

USER GUIDE

# Essential Studio for EJ2 React

---

Version - v25.1.35 | Release Date - March 15, 2024

Signature .....	29
Getting Started.....	29
Dependencies.....	29
Setup your development environment .....	29
Adding Syncfusion packages .....	29
Adding Style sheet to the Application.....	30
Add Signature to the project.....	30
Run the application .....	30
Customization in React Signature component .....	31
Stroke Width .....	31
Stroke Color .....	32
Background Color .....	33
Background Image .....	34
See Also .....	36
Open save in React Signature component.....	36
Open Signature .....	36
Save Signature.....	37
Save with Background.....	40
Draw in React Signature component .....	42
Draw .....	42
User interaction in React Signature component .....	44
Undo and Redo .....	44
Clear .....	44
Disabled.....	44
ReadOnly .....	44
Accessibility in React Signature component .....	48
Keyboard interaction .....	49
Ensuring accessibility .....	50
See also .....	50
How To .....	50
Integration toolbar in React Signature component .....	50
Skeleton .....	61
Getting Started.....	61
Dependencies.....	61
Setup your development environment .....	61

Adding Syncfusion packages .....	61
Adding style sheet to the Application .....	62
Add Skeleton to the Application .....	62
Run the application .....	62
Shapes in React Skeleton component.....	63
Circle skeleton shape .....	63
Square skeleton shape .....	63
Rectangle skeleton shape .....	64
Text skeleton shape .....	64
Shimmer effect in React Skeleton component .....	66
Styles in React Skeleton component.....	68
cssClass.....	68
Visible .....	69
Accessibility in React Skeleton component .....	70
WAI-ARIA attributes.....	71
Ensuring accessibility .....	71
See also .....	71
Smithchart.....	71
Getting Started.....	71
Dependencies.....	71
Installation and configuration.....	72
Add Smith Chart to the Project.....	72
Module Injection.....	73
Add Series to Smithchart .....	74
Add title to SmithChart .....	76
Enable Marker to Smithchart.....	78
Enable DataLabel to Smithchart Marker.....	80
Enable Legend for Smithchart.....	83
Enable Tooltip for Smithchart Series .....	85
Working with data in React Smithchart component .....	87
Data Binding.....	87
Smithchart dimensions in React Smithchart component .....	89
Size for container .....	89
Size for Smith chart .....	91
Title subtitle in React Smithchart component .....	93

Enable title .....	93
Trim title.....	95
Smithchart axis in React Smithchart component.....	96
Labels customization.....	97
Gridlines .....	100
Axis line .....	102
Smithchart legend in React Smithchart component.....	105
Position and alignment .....	105
Customization .....	112
Toggle visibility.....	118
Smithchart tooltip in React Smithchart component.....	121
Smithchart marker in React Smithchart component.....	122
Marker.....	122
Data labels.....	126
Smithchart series in React Smithchart component .....	129
points or data source .....	129
Series customization .....	131
Smithchart print in React Smithchart component.....	132
Print.....	132
Export.....	134
Accessibility in React Smithchart component.....	135
WAI-ARIA attributes.....	136
Keyboard interaction .....	136
Ensuring accessibility .....	136
See also .....	136
Sparkline .....	137
Getting Started.....	137
Dependencies.....	137
Installation and Configuration .....	137
Add Sparkline to the Project .....	138
Module Injection.....	138
Bind data source to Sparkline .....	139
Change the type of Sparkline .....	140
Enable tooltip for Sparkline .....	141
Sparkline dimensions in React Sparkline component.....	142



Size for container .....	142
Size for sparkline .....	143
Sparkline types in React Sparkline component .....	146
Axis customization in React Sparkline component .....	151
Change value type of the sparkline.....	152
Change min and max values of axis .....	155
Change value of axis.....	157
Axis line customization .....	158
Special points customization in React Sparkline component .....	160
Range band in React Sparkline component .....	162
Range band customization.....	162
Multiple range band customization .....	163
Marker in React Sparkline component .....	164
Adding marker to the sparkline .....	164
Adding marker to special point.....	165
Customizing markers.....	166
Data labels in React Sparkline component .....	167
Enable data label.....	167
Customize data label.....	168
Format data label text.....	169
User interaction in React Sparkline component .....	171
Tooltip .....	171
Track line .....	175
Appearance in React Sparkline component.....	176
Sparkline border.....	176
Sparkline padding.....	177
Sparkline area customization.....	178
Sparkline theme .....	179
Localization in React Sparkline component .....	180
Tooltip format .....	180
Accessibility in React Sparkline component .....	181
WAI-ARIA attributes.....	182
Keyboard interaction .....	182
Ensuring accessibility .....	183
See also .....	183

Ej1 api migration in React Sparkline component.....	183
Sparkline Types .....	183
Databinding.....	183
Markers .....	184
Data labels.....	185
Range band .....	186
Special points customization .....	186
Axis customization .....	187
Appearance customization .....	188
Tooltip .....	190
Rendering.....	191
Localization .....	191
Methods.....	192
Events.....	192
SpeedDial .....	194
Getting Started with React Speed Dial Control.....	194
Dependencies.....	194
Installation and Configuration .....	194
Adding Syncfusion packages .....	195
Adding Speed Dial component to the Application.....	195
Adding CSS Reference .....	195
Running the application.....	195
Items in React Speed dial component .....	197
Icons in Speed Dial items .....	197
Animation.....	201
Template .....	202
Positions in React Speed dial component.....	202
Opens items on hover .....	204
Programmatically show/hide Speed Dial items .....	204
Programmatically refresh the position .....	206
Display modes in React Speed dial component.....	208
Linear display mode .....	208
Radial display mode (Radial Menu) .....	209
Radial menu in React Speed dial component .....	209
Radial menu direction .....	209

Radial menu start and end angle .....	210
Offset.....	211
Template in React Speed dial component .....	213
Item template .....	213
Popup template .....	215
Styles in React Speed dial component .....	216
SpeedDial button .....	217
Disabled.....	219
cssClass.....	220
Visible .....	221
Tooltip .....	221
Opens on hover.....	222
Modal in React Speed dial component .....	223
Event in React Speed dial component .....	224
clicked .....	225
created .....	225
beforeOpen .....	226
onOpen .....	227
beforeClose .....	227
onClose.....	228
beforeItemRender .....	229
Accessibility in React Speed dial component.....	231
WAI-ARIA attributes.....	232
Keyboard interaction .....	232
Ensuring accessibility .....	232
See also .....	233
Spinner .....	233
Getting Started.....	233
Create the Spinner globally.....	233
Template in React Spinner component .....	235
Types in React Spinner component .....	239
Style in React Spinner component.....	243
Customizing the spinner .....	243
SplitButton .....	244
Getting Started.....	244

Dependencies.....	244
Installation and configuration.....	244
Adding syncfusion packages .....	245
Adding CSS reference.....	245
Adding SplitButton component .....	245
Binding data source .....	246
Run the application .....	246
See Also .....	248
Icons and separator in React Split button component .....	248
SplitButton icons .....	248
Separator.....	250
See Also .....	252
Popup items in React Split button component.....	252
Icons .....	252
Template .....	253
See Also .....	256
Accessibility in React SplitButton component .....	256
WAI-ARIA attributes.....	257
Keyboard interaction .....	258
Ensuring accessibility .....	258
See also .....	258
Style and appearance in React Split button component .....	258
How To .....	258
Create right to left splitbutton in React Split button component .....	258
Group items in popup in React Split button component.....	260
Open a dialog on popup item click in React Split button component .....	262
Set the disabled state in React Split button component .....	264
Underline a character in a text in React Split button component .....	265
Ej1 api migration in React Split button component.....	267
Properties.....	267
Methods .....	268
Events.....	269
Splitter.....	270
Getting Started.....	270
Dependencies.....	270

Installation and configuration.....	271
Adding Splitter to the project .....	272
Orientation.....	273
Load content to the pane.....	274
See Also .....	277
Getting Started.....	277
Dependencies.....	277
Installation and configuration.....	277
Adding Splitter to the project .....	278
Orientation.....	280
Load content to the pane.....	280
See Also .....	283
Split panes in React Splitter component.....	283
Horizontal layout.....	283
Vertical layout.....	286
Multiple panes .....	290
Separator.....	295
Nested Splitter .....	298
Add or remove pane .....	306
See Also .....	314
Expand collapse in React Splitter component .....	314
Collapsible panes .....	314
Control the expand and collapse action in programmatic manner .....	318
Specify initial state to panes .....	320
See Also .....	324
Resize in React Splitter component .....	324
Min and Max validation .....	324
Prevent resizing.....	328
Refresh content on resizing .....	332
Customize the resize grip and cursor.....	332
See Also .....	335
Pane content in React Splitter component.....	335
Template .....	335
React UI components.....	339
Plain content .....	339

HTML Markup .....	341
Pane content using selector.....	343
Pane sizing in React Splitter component .....	346
Pane size in pixel .....	346
Pane size in percentage .....	347
Auto size panes .....	349
Fixed pane .....	352
Different layouts in React Splitter component .....	356
Code editor style layout.....	356
Outlook style layout.....	361
See Also .....	369
Style in React Splitter component .....	369
Customizing the split bar .....	369
Customizing the split bar resize handle .....	370
Customizing the split bar arrows .....	371
To hide the resize handle in Splitter .....	372
Globalization in React Splitter component .....	372
RTL.....	372
See Also .....	374
Accessibility in React Splitter component.....	374
Keyboard interaction .....	375
Ensuring accessibility .....	375
See also .....	375
Ej1 api migration in React Splitter component.....	375
Common.....	375
Accessibility and Localization.....	376
Control State .....	377
State Maintenance.....	377
Pane Properties.....	377
Animation.....	379
Spreadsheet .....	379
Overview of the React Spreadsheet component.....	379
Key features .....	379
Getting Started.....	381
Dependencies.....	381

Setup for Local Development.....	381
Adding Syncfusion packages .....	382
Adding CSS reference.....	382
Adding Spreadsheet component .....	382
Run the application .....	382
See Also .....	383
Creating a Next.js Application Using Syncfusion React Components.....	383
What is Next.js? .....	383
Prerequisites .....	383
Create a Next.js application .....	384
Install Syncfusion React packages.....	384
Import Syncfusion CSS styles .....	385
Add Syncfusion React component .....	385
Run the application .....	391
Data binding in React Spreadsheet component .....	391
Local data .....	391
Remote data.....	394
Cell data binding .....	455
Dynamic data binding and Datasource change event .....	478
Note .....	486
See Also .....	487
Open save in React Spreadsheet component.....	487
Open.....	487
Save .....	492
Server Configuration .....	596
Server Dependencies .....	597
Note .....	597
See Also .....	597
Worksheet in React Spreadsheet component .....	597
Add sheet .....	597
Delete sheet .....	601
Rename sheet .....	602
Headers .....	602
Gridlines .....	602
Sheet visibility .....	605

Note .....	610
See Also .....	610
Cell range in React Spreadsheet component.....	610
Wrap text .....	611
Merge cells.....	616
Data Validation.....	625
Auto Fill .....	632
Clear .....	637
Note .....	642
See Also .....	642
Editing in React Spreadsheet component.....	642
Edit cell.....	642
Save cell.....	642
Cancel editing.....	642
Limitations.....	648
Note .....	648
See Also .....	648
Context menu in React Spreadsheet component.....	648
Context Menu Items in Row Cell.....	649
Context Menu Items in Row Header / Column Header .....	649
Context Menu Items in Pager .....	649
Context Menu Customization .....	650
Note .....	653
See Also .....	653
Formulas in React Spreadsheet component.....	653
Usage.....	653
Culture-Based Argument Separator.....	653
Create User Defined Functions / Custom Functions.....	672
Formula bar .....	687
Named Ranges .....	687
Supported Formulas.....	695
Formula Error Dialog.....	697
Note .....	698
See Also .....	698
Formatting in React Spreadsheet component.....	698



Number Formatting .....	699
Text and cell formatting .....	705
Conditional Formatting .....	712
Note .....	717
See Also .....	717
Illustrations in React Spreadsheet component.....	717
Image.....	717
Chart.....	793
Note .....	804
See Also .....	805
Rows and columns in React Spreadsheet component.....	805
Insert .....	805
Delete.....	814
Limitations of insert and delete .....	819
Hide and show .....	819
Size .....	824
Changing text in column headers .....	864
Note .....	865
See Also .....	865
Sort in React Spreadsheet component .....	865
Sort by cell value .....	865
Data contains header .....	886
Case sensitive sort.....	886
Sort multiple columns .....	887
Custom sort comparer .....	908
Known error validations.....	908
Limitations.....	909
Note .....	909
See Also .....	909
Filter in React Spreadsheet component .....	909
Apply filter on UI .....	909
Filter by criteria.....	909
Filter by cell value .....	930
Clear filter.....	930
Clear filter on a field.....	930

Reapply filter.....	930
Known error validations.....	930
Limitations.....	930
Note .....	931
See Also .....	931
Link in React Spreadsheet component .....	931
Insert Link.....	931
Edit Hyperlink.....	931
Remove Hyperlink.....	931
How to change target attribute .....	932
Limitations.....	938
Note .....	938
See Also .....	938
Clipboard in React Spreadsheet component .....	938
Cut.....	938
Copy .....	938
Paste.....	939
Prevent the paste functionality .....	960
Limitations.....	981
Note .....	981
Selection in React Spreadsheet component .....	981
Cell selection .....	982
Row selection .....	982
Column selection .....	986
How to remove selection in the spreadsheet.....	991
Limitations.....	995
Note .....	995
Scrolling in React Spreadsheet component .....	995
Finite Scrolling.....	995
Virtual Scrolling.....	995
Finite scrolling with defined rows and columns .....	996
Note .....	999
Protect sheet in React Spreadsheet component .....	999
Protect Sheet .....	1000
Unprotect Sheet.....	1006

Unlock the particular cells in the protected sheet.....	1006
Protect Workbook.....	1013
Unprotect Workbook.....	1019
Note .....	1020
See Also.....	1020
Searching in React Spreadsheet component.....	1020
Find.....	1020
Replace.....	1020
Go to.....	1021
Limitations.....	1026
Note .....	1026
Undo redo in React Spreadsheet component.....	1026
Undo.....	1026
Redo .....	1027
Update custom actions in UndoRedo collection.....	1027
Note .....	1048
See Also .....	1048
Accessibility in React Spreadsheet component .....	1048
WAI-ARIA attributes.....	1049
Keyboard interaction .....	1050
Ensuring accessibility .....	1051
See also .....	1052
Keyboard shortcuts in React Spreadsheet component .....	1052
Note .....	1053
See Also.....	1054
Ribbon in React Spreadsheet component .....	1054
Ribbon Customization.....	1054
Note .....	1064
See Also.....	1064
Global local in React Spreadsheet component.....	1064
Localization .....	1064
Internationalization.....	1079
Right to left (RTL) .....	1082
Note .....	1087
See Also.....	1087

Freeze pane in React Spreadsheet component .....	1087
Apply freezepanes on UI .....	1087
FrozenRows .....	1087
FrozenColumns .....	1087
Limitations.....	1108
Note .....	1108
See Also .....	1108
Template in React Spreadsheet component .....	1108
Note .....	1115
See Also .....	1115
Use Cases .....	1115
Collaborative editing in React Spreadsheet component .....	1115
How To .....	1122
Sort a range by custom list in React Spreadsheet component .....	1122
Print in React Spreadsheet component .....	1143
Create a object structure in React Spreadsheet component .....	1153
Mobile responsiveness in React Spreadsheet component .....	1197
Feature list in React Spreadsheet component.....	1198
See Also .....	1199
Stepper .....	1199
Getting Started.....	1199
Dependencies.....	1199
Installation and Configuration .....	1200
Adding Syncfusion packages .....	1200
Adding CSS Reference .....	1201
Adding Stepper component to the application .....	1201
Running the application .....	1202
Configure icon and label .....	1203
Steps in React Stepper component.....	1206
Adding steps.....	1206
Optional steps .....	1211
Disabling steps .....	1214
Setting active step.....	1217
Step status.....	1220
Step styling.....	1223

Step validation .....	1226
Step types in React Stepper component .....	1226
Default type .....	1226
Label type.....	1229
Indicator type.....	1236
Orientations in React Stepper component .....	1239
Horizontal.....	1239
Vertical .....	1242
Linear flow in React Stepper component .....	1245
Steps validation in React Stepper component.....	1248
Template in React Stepper component .....	1251
Tooltip in React Stepper component .....	1254
Tooltip template .....	1257
Animation in React Stepper component.....	1261
Globalization in React Stepper component .....	1262
Localization .....	1262
RTL.....	1265
Accessibility in React Stepper component.....	1268
WAI-ARIA attributes.....	1269
Keyboard interaction .....	1269
Ensuring accessibility .....	1270
See also .....	1270
Events in React Stepper component.....	1270
created .....	1270
stepChanged .....	1271
stepChanging.....	1273
stepClick .....	1274
beforeStepRender.....	1275
Stock Chart.....	1277
Getting Started.....	1277
Dependencies.....	1277
Installation and configuration.....	1277
Add Stock Chart to the Project.....	1278
Module Injection.....	1280
Populate Chart with Data.....	1281

Add Stock Chart Title.....	1284
Add Crosshair .....	1286
Add Trackball.....	1288
Working with data in React Stock chart component .....	1290
Local Data.....	1290
See Also .....	1292
Chart dimensions in React Stock chart component.....	1293
Size for Container.....	1293
Size for Stock Chart .....	1295
Axis types in React Stock chart component.....	1299
DateTime axis.....	1299
DateTimeCategory axis .....	1301
Logarithmic axis .....	1303
See also .....	1305
Axis customization in React Stock chart component.....	1305
Axis Crossing .....	1305
Title .....	1307
Tick Lines Customization.....	1308
Grid Lines Customization .....	1310
Multiple Axis .....	1312
Inversed Axis .....	1314
Opposed Position .....	1315
Series types in React Stock chart component.....	1317
Line.....	1317
Spline.....	1317
Hilo .....	1317
HiloOpenClose.....	1317
HollowCandle .....	1317
Candle .....	1318
Trend lines in React Stock chart component .....	1319
Linear.....	1319
Exponential .....	1319
Logarithmic .....	1320
Polynomial .....	1320
Power .....	1320

Moving Average .....	1320
Technical indicators in React Stock chart component .....	1322
Accumulation Distribution .....	1322
Average True Range (ATR) .....	1322
Exponential Moving Average (EMA) .....	1322
Momentum .....	1322
Moving Average Convergence Divergence (MACD) .....	1322
Relative Strength Index (RSI).....	1322
Simple Moving Average (SMA) .....	1322
Stochastic .....	1323
Triangular Moving Average (TMA).....	1323
Bollinger Band.....	1323
Tool tip in React Stock chart component.....	1325
Default tooltip.....	1325
Format the tooltip.....	1327
Position the tooltip .....	1329
Customize the appearance of the tooltip .....	1331
Cross hair in React Stock chart component .....	1333
Tooltip for axis .....	1334
Customization .....	1336
Legend in React Stock chart component .....	1338
Position and Alignment.....	1338
Customization .....	1343
Collapsing Legend Item .....	1349
Legend Title.....	1350
Period selector in React Stock chart component.....	1352
Periods .....	1352
Visibility of period selector .....	1355
Range selector in React Stock chart component .....	1356
Export print in React Stock chart component.....	1358
Disable Export and print .....	1360
Appearance in React Stock chart component.....	1362
Stock Chart Title .....	1362
Stock Chart Theme .....	1366
See Also .....	1368

Stock events in React Stock chart component.....	1368
See Also .....	1379
Accessibility in React Stock chart component .....	1379
WAI-ARIA attributes.....	1380
Keyboard interaction .....	1380
Ensuring accessibility .....	1380
See also .....	1381
Switch.....	1381
Getting Started.....	1381
Dependencies.....	1381
Installation and Configuration .....	1381
Adding Syncfusion packages .....	1381
Adding CSS Reference .....	1382
Adding Switch component to the Application .....	1382
Run the application .....	1382
See Also .....	1383
Accessibility in React Switch component.....	1383
WAI-ARIA attributes.....	1384
Keyboard interaction .....	1384
Ensuring accessibility .....	1385
See also .....	1385
Style and appearance in React Switch component.....	1385
How To .....	1385
Set text in React Switch component .....	1385
Change size in React Switch component .....	1386
Customize the appearance of a switch in React Switch component .....	1387
Enable ripple for switch label in React Switch component.....	1390
Enable rtl in React Switch component.....	1391
Set disabled state in React Switch component.....	1392
Submit name and value in form in React Switch component.....	1393
Change switch state using toggle method in React Switch component.....	1394
Tab .....	1395
Getting Started.....	1395
Dependencies.....	1395
Setup for Local Development.....	1396



Adding Syncfusion packages .....	1397
Adding CSS reference.....	1397
Initialize the Tab using JSON items collection.....	1397
Initialize the Tab using HTML elements .....	1399
See Also .....	1402
Adaptive in React Tab component.....	1402
Scrollable.....	1402
Popup .....	1408
See Also .....	1414
Header in React Tab component .....	1414
Styles .....	1414
Icon positions .....	1418
See Also .....	1422
Localization in React Tab component .....	1422
Loading translations.....	1422
Orientation in React Tab component .....	1425
Drag and drop in React Tab component.....	1432
Drag and drop item between tabs.....	1435
Drag and drop items to external source .....	1442
Drag and drop items from external source.....	1446
Accessibility in React Tab component .....	1451
ARIA attributes.....	1452
Keyboard interaction .....	1453
Ensuring accessibility .....	1453
See also .....	1453
Style in React Tab component .....	1453
Customizing Tab.....	1453
Customizing the Tab items.....	1454
Customizing Tab's header .....	1454
Customizing Tab's header icon .....	1454
Customizing Tab's content.....	1454
Customizing the hover state of Tab control.....	1455
Customizing selected item of Tab control .....	1455
How To .....	1456
Load content through post in React Tab component .....	1456

Prevent content swipe selection in React Tab component .....	1458
Customize selected tab styles in React Tab component.....	1461
Customize tab scroll step in React Tab component.....	1464
Create wizard using tab in React Tab component .....	1469
Load tab with data source in React Tab component .....	1496
Add nested tabs in React Tab component.....	1498
Set state persistence of the tab component in React Tab component .....	1506
Set custom animation in React Tab component.....	1509
Load tab items dynamically in React Tab component .....	1513
Create collapsible tabs in React Tab component.....	1517
Customize tab content height in React Tab component .....	1522
Render other components in tab using template in React Tab component .....	1526
Adding dynamic items with content reuse in React Tab component.....	1531
Reorder active tab in React Tab component .....	1535
Display tool tip on tab header in React Tab component .....	1543
Tab selection in React Tab component.....	1546
React hooks tab in React Tab component .....	1553
Tab key navigation in React Tab component.....	1554
Ej1 api migration in React Tab component.....	1560
Accessibility and Localization.....	1560
AjaxSettings.....	1561
Animation.....	1561
Header.....	1562
Items .....	1562
Common.....	1564
TextArea .....	1566
Getting Started with React TextArea Component .....	1566
Dependencies.....	1566
Installation and configuration.....	1566
Adding TextArea to the application .....	1567
Running the application .....	1568
Getting and setting values .....	1568
Floating Label in React TextArea Component.....	1571
Placeholder with localization .....	1572
Rows and Columns in React TextArea Component .....	1574

Resize in React TextArea Component.....	1575
Width of react TextArea component .....	1576
Maximum Length in React TextArea Component.....	1577
Form Support in React TextArea Component.....	1578
Integration of react TextArea component with FormValidator component .....	1579
Sizing in React TextArea Component.....	1582
Filled and outline mode .....	1583
Custom styling with cssClass API in TextArea .....	1584
Setting the disabled state in TextArea .....	1585
Set the readonly TextArea .....	1586
Set the rounded corner in TextArea .....	1586
Static clear button in TextArea .....	1587
Customize the TextArea background color and text color .....	1588
Change the floating label color of the TextArea .....	1588
Adding mandatory asterisk to placeholder.....	1590
Events in React TextArea Component .....	1591
Created event.....	1591
Input event.....	1591
Change event .....	1592
Focus event .....	1593
Blur event.....	1594
Destroyed event.....	1594
Methods in React TextArea Component .....	1595
FocusIn method .....	1595
FocusOut method .....	1596
GetPersistData method.....	1597
TextBox .....	1598
Getting Started.....	1598
Dependencies.....	1598
Installation and Configuration .....	1598
Adding TextBox to the application.....	1599
Adding icons to the TextBox .....	1600
Floating Label .....	1605
See Also .....	1607
Groups in React Textbox component .....	1607

With icon and floating label .....	1609
With clear button and floating label .....	1617
Floating Label without required attribute .....	1618
Multi-line input with floating label .....	1623
See Also .....	1625
Sizing in React Textbox component .....	1625
Validation in React Textbox component.....	1631
Multiline in React Textbox component.....	1634
Create multiline textbox .....	1634
Implementing floating label .....	1635
Auto resizing .....	1637
Disable resizing .....	1640
Limit the text length.....	1641
Count characters.....	1643
Accessibility in React Textbox component.....	1646
WAI-ARIA attributes.....	1647
Ensuring accessibility .....	1647
See also .....	1647
React functional component in React Textbox component .....	1647
Style appearance in React Textbox component .....	1651
Customizing the appearance of TextBox wrapper element .....	1651
Customizing the TextBox placeholder .....	1651
How To .....	1651
Set the rounded corner in React Textbox component .....	1651
Set the disabled state in React Textbox component .....	1654
Set the read only textbox in React Textbox component .....	1654
Add textbox programmatically in React Textbox component .....	1656
Add floating label to textbox programmatically in React Textbox component.....	1657
Change the floating label color of the textbox in React Textbox component .....	1660
Change the color of the textbox based on its value in React Textbox component .....	1662
Add floating label to read only textbox in React Textbox component .....	1665
Customize the textbox background color and text color in React Textbox component .....	1667
Migration from css textbox to react textbox in React Textbox component.....	1668
Normal textbox .....	1668
Floating label textbox.....	1669

Timeline.....	1669
Getting Started.....	1669
Dependencies.....	1669
Installation and Configuration .....	1670
Adding Syncfusion packages .....	1670
Adding CSS Reference .....	1670
Adding Timeline component to the application .....	1671
Running the application.....	1671
Items in React Timeline component .....	1672
Adding content.....	1673
Adding opposite content .....	1674
Dot item .....	1675
Disabling items.....	1676
Css class.....	1677
Orientations in React Timeline component.....	1677
Vertical .....	1677
Horizontal.....	1679
Alignment in React Timeline component .....	1680
Before.....	1680
After .....	1681
Alternate .....	1682
Alternate reverse .....	1683
Reverse in React Timeline component .....	1684
Template in React Timeline component.....	1686
Customization in React Timeline component.....	1689
Connector styling .....	1689
Dot styling .....	1692
Accessibility in React Timeline component .....	1699
WAI-ARIA attributes.....	1699
Ensuring accessibility .....	1700
See also .....	1700
Events in React Timeline component .....	1700
created .....	1700
beforeItemRender .....	1701
TimePicker.....	1702

Getting started .....	1702
Dependencies.....	1702
Installation and configuration.....	1703
Adding Syncfusion packages .....	1703
Adding Style sheet to the Application.....	1703
Adding TimePicker component to the Application .....	1703
Run the application .....	1704
Setting the value, min, and max time .....	1706
Setting the time format .....	1708
See Also .....	1709
Time range in React Timepicker component .....	1709
Time masking in React Timepicker component .....	1711
Configure Mask Placeholder .....	1714
Strict mode in React Timepicker component .....	1717
Globalization in React Timepicker component .....	1721
Right-To-Left .....	1822
Accessibility in React Timepicker component.....	2405
WAI-ARIA attributes.....	2406
Keyboard Interaction .....	2407
Ensuring accessibility .....	2409
See also .....	2410
Style appearance in React Timepicker component .....	2410
Customizing the appearance of TimePicker wrapper element .....	2410
Customizing the TimePicker icon element.....	2410
Customizing the TimePicker popup .....	2410
Customizing the TimePicker popup content.....	2411
Full screen mode support in mobiles and tablets.....	2411
How To .....	2413
Css customization in React Timepicker component .....	2413
Client side validation using form validator in React Timepicker component .....	2414
Ej1 api migration in React Timepicker component.....	2415
Time Selection.....	2415
Time Format.....	2416
Time Range.....	2416
Disabled Time Ranges .....	2416

Customization .....	2416
Accessibility .....	2417
Persistence .....	2417
Validation .....	2418
Common .....	2418
Globalization .....	2419
Strict Mode .....	2420
Open and Close .....	2420
Toast .....	2420
Getting Started .....	2420
Dependencies .....	2420
Installation and configuration .....	2421
Initialize the Toast with message .....	2422
Initialize the Toast with target .....	2424
See Also .....	2426
Getting Started .....	2426
Dependencies .....	2426
Installation and configuration .....	2427
Initialize the Toast with message .....	2428
Initialize the Toast with target .....	2429
See Also .....	2430
Config in React Toast component .....	2430
Title and content template .....	2430
Specifying custom target .....	2432
Close Button .....	2433
Progress bar .....	2433
Newest on top .....	2433
Width and height .....	2435
See Also .....	2443
Position in React Toast component .....	2443
Predefined .....	2443
Custom .....	2444
See Also .....	2458
Action buttons in React Toast component .....	2458
See Also .....	2461

Timeout in React Toast component.....	2461
Visual representation of timeOut .....	2461
How long the toast displayed in page.....	2461
Static toast .....	2464
See Also .....	2467
Template in React Toast component.....	2467
See Also .....	2470
Style in React Toast component .....	2470
Customizing the toast title .....	2470
Customizing the toast content.....	2470
Customizing the toast icon.....	2471
Customizing the toast background .....	2471
Animation in React Toast component .....	2471
Toast services in React Toast component.....	2477
Show Toast with predefined types .....	2477
Show Toast with ToastModel .....	2481
Accessibility in React Toast component.....	2484
WAI-ARIA attributes.....	2485
Ensuring accessibility .....	2486
See also .....	2486
How To .....	2487
Prevent duplicate toast display in React Toast component .....	2487
Restrict the maximum toast to show in React Toast component.....	2492
Customize progress bar theme and sizing in React Toast component.....	2496
Play an audio before open the toast in React Toast component .....	2501
Show different types of toast in React Toast component .....	2504
Show multiple toasts in various positions in React Toast component .....	2508
Close the toast with click tap in React Toast component.....	2512
Add dynamic template in React Toast component.....	2515
Prevent toast close with mobile swipe in React Toast component.....	2519



## Signature

### Getting Started

The following section explains the required steps to build the Signature component with its basic usage in step-by-step procedure.

To get start quickly with Signature Component using React, you can check on this video:

### Dependencies

The following list of dependencies are required to use the Signature component in your application.

```
`javascript
|-- @syncfusion/ej2-react-base
`
```

### Setup your development environment

You can use [Create-react-app](#) to setup the applications.

To install `create-react-app` run the following command.

```
`bash
npm install -g create-react-app
`
```

To set-up a React application in TypeScript environment, run the following command.

```
`bash
npx create-react-app my-app --template typescript
cd my-app
npm start
`
```

To set-up a React application in JavaScript environment, run the following command.

```
`bash
npx create-react-app my-app
cd my-app
npm start
`
```

### Adding Syncfusion packages

All the available Essential JS 2 packages are published in [npmjs.com](https://www.npmjs.com) public registry.

You can choose the component that you want to install. For this application, we are going to use Signature component.

To install Signature component, use the following command

```
`bash
```

```
npm install @syncfusion/ej2-react-inputs --save
```

The above command installs [Signature dependencies](#) which are required to render the component in the **React** environment.

### Adding Style sheet to the Application

Add Signature component's styles as given below in **App.css**.

```
`css
@import "../node_modules/@syncfusion/ej2-base/styles/material.css";
@import "../node_modules/@syncfusion/ej2-inputs/styles/material.css";
```

### Add Signature to the project

Now, you can create **Signature** component in the application. Add **Signature** component in **src/App.tsx** file using the following code snippet.

```
`ts
import { SignatureComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
function App() {
  return (<div style={{marginTop: '150px'}}>
    <SignatureComponent id='signature'></SignatureComponent>
  </div>);
}
export default App;
```

### Run the application

Use the **npm run start** command to run the application in the browser.

```
npm run start
```

The following example shows the default Signature.

### **INDEX.JSX**

```
import { SignatureComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// initializes Signature component.
function App() {
  return (<div id='container'>
    <div id='signature-control'>
```

```

        <h4>Sign here</h4>
        <SignatureComponent id='signature' />
      </div>
    </div>);
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

### INDEX.TSX

```

import { SignatureComponent } from '@syncfusion/ej2-react-inputs';
import * as React from "react";
import * as ReactDOM from "react-dom";
// initializes Signature component.
function App() {
  return (
    <div id='container'>
      <div id='signature-control'>
        <h4>Sign here</h4>
        <SignatureComponent id='signature' />
      </div>
    </div>
  );
};
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

### Customization in React Signature component

The Signature component draws stroke/path using `moveTo()` and `lineTo()` methods to connect one or more points while drawing in canvas. The stroke width can be modified by using its color and width. And the background can be modified by using its background color and background image.

#### Stroke Width

The variable stroke width is based on the values of [maxStrokeWidth](#), [minStrokeWidth](#) and [velocity](#) for smoother and realistic signature. The default value of minimum stroke width is set as 0.5, maximum stroke width is set as 2.5 and velocity is set as 0.7.

In the following example, minimum stroke width is set as 0.5, maximum stroke width is set as 3 and velocity is set as 0.7.

### INDEX.JSX

```

import { SignatureComponent } from '@syncfusion/ej2-react-inputs';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (<div id='container'>
    <div className='signature-control'>
      <h4>Sign here</h4>
      <SignatureComponent id='signature' maxStrokeWidth={3}
minStrokeWidth={0.5} velocity={0.7}/>
    </div>
  </div>);
}
```

```
;
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

### INDEX.TSX

```
import { SignatureComponent } from '@syncfusion/ej2-react-inputs';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (
        <div id='container'>
            <div className='signature-control'>
                <h4>Sign here</h4>
                <SignatureComponent id='signature' maxStrokeWidth= {3}
minStrokeWidth= {0.5} velocity= {0.7}/>
            </div>
        </div>
    );
};
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

### Stroke Color

Color of the stroke can be specified by using [strokeColor](#) property and it accepts hexadecimal code, RGB, and text. The default value of this property is “#000000”.

### INDEX.JSX

```
import { SignatureComponent } from '@syncfusion/ej2-react-inputs';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { getComponent } from '@syncfusion/ej2-base';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    function clickEventHandler() {
        let signature = getComponent(document.getElementById('signature'),
'signature');
        let color = document.getElementById('text').value;
        signature.strokeColor = color;
    }
    return (<div id='container'>
        <div id="signature-base-control">
            <div id="input">
                <input type="text" id="text" placeholder="Enter the
Stroke Color Value"></input>
                <ButtonComponent onClick={clickEventHandler}
isPrimary={true}>Set Stroke Color</ButtonComponent>
            </div>
            <div id="signature-control">
                <SignatureComponent id='signature' />
            </div>
        </div>
    </div>);
}
```

```
;
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

### INDEX.TSX

```
import { SignatureComponent } from '@syncfusion/ej2-react-inputs';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { getComponent, closest } from '@syncfusion/ej2-base';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  function clickEventHandler(): void {
    let signature: SignatureComponent =
      getComponent(document.getElementById('signature'), 'signature');
    let color: string = document.getElementById('text').value;
    signature.strokeColor = color;
  }
  return (
    <div id='container'>
      <div id="signature-base-control">
        <div id="input">
          <input type="text" id="text" placeholder="Enter the
Stroke Color Value"></input>
          <ButtonComponent onClick={clickEventHandler}
isPrimary={true}>Set Stroke Color</ButtonComponent>
        </div>
        <div id="signature-control">
          <SignatureComponent id='signature' />
        </div>
      </div>
    </div>
  );
};
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

### Background Color

Background color of a signature can be specified by using [backgroundColor](#) property and it accepts hexadecimal code, RGB, and text. The default value of this property is "#ffffff".

### INDEX.JSX

```
import { SignatureComponent } from '@syncfusion/ej2-react-inputs';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { getComponent } from '@syncfusion/ej2-base';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  function clickEventHandler() {
    let signature = getComponent(document.getElementById('signature'),
'signature');
    let bgColor = document.getElementById('text').value;
    signature.backgroundColor = bgColor;
  }
}
```

```

    return (<div id='container'>
      <div id="signature-base-control">
        <div id="input">
          <input type="text" id="text" placeholder="Enter the
Background Color Value"></input>
          <ButtonComponent onClick={clickEventHandler}
isPrimary={true}>Set Background Color</ButtonComponent>
        </div>
        <div id="signature-control">
          <SignatureComponent id='signature' />
        </div>
      </div>
    </div>);
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

### INDEX.TSX

```

import { SignatureComponent } from '@syncfusion/ej2-react-inputs';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { getComponent, closest } from '@syncfusion/ej2-base';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  function clickEventHandler(): void {
    let signature: SignatureComponent =
getComponent(document.getElementById('signature'), 'signature');
    let bgColor: string = document.getElementById('text').value;
    signature.backgroundColor = bgColor;
  }
  return (
    <div id='container'>
      <div id="signature-base-control">
        <div id="input">
          <input type="text" id="text" placeholder="Enter the
Background Color Value"></input>
          <ButtonComponent onClick={clickEventHandler}
isPrimary={true}>Set Background Color</ButtonComponent>
        </div>
        <div id="signature-control">
          <SignatureComponent id='signature' />
        </div>
      </div>
    </div>
  );
};
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

### Background Image

Background image of a signature can be specified by using [backgroundImage](#) property. The background image can be set by either hosting the image in our local IIS or online image.

**INDEX.JSX**

```

import { SignatureComponent } from '@syncfusion/ej2-react-inputs';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { getComponent } from '@syncfusion/ej2-base';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    function clickEventHandler() {
        let signature = getComponent(document.getElementById('signature'),
'signature');
        let url = document.getElementById('text').value;
        signature.backgroundImage = url;
    }
    return (<div id='container'>
        <div id="signature-base-control">
            <div id="input">
                <input type="text" id="text" placeholder="Enter the URL
of the background Image"></input>
                <ButtonComponent onClick={clickEventHandler}
isPrimary={true}>Set Background Image</ButtonComponent>
            </div>
            <div id="signature-control">
                <SignatureComponent id='signature' />
            </div>
        </div>
    </div>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

**INDEX.TSX**

```

import { SignatureComponent } from '@syncfusion/ej2-react-inputs';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { getComponent, closest } from '@syncfusion/ej2-base';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    function clickEventHandler(): void {
        let signature: SignatureComponent =
getComponent(document.getElementById('signature'), 'signature');
        let url: string = document.getElementById('text').value;
        signature.backgroundImage = url;
    }
    return (
        <div id='container'>
            <div id="signature-base-control">
                <div id="input">
                    <input type="text" id="text" placeholder="Enter the URL
of the background Image"></input>
                    <ButtonComponent onClick={clickEventHandler}
isPrimary={true}>Set Background Image</ButtonComponent>
                </div>
                <div id="signature-control">

```

```

        <SignatureComponent id='signature' />
      </div>
    </div>
  </div>
);
};
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

See Also

- [Save With Background](#)

### Open save in React Signature component

The Signature component supports to open the signature by using hosted/online URL or base64. And it also supports various save options like image, base64, and blob.

### Open Signature

The signature component opens a pre-drawn signature as either base64 or hosted/ online URL using the [load](#) method. It supports the PNG, JPEG, and SVG image's base64.

### INDEX.JSX

```

import { SignatureComponent } from '@syncfusion/ej2-react-inputs';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { getComponent } from '@syncfusion/ej2-base';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  function clickEventHandler() {
    let signature = getComponent(document.getElementById('signature'),
'signature');
    let sign = document.getElementById('text').value;
    signature.load(sign);
  }
  return (<div id='container'>
    <div id="signature-base-control">
      <div id="input">
        <input type="text" id="text" placeholder="Enter the
Base64 or URL of signature"></input>
        <ButtonComponent onClick={clickEventHandler}
isPrimary={true}>Open</ButtonComponent>
      </div>
      <div id="signature-control">
        <SignatureComponent id='signature' />
      </div>
    </div>
  </div>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

### INDEX.TSX



```
import { SignatureComponent } from '@syncfusion/ej2-react-inputs';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { getComponent, closest } from '@syncfusion/ej2-base';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    function clickEventHandler(): void {
        let signature: SignatureComponent =
getComponent(document.getElementById('signature'), 'signature');
        let sign: string = document.getElementById('text').value;
        signature.load(sign);
    }
    return (
        <div id='container'>
            <div id="signature-base-control">
                <div id="input">
                    <input type="text" id="text" placeholder="Enter the
Base64 or URL of signature"></input>
                    <ButtonComponent onClick={clickEventHandler}
isPrimary={true}>Open</ButtonComponent>
                </div>
                <div id="signature-control">
                    <SignatureComponent id='signature' />
                </div>
            </div>
        </div>
    );
};
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

### Save Signature

The Signature component saves the signature as base64, blob, and image like PNG, JPEG, and SVG.

#### Save as Base64

The `getSignature` method is used to get the signature as base64 with the PNG, JPEG, and SVG type. This can be loaded to signature using [load](#) method.

### INDEX.JSX

```
import { SignatureComponent } from '@syncfusion/ej2-react-inputs';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { DialogComponent } from '@syncfusion/ej2-react-popups';
import { getComponent } from '@syncfusion/ej2-base';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    let dialogInstance;
    const animationSettings = { effect: 'Zoom', duration: 400, delay: 0 };
    function getSign() {
        let signature = getComponent(document.getElementById('signature'),
'signature');
        dialogInstance.content = signature.getSignature();
        dialogInstance.show();
    }
    return (<div id='container'>
```

```

        <div id="signature-control">
            <SignatureComponent id='signature' />
        </div>
        <ButtonComponent id="save" cssClass='e-primary' content='Save as
Base64' onClick={getSign}/>
        <DialogComponent id="defaultdialog" showCloseIcon={true}
animationSettings={animationSettings} ref={dialog => dialogInstance =
dialog} height='auto' header='Base64 of the signature' visible={false}
width='80%' />
        </div>);
    }
;
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

## INDEX.TSX

```

import { SignatureComponent } from '@syncfusion/ej2-react-inputs';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { DialogComponent } from '@syncfusion/ej2-react-popups';
import { getComponent, closest } from '@syncfusion/ej2-base';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    let dialogInstance: DialogComponent;
    const animationSettings: any = { effect: 'Zoom', duration: 400, delay:
0 };
    function getSign() {
        let signature: Signature =
getComponent(document.getElementById('signature'), 'signature');
        dialogInstance.content = signature.getSignature();
        dialogInstance.show();
    }
    return (
        <div id='container'>
            <div id="signature-control">
                <SignatureComponent id='signature' />
            </div>
            <ButtonComponent id="save" cssClass='e-primary' content='Save as
Base64' onClick={getSign}/>
            <DialogComponent id="defaultdialog" showCloseIcon={true}
animationSettings={animationSettings} ref={dialog => dialogInstance =
dialog} height='auto' header='Base64 of the signature' visible={false}
width='80%' />
            </div>
        );
    };
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

### Save as Blob

The [saveAsBlob](#) method is used to save the signature as Blob. It is defined as the chunk of binary data stored as a single entity in a database system.

*Save as Image*

The [save](#) method is used to save the signature as an image. And it accepts file name and file type as parameter. The file type parameter supports PNG, JPEG, and SVG and the default file type is PNG.

**INDEX.JSX**

```
import { SignatureComponent } from '@syncfusion/ej2-react-inputs';
import { SplitButtonComponent } from '@syncfusion/ej2-react-splitbuttons';
import { getComponent } from '@syncfusion/ej2-base';
import * as React from "react";
import * as ReactDOM from "react-dom";

function App() {
  let items;
  items = [
    {
      text: 'Png'
    },
    {
      text: 'Jpeg'
    },
    {
      text: 'Svg'
    }
  ];
  function onSelect(args) {
    let signature = getComponent(document.getElementById('signature'),
    'signature');
    signature.save(args.item.text, 'Signature');
  }
  return (<div id='container'>
    <div>
      <span>Sign here</span>
      <SplitButtonComponent content="Save" iconCss='e-sign-icons
e-save' items={items} select={onSelect}/>
    </div>
    <div id="signature-control">
      <SignatureComponent id='signature' />
    </div>
  </div>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

**INDEX.TSX**

```
import { SignatureComponent } from '@syncfusion/ej2-react-inputs';
import { MenuEventArgs, SplitButton, SplitButtonComponent } from
 '@syncfusion/ej2-react-splitbuttons';
import { getComponent, closest } from '@syncfusion/ej2-base';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  let items: { text: string; }[];
  items = [
    {
```

```

        text: 'Png'
      },
      {
        text: 'Jpeg'
      },
      {
        text: 'Svg'
      }
    ]];
    function onSelect(args: MenuEventArgs ): void {
      let signature: Signature =
        getComponent(document.getElementById('signature'), 'signature');
      signature.save(args.item.text as SignatureFileType, 'Signature');
    }
    return (
      <div id='container'>
        <div>
          <span>Sign here</span>
          <SplitButtonComponent content="Save" iconCss='e-sign-icons
e-save' items={items} select={onSelect} />
        </div>
        <div id="signature-control">
          <SignatureComponent id='signature' />
        </div>
      </div>
    );
  };
  export default App;
  ReactDOM.render(<App />, document.getElementById('element'));
```

### Save with Background

The [saveWithBackground](#) property is used to save the signature with its background and its default value is true. So, by default the signature is saved with its background.

In the following sample, the background color is set as 'rgb(103 58 183)' and save with background as true.

### INDEX.JSX

```

import { SignatureComponent } from '@syncfusion/ej2-react-inputs';
import { SplitButtonComponent } from '@syncfusion/ej2-react-splitbuttons';
import { getComponent } from '@syncfusion/ej2-base';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  let items;
  items = [
    {
      text: 'Png'
    },
    {
      text: 'Jpeg'
    },
    {
      text: 'Svg'
    }
  ];
};
```

```

function onSelect(args) {
    let signature = getComponent(document.getElementById('signature'),
    'signature');
    signature.save(args.item.text, 'Signature');
}
return (<div id='container'>
    <div>
        <span>Sign here</span>
        <SplitButtonComponent content="Save" iconCss='e-sign-icons
e-save' items={items} select={onSelect}/>
    </div>
    <div id="signature-control">
        <SignatureComponent id='signature' backgroundColor="rgb(103
58 183)" saveWithBackground={true}/>
    </div>
</div>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

## INDEX.TSX

```

import { SignatureComponent } from '@syncfusion/ej2-react-inputs';
import { MenuEventArgs, SplitButton, SplitButtonComponent } from
 '@syncfusion/ej2-react-splitbuttons';
import { getComponent, closest } from '@syncfusion/ej2-base';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    let items: { text: string; }[];
    items = [
        {
            text: 'Png'
        },
        {
            text: 'Jpeg'
        },
        {
            text: 'Svg'
        }
    ];
    function onSelect(args: MenuEventArgs ): void {
        let signature: Signature =
        getComponent(document.getElementById('signature'), 'signature');
        signature.save(args.item.text as SignatureFileType, 'Signature');
    }
    return (
        <div id='container'>
            <div>
                <span>Sign here</span>
                <SplitButtonComponent content="Save" iconCss='e-sign-icons
e-save' items={items} select={onSelect} />
            </div>
            <div id="signature-control">
                <SignatureComponent id='signature' backgroundColor="rgb(103
58 183)" saveWithBackground={true}/>
            </div>
        </div>
    );
}

```

```

        </div>
    </div>
    );
};
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

## Draw in React Signature component

### Draw

The [draw](#) method is used to draw a text as signature with different font families like Arial, Serif, with different font sizes. It accepts text, font family, font size as its parameters. The default font family is "Arial", and the default font size is "30".

### INDEX.JSX

```

import { SignatureComponent } from '@syncfusion/ej2-react-inputs';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { getComponent } from '@syncfusion/ej2-base';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const fontData = ['Arial', 'Serif', 'Sans-serif', 'Cursive', 'Fantasy'];
    const sizeData = [20, 30, 40, 50];
    function clickEventHandler() {
        let signature = getComponent(document.getElementById('signature'),
        'signature');
        let text = document.getElementById('text').value;
        let font = document.getElementById("font").value;
        let size = document.getElementById("size").value;
        signature.draw(text, font, size);
    }
    return (
        <div id='container'>
            <div className='signature-control'>
                <SignatureComponent id='signature' />
            </div>
            <div id='input'>
                <table>
                    <tbody>
                        <tr>
                            <td><div>Enter the Text:</div></td>
                            <td><input type="text" id="text"
placeholder="Enter the Text"></input></td>
                        </tr>
                        <tr>
                            <td><div>Font Family:</div></td>
                            <td>
                                <DropDownListComponent id='font'
dataSource={fontData} value='Arial' popupHeight='200px' />
                            </td>
                        </tr>
                        <tr>
                            <td><div>Font Size:</div></td>
                            <td>

```

```

                                <DropDownListComponent id='size'
dataSource={sizeData} value={20} popupHeight='200px' />
                                </td>
                            </tr>
                        </tbody>
                    </table>
                    <ButtonComponent onClick={clickEventHandler}
isPrimary={true}>Draw</ButtonComponent>
                </div>
            </div>);
    }
;
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

## INDEX.TSX

```

import { SignatureComponent } from '@syncfusion/ej2-react-inputs';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { getComponent, closest } from '@syncfusion/ej2-base';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const fontData: string[] = ['Arial', 'Serif', 'Sans-serif', 'Cursive',
'Fantasy'];
    const sizeData: number[] = [20, 30, 40, 50];
    function clickEventHandler(): void {
        let signature: SignatureComponent =
getComponent(document.getElementById('signature'), 'signature');
        let text: string = document.getElementById('text').value;
        let font: string = document.getElementById("font").value;
        let size: number = document.getElementById("size").value;
        signature.draw(text, font, size);
    }
    return (
        <div id='container'>
            <div className='signature-control'>
                <SignatureComponent id='signature' />
            </div>
            <div id='input'>
                <table>
                    <tbody>
                        <tr>
                            <td><div>Enter the Text:</div></td>
                            <td><input type="text" id="text"
placeholder="Enter the Text"></input></td>
                        </tr>
                        <tr>
                            <td><div>Font Family:</div></td>
                            <td>
                                <DropDownListComponent id='font'
dataSource={fontData} value='Arial' popupHeight='200px' />
                            </td>
                        </tr>
                        <tr>

```

```

                <td><div>Font Size:</div></td>
                <td>
                    <DropDownListComponent id='size'
dataSource={sizeData} value={20} popupHeight='200px' />
                </td>
            </tr>
        </tbody>
    </table>
    <ButtonComponent onClick={clickEventHandler}
isPrimary={true}>Draw</ButtonComponent>
    </div>
</div>
);
};
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

### User interaction in React Signature component

The below interactions were available in Signature, and we can walk through one by one.

- Undo and Redo
- Clear
- Disabled
- ReadOnly

#### Undo and Redo

In the Signature, every action can be maintained as a snap for undo and redo operations. And maintained SnapIndex for indexing the snap collection.

The [undo](#) method reverts the last action of signature by decreasing SnapIndex value to index previous snap. Here, [canUndo](#) method is used to ensure whether undo can be performed or not.

The [redo](#) method reverts the last undo action of the signature by increasing the SnapIndex to get the next snap. Here, [canRedo](#) method is used to ensure whether redo can be performed or not.

#### Clear

The [clear](#) method is used to clear the signature and makes the canvas empty. This is also considered in Undo/ Redo. Here, [isEmpty](#) method is used to ensure whether the signature is empty or not.

#### Disabled

The [disabled](#) property is used to enable/disable the signature component. In the disabled state, the user is not allowed to draw signature. And it can't be focused until the user enabled the signature.

#### ReadOnly

The [isReadOnly](#) property is used to enable/disable the ReadOnly Signature. It can be focused but it prevents drawing in Signature.

The following sample explains about user interactions available in signature.

#### INDEX.JSX

```

import * as ReactDOM from 'react-dom';
import * as React from 'react';
import { SignatureComponent } from '@syncfusion/ej2-react-inputs';

```



```
import { ButtonComponent, CheckBoxComponent } from '@syncfusion/ej2-react-buttons';
import { getComponent } from '@syncfusion/ej2-base';
function App() {
    function undoBtnCreated() {
        document.getElementById('undo').addEventListener('click',
undoBtnClick);
    }
    function undoBtnClick() {
        let signature = getComponent(document.getElementById('signature'),
'signature');
        if (!signature.disabled && !signature.isReadOnly) {
            signature.undo();
        }
    }
    function redoBtnCreated() {
        document.getElementById('redo').addEventListener('click',
redoBtnClick);
    }
    function redoBtnClick() {
        let signature = getComponent(document.getElementById('signature'),
'signature');
        if (!signature.disabled && !signature.isReadOnly) {
            signature.redo();
        }
    }
    function clrBtnCreated() {
        document.getElementById('clear').addEventListener('click',
clrBtnClick);
    }
    function clrBtnClick() {
        let signature = getComponent(document.getElementById('signature'),
'signature');
        if (!signature.disabled && !signature.isReadOnly) {
            signature.clear();
        }
    }
    function disableChange(args) {
        let signature = getComponent(document.getElementById('signature'),
'signature');
        signature.disabled = args.checked;
    }
    function readOnlyChange(args) {
        let signature = getComponent(document.getElementById('signature'),
'signature');
        signature.isReadOnly = args.checked;
    }
    function change(args) {
        let signature = getComponent(document.getElementById('signature'),
'signature');
        let undoButton = getComponent(document.getElementById('undo'),
'btn');
        let redoButton = getComponent(document.getElementById('redo'),
'btn');
        let clearButton = getComponent(document.getElementById('clear'),
'btn');
        if (!signature.disabled && !signature.isReadOnly) {
```

```

        if (signature.canUndo()) {
            undoButton.disabled = false;
        }
        else {
            undoButton.disabled = true;
        }
        if (signature.canRedo()) {
            redoButton.disabled = false;
        }
        else {
            redoButton.disabled = true;
        }
        if (!signature.isEmpty()) {
            clearButton.disabled = false;
        }
        else {
            clearButton.disabled = true;
        }
    }
}
return (<div id='container'>
    <div id="option">
        <table>
            <tr>
                <td>
                    <ButtonComponent id='undo' cssClass='e-primary'
content='UNDO' disabled={true} created={undoBtnCreated.bind(this)} />
                </td><td>
                    <ButtonComponent id='redo' cssClass='e-primary'
content='REDO' disabled={true} created={redoBtnCreated.bind(this)} />
                </td><td>
                    <ButtonComponent id='clear' cssClass='e-primary'
content='CLEAR' disabled={true} created={clrBtnCreated.bind(this)} />
                </td><td>
                    <div id="checkbox1">
                        <CheckBoxComponent label="Disabled"
change={disableChange} />
                    </div>
                    <div id="checkbox2"><CheckBoxComponent
label="ReadOnly" change={readOnlyChange} /></div>
                </td>
            </tr>
        </table>
    </div>
    <div id="signature-control">
        <SignatureComponent id='signature' change={change} />
    </div>
</div>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

**INDEX.TSX**

```
import * as ReactDOM from 'react-dom';
```

```
import * as React from 'react';
import { SignatureFileType, Signature, SignatureComponent,
SignatureChangeEventArgs } from '@syncfusion/ej2-react-inputs';
import { ButtonComponent, CheckBoxComponent } from '@syncfusion/ej2-react-
buttons';
import { getComponent, closest } from '@syncfusion/ej2-base';
import { ChangeEventArgs, CheckBox } from '@syncfusion/ej2-buttons';
function App() {
  function undoBtnCreated() {
    document.getElementById('undo').addEventListener('click', undoBtnClick);
  }
  function undoBtnClick() {
    let signature: Signature =
getComponent(document.getElementById('signature'), 'signature');
    if (!signature.disabled && !signature.isReadOnly) {
      signature.undo();
    }
  }
  function redoBtnCreated() {
    document.getElementById('redo').addEventListener('click', redoBtnClick);
  }
  function redoBtnClick() {
    let signature: Signature =
getComponent(document.getElementById('signature'), 'signature');
    if (!signature.disabled && !signature.isReadOnly) {
      signature.redo();
    }
  }
  function clrBtnCreated() {
    document.getElementById('clear').addEventListener('click', clrBtnClick);
  }
  function clrBtnClick() {
    let signature: Signature =
getComponent(document.getElementById('signature'), 'signature');
    if (!signature.disabled && !signature.isReadOnly) {
      signature.clear();
    }
  }
  function disableChange(args: ChangeEventArgs) {
    let signature: Signature =
getComponent(document.getElementById('signature'), 'signature');
    signature.disabled = args.checked;
  }
  function readOnlyChange(args: ChangeEventArgs) {
    let signature: Signature =
getComponent(document.getElementById('signature'), 'signature');
    signature.isReadOnly = args.checked;
  }
  function change(args: SignatureChangeEventArgs) {
    let signature: Signature =
getComponent(document.getElementById('signature'), 'signature');
    let undoButton: Button = getComponent(document.getElementById('undo'),
'btn');
    let redoButton: Button = getComponent(document.getElementById('redo'),
'btn');
    let clearButton: Button = getComponent(document.getElementById('clear'),
'btn');
```

```

    if (!signature.disabled && !signature.isReadOnly) {
      if (signature.canUndo()) {
        undoButton.disabled = false;
      } else {
        undoButton.disabled = true;
      }
      if (signature.canRedo()) {
        redoButton.disabled = false;
      } else {
        redoButton.disabled = true;
      }
      if (!signature.isEmpty()) {
        clearButton.disabled = false;
      } else {
        clearButton.disabled = true;
      }
    }
  }
  return (<div id='container'>
    <div id="option">
      <table>
        <tr>
          <td>
            <ButtonComponent id='undo' cssClass='e-primary'
content='UNDO' disabled={true} created={undoBtnCreated.bind(this)} />
          </td><td>
            <ButtonComponent id='redo' cssClass='e-primary'
content='REDO' disabled={true} created={redoBtnCreated.bind(this)} />
          </td><td>
            <ButtonComponent id='clear' cssClass='e-primary'
content='CLEAR' disabled={true} created={clrBtnCreated.bind(this)} />
          </td><td>
            <div id="checkbox1">
              <CheckBoxComponent label="Disabled"
change={disableChange} />
            </div>
            <div id="checkbox2"><CheckBoxComponent
label="ReadOnly" change={readOnlyChange} /></div>
          </td>
        </tr>
      </table>
    </div>
    <div id="signature-control">
      <SignatureComponent id='signature' change={change} />
    </div>
  </div>
  )
};
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

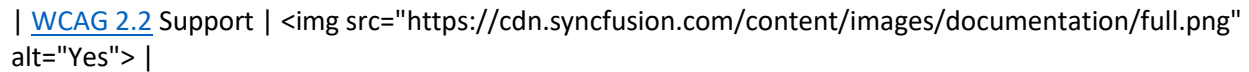
## Accessibility in React Signature component

The Signature component followed the accessibility guidelines and standards, including [ADA, Section 508](#), [WCAG 2.2](#) standards, and [WCAG roles](#) that are commonly used to evaluate accessibility.

The accessibility compliance for the Signature component is outlined below.

| Accessibility Criteria | Compatibility |

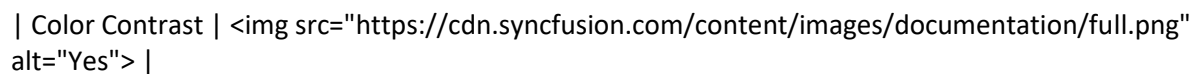
| -- | -- |

| [WCAG 2.2](#) Support |  alt="Yes"> |

| [Section 508](#) Support |  alt="Yes"> |

| Screen Reader Support |  alt="Yes"> |

| Right-To-Left Support | Not Applicable |

| Color Contrast |  alt="Yes"> |

| Mobile Device Support |  alt="Yes"> |

| Keyboard Navigation Support |  alt="Yes"> |

| [Accessibility Checker](#) Validation |  alt="Yes"> |

| [Axe-core](#) Accessibility Validation |  alt="Yes"> |

<style>

.post .post-content img {

display: inline-block;

margin: 0.5em 0;

}

</style>

<div> - All features of the component meet the requirement.</div>

<div> - Some features of the component do not meet the requirement.</div>

<div> - The component does not meet the requirement.</div>

### Keyboard interaction

The Signature component followed the [keyboard interaction] guideline, making it easy for people who use assistive technologies (AT) and those who completely rely on keyboard navigation. The following keyboard shortcuts are supported by the Signature component.

| **Press** | **To do this** |

| --- | --- |

| Ctrl + Z | Undo the last action. |

| Ctrl + Y | Redo the last action. |

| Ctrl + S | To save the signature. |

| delete | Erases all the signature strokes signed by user. |

### Ensuring accessibility

The Signature component's accessibility levels are ensured through an [accessibility-checker](#) and [axe-core](#) software tools during automated testing.

The accessibility compliance of the Signature component is shown in the following sample. Open the [sample](#) in a new window to evaluate the accessibility of the Signature component with accessibility tools.

See also

- [Accessibility in Syncfusion React components](#)

### How To

#### Integration toolbar in React Signature component

The Signature component integrates with the toolbar and the interaction performed using the `change` event of the toolbar.

In that, [canUndo](#), [canRedo](#) and [Link to the Video](#) methods were used to enable/disable undo, redo, and clear buttons.

#### INDEX.JSX

```
import * as ReactDOM from 'react-dom';
import * as React from 'react';
import { ColorPickerComponent } from '@syncfusion/ej2-react-inputs';
import { SignatureComponent } from '@syncfusion/ej2-react-inputs';
import { getComponent, createElement, addClass } from '@syncfusion/ej2-base';
import { ItemDirective, ItemsDirective, ToolbarComponent } from '@syncfusion/ej2-react-navigations';
import { CheckBox } from '@syncfusion/ej2-buttons';
import { SplitButtonComponent } from '@syncfusion/ej2-react-splitbuttons';
import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
function App() {
    let signature;
    let disabledTemplate = new CheckBox({ label: 'Disabled', checked: false, change: change });
    function change(args) {
        let signature = getComponent(document.getElementById('signature'), 'signature');
        signature.disabled = args.checked;
    }
    function onCreated() {
        let strokeColor = getComponent(document.getElementById('stroke-color'), 'colorpicker');
    }
}
```

```

        let bgColor = getComponent(document.getElementById('bg-color'),
'colorpicker');
        addClass([strokeColor.element.nextElementSibling.querySelector('.e-
selected-color')], 'e-sign-icons');
        addClass([bgColor.element.nextElementSibling.querySelector('.e-
selected-color')], 'e-sign-icons');
        document.getElementById('save-option').addEventListener('click',
saveBtnClick);
        clearButton();
        let toolbarItems = document.querySelectorAll('.e-toolbar .e-
toolbar-items .e-toolbar-item .e-tbar-btn.e-tbtn-txt');
        for (var i = 0; i < toolbarItems.length; i++) {
            if (toolbarItems[i].children[0].classList.contains('e-undo')) {
                let undoButton = getComponent(toolbarItems[i], 'btn');
                undoButton.disabled = true;
            }
            if (toolbarItems[i].children[0].classList.contains('e-redo')) {
                let redoButton = getComponent(toolbarItems[i], 'btn');
                redoButton.disabled = true;
            }
        }
    }
    function onClicked(args) {
        let signature = getComponent(document.getElementById('signature'),
'signature');
        let saveBtn = getComponent(document.getElementById("save-option"),
'split-btn');
        if (signature.disabled && args.item.tooltipText != 'Disabled') {
            return;
        }
        switch (args.item.tooltipText) {
            case 'Undo (Ctrl + Z)':
                if (signature.canUndo()) {
                    signature.undo();
                    updateUndoRedo();
                    updateSaveBtn();
                }
                break;
            case 'Redo (Ctrl + Y)':
                if (signature.canRedo()) {
                    signature.redo();
                    updateUndoRedo();
                    updateSaveBtn();
                }
                break;
            case 'Clear':
                signature.clear();
                if (signature.isEmpty()) {
                    clearButton();
                    saveBtn.disabled = true;
                }
                break;
        }
    }
    function onChange() {
        let signature = getComponent(document.getElementById('signature'),
'signature');

```

```

        let saveBtn = getComponent(document.getElementById("save-option"),
'split-btn');
        if (!signature.isEmpty()) {
            clearButton();
            saveBtn.disabled = false;
        }
        updateUndoRedo();
    }
    function saveBtnClick() {
        let signature = getComponent(document.getElementById("signature"),
'signature');
        signature.save();
    }
    function clearButton() {
        let signature = getComponent(document.getElementById('signature'),
'signature');
        let tlItems = document.querySelectorAll('.e-toolbar .e-toolbar-items
.e-toolbar-item .e-tbar-btn.e-tbtn-txt');
        for (var i = 0; i < tlItems.length; i++) {
            if (tlItems[i].children[0].classList.contains('e-clear')) {
                let clrBtn = getComponent(tlItems[i], 'btn');
                if (signature.isEmpty()) {
                    clrBtn.disabled = true;
                }
                else {
                    clrBtn.disabled = false;
                }
            }
        }
    }
    function updateSaveBtn() {
        let signature = getComponent(document.getElementById('signature'),
'signature');
        let saveBtn = getComponent(document.getElementById("save-option"),
'split-btn');
        if (signature.isEmpty()) {
            saveBtn.disabled = true;
        }
        else {
            saveBtn.disabled = false;
        }
    }
    function updateUndoRedo() {
        let signature = getComponent(document.getElementById('signature'),
'signature');
        let undoButton;
        let redoButton;
        let tlItems = document.querySelectorAll('.e-toolbar .e-toolbar-items
.e-toolbar-item .e-tbar-btn.e-tbtn-txt');
        for (var i = 0; i < tlItems.length; i++) {
            if (tlItems[i].children[0].classList.contains('e-undo')) {
                undoButton = getComponent(tlItems[i], 'btn');
            }
            if (tlItems[i].children[0].classList.contains('e-redo')) {
                redoButton = getComponent(tlItems[i], 'btn');
            }
        }
    }
}

```



```

        if (signature.canUndo()) {
            undoButton.disabled = false;
        }
        else {
            undoButton.disabled = true;
        }
        if (signature.canRedo()) {
            redoButton.disabled = false;
        }
        else {
            redoButton.disabled = true;
        }
    }
    function saveTemplate() {
        let items;
        items = [
            {
                text: 'Png'
            },
            {
                text: 'Jpeg'
            },
            {
                text: 'Svg'
            }
        ];
        function onSelect(args) {
            let signature =
            getComponent(document.getElementById('signature'), 'signature');
            signature.save(args.item.text, 'Signature');
        }
        return (<div>
            <SplitButtonComponent content="Save" id="save-option"
items={items} iconCss='e-sign-icons e-save' select={onSelect}
disabled={true}/>
            </div>);
    }
    function strokeColorTemplate() {
        let presets;
        presets = {
            'custom': ['#000000', '#e91e63', '#9c27b0', '#673ab7',
'#2196f3', '#03a9f4', '#00bcd4',
'#009688', '#8bc34a', '#cddc39', '#ffeb3b', '#ffc107']
        };
        function tileRender(args) {
            args.element.classList.add('e-circle-palette');
            args.element.appendChild(createElement('span', { className: 'e-
circle-selection' }));
        }
        function strokeColorChanged(args) {
            let signature =
            getComponent(document.getElementById('signature'), 'signature');
            let strokeColor = getComponent(document.getElementById('stroke-
color'), 'colorpicker');
            if (signature.disabled) {
                return;
            }
        }
    }

```

```

        let selElem =
strokeColor.element.nextElementSibling.querySelector('.e-selected-color');
        selElem.style.borderBottomColor = args.currentValue.rgba;
        signature.strokeColor = args.currentValue.rgba;
    }
    return (<div>
        <ColorPickerComponent id="stroke-color" mode='Palette'
cssClass='e-stroke-color' modeSwitcher={false} showButtons={false}
columns={4} presetColors={presets} beforeTileRender={tileRender}
change={strokeColorChanged}></ColorPickerComponent>
    </div>);
    }
    function bgColorTemplate() {
        let presets;
        presets = {
            'custom': ['#ffffff', '#f44336', '#e91e63', '#9c27b0',
'#673ab7', '#2196f3', '#03a9f4', '#00bcd4',
            '#009688', '#8bc34a', '#cddc39', '#ffeb3b']
        };
        function beforeTileRender(args) {
            args.element.classList.add('e-circle-palette');
            args.element.appendChild(createElement('span', { className: 'e-
circle-selection' }));
        }
        function bgColorChanged(args) {
            let signature =
getComponent(document.getElementById('signature'), 'signature');
            let bgColor = getComponent(document.getElementById('bg-color'),
'colorpicker');
            if (signature.disabled) {
                return;
            }
            let selElem =
bgColor.element.nextElementSibling.querySelector('.e-selected-color');
            signature.backgroundColor = args.currentValue.rgba;
            selElem.style.borderBottomColor = args.currentValue.rgba;
        }
        return (<div>
            <ColorPickerComponent id="bg-color" noColor={true} mode='Palette'
cssClass='e-bg-color' modeSwitcher={false} showButtons={false} columns={4}
presetColors={presets} beforeTileRender={beforeTileRender}
change={bgColorChanged}></ColorPickerComponent>
        </div>);
    }
    function strokeWidthTemplate() {
        let data = [1, 2, 3, 4, 5];
        let value = 2;
        function strokeWidthChanged(args) {
            let signature =
getComponent(document.getElementById('signature'), 'signature');
            signature.maxStrokeWidth = args.value;
        }
        return (<div>
            <DropDownListComponent id="ddlelement" dataSource={data}
value={value} width="60" change={strokeWidthChanged}/>
        </div>);
    }
}

```

```

    return (<div className='control-pane'>
      <div className="col-lg-12 control-section">
        <div id="signature-toolbar-control">
          <ToolbarComponent id='toolbar' created={onCreated}
            clicked={onClicked}>
            <ItemsDirective>
              <ItemDirective text='Undo' prefixIcon='e-icons e-undo' tooltipText='Undo (Ctrl + Z)' />
              <ItemDirective text='Redo' prefixIcon='e-icons e-redo' tooltipText='Redo (Ctrl + Y)' />
              <ItemDirective type='Separator' />
              <ItemDirective tooltipText='Save (Ctrl + S)' type='Button' template={saveTemplate} />
              <ItemDirective type='Separator' />
              <ItemDirective tooltipText='Stroke Color' template={strokeColorTemplate} />
              <ItemDirective type='Separator' />
              <ItemDirective tooltipText='Background Color' template={bgColorTemplate} />
              <ItemDirective type='Separator' />
              <ItemDirective tooltipText='Stroke Width' template={strokeWidthTemplate} />
              <ItemDirective type='Separator' />
              <ItemDirective text='Clear' prefixIcon='e-sign-icons e-clear' tooltipText='Clear' />
              <ItemDirective tooltipText='Disabled' type='Input' template={disabledTemplate} align='Right' />
            </ItemsDirective>
          </ToolbarComponent>
          <div id="signature-control">
            <SignatureComponent maxStrokeWidth={2} id="signature" change={onChange}></SignatureComponent>
          </div>
        </div>
      </div>
    </div>);
  }
  export default App;
  ReactDOM.render(<App />, document.getElementById('element'));

```

## INDEX.TSX

```

import * as ReactDOM from 'react-dom';
import * as React from 'react';
import { ColorPicker, ColorPickerComponent, ColorPickerEventArgs, PaletteTileEventArgs } from '@syncfusion/ej2-react-inputs';
import { SignatureFileType, Signature, SignatureComponent } from '@syncfusion/ej2-react-inputs';
import { Button } from '@syncfusion/ej2-react-buttons';
import { GetComponent, createElement, addClass } from '@syncfusion/ej2-base';
import { ClickEventArgs, ItemDirective, ItemsDirective, ToolbarComponent } from '@syncfusion/ej2-react-navigations';
import { ChangeEventArgs, CheckBox } from '@syncfusion/ej2-buttons';
import { MenuEventArgs, SplitButton, SplitButtonComponent } from '@syncfusion/ej2-react-splitbuttons';

```

```

import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
function App() {
    let signature: SignatureComponent;
    let disabledTemplate: CheckBox = new CheckBox({ label: 'Disabled',
checked: false, change: change });
    function change(args: ChangeEventArgs): void {
        let signature: Signature =
getComponent(document.getElementById('signature'), 'signature');
        signature.disabled = args.checked;
    }
    function onCreated(): void {
        let strokeColor: ColorPicker =
getComponent(document.getElementById('stroke-color'), 'colorpicker');
        let bgColor: ColorPicker = getComponent(document.getElementById('bg-
color'), 'colorpicker');
        addClass([strokeColor.element.nextElementSibling.querySelector('.e-
selected-color')], 'e-sign-icons');
        addClass([bgColor.element.nextElementSibling.querySelector('.e-
selected-color')], 'e-sign-icons');
        document.getElementById('save-option').addEventListener('click',
saveBtnClick);
        clearButton();
        let toolbarItems: NodeListOf<Element> =
document.querySelectorAll('.e-toolbar .e-toolbar-items .e-toolbar-item .e-
tbar-btn.e-tbtn-txt');
        for (var i = 0; i < toolbarItems.length; i++) {
            if (toolbarItems[i].children[0].classList.contains('e-undo')) {
                let undoButton: Button = getComponent(toolbarItems[i] as
HTMLElement, 'btn');
                undoButton.disabled = true;
            }
            if (toolbarItems[i].children[0].classList.contains('e-redo')) {
                let redoButton: Button = getComponent(toolbarItems[i] as
HTMLElement, 'btn');
                redoButton.disabled = true;
            }
        }
    }
    function onClicked(args: ClickEventArgs): void {
        let signature: Signature =
getComponent(document.getElementById('signature'), 'signature');
        let saveBtn: SplitButton = getComponent(document.getElementById("save-
option"), 'split-btn');
        if (signature.disabled && args.item.tooltipText != 'Disabled') {
            return;
        }
        switch (args.item.tooltipText) {
            case 'Undo (Ctrl + Z)':
                if (signature.canUndo()) {
                    signature.undo();
                    updateUndoRedo();
                    updateSaveBtn();
                }
                break;
            case 'Redo (Ctrl + Y)':
                if (signature.canRedo()) {
                    signature.redo();
                }
        }
    }
}

```

```

        updateUndoRedo();
        updateSaveBtn();
    }
    break;
case 'Clear':
    signature.clear();
    if (signature.isEmpty()) {
        clearButton();
        saveBtn.disabled = true;
    }
    break;
}
}
function onChange() {
    let signature: Signature =
getComponent(document.getElementById('signature'), 'signature');
    let saveBtn: SplitButton = getComponent(document.getElementById("save-
option"), 'split-btn');
    if (!signature.isEmpty()) {
        clearButton();
        saveBtn.disabled = false;
    }
    updateUndoRedo();
}
function saveBtnClick(): void {
    let signature: Signature =
getComponent(document.getElementById("signature"), 'signature');
    signature.save();
}
function clearButton() {
    let signature: Signature =
getComponent(document.getElementById('signature'), 'signature');
    let tlItems: NodeListOf<Element> = document.querySelectorAll('.e-
toolbar .e-toolbar-items .e-toolbar-item .e-tbar-btn.e-tbtn-txt');
    for (var i = 0; i < tlItems.length; i++) {
        if (tlItems[i].children[0].classList.contains('e-clear')) {
            let clrBtn: Button = getComponent(tlItems[i] as HTMLElement,
'btn');
            if (signature.isEmpty()) {
                clrBtn.disabled = true;
            } else {
                clrBtn.disabled = false;
            }
        }
    }
}
function updateSaveBtn() {
    let signature: Signature =
getComponent(document.getElementById('signature'), 'signature');
    let saveBtn: SplitButton = getComponent(document.getElementById("save-
option"), 'split-btn');
    if (signature.isEmpty()) {
        saveBtn.disabled = true;
    } else {
        saveBtn.disabled = false;
    }
}
}

```

```

function updateUndoRedo() {
    let signature: Signature =
    getComponent(document.getElementById('signature'), 'signature');
    let undoButton: Button; let redoButton: Button
    let tlItems: NodeListOf<Element> = document.querySelectorAll('.e-
toolbar .e-toolbar-items .e-toolbar-item .e-tbar-btn.e-tbtn-txt');
    for (var i = 0; i < tlItems.length; i++) {
        if (tlItems[i].children[0].classList.contains('e-undo')) {
            undoButton = getComponent(tlItems[i] as HTMLElement, 'btn');
        }
        if (tlItems[i].children[0].classList.contains('e-redo')) {
            redoButton = getComponent(tlItems[i] as HTMLElement, 'btn');
        }
    }
    if (signature.canUndo()) {
        undoButton.disabled = false;
    } else {
        undoButton.disabled = true;
    }
    if (signature.canRedo()) {
        redoButton.disabled = false;
    } else {
        redoButton.disabled = true;
    }
}
function saveTemplate() {
    let items: { text: string; }[];
    items = [
        {
            text: 'Png'
        },
        {
            text: 'Jpeg'
        },
        {
            text: 'Svg'
        }
    ];
    function onSelect(args: MenuEventArgs): void {
        let signature: Signature =
        getComponent(document.getElementById('signature'), 'signature');
        signature.save(args.item.text as SignatureFileType, 'Signature');
    }
    return (<div >
        <SplitButtonComponent content="Save" id="save-option"
items={items} iconCss='e-sign-icons e-save' select={onSelect}
disabled={true} />
        </div>);
}
function strokeColorTemplate() {
    let presets: any;
    presets = {
        'custom': ['#000000', '#e91e63', '#9c27b0', '#673ab7', '#2196f3',
'#03a9f4', '#00bcd4',
        '#009688', '#8bc34a', '#cddc39', '#ffeb3b', '#ffc107']
    };
    function tileRender(args: PaletteTileEventArgs): void {

```

```

        args.element.classList.add('e-circle-palette');
        args.element.appendChild(createElement('span', { className: 'e-
circle-selection' }));
    }
    function strokeColorChanged(args: ColorPickerEventArgs): void {
        let signature: Signature =
getComponent(document.getElementById('signature'), 'signature');
        let strokeColor: ColorPicker =
getComponent(document.getElementById('stroke-color'), 'colorpicker');
        if (signature.disabled) {
            return;
        }
        let selElem: HTMLElement =
strokeColor.element.nextElementSibling.querySelector('.e-selected-color') as
HTMLElement;
        selElem.style.borderBottomColor = args.currentValue.rgba;
        signature.strokeColor = args.currentValue.rgba;
    }
    return (<div >
        <ColorPickerComponent id="stroke-color" mode='Palette'
cssClass='e-stroke-color' modeSwitcher={false} showButtons={false}
columns={4} presetColors={presets} beforeTileRender={tileRender}
change={strokeColorChanged}></ColorPickerComponent>
    </div>);
}

function bgColorTemplate() {
    let presets: any;
    presets = {
        'custom': ['#ffffff', '#f44336', '#e91e63', '#9c27b0', '#673ab7',
'#2196f3', '#03a9f4', '#00bcd4',
        '#009688', '#8bc34a', '#cddc39', '#ffeb3b']
    };
    function beforeTileRender(args: PaletteTileEventArgs): void {
        args.element.classList.add('e-circle-palette');
        args.element.appendChild(createElement('span', { className: 'e-
circle-selection' }));
    }
    function bgColorChanged(args: ColorPickerEventArgs): void {
        let signature: Signature =
getComponent(document.getElementById('signature'), 'signature');
        let bgColor: ColorPicker =
getComponent(document.getElementById('bg-color'), 'colorpicker');
        if (signature.disabled) {
            return;
        }
        let selElem: HTMLElement =
bgColor.element.nextElementSibling.querySelector('.e-selected-color') as
HTMLElement;
        signature.backgroundColor = args.currentValue.rgba;
        selElem.style.borderBottomColor = args.currentValue.rgba;
    }
    return (<div >
        <ColorPickerComponent id="bg-color" noColor={true} mode='Palette'
cssClass='e-bg-color' modeSwitcher={false} showButtons={false} columns={4}
presetColors={presets} beforeTileRender={beforeTileRender}
change={bgColorChanged}></ColorPickerComponent>

```

```

    </div>);
  }
  function strokeWidthTemplate() {
    let data: any = [1, 2, 3, 4, 5];
    let value: any = 2;
    function strokeWidthChanged(args: any): void {
      let signature: Signature =
        getComponent(document.getElementById('signature'), 'signature');
      signature.maxStrokeWidth = args.value;
    }
    return (<div >
      <DropDownListComponent id="ddlelement" dataSource={data}
value={value} width="60" change={strokeWidthChanged} />
    </div>);
  }
  return (
    <div className='control-pane'>
      <div className="col-lg-12 control-section">
        <div id="signature-toolbar-control">
          <ToolbarComponent id='toolbar' created={onCreated}
clicked={onClicked}>
            <ItemsDirective>
              <ItemDirective text='Undo' prefixIcon='e-icons e-
undo' tooltipText='Undo (Ctrl + Z)' />
              <ItemDirective text='Redo' prefixIcon='e-icons e-
redo' tooltipText='Redo (Ctrl + Y)' />
              <ItemDirective type='Separator' />
              <ItemDirective tooltipText='Save (Ctrl + S)'
type='Button' template={saveTemplate} />
              <ItemDirective type='Separator' />
              <ItemDirective tooltipText='Stroke Color'
template={strokeColorTemplate} />
              <ItemDirective type='Separator' />
              <ItemDirective tooltipText='Background Color'
template={bgColorTemplate} />
              <ItemDirective type='Separator' />
              <ItemDirective tooltipText='Stroke Width'
template={strokeWidthTemplate} />
              <ItemDirective type='Separator' />
              <ItemDirective text='Clear' prefixIcon='e-sign-
icons e-clear' tooltipText='Clear' />
              <ItemDirective tooltipText='Disabled' type='Input'
template={disabledTemplate} align='Right' />
            </ItemsDirective>
          </ToolbarComponent>
          <div id="signature-control">
            <SignatureComponent maxStrokeWidth={2} id="signature"
change={onChange}></SignatureComponent>
          </div>
        </div>
      </div>
    </div>
  )
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```



## Skeleton

### Getting Started

This section explains how to create a simple Skeleton, and configure its available functionalities in React.

To get started quickly with the React Skeleton component, you can check out this video.

### Dependencies

The following list of dependencies are required to use the Skeleton component in your application.

```
`js
|-- @syncfusion/ej2-react-notifications
|-- @syncfusion/ej2-react-base
|-- @syncfusion/ej2-notifications
|-- @syncfusion/ej2-base
`
```

### Setup your development environment

You can use [Create-react-app](#) to setup the applications.

To install `create-react-app` run the following command.

```
`bash
npm install -g create-react-app
`
```

Start a new project using create-react-app command as follows

```
`bash
create-react-app quickstart --scripts-version=react-scripts-ts
cd quickstart
`
```

```
`bash
create-react-app quickstart
cd quickstart
`
```

'react-scripts-ts' is used for creating React app with typescript.

### Adding Syncfusion packages

All the available Essential JS 2 packages are published in [npmjs.com](https://www.npmjs.com) public registry.

To install `Skeleton` component, use the following command

```
`bash
npm install @syncfusion/ej2-react-notifications --save
`
```

The above command installs [Skeleton dependencies](#) which are required to render the component in the **React** environment.

#### Adding style sheet to the Application

Add Skeleton component's styles as given below in **App.css**.

```
`css
@import "../node_modules/@syncfusion/ej2-base/styles/material.css";
@import "../node_modules/@syncfusion/ej2-notifications/styles/material.css";
`
```

#### Add Skeleton to the Application

Now, you can create **Skeleton** component in the application. Add **Skeleton** component in **src/App.tsx** file using the following code snippet.

```
`ts
import { SkeletonComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import * as ReactDOM from 'react-dom';
function App() {
  return (
    <SkeletonComponent height='15px'></SkeletonComponent>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
`
```

#### Run the application

Run the application in the browser using the following command:

```
`
npm start
`
```

The following example shows a basic Skeleton component.

#### **INDEX.JSX**

```
import { SkeletonComponent } from '@syncfusion/ej2-react-notifications';
```

```
import * as React from "react";
import * as ReactDOM from 'react-dom';
function App() {
    return (<SkeletonComponent height='15px'></SkeletonComponent>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

**INDEX.TSX**

```
import { SkeletonComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import * as ReactDOM from 'react-dom';
function App() {
    return (
        <SkeletonComponent height='15px'></SkeletonComponent>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

[Shapes in React Skeleton component](#)

The Skeleton control support various built-in shape variants to design layout of the page. You can use the [shape](#) property to create a preview of any layout.

The Skeleton component supports the following content shapes:

[Circle skeleton shape](#)

```
`ts
```

```
import { SkeletonComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
function App() {
    return (
        <SkeletonComponent shape= 'Circle' width= "48px"></SkeletonComponent>
    );
}
export default App;
```

[Square skeleton shape](#)

```
`ts
```

```
import { SkeletonComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
function App() {
    return (
        <SkeletonComponent shape= 'Square' width= "48px"></SkeletonComponent>);
```

```

}
export default App;
`ts

Rectangle skeleton shape
`ts
import { SkeletonComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
function App() {
return (
<SkeletonComponent shape= 'Rectangle' width= "50px" height='32px'></SkeletonComponent>);
}
export default App;
`ts

```

#### Text skeleton shape

```

`ts
import { SkeletonComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
function App() {
return (
<SkeletonComponent shape= 'Text' width= "50%" height="15px"></SkeletonComponent>);
}
export default App;
`ts

```

Below example demonstrates the above functionalities of a Skeleton component.

#### INDEX.JSX

```

import { SkeletonComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import * as ReactDOM from 'react-dom';
function App() {
return (
<div id="skeletonCard">
<div className='cardProfile'>
<SkeletonComponent id='cardProfile' shape='Circle'
width="60px"></SkeletonComponent>
</div>
<div className="cardinfo">
<SkeletonComponent id="text1" width="30%"
height='15px'></SkeletonComponent><br />
<SkeletonComponent id="text2" width="15%"
height='15px'></SkeletonComponent>

```

```

        </div>
        <div className="cardContent">
            <SkeletonComponent id="cardImage" shape="Rectangle"
width="100%" height='150px'></SkeletonComponent>
        </div>
        <div className="cardoptions">
            <SkeletonComponent id="rightOption" shape="Rectangle"
width="20%" height='32px'></SkeletonComponent>
            <SkeletonComponent id="leftOption" shape="Rectangle"
width="20%" height='32px'></SkeletonComponent>
        </div>
    </div>
</div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

### INDEX.TSX

```

import { SkeletonComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import * as ReactDOM from 'react-dom';
function App() {
    return (
        <div>
            <div id="skeletonCard">
                <div className='cardProfile'>
                    <SkeletonComponent id='cardProfile' shape= 'Circle' width=
"60px"></SkeletonComponent>
                </div>
                <div className="cardinfo">
                    <SkeletonComponent id="text1" width="30%"
height='15px'></SkeletonComponent><br/>
                    <SkeletonComponent id="text2" width="15%"
height='15px'></SkeletonComponent>
                </div>
                <div className="cardContent">
                    <SkeletonComponent id="cardImage" shape="Rectangle"
width="100%" height='150px'></SkeletonComponent>
                </div>
                <div className="cardoptions">
                    <SkeletonComponent id="rightOption" shape="Rectangle"
width="20%" height='32px'></SkeletonComponent>
                    <SkeletonComponent id="leftOption" shape="Rectangle"
width="20%" height='32px'></SkeletonComponent>
                </div>
            </div>
        </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

### INDEX.CSS

```

/* Represents the styles for loader */
#loader {

```

```

        color: #008cff;
        height: 40px;
        left: 45%;
        position: absolute;
        top: 45%;
        width: 30%;
    }
    #skeletonCard {
        padding: 10px;
        line-height: inherit;
        height: 330px;
    }
    #skeletonCard .cardProfile {
        float: left;
        margin-right: 15px;
    }
    #skeletonCard .cardinfo {
        margin-top: 10px;
        overflow: hidden;
    }
    #skeletonCard .cardContent {
        margin: 20px 0px 20px;
    }
    #skeletonCard .cardoptions {
        display: flex;
        justify-content: space-between;
    }
}

```

### Shimmer effect in React Skeleton component

You can use the [shimmerEffect](#) property to change animation effect in the skeleton control. Skeleton supports **Wave**, **Pulse** and **Fade** effects and by default, the **shimmerEffect** is set to **Wave** effect.

`ts

```

import { SkeletonComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
function App() {
    return (
        <SkeletonComponent shape= 'Circle' width= "60px" shimmerEffect= 'Pulse'></SkeletonComponent>;
    )
}
export default App;
`

```

Below example demonstrates a list with pulse effect skeleton.

#### INDEX.JSX

```

import { SkeletonComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import * as ReactDOM from 'react-dom';
function App() {
    return <div>

```

```

        <ul id="skeleton-list" className="e-card">
          <li>
            <div className='cardProfile'>
              <SkeletonComponent shape="Circle" width="40px"
shimmerEffect='Pulse'></SkeletonComponent>
            </div>
            <div>
              <SkeletonComponent width="60%" height='15px'
shimmerEffect='Pulse'></SkeletonComponent><br></br>
              <SkeletonComponent width="40%" height='15px'
shimmerEffect='Pulse'></SkeletonComponent>
            </div>
          </li>
          <li>
            <div className='cardProfile'>
              <SkeletonComponent shape="Circle" width="40px"
shimmerEffect='Pulse'></SkeletonComponent>
            </div>
            <div>
              <SkeletonComponent width="60%" height='15px'
shimmerEffect='Pulse'></SkeletonComponent><br></br>
              <SkeletonComponent width="40%" height='15px'
shimmerEffect='Pulse'></SkeletonComponent>
            </div>
          </li>
        </ul>
      </div>);
    }
    export default App;
    ReactDOM.render(<App />, document.getElementById('element'));

```

## INDEX.TSX

```

import { SkeletonComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import * as ReactDOM from 'react-dom';
function App() {
  return (
    <div>
      <ul id="skeleton-list" className="e-card">
        <li>
          <div className='cardProfile'>
            <SkeletonComponent shape="Circle" width="40px"
shimmerEffect='Pulse'></SkeletonComponent>
          </div>
          <div>
            <SkeletonComponent width="60%" height='15px'
shimmerEffect='Pulse'></SkeletonComponent><br></br>
            <SkeletonComponent width="40%" height='15px'
shimmerEffect='Pulse'></SkeletonComponent>
          </div>
        </li>
        <li>
          <div className='cardProfile'>
            <SkeletonComponent shape="Circle" width="40px"
shimmerEffect='Pulse'></SkeletonComponent>

```

```

        </div>
        <div>
          <SkeletonComponent width="60%" height='15px'
shimmerEffect='Pulse'></SkeletonComponent><br></br>
          <SkeletonComponent width="40%" height='15px'
shimmerEffect='Pulse'></SkeletonComponent>
        </div>
      </li>
    </ul>
  </div>
);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

### INDEX.CSS

```

/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
#skeleton-list {
  padding-left: 12px;
  padding-top: 7px;
  line-height: inherit;
}
#skeleton-list li {
  list-style: none;
  display: flow-root;
  margin-bottom: 9px;
}
.cardProfile {
  float: left;
  margin-right: 15px;
}

```

### Styles in React Skeleton component

You can customize skeleton control in the below ways.

#### cssClass

You can customize the style of a Skeleton control by using [cssClass](#). The appearance of React Skeleton can be customized by changing the wave color, background color, width, and height. For detailed information, refer [index.css](#) file below.

### INDEX.JSX

```

import { SkeletonComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import * as ReactDOM from 'react-dom';
function App() {

```



```

    return (<SkeletonComponent shape='Circle' width="60px" cssClass='e-
customize'></SkeletonComponent>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

### INDEX.TSX

```

import { SkeletonComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import * as ReactDOM from 'react-dom';
function App() {
    return (
        <SkeletonComponent shape= 'Circle' width= "60px" cssClass='e-
customize'></SkeletonComponent>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

### INDEX.CSS

```

/* Represents the styles for loader */
#loader {
    color: #008cff;
    height: 40px;
    left: 45%;
    position: absolute;
    top: 45%;
    width: 30%;
}
/* Represents the styles for shimmer-wave */
.e-customize.e-skeleton.e-shimmer-wave::after{
    background-image: linear-gradient(90deg, transparent calc(50% - 100px),
    rgb(30 128 234 / 50%) 50%,
    transparent calc(50% + 100px));
}
/* Represents the background-color for skeleton-circle */
.e-customize.e-skeleton.e-skeleton-circle{
    background-color: #a8c1f2;
}

```

### Visible

You can use the [visible](#) property which defines the visible state of Skeleton.

`ts

```

import { SkeletonComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
function App() {
    return (
        <SkeletonComponent shape= 'Circle' width= "60px" visible={false}></SkeletonComponent>);
}

```

```

}
export default App;
`

```

### Accessibility in React Skeleton component

The Skeleton component followed the accessibility guidelines and standards, including [ADA, Section 508](#), [WCAG 2.2](#) standards, and [WCAG roles](#) that are commonly used to evaluate accessibility.

The accessibility compliance for the Skeleton component is outlined below.

Accessibility Criteria	Compatibility
<a href="#">WCAG 2.2</a> Support 	
<a href="#">Section 508</a> Support 	
Screen Reader Support 	
Right-To-Left Support 	
Color Contrast 	
Mobile Device Support 	
Keyboard Navigation Support 	
<a href="#">Accessibility Checker</a> Validation 	
<a href="#">Axe-core</a> Accessibility Validation 	

```

<style>
.post .post-content img {
display: inline-block;
margin: 0.5em 0;
}
</style>

<div> - All
features of the component meet the requirement.</div>

<div> - Some features of the component do not meet the requirement.</div>

```

```
<div> - The component does not meet the requirement.</div>
```

### WAI-ARIA attributes

The Skeleton component followed the [WAI-ARIA](#) patterns to meet the accessibility. The following ARIA attributes are used in the Skeleton component:

- | Properties | Functionality |
- | ----- | ----- |
- | **role=alert** | Used to convey important, time-sensitive or contextual message to the user. |
- | **aria-label** | Attribute provides the text label for the Skeleton. |
- | **aria-live** | Attributes which indicates the content changes which are not interactable are live regions. |
- | **aria-busy** | Set to true until loading is complete, then set to false. |

### Ensuring accessibility

The Skeleton component's accessibility levels are ensured through an [accessibility-checker](#) and [axe-core](#) software tools during automated testing.

The accessibility compliance of the Skeleton component is shown in the following sample. Open the [sample](#) in a new window to evaluate the accessibility of the Skeleton component with accessibility tools.

See also

- [Accessibility in Syncfusion React componentsLink to the Video](#)

## Smithchart

### Getting Started

This section explains you the steps required to create a simple Smith Chart and demonstrate the basic usage of the Smith Chart control.

To get start quickly with React Smith Chart, you can check on this video:

### Dependencies

Below is the list of minimum dependencies required to use the Smith Chart.

```
`ts
|-- @syncfusion/ej2-react-charts
|-- @syncfusion/ej2-charts
|-- @syncfusion/ej2-base
|-- @syncfusion/ej2-data
|-- @syncfusion/ej2-svg-base
|-- @syncfusion/ej2-pdf-export
|-- @syncfusion/ej2-compression
|-- @syncfusion/ej2-file-utils
```

```
|-- @syncfusion/ej2-react-base  
,
```

### Installation and configuration

You can use [create-react-app](#) to setup the applications.

To install `create-react-app` run the following command.

```
,
```

```
npm install -g create-react-app
```

```
,
```

- To set-up a React application in TypeScript environment, run the following command.

```
,
```

```
create-react-app quickstart --template typescript
```

```
cd quickstart
```

```
npm start
```

```
,
```

- To set-up a React application in JavaScript environment, run the following command.

```
,
```

```
create-react-app quickstart
```

```
cd quickstart
```

```
npm start
```

```
,
```

- Install Syncfusion packages using below command.

```
,
```

```
npm install @syncfusion/ej2-react-charts --save
```

```
,
```

### Add Smith Chart to the Project

Now, you can start adding Smith Chart component in the application. For getting started, add the Smith Chart component in `src/App.tsx` file using following code.

```
`ts
```

```
import * as React from "react";
```

```
import { SmithchartComponent } from '@syncfusion/ej2-react-charts';
```

```
function App() {
```

```
return ( <SmithchartComponent></SmithchartComponent> );
}
export default App;
```

Now use the `npm start` command to run the application in the browser.

```
npm start
```

The below example shows basic Smithchart.

### INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent } from '@syncfusion/ej2-react-charts';
function App() {
    return (<SmithchartComponent id="smithchart"></SmithchartComponent>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));
```

### INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent } from '@syncfusion/ej2-react-charts';
function App() {
    return ( <SmithchartComponent id="smithchart"></SmithchartComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));
```

## Module Injection

Smithchart component are segregated into individual feature-wise modules. In order to use a particular feature,

you need to inject its feature module using `<Inject services={} />` method. In the current application, we are

going to modify the above basic smithchart to visualize transmission lines.

For this application we are going to use tooltip and legend feature of the smithchart. Please find relevant

feature module name and description as follows.

- SmithchartLegend - Inject this provider to use legend feature.
- TooltipRender - Inject this provider to use tooltip feature.

Now import the above mentioned modules from smithchart package and inject it into the Smithchart component using

`<Inject services={} />` method.

```
`ts
import * as React from "react";
import { SmithchartComponent } from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SmithchartComponent>
    <Inject services={[SmithchartLegend, TooltipRender]} />
  </SmithchartComponent> );
}
export default App;
`
```

### Add Series to Smithchart

Smithchart had two type of specification for adding series.

- **dataSource** - Using this, Data object can bind directly by specifying the resistance and reactance values, series add to smithchart.
- **points** - Using this, collection of resistance and reactance values can bind directly to render series.

Below sample demonstrate adding two series to smithchart both ways.

- First series Transmission1 shows dataSource bound series.
- Second series Transmission2 shows points bound series.

### INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
  return (<SmithchartComponent id='smithchart'>
    <SmithchartSeriesCollectionDirective>
      <SmithchartSeriesDirective points=[
        { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
      },
        { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },
        { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },
        { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },
```

```

        { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
        { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },
    ]] name='Transmission1'>
        </SmithchartSeriesDirective>
        <SmithchartSeriesDirective points=[
        { resistance: 20, reactance: -50 }, { resistance: 10, reactance:
-10 },
        { resistance: 9, reactance: -4.5 }, { resistance: 8, reactance:
-3.5 },
        { resistance: 7, reactance: -2.5 }, { resistance: 6, reactance:
-1.5 },
        { resistance: 5, reactance: -1 }, { resistance: 4.5, reactance:
-0.5 },
        { resistance: 3.5, reactance: 0 }, { resistance: 2.5, reactance:
0.4 },
        { resistance: 2, reactance: 0.5 }, { resistance: 1.5, reactance:
0.5 },
        { resistance: 1, reactance: 0.4 }, { resistance: 0.5, reactance:
0.2 },
        { resistance: 0.3, reactance: 0.1 }, { resistance: 0, reactance:
0.05 },
    ]] name='Transmission2'>
        </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
</SmithchartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
    return ( <SmithchartComponent id='smithchart'>
        <SmithchartSeriesCollectionDirective>
            <SmithchartSeriesDirective
                points=[
                    { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
                    { resistance: 6, reactance: 4.5 }, {
resistance: 4.5, reactance: 2 },
                    { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
                    { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },
                    { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
                    { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
                ]] name='Transmission1'
            </SmithchartSeriesDirective>
        </SmithchartSeriesCollectionDirective>
    </SmithchartComponent> );
}

```

```

    >
    </SmithchartSeriesDirective>
    <SmithchartSeriesDirective
      points=[
        { resistance: 20, reactance: -50 }, {
resistance: 10, reactance: -10 },
        { resistance: 9, reactance: -4.5 }, {
resistance: 8, reactance: -3.5 },
        { resistance: 7, reactance: -2.5 }, {
resistance: 6, reactance: -1.5 },
        { resistance: 5, reactance: -1 }, {
resistance: 4.5, reactance: -0.5 },
        { resistance: 3.5, reactance: 0 }, {
resistance: 2.5, reactance: 0.4 },
        { resistance: 2, reactance: 0.5 }, {
resistance: 1.5, reactance: 0.5 },
        { resistance: 1, reactance: 0.4 }, {
resistance: 0.5, reactance: 0.2 },
        { resistance: 0.3, reactance: 0.1 }, {
resistance: 0, reactance: 0.05 },
      ] name='Transmission2'
    >
    </SmithchartSeriesDirective>
  </SmithchartSeriesCollectionDirective>
</SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

### Add title to SmithChart

smithchart **title** API used to add title for smithchart. In that **text** API used to set text of the title.

API **visible** used to toggle the title.

### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
  return (<SmithchartComponent id='smithchart' title={{
    visible: true,
    text: 'Transmission lines applied for both impedance and
admittance'
  }}>
    <SmithchartSeriesCollectionDirective>
      <SmithchartSeriesDirective points=[
        { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
      },
        { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },
        { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },

```



```

    { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },
    { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
    { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },
  ]] name='Transmission1'>
    </SmithchartSeriesDirective>
    <SmithchartSeriesDirective points=[
    { resistance: 20, reactance: -50 }, { resistance: 10, reactance:
-10 },
    { resistance: 9, reactance: -4.5 }, { resistance: 8, reactance:
-3.5 },
    { resistance: 7, reactance: -2.5 }, { resistance: 6, reactance:
-1.5 },
    { resistance: 5, reactance: -1 }, { resistance: 4.5, reactance:
-0.5 },
    { resistance: 3.5, reactance: 0 }, { resistance: 2.5, reactance:
0.4 },
    { resistance: 2, reactance: 0.5 }, { resistance: 1.5, reactance:
0.5 },
    { resistance: 1, reactance: 0.4 }, { resistance: 0.5, reactance:
0.2 },
    { resistance: 0.3, reactance: 0.1 }, { resistance: 0, reactance:
0.05 },
  ]] name='Transmission2'>
    </SmithchartSeriesDirective>
  </SmithchartSeriesCollectionDirective>
</SmithchartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));
{% enddraw %}

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SmithchartComponent id='smithchart'
    title = { {
      visible: true,
      text: 'Transmission lines applied for both
impedance and admittance'
    }} >
    <SmithchartSeriesCollectionDirective>
      <SmithchartSeriesDirective
        points=[
          { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
          { resistance: 6, reactance: 4.5 }, {
resistance: 4.5, reactance: 2 },

```

```

    { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
    { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },
    { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
    { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
    ]} name='Transmission1'
  >
</SmithchartSeriesDirective>
<SmithchartSeriesDirective
  points=[
    { resistance: 20, reactance: -50 }, {
resistance: 10, reactance: -10 },
    { resistance: 9, reactance: -4.5 }, {
resistance: 8, reactance: -3.5 },
    { resistance: 7, reactance: -2.5 }, {
resistance: 6, reactance: -1.5 },
    { resistance: 5, reactance: -1 }, {
resistance: 4.5, reactance: -0.5 },
    { resistance: 3.5, reactance: 0 }, {
resistance: 2.5, reactance: 0.4 },
    { resistance: 2, reactance: 0.5 }, {
resistance: 1.5, reactance: 0.5 },
    { resistance: 1, reactance: 0.4 }, {
resistance: 0.5, reactance: 0.2 },
    { resistance: 0.3, reactance: 0.1 }, {
resistance: 0, reactance: 0.05 },
    ]} name='Transmission2'
  >
</SmithchartSeriesDirective>
</SmithchartSeriesCollectionDirective>
</SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

### Enable Marker to Smithchart

To use series marker and its customization in smithchart, use series `marker`. To display marker for particular series, need to specify `marker visible` as true. Below sample marker enabled for first series only.

#### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
  return (<SmithchartComponent id='smithchart' title={{
    visible: true,
    text: 'Transmission lines applied for both impedance and
admittance'
  }}>

```

```

        <SmithchartSeriesCollectionDirective>
          <SmithchartSeriesDirective points=[
            { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
          },
            { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },
            { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },
            { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },
            { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
            { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },
          ] name='Transmission1'>
        </SmithchartSeriesDirective>
        <SmithchartSeriesDirective points=[
          { resistance: 20, reactance: -50 }, { resistance: 10, reactance:
-10 },
          { resistance: 9, reactance: -4.5 }, { resistance: 8, reactance:
-3.5 },
          { resistance: 7, reactance: -2.5 }, { resistance: 6, reactance:
-1.5 },
          { resistance: 5, reactance: -1 }, { resistance: 4.5, reactance:
-0.5 },
          { resistance: 3.5, reactance: 0 }, { resistance: 2.5, reactance:
0.4 },
          { resistance: 2, reactance: 0.5 }, { resistance: 1.5, reactance:
0.5 },
          { resistance: 1, reactance: 0.4 }, { resistance: 0.5, reactance:
0.2 },
          { resistance: 0.3, reactance: 0.1 }, { resistance: 0, reactance:
0.05 },
        ] name='Transmission2' marker={{ shape: 'Circle', visible: true,
border: { width: 2 } }}>
        </SmithchartSeriesDirective>
      </SmithchartSeriesCollectionDirective>
    </SmithchartComponent>);
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));
{% enddraw %}

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SmithchartComponent id='smithchart'
    title = { {
      visible: true,
      text: 'Transmission lines applied for both
impedance and admittance'

```

```

    } } >
    <SmithchartSeriesCollectionDirective>
      <SmithchartSeriesDirective
        points=[
          { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
          { resistance: 6, reactance: 4.5 }, {
resistance: 4.5, reactance: 2 },
          { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
          { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },
          { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
          { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
        ]} name='Transmission1'
      >
    </SmithchartSeriesDirective>
    <SmithchartSeriesDirective
      points=[
        { resistance: 20, reactance: -50 }, {
resistance: 10, reactance: -10 },
        { resistance: 9, reactance: -4.5 }, {
resistance: 8, reactance: -3.5 },
        { resistance: 7, reactance: -2.5 }, {
resistance: 6, reactance: -1.5 },
        { resistance: 5, reactance: -1 }, {
resistance: 4.5, reactance: -0.5 },
        { resistance: 3.5, reactance: 0 }, {
resistance: 2.5, reactance: 0.4 },
        { resistance: 2, reactance: 0.5 }, {
resistance: 1.5, reactance: 0.5 },
        { resistance: 1, reactance: 0.4 }, {
resistance: 0.5, reactance: 0.2 },
        { resistance: 0.3, reactance: 0.1 }, {
resistance: 0, reactance: 0.05 },
        ]} name='Transmission2' marker={ { shape:
'Circle', visible: true, border: { width: 2 } } }
      >
    </SmithchartSeriesDirective>
  </SmithchartSeriesCollectionDirective>
</SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

### Enable DataLabel to Smithchart Marker

To use marker dataLabel and it's customization in smithchart, use marker dataLabel. To display dataLabel for particular series marker, need to specify dataLabel visible as true in that series marker. Below sample dataLabel enabled for first series.

### INDEX.JSX

```

{% raw %}
import * as React from "react";

```

```

import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
    return (<SmithchartComponent id='smithchart' title={{
        visible: true,
        text: 'Transmission lines applied for both impedance and
admittance'
    }}>
        <SmithchartSeriesCollectionDirective>
            <SmithchartSeriesDirective points=([
                { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
},
                { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },
                { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },
                { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },
                { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
                { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },
            ]] name='Transmission1' marker={{ shape: 'Circle', dataLabel: {
visible: true }, visible: true, border: { width: 2 } }}>
                </SmithchartSeriesDirective>
                <SmithchartSeriesDirective points=([
-10 },
                { resistance: 9, reactance: -4.5 }, { resistance: 8, reactance:
-3.5 },
                { resistance: 7, reactance: -2.5 }, { resistance: 6, reactance:
-1.5 },
                { resistance: 5, reactance: -1 }, { resistance: 4.5, reactance:
-0.5 },
                { resistance: 3.5, reactance: 0 }, { resistance: 2.5, reactance:
0.4 },
                { resistance: 2, reactance: 0.5 }, { resistance: 1.5, reactance:
0.5 },
                { resistance: 1, reactance: 0.4 }, { resistance: 0.5, reactance:
0.2 },
                { resistance: 0.3, reactance: 0.1 }, { resistance: 0, reactance:
0.05 },
            ]] name='Transmission2'>
                </SmithchartSeriesDirective>
            </SmithchartSeriesCollectionDirective>
        </SmithchartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));
{% endraw %}

```

**INDEX.TSX**

```
import * as React from "react";
```

```

import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SmithchartComponent id='smithchart'
    title = { {
      visible: true,
      text: 'Transmission lines applied for both
impedance and admittance'
    } } >
    <SmithchartSeriesCollectionDirective>
      <SmithchartSeriesDirective
        points=[
          { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
          { resistance: 6, reactance: 4.5 }, {
resistance: 4.5, reactance: 2 },
          { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
          { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },
          { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
          { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
        ]} name='Transmission1' marker={ { shape:
'Circle', dataLabel: { visible: true }, visible: true, border: { width: 2 }
} }
      >
    </SmithchartSeriesDirective>
    <SmithchartSeriesDirective
      points=[
        { resistance: 20, reactance: -50 }, {
resistance: 10, reactance: -10 },
        { resistance: 9, reactance: -4.5 }, {
resistance: 8, reactance: -3.5 },
        { resistance: 7, reactance: -2.5 }, {
resistance: 6, reactance: -1.5 },
        { resistance: 5, reactance: -1 }, {
resistance: 4.5, reactance: -0.5 },
        { resistance: 3.5, reactance: 0 }, {
resistance: 2.5, reactance: 0.4 },
        { resistance: 2, reactance: 0.5 }, {
resistance: 1.5, reactance: 0.5 },
        { resistance: 1, reactance: 0.4 }, {
resistance: 0.5, reactance: 0.2 },
        { resistance: 0.3, reactance: 0.1 }, {
resistance: 0, reactance: 0.05 },
      ]} name='Transmission2'
    >
    </SmithchartSeriesDirective>
  </SmithchartSeriesCollectionDirective>
</SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

### Enable Legend for Smithchart

Smithchart had a legend feature, which is used to denote the correspond series. To enable the legend, need to inject `SmithchartLegend` module using `Smithchart.Inject(SmithchartLegend)` method and `smithchart legendSettings visible` as `true`. Following example sample shows enabling legend for smithchart. Series name can customize using series `name`.

#### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective, Inject, SmithchartLegend } from '@syncfusion/ej2-react-charts';
function App() {
    return (<SmithchartComponent id='smithchart' legendSettings={{ visible:
true, shape: 'Circle' }} title={{
        visible: true,
        text: 'Transmission lines applied for both impedance and
admittance'
    }}>
        <Inject services=[[SmithchartLegend]]/>
        <SmithchartSeriesCollectionDirective>
            <SmithchartSeriesDirective points=[[
                { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
            },
                { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },
                { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },
                { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },
                { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
                { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },
            ]] name='Transmission1' marker={{ shape: 'Circle', dataLabel: {
visible: true }, visible: true, border: { width: 2 } }}>
                </SmithchartSeriesDirective>
                <SmithchartSeriesDirective points=[[
                    { resistance: 20, reactance: -50 }, { resistance: 10, reactance:
-10 },
                    { resistance: 9, reactance: -4.5 }, { resistance: 8, reactance:
-3.5 },
                    { resistance: 7, reactance: -2.5 }, { resistance: 6, reactance:
-1.5 },
                    { resistance: 5, reactance: -1 }, { resistance: 4.5, reactance:
-0.5 },
                    { resistance: 3.5, reactance: 0 }, { resistance: 2.5, reactance:
0.4 },
                    { resistance: 2, reactance: 0.5 }, { resistance: 1.5, reactance:
0.5 },
                    { resistance: 1, reactance: 0.4 }, { resistance: 0.5, reactance:
0.2 },
                ]]
```

```

        { resistance: 0.3, reactance: 0.1 }, { resistance: 0, reactance:
0.05 },
      ]} name='Transmission2'>
        </SmithchartSeriesDirective>
      </SmithchartSeriesCollectionDirective>
    </SmithchartComponent>);
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));
{% enddraw %}

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective, Inject, SmithchartLegend } from '@syncfusion/ej2-
react-charts';
function App() {
  return ( <SmithchartComponent id='smithchart'
    legendSettings={ { visible: true, shape: 'Circle' } }
  >
    title = { {
      visible: true,
      text: 'Transmission lines applied for both
impedance and admittance'
    } } >
    <Inject services={[SmithchartLegend]} />
    <SmithchartSeriesCollectionDirective>
      <SmithchartSeriesDirective
        points=[
          { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
          { resistance: 6, reactance: 4.5 }, {
resistance: 4.5, reactance: 2 },
          { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
          { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },
          { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
          { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
        ]} name='Transmission1' marker={ { shape:
'Circle', dataLabel: { visible: true }, visible: true, border: { width: 2 }
      } }
      >
    </SmithchartSeriesDirective>
    <SmithchartSeriesDirective
      points=[
        { resistance: 20, reactance: -50 }, {
resistance: 10, reactance: -10 },
        { resistance: 9, reactance: -4.5 }, {
resistance: 8, reactance: -3.5 },

```



```

    { resistance: 7, reactance: -2.5 }, {
    resistance: 6, reactance: -1.5 },
    { resistance: 5, reactance: -1 }, {
    resistance: 4.5, reactance: -0.5 },
    { resistance: 3.5, reactance: 0 }, {
    resistance: 2.5, reactance: 0.4 },
    { resistance: 2, reactance: 0.5 }, {
    resistance: 1.5, reactance: 0.5 },
    { resistance: 1, reactance: 0.4 }, {
    resistance: 0.5, reactance: 0.2 },
    { resistance: 0.3, reactance: 0.1 }, {
    resistance: 0, reactance: 0.05 },
    ]} name='Transmission2'
  >
</SmithchartSeriesDirective>
</SmithchartSeriesCollectionDirective>
</SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

### Enable Tooltip for Smithchart Series

Smithchart had a tooltip feature, which is used to show the current point's values. To enable the tooltip, need to inject `TooltipRender` module using `Smithchart.Inject(TooltipRender)` method and smithchart series `tooltip visible` as true. Following example sample shows enabling tooltip for smithchart series collection.

### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective, Inject, SmithchartLegend, TooltipRender } from
'@syncfusion/ej2-react-charts';
function App() {
  return (<SmithchartComponent id='smithchart' legendSettings={{ visible:
true, shape: 'Circle' }} title={{
    visible: true,
    text: 'Transmission lines applied for both impedance and
admittance'
  }}>
    <Inject services={[SmithchartLegend,
TooltipRender]}/>
    <SmithchartSeriesCollectionDirective>
      <SmithchartSeriesDirective points=[
        { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
      },
        { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },
        { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },
        { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },

```

```

        { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
        { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },
    ]} name='Transmission1' tooltip={{ visible: true }} marker={{ shape:
'Circle', dataLabel: { visible: true }, visible: true, border: { width: 2 }
}}>
        </SmithchartSeriesDirective>
        <SmithchartSeriesDirective points=[
    { resistance: 20, reactance: -50 }, { resistance: 10, reactance:
-10 },
    { resistance: 9, reactance: -4.5 }, { resistance: 8, reactance:
-3.5 },
    { resistance: 7, reactance: -2.5 }, { resistance: 6, reactance:
-1.5 },
    { resistance: 5, reactance: -1 }, { resistance: 4.5, reactance:
-0.5 },
    { resistance: 3.5, reactance: 0 }, { resistance: 2.5, reactance:
0.4 },
    { resistance: 2, reactance: 0.5 }, { resistance: 1.5, reactance:
0.5 },
    { resistance: 1, reactance: 0.4 }, { resistance: 0.5, reactance:
0.2 },
    { resistance: 0.3, reactance: 0.1 }, { resistance: 0, reactance:
0.05 },
    ]} name='Transmission2' tooltip={{ visible: true }}>
        </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
</SmithchartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));
{% enddraw %}

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective, Inject, SmithchartLegend, TooltipRender } from
'@syncfusion/ej2-react-charts';
function App() {
    return ( <SmithchartComponent id='smithchart'
        legendSettings={ { visible: true, shape: 'Circle' } }
    )
        title = { {
            visible: true,
            text: 'Transmission lines applied for both
impedance and admittance'
        } } >
        <Inject services=[[SmithchartLegend, TooltipRender]]
    />
        <SmithchartSeriesCollectionDirective>
            <SmithchartSeriesDirective
                points=[

```

```

resistance: 8, reactance: 6 },
resistance: 4.5, reactance: 2 },
resistance: 2.5, reactance: 1.3 },
resistance: 1.5, reactance: 1 },
resistance: 0.5, reactance: 0.4 },
resistance: 0, reactance: 0.15 },
    ]} name='Transmission1' tooltip={ { visible:
true } } marker={ { shape: 'Circle', dataLabel: { visible: true }, visible:
true, border: { width: 2 } } }
    >
    </SmithchartSeriesDirective>
    <SmithchartSeriesDirective
      points=[
        { resistance: 20, reactance: -50 }, {
resistance: 10, reactance: -10 },
        { resistance: 9, reactance: -4.5 }, {
resistance: 8, reactance: -3.5 },
        { resistance: 7, reactance: -2.5 }, {
resistance: 6, reactance: -1.5 },
        { resistance: 5, reactance: -1 }, {
resistance: 4.5, reactance: -0.5 },
        { resistance: 3.5, reactance: 0 }, {
resistance: 2.5, reactance: 0.4 },
        { resistance: 2, reactance: 0.5 }, {
resistance: 1.5, reactance: 0.5 },
        { resistance: 1, reactance: 0.4 }, {
resistance: 0.5, reactance: 0.2 },
        { resistance: 0.3, reactance: 0.1 }, {
resistance: 0, reactance: 0.05 },
      ]} name='Transmission2' tooltip={ { visible:
true } }
    >
    </SmithchartSeriesDirective>
  </SmithchartSeriesCollectionDirective>
</SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

### Working with data in React Smithchart component

The Smith chart can visualize the data bound from local data. The data bound for the Smith chart should be an array of object and should contain the field resistance and rectangle. This data should be bound to points or data source in the Smith chart.

#### Data Binding

You can bind simple JSON data to Smith chart using the point property in series. The JSON data should contain [resistance] and [reactance] fields. This JSON data should be bound to points or data source in the Smith chart. You can add any number of JSON or data source to points.

**INDEX.JSX**

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
    return (<SmithchartComponent id='smithchart'>
        <SmithchartSeriesCollectionDirective>
            <SmithchartSeriesDirective points=[
                { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
            },
                { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },
                { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },
                { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },
                { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
                { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },
            ]] name='Transmission1'>
                </SmithchartSeriesDirective>
                <SmithchartSeriesDirective points=[
                    { resistance: 20, reactance: -50 }, { resistance: 10, reactance:
-10 },
                    { resistance: 9, reactance: -4.5 }, { resistance: 8, reactance:
-3.5 },
                    { resistance: 7, reactance: -2.5 }, { resistance: 6, reactance:
-1.5 },
                    { resistance: 5, reactance: -1 }, { resistance: 4.5, reactance:
-0.5 },
                    { resistance: 3.5, reactance: 0 }, { resistance: 2.5, reactance:
0.4 },
                    { resistance: 2, reactance: 0.5 }, { resistance: 1.5, reactance:
0.5 },
                    { resistance: 1, reactance: 0.4 }, { resistance: 0.5, reactance:
0.2 },
                    { resistance: 0.3, reactance: 0.1 }, { resistance: 0, reactance:
0.05 },
                ]] name='Transmission2'>
                    </SmithchartSeriesDirective>
                </SmithchartSeriesCollectionDirective>
            </SmithchartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

**INDEX.TSX**

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';

```

```

function App() {
  return ( <SmithchartComponent id='smithchart'>
    <SmithchartSeriesCollectionDirective>
      <SmithchartSeriesDirective
        points=[
          { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
          { resistance: 6, reactance: 4.5 }, {
resistance: 4.5, reactance: 2 },
          { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
          { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },
          { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
          { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
        ]} name='Transmission1'
      >
    </SmithchartSeriesDirective>
    <SmithchartSeriesDirective
      points=[
          { resistance: 20, reactance: -50 }, {
resistance: 10, reactance: -10 },
          { resistance: 9, reactance: -4.5 }, {
resistance: 8, reactance: -3.5 },
          { resistance: 7, reactance: -2.5 }, {
resistance: 6, reactance: -1.5 },
          { resistance: 5, reactance: -1 }, {
resistance: 4.5, reactance: -0.5 },
          { resistance: 3.5, reactance: 0 }, {
resistance: 2.5, reactance: 0.4 },
          { resistance: 2, reactance: 0.5 }, {
resistance: 1.5, reactance: 0.5 },
          { resistance: 1, reactance: 0.4 }, {
resistance: 0.5, reactance: 0.2 },
          { resistance: 0.3, reactance: 0.1 }, {
resistance: 0, reactance: 0.05 },
        ]} name='Transmission2'
      >
    </SmithchartSeriesDirective>
  </SmithchartSeriesCollectionDirective>
</SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

### Smithchart dimensions in React Smithchart component

You can render the Smith chart corresponding to its container size. To set the size for the Smith chart, use the width and height properties.

#### Size for container

To render the Smith chart to its container size, specify the width and height of the Smith chart's container using inline or CSS as demonstrated in the following code sample.

```
`ts
<div id='container'>

<div id='smithchart' style="width:650px; height:350px;"></div>

</div>
`
```

### INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
    return (<SmithchartComponent id='smithchart'>
        <SmithchartSeriesCollectionDirective>
            <SmithchartSeriesDirective points=[
                { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
            },
                { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },
                { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },
                { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },
                { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
                { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },
            ] name='Transmission1'>
            </SmithchartSeriesDirective>
        </SmithchartSeriesCollectionDirective>
    </SmithchartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));
```

### INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
    return ( <SmithchartComponent id='smithchart'>
        <SmithchartSeriesCollectionDirective>
            <SmithchartSeriesDirective
                points=[
                    { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
                    { resistance: 6, reactance: 4.5 }, {
resistance: 4.5, reactance: 2 },
                    { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
```

```

        { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },
        { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
        { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
      ]} name='Transmission1'
    >
    </SmithchartSeriesDirective>
  </SmithchartSeriesCollectionDirective>
</SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

### Size for Smith chart

<!-- markdownlint-disable MD036 -->

You can also set the size for Smith chart in pixels or percentage directly using the `[width]` and `[height]` properties.

#### *In pixel*

In the width and height properties of Smith chart, directly set the values in pixels as demonstrated in the following code sample.

#### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
  return (<SmithchartComponent id='smithchart' width="650px"
height="300px">
    <SmithchartSeriesCollectionDirective>
      <SmithchartSeriesDirective points={[
        { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
},
        { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },
        { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },
        { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },
        { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
        { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },
      ]} name='Transmission1'>
      </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
  </SmithchartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

**INDEX.TSX**

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
    return ( <SmithchartComponent id='smithchart' width="650px"
height="300px">
        <SmithchartSeriesCollectionDirective>
            <SmithchartSeriesDirective
                points=[
                    { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
                    { resistance: 6, reactance: 4.5 }, {
resistance: 4.5, reactance: 2 },
                    { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
                    { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },
                    { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
                    { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
                ]} name='Transmission1'
            >
        </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
</SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));
```

*In percentage*

You can also specify the width and height properties of the Smith chart in percentage. If you mention the width and height in percentage, then Smith chart will be rendered to its container size. The following code sample demonstrates how to set the width and height properties of the Smith chart in percentage.

**INDEX.JSX**

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
    return ( <SmithchartComponent id='smithchart' width="90%" height="85%">
        <SmithchartSeriesCollectionDirective>
            <SmithchartSeriesDirective points=[
                { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
},
                { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },
                { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },
            ]} name='Transmission1'
            >
        </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
</SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));
```



```

        { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },
        { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
        { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },
    ]] name='Transmission1'>
        </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
</SmithchartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));
```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
    return ( <SmithchartComponent id='smithchart' width="90%" height="85%">
        <SmithchartSeriesCollectionDirective>
            <SmithchartSeriesDirective
                points=[
                    { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
                    { resistance: 6, reactance: 4.5 }, {
resistance: 4.5, reactance: 2 },
                    { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
                    { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },
                    { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
                    { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
                ]] name='Transmission1'
            >
        </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
</SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));
```

## Title subtitle in React Smithchart component

### Enable title

The title and subtitle are used to describe the information about the data being plotted in the Smith chart. You can set the title and subtitle of the Smith chart using the `[text]` property. By default, the visibility of the title and subtitle is enabled. The following code sample demonstrates how to simply set text to title and subtitle.

**INDEX.JSX**

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
    function load(args) {
        args.smithchart.title.text = 'Impedance Transmission';
        args.smithchart.title.visible = true;
        args.smithchart.title.subtitle.text = 'Transmission';
    }
    return (<SmithchartComponent id='smithchart' load={load.bind(this)}>
        <SmithchartSeriesCollectionDirective>
            <SmithchartSeriesDirective points=[
                { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
            },
                { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },
                { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },
                { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },
                { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
                { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },
            ] name='Transmission1'>
            </SmithchartSeriesDirective>
        </SmithchartSeriesCollectionDirective>
    </SmithchartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

**INDEX.TSX**

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
    function load(args: ISmithchartLoadedEventArgs): void {
        args.smithchart.title.text = 'Impedance Transmission';
        args.smithchart.title.visible = true;
        args.smithchart.title.subtitle.text = 'Transmission';
    }
    return ( <SmithchartComponent id='smithchart' load={load.bind(this)}>
        <SmithchartSeriesCollectionDirective>
            <SmithchartSeriesDirective
                points=[
                    { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
                    { resistance: 6, reactance: 4.5 }, {
resistance: 4.5, reactance: 2 },

```

```

        { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
        { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },
        { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
        { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
      ]} name='Transmission1'
    >
      </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
  </SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

### Trim title

Both the title and subtitle of the Smith chart can be trimmed if they exceed the certain length. Trimming is enabled using the [enableTrim] property for title and subtitle. The length for title and subtitle can be changed using the [maximumWidth] property. You can also customize the font, alignment, and visibility of title and subtitle using the [font], [textAlignment], and [visibility] properties.

### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
  function load(args) {
    args.smithchart.title.text = 'Impedance Transmission';
    args.smithchart.title.visible = true;
    args.smithchart.title.subtitle.text = 'Transmission';
    args.smithchart.title.enableTrim = true;
    args.smithchart.title.maximumWidth = 70;
    args.smithchart.title.subtitle.textAlignment = 'Far';
    args.smithchart.title.subtitle.enableTrim = true;
    args.smithchart.title.subtitle.maximumWidth = 40;
  }
  return (<SmithchartComponent id='smithchart' load={load.bind(this)}>
    <SmithchartSeriesCollectionDirective>
      <SmithchartSeriesDirective points=[
        { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
},
        { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },
        { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },
        { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },
        { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
        { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },

```

```

    ]] name='Transmission1'>
        </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
</SmithchartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
    function load(args: ISmithchartLoadedEventArgs): void {
        args.smithchart.title.text = 'Impedance Transmission';
        args.smithchart.title.visible = true;
        args.smithchart.title.subtitle.text = 'Transmission';
        args.smithchart.title.enableTrim = true;
        args.smithchart.title.maximumWidth = 70;
        args.smithchart.title.subtitle.textAlignment = 'Far';
        args.smithchart.title.subtitle.enableTrim = true;
        args.smithchart.title.subtitle.maximumWidth = 40;
    }
    return ( <SmithchartComponent id='smithchart' load={load.bind(this)}>
        <SmithchartSeriesCollectionDirective>
            <SmithchartSeriesDirective
                points=[
                    { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
                    { resistance: 6, reactance: 4.5 }, {
resistance: 4.5, reactance: 2 },
                    { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
                    { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },
                    { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
                    { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
                ]} name='Transmission1'
            >
                </SmithchartSeriesDirective>
            </SmithchartSeriesCollectionDirective>
        </SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

## Smithchart axis in React Smithchart component

Like chart, Smith chart also has support for two types of axis:

- Horizontal axis: Axis drawn as straight line in horizontal direction of the chart.
- Radial axis: Axis drawn as circular path.

### Labels customization

Axis labels are used to denote the kind of data that is bound to Smith chart. Using axis labels, you can easily identify the interval, in which the chart is rendered. The following properties are used to customize the axis labels for horizontal and radial axes:

- `[labelPosition]`: Places the labels either inside or outside the axis line.
- `[labelIntersectAction]`: Hides the labels when intersect with other labels.
- `[labelStyle]`: Customizes the properties such as font size, family, weight, and opacity.

### INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
  function load(args) {
    args.smithchart.radialAxis = {
      labelPosition: 'Inside',
      labelIntersectAction: 'Hide',
      labelStyle: {
        fontFamily: 'Times New Roman',
        fontWeight: 'bold',
        fontStyle: 'Italic',
        opacity: 0.75,
        size: '14px'
      }
    };
    args.smithchart.horizontalAxis = {
      labelPosition: 'Inside',
      labelIntersectAction: 'Hide',
      labelStyle: {
        fontFamily: 'Times New Roman',
        fontWeight: 'bold',
        fontStyle: 'Italic',
        opacity: 0.75,
        size: '14px'
      }
    };
  };
  return (<SmithchartComponent id='smithchart' load={load.bind(this)}>
    <SmithchartSeriesCollectionDirective>
      <SmithchartSeriesDirective points=[
        { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
      },
        { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },
        { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },
        { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },
```

```

        { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
        { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },
    ]] name='Transmission1'>
        </SmithchartSeriesDirective>
        <SmithchartSeriesDirective points=[
        { resistance: 20, reactance: -50 }, { resistance: 10, reactance:
-10 },
        { resistance: 9, reactance: -4.5 }, { resistance: 8, reactance:
-3.5 },
        { resistance: 7, reactance: -2.5 }, { resistance: 6, reactance:
-1.5 },
        { resistance: 5, reactance: -1 }, { resistance: 4.5, reactance:
-0.5 },
        { resistance: 3.5, reactance: 0 }, { resistance: 2.5, reactance:
0.4 },
        { resistance: 2, reactance: 0.5 }, { resistance: 1.5, reactance:
0.5 },
        { resistance: 1, reactance: 0.4 }, { resistance: 0.5, reactance:
0.2 },
        { resistance: 0.3, reactance: 0.1 }, { resistance: 0, reactance:
0.05 },
    ]] name='Transmission2'>
        </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
</SmithchartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
    function load(args: ISmithchartLoadedEventArgs): void {
        args.smithchart.radialAxis = {
            labelPosition: 'Inside',
            labelIntersectAction: 'Hide',
            labelStyle: {
                fontFamily: 'Times New Roman',
                fontWeight: 'bold',
                fontStyle: 'Italic',
                opacity: 0.75,
                size: '14px'
            }
        };
        args.smithchart.horizontalAxis = {
            labelPosition: 'Inside',
            labelIntersectAction: 'Hide',
            labelStyle: {
                fontFamily: 'Times New Roman',

```

```

        fontWeight: 'bold',
        fontStyle: 'Italic',
        opacity: 0.75,
        size: '14px'
    }
    };
}
}
return ( <SmithchartComponent id='smithchart' load={load.bind(this)}>
    <SmithchartSeriesCollectionDirective>
        <SmithchartSeriesDirective
            points=[
                { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
                { resistance: 6, reactance: 4.5 }, {
resistance: 4.5, reactance: 2 },
                { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
                { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },
                { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
                { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
            ]} name='Transmission1'
        >
    </SmithchartSeriesDirective>
    <SmithchartSeriesDirective
        points=[
            { resistance: 20, reactance: -50 }, {
resistance: 10, reactance: -10 },
            { resistance: 9, reactance: -4.5 }, {
resistance: 8, reactance: -3.5 },
            { resistance: 7, reactance: -2.5 }, {
resistance: 6, reactance: -1.5 },
            { resistance: 5, reactance: -1 }, {
resistance: 4.5, reactance: -0.5 },
            { resistance: 3.5, reactance: 0 }, {
resistance: 2.5, reactance: 0.4 },
            { resistance: 2, reactance: 0.5 }, {
resistance: 1.5, reactance: 0.5 },
            { resistance: 1, reactance: 0.4 }, {
resistance: 0.5, reactance: 0.2 },
            { resistance: 0.3, reactance: 0.1 }, {
resistance: 0, reactance: 0.05 },
            ]} name='Transmission2'
        >
    </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
</SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

### Gridlines

To make the data easier to read in a chart that displays axes, display horizontal axis and radial axis gridlines. Gridlines are extended from any horizontal axis or radial axis across the plot area of the Smith chart.

Both horizontal and radial axes have support for major and minor gridlines. Major gridlines are drawn from the position in which labels are rendered. Minor gridlines are drawn between two major gridlines as per the count you set in settings.

You can customize the major and minor gridlines using the following properties:

- `[width]`: Customizes the width of gridlines.
- `[dashArray]`: Decides whether the gridlines have to be rendered as normal line or dashed line.
- `[visible]`: Enables or disables the visibility of the gridlines.
- `[opacity]`: Customizes the opacity of the major gridlines.
- `[count]`: Customizes the count of the minor gridlines.

### INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
  function load(args) {
    args.smithchart.horizontalAxis = {
      majorGridLines: {
        visible: true,
        opacity: 0.8,
        width: 10
      }
    };
  }
  return (<SmithchartComponent id='smithchart' load={load.bind(this)}>
    <SmithchartSeriesCollectionDirective>
      <SmithchartSeriesDirective points=[
        { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
      },
        { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },
        { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },
        { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },
        { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
        { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },
      ] name='Transmission1'>
        </SmithchartSeriesDirective>
        <SmithchartSeriesDirective points=[
          { resistance: 20, reactance: -50 }, { resistance: 10, reactance:
-10 },
```



```

        { resistance: 9, reactance: -4.5 }, { resistance: 8, reactance:
-3.5 },
        { resistance: 7, reactance: -2.5 }, { resistance: 6, reactance:
-1.5 },
        { resistance: 5, reactance: -1 }, { resistance: 4.5, reactance:
-0.5 },
        { resistance: 3.5, reactance: 0 }, { resistance: 2.5, reactance:
0.4 },
        { resistance: 2, reactance: 0.5 }, { resistance: 1.5, reactance:
0.5 },
        { resistance: 1, reactance: 0.4 }, { resistance: 0.5, reactance:
0.2 },
        { resistance: 0.3, reactance: 0.1 }, { resistance: 0, reactance:
0.05 },
    ]} name='Transmission2'>
        </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
</SmithchartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
    function load(args: ISmithchartLoadedEventArgs): void {
        args.smithchart.horizontalAxis = {
            majorGridLines: {
                visible: true,
                opacity: 0.8,
                width: 10
            }
        };
    }
    return ( <SmithchartComponent id='smithchart' load={load.bind(this)}>
        <SmithchartSeriesCollectionDirective>
            <SmithchartSeriesDirective
                points=[
                    { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
                    { resistance: 6, reactance: 4.5 }, {
resistance: 4.5, reactance: 2 },
                    { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
                    { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },
                    { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
                    { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
                ]} name='Transmission1'

```

```

    >
    </SmithchartSeriesDirective>
    <SmithchartSeriesDirective
      points=[
        { resistance: 20, reactance: -50 }, {
resistance: 10, reactance: -10 },
        { resistance: 9, reactance: -4.5 }, {
resistance: 8, reactance: -3.5 },
        { resistance: 7, reactance: -2.5 }, {
resistance: 6, reactance: -1.5 },
        { resistance: 5, reactance: -1 }, {
resistance: 4.5, reactance: -0.5 },
        { resistance: 3.5, reactance: 0 }, {
resistance: 2.5, reactance: 0.4 },
        { resistance: 2, reactance: 0.5 }, {
resistance: 1.5, reactance: 0.5 },
        { resistance: 1, reactance: 0.4 }, {
resistance: 0.5, reactance: 0.2 },
        { resistance: 0.3, reactance: 0.1 }, {
resistance: 0, reactance: 0.05 },
      ] name='Transmission2'
    >
    </SmithchartSeriesDirective>
  </SmithchartSeriesCollectionDirective>
</SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

### Axis line

Axis line is a line in Smith chart that can be configured to denote the axis. You can customize the axis line using the following properties:

- **[width]**: Customize the width of the axis line.
- **[dashArray]**: Renders the axis line as dashed line.
- **[visible]**: Enables or disable the visibility of the axis line.

By default, visibility of the axis line is true. You can customize its visibility by using the visible property in axis line.

### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
  function load(args) {
    args.smithchart.horizontalAxis = {
      majorGridLines: {
        visible: true,
        opacity: 0.8,
        width: 10
      },

```

```

        axisLine: {
            width: 10,
            visible: false
        }
    };
}
return (<SmithchartComponent id='smithchart' load={load.bind(this)}>
    <SmithchartSeriesCollectionDirective>
        <SmithchartSeriesDirective points=[
            { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
        },
            { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },
            { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },
            { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },
            { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
            { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },
        ]] name='Transmission1'>
            </SmithchartSeriesDirective>
            <SmithchartSeriesDirective points=[
-10 },
            { resistance: 9, reactance: -4.5 }, { resistance: 8, reactance:
-3.5 },
            { resistance: 7, reactance: -2.5 }, { resistance: 6, reactance:
-1.5 },
            { resistance: 5, reactance: -1 }, { resistance: 4.5, reactance:
-0.5 },
            { resistance: 3.5, reactance: 0 }, { resistance: 2.5, reactance:
0.4 },
            { resistance: 2, reactance: 0.5 }, { resistance: 1.5, reactance:
0.5 },
            { resistance: 1, reactance: 0.4 }, { resistance: 0.5, reactance:
0.2 },
            { resistance: 0.3, reactance: 0.1 }, { resistance: 0, reactance:
0.05 },
        ]] name='Transmission2'>
            </SmithchartSeriesDirective>
        </SmithchartSeriesCollectionDirective>
    </SmithchartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {

```

```

function load(args: ISmithchartLoadedEventArgs): void {
    args.smithchart.horizontalAxis = {
        majorGridLines: {
            visible: true,
            opacity: 0.8,
            width: 10
        },
        axisLine: {
            width: 10,
            visible: false
        }
    };
}

return ( <SmithchartComponent id='smithchart' load={load.bind(this)}>
    <SmithchartSeriesCollectionDirective>
        <SmithchartSeriesDirective
            points=[
                { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
                { resistance: 6, reactance: 4.5 }, {
resistance: 4.5, reactance: 2 },
                { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
                { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },
                { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
                { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
            ] name='Transmission1'
        >
    </SmithchartSeriesDirective>
    <SmithchartSeriesDirective
        points=[
            { resistance: 20, reactance: -50 }, {
resistance: 10, reactance: -10 },
            { resistance: 9, reactance: -4.5 }, {
resistance: 8, reactance: -3.5 },
            { resistance: 7, reactance: -2.5 }, {
resistance: 6, reactance: -1.5 },
            { resistance: 5, reactance: -1 }, {
resistance: 4.5, reactance: -0.5 },
            { resistance: 3.5, reactance: 0 }, {
resistance: 2.5, reactance: 0.4 },
            { resistance: 2, reactance: 0.5 }, {
resistance: 1.5, reactance: 0.5 },
            { resistance: 1, reactance: 0.4 }, {
resistance: 0.5, reactance: 0.2 },
            { resistance: 0.3, reactance: 0.1 }, {
resistance: 0, reactance: 0.05 },
            ] name='Transmission2'
        >
    </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
    </SmithchartComponent> );
};

export default App;

```

```
ReactDOM.render(<App />, document.getElementById('smithchart'));
```

<!-- markdownlint-disable MD036 -->

## Smithchart legend in React Smithchart component

Legend is a key used in smithchart; it contains symbols and descriptions. The legends provide a valuable information for interpreting what the Smith chart displays. They can be represented in various colors, shapes, or other identifiers based on the data. The legends are used to denote the series rendered in the Smith chart.

### Position and alignment

#### Position

By default, the visibility of the legend is false. To enable the legend, set the visible property to true in legendSettings. The default position for the legend is bottom. By using the [position] property, you can either place the legend at the bottom, top, right, or left of the Smith chart. To use the legend in Smith chart, import and inject the SmithchartLegend from chart.

#### INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { Inject, SmithchartComponent, SmithchartLegend,
SmithchartSeriesCollectionDirective, SmithchartSeriesDirective } from
'@syncfusion/ej2-react-charts';
function App() {
  function load(args) {
    args.smithchart.legendSettings = {
      visible: true,
      position: 'Top'
    };
  }
  return (<SmithchartComponent id='smithchart' load={load.bind(this)}>
    <Inject services={[SmithchartLegend]} />
    <SmithchartSeriesCollectionDirective>
      <SmithchartSeriesDirective points=[
        { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
      },
        { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },
        { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },
        { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },
        { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
        { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },
      ] name='Transmission1'>
        </SmithchartSeriesDirective>
        <SmithchartSeriesDirective points=[
          { resistance: 20, reactance: -50 }, { resistance: 10, reactance:
-10 },
          { resistance: 9, reactance: -4.5 }, { resistance: 8, reactance:
-3.5 },
```

```

        { resistance: 7, reactance: -2.5 }, { resistance: 6, reactance:
-1.5 },
        { resistance: 5, reactance: -1 }, { resistance: 4.5, reactance:
-0.5 },
        { resistance: 3.5, reactance: 0 }, { resistance: 2.5, reactance:
0.4 },
        { resistance: 2, reactance: 0.5 }, { resistance: 1.5, reactance:
0.5 },
        { resistance: 1, reactance: 0.4 }, { resistance: 0.5, reactance:
0.2 },
        { resistance: 0.3, reactance: 0.1 }, { resistance: 0, reactance:
0.05 },
    ]] name='Transmission2'>
        </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
</SmithchartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { Inject, SmithchartComponent, SmithchartLegend,
SmithchartSeriesCollectionDirective, SmithchartSeriesDirective } from
'@syncfusion/ej2-react-charts';
function App() {
    function load(args: ISmithchartLoadedEventArgs): void {
        args.smithchart.legendSettings = {
            visible: true,
            position: 'Top'
        };
    }
    return ( <SmithchartComponent id='smithchart' load={load.bind(this)}>
        <Inject services={[SmithchartLegend]} />
        <SmithchartSeriesCollectionDirective>
            <SmithchartSeriesDirective
                points=[
                    { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
                    { resistance: 6, reactance: 4.5 }, {
resistance: 4.5, reactance: 2 },
                    { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
                    { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },
                    { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
                    { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
                ]] name='Transmission1'
            >
        </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>

```

```

                                points=[
                                { resistance: 20, reactance: -50 }, {
resistance: 10, reactance: -10 },
                                { resistance: 9, reactance: -4.5 }, {
resistance: 8, reactance: -3.5 },
                                { resistance: 7, reactance: -2.5 }, {
resistance: 6, reactance: -1.5 },
                                { resistance: 5, reactance: -1 }, {
resistance: 4.5, reactance: -0.5 },
                                { resistance: 3.5, reactance: 0 }, {
resistance: 2.5, reactance: 0.4 },
                                { resistance: 2, reactance: 0.5 }, {
resistance: 1.5, reactance: 0.5 },
                                { resistance: 1, reactance: 0.4 }, {
resistance: 0.5, reactance: 0.2 },
                                { resistance: 0.3, reactance: 0.1 }, {
resistance: 0, reactance: 0.05 },
                                ]} name='Transmission2'
                                >
                                </SmithchartSeriesDirective>
                                </SmithchartSeriesCollectionDirective>
                                </SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

### Custom position

The legends are not limited to place at the default positions only. You can place a legend anywhere in the Smith chart by setting the position as custom in legendSettings and specifying the x and y-coordinates using the x and y properties.

### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { Inject, SmithchartComponent, SmithchartLegend,
SmithchartSeriesCollectionDirective, SmithchartSeriesDirective } from
'@syncfusion/ej2-react-charts';
function App() {
    function load(args) {
        args.smithchart.legendSettings = {
            position: 'Custom',
            location: {
                x: 80,
                y: 100
            }
        };
    }
    return (<SmithchartComponent id='smithchart' load={load.bind(this)}>
        <Inject services={[SmithchartLegend]}/>
        <SmithchartSeriesCollectionDirective>
            <SmithchartSeriesDirective points=[
                { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
            },
                { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },

```

```

    { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },
    { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },
    { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
    { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },
  ]] name='Transmission1'>
    </SmithchartSeriesDirective>
    <SmithchartSeriesDirective points=[
    { resistance: 20, reactance: -50 }, { resistance: 10, reactance:
-10 },
    { resistance: 9, reactance: -4.5 }, { resistance: 8, reactance:
-3.5 },
    { resistance: 7, reactance: -2.5 }, { resistance: 6, reactance:
-1.5 },
    { resistance: 5, reactance: -1 }, { resistance: 4.5, reactance:
-0.5 },
    { resistance: 3.5, reactance: 0 }, { resistance: 2.5, reactance:
0.4 },
    { resistance: 2, reactance: 0.5 }, { resistance: 1.5, reactance:
0.5 },
    { resistance: 1, reactance: 0.4 }, { resistance: 0.5, reactance:
0.2 },
    { resistance: 0.3, reactance: 0.1 }, { resistance: 0, reactance:
0.05 },
  ]] name='Transmission2'>
    </SmithchartSeriesDirective>
  </SmithchartSeriesCollectionDirective>
</SmithchartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { Inject, SmithchartComponent, SmithchartLegend,
SmithchartSeriesCollectionDirective, SmithchartSeriesDirective } from
'@syncfusion/ej2-react-charts';
function App() {
  function load(args: ISmithchartLoadedEventArgs): void {
    args.smithchart.legendSettings = {
      position: 'Custom',
      location:{
        x: 80,
        y: 100
      }
    };
  }
  return ( <SmithchartComponent id='smithchart' load={load.bind(this)}>
    <Inject services={[SmithchartLegend]} />
    <SmithchartSeriesCollectionDirective>

```



```

        <SmithchartSeriesDirective
            points=[
                { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
                { resistance: 6, reactance: 4.5 }, {
resistance: 4.5, reactance: 2 },
                { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
                { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },
                { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
                { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
            ] name='Transmission1'
        >
    </SmithchartSeriesDirective>
    <SmithchartSeriesDirective
        points=[
            { resistance: 20, reactance: -50 }, {
resistance: 10, reactance: -10 },
            { resistance: 9, reactance: -4.5 }, {
resistance: 8, reactance: -3.5 },
            { resistance: 7, reactance: -2.5 }, {
resistance: 6, reactance: -1.5 },
            { resistance: 5, reactance: -1 }, {
resistance: 4.5, reactance: -0.5 },
            { resistance: 3.5, reactance: 0 }, {
resistance: 2.5, reactance: 0.4 },
            { resistance: 2, reactance: 0.5 }, {
resistance: 1.5, reactance: 0.5 },
            { resistance: 1, reactance: 0.4 }, {
resistance: 0.5, reactance: 0.2 },
            { resistance: 0.3, reactance: 0.1 }, {
resistance: 0, reactance: 0.05 },
            ] name='Transmission2'
        >
    </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
</SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

### Legend alignment

By default, the legends are aligned in center position. Using the `alignment` property, you can align the legend in near and far locations of the Smith chart.

### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { Inject, SmithchartComponent, SmithchartLegend,
SmithchartSeriesCollectionDirective, SmithchartSeriesDirective } from
'@syncfusion/ej2-react-charts';
function App() {

```

```

function load(args) {
  args.smithchart.legendSettings = {
    visible: true,
    position: 'Top',
    alignment: 'Near'
  };
}
return (<SmithchartComponent id='smithchart' load={load.bind(this)}>
  <Inject services={[SmithchartLegend]}/>
  <SmithchartSeriesCollectionDirective>
    <SmithchartSeriesDirective points=[
      { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
    },
    { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },
    { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },
    { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },
    { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
    { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },
  ] name='Transmission1'>
    </SmithchartSeriesDirective>
    <SmithchartSeriesDirective points=[
-10 },
    { resistance: 9, reactance: -4.5 }, { resistance: 8, reactance:
-3.5 },
    { resistance: 7, reactance: -2.5 }, { resistance: 6, reactance:
-1.5 },
    { resistance: 5, reactance: -1 }, { resistance: 4.5, reactance:
-0.5 },
    { resistance: 3.5, reactance: 0 }, { resistance: 2.5, reactance:
0.4 },
    { resistance: 2, reactance: 0.5 }, { resistance: 1.5, reactance:
0.5 },
    { resistance: 1, reactance: 0.4 }, { resistance: 0.5, reactance:
0.2 },
    { resistance: 0.3, reactance: 0.1 }, { resistance: 0, reactance:
0.05 },
  ] name='Transmission2'>
    </SmithchartSeriesDirective>
  </SmithchartSeriesCollectionDirective>
</SmithchartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

**INDEX.TSX**

```

import * as React from "react";
import * as ReactDOM from "react-dom";

```

```

import { Inject, SmithchartComponent, SmithchartLegend,
SmithchartSeriesCollectionDirective, SmithchartSeriesDirective } from
'@syncfusion/ej2-react-charts';
function App() {
    function load(args: ISmithchartLoadedEventArgs): void {
        args.smithchart.legendSettings = {
            visible: true,
            position: 'Top',
            alignment: 'Near'
        };
    }
    return ( <SmithchartComponent id='smithchart' load={load.bind(this)}>
        <Inject services={[SmithchartLegend]} />
        <SmithchartSeriesCollectionDirective>
            <SmithchartSeriesDirective
                points=[
                    { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
                    { resistance: 6, reactance: 4.5 }, {
resistance: 4.5, reactance: 2 },
                    { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
                    { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },
                    { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
                    { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
                ] name='Transmission1'
            >
        </SmithchartSeriesDirective>
        <SmithchartSeriesDirective
            points=[
                { resistance: 20, reactance: -50 }, {
resistance: 10, reactance: -10 },
                { resistance: 9, reactance: -4.5 }, {
resistance: 8, reactance: -3.5 },
                { resistance: 7, reactance: -2.5 }, {
resistance: 6, reactance: -1.5 },
                { resistance: 5, reactance: -1 }, {
resistance: 4.5, reactance: -0.5 },
                { resistance: 3.5, reactance: 0 }, {
resistance: 2.5, reactance: 0.4 },
                { resistance: 2, reactance: 0.5 }, {
resistance: 1.5, reactance: 0.5 },
                { resistance: 1, reactance: 0.4 }, {
resistance: 0.5, reactance: 0.2 },
                { resistance: 0.3, reactance: 0.1 }, {
resistance: 0, reactance: 0.05 },
            ] name='Transmission2'
        >
        </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
    </SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

## Customization

### Legend shape

By default, the legends are rendered in a circle shape. The color of the shape will be same as the series color. Using the [shape] property in legend settings, you can change the shape of the legend to rectangle, triangle, and so on.

### INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { Inject, SmithchartComponent, SmithchartLegend,
SmithchartSeriesCollectionDirective, SmithchartSeriesDirective } from
'@syncfusion/ej2-react-charts';
function App() {
  function load(args) {
    args.smithchart.legendSettings = {
      visible: true,
      position: 'Top',
      shape: 'Rectangle'
    };
  }
  return (<SmithchartComponent id='smithchart' load={load.bind(this)}>
    <Inject services={[SmithchartLegend]}/>
    <SmithchartSeriesCollectionDirective>
      <SmithchartSeriesDirective points=[
        { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
      },
        { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },
        { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },
        { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },
        { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
        { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },
      ] name='Transmission1'>
      </SmithchartSeriesDirective>
      <SmithchartSeriesDirective points=[
        { resistance: 20, reactance: -50 }, { resistance: 10, reactance:
-10 },
        { resistance: 9, reactance: -4.5 }, { resistance: 8, reactance:
-3.5 },
        { resistance: 7, reactance: -2.5 }, { resistance: 6, reactance:
-1.5 },
        { resistance: 5, reactance: -1 }, { resistance: 4.5, reactance:
-0.5 },
        { resistance: 3.5, reactance: 0 }, { resistance: 2.5, reactance:
0.4 },
        { resistance: 2, reactance: 0.5 }, { resistance: 1.5, reactance:
0.5 },
        { resistance: 1, reactance: 0.4 }, { resistance: 0.5, reactance:
0.2 },
      ] name='Transmission2'>
      </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
  </SmithchartComponent>);
}
```

```

        { resistance: 0.3, reactance: 0.1 }, { resistance: 0, reactance:
0.05 },
      ]} name='Transmission2'>
        </SmithchartSeriesDirective>
      </SmithchartSeriesCollectionDirective>
    </SmithchartComponent>);
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { Inject, SmithchartComponent, SmithchartLegend,
SmithchartSeriesCollectionDirective, SmithchartSeriesDirective } from
'@syncfusion/ej2-react-charts';
function App() {
  function load(args: ISmithchartLoadedEventArgs): void {
    args.smithchart.legendSettings = {
      visible: true,
      position: 'Top',
      shape: 'Rectangle'
    };
  }
  return ( <SmithchartComponent id='smithchart' load={load.bind(this)}>
    <Inject services={[SmithchartLegend]} />
    <SmithchartSeriesCollectionDirective>
      <SmithchartSeriesDirective
        points=[
          { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
          { resistance: 6, reactance: 4.5 }, {
resistance: 4.5, reactance: 2 },
          { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
          { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },
          { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
          { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
        ]} name='Transmission1'
      >
    </SmithchartSeriesDirective>
    <SmithchartSeriesDirective
      points=[
          { resistance: 20, reactance: -50 }, {
resistance: 10, reactance: -10 },
          { resistance: 9, reactance: -4.5 }, {
resistance: 8, reactance: -3.5 },
          { resistance: 7, reactance: -2.5 }, {
resistance: 6, reactance: -1.5 },
          { resistance: 5, reactance: -1 }, {
resistance: 4.5, reactance: -0.5 },

```

```

        { resistance: 3.5, reactance: 0 }, {
resistance: 2.5, reactance: 0.4 },
        { resistance: 2, reactance: 0.5 }, {
resistance: 1.5, reactance: 0.5 },
        { resistance: 1, reactance: 0.4 }, {
resistance: 0.5, reactance: 0.2 },
        { resistance: 0.3, reactance: 0.1 }, {
resistance: 0, reactance: 0.05 },
    ]} name='Transmission2'
    >
    </SmithchartSeriesDirective>
  </SmithchartSeriesCollectionDirective>
</SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

### Legend size

By default, a legend takes 20 to 25% of the chart's height horizontally when it is placed at the top or bottom position and 20 to 25% of the width vertically when placing at the left or right. You can change the legend size using the `[width]` and `[height]` properties of the `legendSettings`.

### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { Inject, SmithchartComponent, SmithchartLegend,
SmithchartSeriesCollectionDirective, SmithchartSeriesDirective } from
'@syncfusion/ej2-react-charts';
function App() {
  function load(args) {
    args.smithchart.legendSettings = {
      visible: true,
      position: 'Top',
      height: 100,
      width: 200
    };
  }
  return (<SmithchartComponent id='smithchart' load={load.bind(this)}>
    <Inject services={[SmithchartLegend]} />
    <SmithchartSeriesCollectionDirective>
      <SmithchartSeriesDirective points=[
        { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
},
        { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },
        { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },
        { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },
        { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
        { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },
      ]} name='Transmission1'>
      </SmithchartSeriesDirective>

```

```

        <SmithchartSeriesDirective points=[
            { resistance: 20, reactance: -50 }, { resistance: 10, reactance:
-10 },
            { resistance: 9, reactance: -4.5 }, { resistance: 8, reactance:
-3.5 },
            { resistance: 7, reactance: -2.5 }, { resistance: 6, reactance:
-1.5 },
            { resistance: 5, reactance: -1 }, { resistance: 4.5, reactance:
-0.5 },
            { resistance: 3.5, reactance: 0 }, { resistance: 2.5, reactance:
0.4 },
            { resistance: 2, reactance: 0.5 }, { resistance: 1.5, reactance:
0.5 },
            { resistance: 1, reactance: 0.4 }, { resistance: 0.5, reactance:
0.2 },
            { resistance: 0.3, reactance: 0.1 }, { resistance: 0, reactance:
0.05 },
        ]} name='Transmission2'>
        </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
</SmithchartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { Inject, SmithchartComponent, SmithchartLegend,
SmithchartSeriesCollectionDirective, SmithchartSeriesDirective } from
'@syncfusion/ej2-react-charts';
function App() {
    function load(args: ISmithchartLoadedEventArgs): void {
        args.smithchart.legendSettings = {
            visible: true,
            position: 'Top',
            height: 100,
            width: 200
        };
    }
    return ( <SmithchartComponent id='smithchart' load={load.bind(this)}>
        <Inject services={[SmithchartLegend]} />
        <SmithchartSeriesCollectionDirective>
            <SmithchartSeriesDirective
                points=[
                    { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
                    { resistance: 6, reactance: 4.5 }, {
resistance: 4.5, reactance: 2 },
                    { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
                    { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },

```

```

        { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
        { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
      ]} name='Transmission1'
    >
    </SmithchartSeriesDirective>
    <SmithchartSeriesDirective
      points=[
        { resistance: 20, reactance: -50 }, {
resistance: 10, reactance: -10 },
        { resistance: 9, reactance: -4.5 }, {
resistance: 8, reactance: -3.5 },
        { resistance: 7, reactance: -2.5 }, {
resistance: 6, reactance: -1.5 },
        { resistance: 5, reactance: -1 }, {
resistance: 4.5, reactance: -0.5 },
        { resistance: 3.5, reactance: 0 }, {
resistance: 2.5, reactance: 0.4 },
        { resistance: 2, reactance: 0.5 }, {
resistance: 1.5, reactance: 0.5 },
        { resistance: 1, reactance: 0.4 }, {
resistance: 0.5, reactance: 0.2 },
        { resistance: 0.3, reactance: 0.1 }, {
resistance: 0, reactance: 0.05 },
      ]} name='Transmission2'
    >
    </SmithchartSeriesDirective>
  </SmithchartSeriesCollectionDirective>
</SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

### Padding

You can customize the space between two legend items and space between legend shape and text as needed. For customizing the space between two legend items, use the `[itemPadding]` property. To control the space between legend shape and text, use the `[shapePadding]` property.

### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { Inject, SmithchartComponent, SmithchartLegend,
SmithchartSeriesCollectionDirective, SmithchartSeriesDirective } from
'@syncfusion/ej2-react-charts';
function App() {
  function load(args) {
    args.smithchart.legendSettings = {
      visible: true,
      position: 'Top',
      itemPadding: 5,
      shapePadding: 10
    };
  }
  return (<SmithchartComponent id='smithchart' load={load.bind(this)}>

```



```

        <Inject services={[SmithchartLegend]} />
        <SmithchartSeriesCollectionDirective>
            <SmithchartSeriesDirective points={[
                { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
            },
                { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
            2 },
                { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
            reactance: 1.3 },
                { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
            1 },
                { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
            0.4 },
                { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
            0.15 },
            ]} name='Transmission1'>
            </SmithchartSeriesDirective>
            <SmithchartSeriesDirective points={[
                { resistance: 20, reactance: -50 }, { resistance: 10, reactance:
            -10 },
                { resistance: 9, reactance: -4.5 }, { resistance: 8, reactance:
            -3.5 },
                { resistance: 7, reactance: -2.5 }, { resistance: 6, reactance:
            -1.5 },
                { resistance: 5, reactance: -1 }, { resistance: 4.5, reactance:
            -0.5 },
                { resistance: 3.5, reactance: 0 }, { resistance: 2.5, reactance:
            0.4 },
                { resistance: 2, reactance: 0.5 }, { resistance: 1.5, reactance:
            0.5 },
                { resistance: 1, reactance: 0.4 }, { resistance: 0.5, reactance:
            0.2 },
                { resistance: 0.3, reactance: 0.1 }, { resistance: 0, reactance:
            0.05 },
            ]} name='Transmission2'>
            </SmithchartSeriesDirective>
        </SmithchartSeriesCollectionDirective>
    </SmithchartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { Inject, SmithchartComponent, SmithchartLegend,
    SmithchartSeriesCollectionDirective, SmithchartSeriesDirective } from
    '@syncfusion/ej2-react-charts';
function App() {
    function load(args: ISmithchartLoadedEventArgs): void {
        args.smithchart.legendSettings = {
            visible: true,
            position: 'Top',
            itemPadding: 5,

```

```

        shapePadding: 10
    };
}
return ( <SmithchartComponent id='smithchart' load={load.bind(this)}>
    <Inject services={[SmithchartLegend]} />
    <SmithchartSeriesCollectionDirective>
        <SmithchartSeriesDirective
            points=[
                { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
                { resistance: 6, reactance: 4.5 }, {
resistance: 4.5, reactance: 2 },
                { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
                { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },
                { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
                { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
            ] name='Transmission1'
        >
    </SmithchartSeriesDirective>
    <SmithchartSeriesDirective
        points=[
            { resistance: 20, reactance: -50 }, {
resistance: 10, reactance: -10 },
            { resistance: 9, reactance: -4.5 }, {
resistance: 8, reactance: -3.5 },
            { resistance: 7, reactance: -2.5 }, {
resistance: 6, reactance: -1.5 },
            { resistance: 5, reactance: -1 }, {
resistance: 4.5, reactance: -0.5 },
            { resistance: 3.5, reactance: 0 }, {
resistance: 2.5, reactance: 0.4 },
            { resistance: 2, reactance: 0.5 }, {
resistance: 1.5, reactance: 0.5 },
            { resistance: 1, reactance: 0.4 }, {
resistance: 0.5, reactance: 0.2 },
            { resistance: 0.3, reactance: 0.1 }, {
resistance: 0, reactance: 0.05 },
            ] name='Transmission2'
        >
    </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
    </SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

### Toggle visibility

By default, the series name is displayed in the legend. You can collapse the visibility of a series by clicking the legend for the particular series and toggle the series visibility to true or false using the `toggleVisibility` property. By default, this property is set to true.

**INDEX.JSX**

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { Inject, SmithchartComponent, SmithchartLegend,
SmithchartSeriesCollectionDirective, SmithchartSeriesDirective } from
'@syncfusion/ej2-react-charts';
function App() {
  function load(args) {
    args.smithchart.legendSettings = {
      visible: true,
      position: 'Top',
      toggleVisibility: false
    };
  }
  return (<SmithchartComponent id='smithchart' load={load.bind(this)}>
    <Inject services={[SmithchartLegend]}/>
    <SmithchartSeriesCollectionDirective>
      <SmithchartSeriesDirective points=[
        { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
      },
        { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },
        { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },
        { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },
        { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
        { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },
      ]} name='Transmission1'>
      </SmithchartSeriesDirective>
      <SmithchartSeriesDirective points=[
-10 },
        { resistance: 9, reactance: -4.5 }, { resistance: 8, reactance:
-3.5 },
        { resistance: 7, reactance: -2.5 }, { resistance: 6, reactance:
-1.5 },
        { resistance: 5, reactance: -1 }, { resistance: 4.5, reactance:
-0.5 },
        { resistance: 3.5, reactance: 0 }, { resistance: 2.5, reactance:
0.4 },
        { resistance: 2, reactance: 0.5 }, { resistance: 1.5, reactance:
0.5 },
        { resistance: 1, reactance: 0.4 }, { resistance: 0.5, reactance:
0.2 },
        { resistance: 0.3, reactance: 0.1 }, { resistance: 0, reactance:
0.05 },
      ]} name='Transmission2'>
      </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
  </SmithchartComponent>);
}
;
export default App;

```

```
ReactDOM.render(<App />, document.getElementById('smithchart'));
```

## INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { Inject, SmithchartComponent, SmithchartLegend,
SmithchartSeriesCollectionDirective, SmithchartSeriesDirective } from
'@syncfusion/ej2-react-charts';
function App() {
  function load(args: ISmithchartLoadedEventArgs): void {
    args.smithchart.legendSettings = {
      visible: true,
      position: 'Top',
      toggleVisibility: false
    };
  }
  return ( <SmithchartComponent id='smithchart' load={load.bind(this)}>
    <Inject services={[SmithchartLegend]} />
    <SmithchartSeriesCollectionDirective>
      <SmithchartSeriesDirective
        points=[
          { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
          { resistance: 6, reactance: 4.5 }, {
resistance: 4.5, reactance: 2 },
          { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
          { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },
          { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
          { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
        ]} name='Transmission1'
      >
    </SmithchartSeriesDirective>
    <SmithchartSeriesDirective
      points=[
          { resistance: 20, reactance: -50 }, {
resistance: 10, reactance: -10 },
          { resistance: 9, reactance: -4.5 }, {
resistance: 8, reactance: -3.5 },
          { resistance: 7, reactance: -2.5 }, {
resistance: 6, reactance: -1.5 },
          { resistance: 5, reactance: -1 }, {
resistance: 4.5, reactance: -0.5 },
          { resistance: 3.5, reactance: 0 }, {
resistance: 2.5, reactance: 0.4 },
          { resistance: 2, reactance: 0.5 }, {
resistance: 1.5, reactance: 0.5 },
          { resistance: 1, reactance: 0.4 }, {
resistance: 0.5, reactance: 0.2 },
          { resistance: 0.3, reactance: 0.1 }, {
resistance: 0, reactance: 0.05 },
        ]} name='Transmission2'
      >
    </SmithchartSeriesDirective>
  </SmithchartComponent>
```

```

        >
        </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
</SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

### Smithchart tooltip in React Smithchart component

The Smith chart displays details about the points using tooltip when the mouse is hovered over the points. By default, the tooltips are disabled. To enable the tooltips, import and inject the TooltipRender module from Smith chart, and set the visible property to true in tooltip settings. You can customize the visibility and appearance of tooltips for each series in the Smith chart.

#### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { Inject, SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective, TooltipRender } from '@syncfusion/ej2-react-
charts';
function App() {
    function load(args) {
        args.smithchart.series[0].tooltip = {
            visible: true
        };
        args.smithchart.series[0].marker = {
            visible: true
        };
    }
    return (<SmithchartComponent id='smithchart' load={load.bind(this)}>
        <Inject services=[[TooltipRender]]/>
        <SmithchartSeriesCollectionDirective>
            <SmithchartSeriesDirective points=[[
                { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
            },
                { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },
                { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },
                { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },
                { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
                { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },
            ]]] name='Transmission1'>
                </SmithchartSeriesDirective>
            </SmithchartSeriesCollectionDirective>
        </SmithchartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

**INDEX.TSX**

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { Inject, SmithchartComponent, SmithchartLegend,
SmithchartSeriesCollectionDirective, SmithchartSeriesDirective,
TooltipRender } from '@syncfusion/ej2-react-charts';
function App() {
    function load(args: ISmithchartLoadedEventArgs): void {
        args.smithchart.series[0].tooltip = {
            visible: true
        };
        args.smithchart.series[0].marker = {
            visible: true
        };
    }
    return ( <SmithchartComponent id='smithchart' load={load.bind(this)}>
        <Inject services={[TooltipRender]} />
        <SmithchartSeriesCollectionDirective>
            <SmithchartSeriesDirective
                points=[
                    { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
                    { resistance: 6, reactance: 4.5 }, {
resistance: 4.5, reactance: 2 },
                    { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
                    { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },
                    { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
                    { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
                ]} name='Transmission1'
            >
        </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
    </SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

<!-- markdownlint-disable MD036 -->

### Smithchart marker in React Smithchart component

The markers and data labels are used to provide information about the data points in the series. You can add a shape to adorn each data point. By default, both the marker and data label are disabled in Smith chart. You can enable them by setting the visible property to true in marker and data label settings.

#### Marker

By default, the visibility of marker is false. You can enable marker by setting the visible property to true in marker settings. Using marker settings, you can customize marker differently for each series in Smith chart.

**INDEX.JSX**

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { Inject, SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective, TooltipRender } from '@syncfusion/ej2-react-
charts';
function App() {
  function load(args) {
    args.smithchart.series[0].marker = {
      visible: true
    };
  }
  return (<SmithchartComponent id='smithchart' load={load.bind(this)}>
    <Inject services={[TooltipRender]}/>
    <SmithchartSeriesCollectionDirective>
      <SmithchartSeriesDirective points=[
        { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
      },
        { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },
        { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },
        { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },
        { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
        { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },
      ] name='Transmission1'>
        </SmithchartSeriesDirective>
      </SmithchartSeriesCollectionDirective>
    </SmithchartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { Inject, SmithchartComponent, SmithchartLegend,
SmithchartSeriesCollectionDirective, SmithchartSeriesDirective,
TooltipRender } from '@syncfusion/ej2-react-charts';
function App() {
  function load(args: ISmithchartLoadedEventArgs): void {
    args.smithchart.series[0].marker = {
      visible: true
    };
  }
  return ( <SmithchartComponent id='smithchart' load={load.bind(this)}>
    <Inject services={[TooltipRender]} />
    <SmithchartSeriesCollectionDirective>
      <SmithchartSeriesDirective
        points=[
          { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },

```

```

resistance: 4.5, reactance: 2 }, { resistance: 6, reactance: 4.5 }, {
resistance: 2.5, reactance: 1.3 }, { resistance: 3.5, reactance: 1.6 }, {
resistance: 1.5, reactance: 1 }, { resistance: 2, reactance: 1.2 }, {
resistance: 0.5, reactance: 0.4 }, { resistance: 1, reactance: 0.8 }, {
resistance: 0, reactance: 0.15 }, { resistance: 0.3, reactance: 0.2 }, {
    ]} name='Transmission1'
    >
    </SmithchartSeriesDirective>
  </SmithchartSeriesCollectionDirective>
</SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

### Marker customization

Using marker settings in series, you can customize the marker for each series differently. You can customize the markers using the following properties differently for each series in the Smith chart:

- **[width]**: Controls the width of the marker.
- **[height]**: Controls the height of the marker.
- **[fill]**: Customizes the fill color of the marker.
- **[opacity]**: Customizes the opacity of the marker.
- **[border]**: Controls the width and color of the marker's border.
- **[shape]**: Changes the shape of the marker.

### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { Inject, SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective, TooltipRender } from '@syncfusion/ej2-react-charts';
function App() {
    function load(args) {
        args.smithchart.series[0].marker = {
            visible: true,
            height: 10,
            width: 10,
            fill: '#ff99ff',
            opacity: 1,
            shape: 'rectangle',
            border: {
                color: '#cc00cc',
                width: 2
            }
        };
    }
    return (<SmithchartComponent id='smithchart' load={load.bind(this)}>

```



```

        <Inject services={[TooltipRender]}/>
        <SmithchartSeriesCollectionDirective>
            <SmithchartSeriesDirective points={[
                { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
            },
                { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
            2 },
                { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
            reactance: 1.3 },
                { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
            1 },
                { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
            0.4 },
                { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
            0.15 },
            ]} name='Transmission1'>
            </SmithchartSeriesDirective>
        </SmithchartSeriesCollectionDirective>
    </SmithchartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { Inject, SmithchartComponent, SmithchartLegend,
SmithchartSeriesCollectionDirective, SmithchartSeriesDirective,
TooltipRender } from '@syncfusion/ej2-react-charts';
function App() {
    function load(args: ISmithchartLoadedEventArgs): void {
        args.smithchart.series[0].marker = {
            visible: true,
            height: 10,
            width: 10,
            fill: '#ff99ff',
            opacity: 1,
            shape: 'rectangle',
            border: {
                color: '#cc00cc',
                width: 2
            }
        };
    }
    return ( <SmithchartComponent id='smithchart' load={load.bind(this)}>
        <Inject services={[TooltipRender]}/>
        <SmithchartSeriesCollectionDirective>
            <SmithchartSeriesDirective
                points={[
                    { resistance: 10, reactance: 25 }, {
            resistance: 8, reactance: 6 },
                    { resistance: 6, reactance: 4.5 }, {
            resistance: 4.5, reactance: 2 },

```

```

        { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
        { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },
        { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
        { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
        ]] name='Transmission1'
      >
    </SmithchartSeriesDirective>
  </SmithchartSeriesCollectionDirective>
</SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

### Data labels

By default, the data labels are disabled. You can enable the data labels by setting the visible property to true in data label settings. For each point in series, a data label is created. The data labels for each series can be customized differently using the data label settings.

### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { Inject, SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective, TooltipRender } from '@syncfusion/ej2-react-
charts';
function App() {
  function load(args) {
    args.smithchart.series[0].marker = {
      dataLabel: {
        visible: true
      }
    };
  }
  return (<SmithchartComponent id='smithchart' load={load.bind(this)}>
    <Inject services={[TooltipRender]}/>
    <SmithchartSeriesCollectionDirective>
      <SmithchartSeriesDirective points=[
        { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
},
        { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },
        { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },
        { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },
        { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
        { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },
      ]] name='Transmission1'>
      </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
  </SmithchartComponent>);
}

```

```

        </SmithchartComponent>);
    }
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { Inject, SmithchartComponent, SmithchartLegend,
SmithchartSeriesCollectionDirective, SmithchartSeriesDirective,
TooltipRender } from '@syncfusion/ej2-react-charts';
function App() {
    function load(args: ISmithchartLoadedEventArgs): void {
        args.smithchart.series[0].marker = {
            dataLabel: {
                visible: true
            }
        };
    }
    return ( <SmithchartComponent id='smithchart' load={load.bind(this)}>
        <Inject services={[TooltipRender]} />
        <SmithchartSeriesCollectionDirective>
            <SmithchartSeriesDirective
                points=[
                    { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
                    { resistance: 6, reactance: 4.5 }, {
resistance: 4.5, reactance: 2 },
                    { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
                    { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },
                    { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
                    { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
                ] name='Transmission1'
            >
            </SmithchartSeriesDirective>
        </SmithchartSeriesCollectionDirective>
    </SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

### Data label customization

Using data label settings in marker, you can customize the data label for each series differently. The following properties are used to customize data labels differently for each series:

- **[fill]**: Changes the fill color of the data label's shape.
- **[opacity]**: Controls the opacity of the data label's shape.
- **[border]**: Customizes the width and color of the border.

- `[textStyle]`: Customizes the font color, width, and size.

### INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { Inject, SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective, TooltipRender } from '@syncfusion/ej2-react-
charts';
function App() {
  function load(args) {
    args.smithchart.series[0].marker = {
      dataLabel: {
        visible: true,
        fill: '#99ffcc',
        opacity: 1,
        border: {
          color: '#1aff8c',
          width: 2,
        }
      }
    };
  }
  return (<SmithchartComponent id='smithchart' load={load.bind(this)}>
    <Inject services={[TooltipRender]}/>
    <SmithchartSeriesCollectionDirective>
      <SmithchartSeriesDirective points=[
        { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
2 },
        { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },
        { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },
        { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },
        { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
        { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },
      ] name='Transmission1'>
        </SmithchartSeriesDirective>
      </SmithchartSeriesCollectionDirective>
    </SmithchartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));
```

### INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { Inject, SmithchartComponent, SmithchartLegend,
SmithchartSeriesCollectionDirective, SmithchartSeriesDirective,
TooltipRender } from '@syncfusion/ej2-react-charts';
function App() {
```

```

function load(args: ISmithchartLoadedEventArgs): void {
    args.smithchart.series[0].marker = {
        dataLabel: {
            visible: true,
            fill: '#99ffcc',
            opacity: 1,
            border: {
                color: '#1aff8c',
                width: 2,
            }
        }
    };
}

return ( <SmithchartComponent id='smithchart' load={load.bind(this)}>
    <Inject services={[TooltipRender]} />
    <SmithchartSeriesCollectionDirective>
        <SmithchartSeriesDirective
            points=[
                { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
                { resistance: 6, reactance: 4.5 }, {
resistance: 4.5, reactance: 2 },
                { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
                { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },
                { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
                { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
            ] name='Transmission1'
        >
    </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
    </SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));

```

### Smithchart series in React Smithchart component

You can add any number of series to Smith chart as needed. By using the series setting, you can add or customize the data. For the points and data source added in the series, line can be drawn. You can customize the each series with marker, data label, animation, opacity, and so on.

#### points or data source

For adding values in the Smith chart, you can use either points or data source in the series. Both the points and data source should be an array of object, which should contain the field names resistance and rectangle.

#### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';

```

```
function App() {
  return (<SmithchartComponent id='smithchart'>
    <SmithchartSeriesCollectionDirective>
      <SmithchartSeriesDirective points=[
        { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
      },
        { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },
        { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },
        { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },
        { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
        { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },
      ]} name='Transmission1'>
      </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
  </SmithchartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));
```

## INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SmithchartComponent id='smithchart'>
    <SmithchartSeriesCollectionDirective>
      <SmithchartSeriesDirective
        points=[
          { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
          { resistance: 6, reactance: 4.5 }, {
resistance: 4.5, reactance: 2 },
          { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
          { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },
          { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
          { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
        ]} name='Transmission1'
      >
      </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
  </SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));
```

### Series customization

Using the following options in series settings, you can customize the each series in Smith chart:

- `[fill]`: Customizes the fill color for the series.
- `[enableSmartLabels]`: Places the data labels on Smith chart without overlapping with each other.
- `[visibility]`: Handles the visibility of the series.
- `[opacity]`: Controls the opacity of the series line.
- `[width]`: Customizes the width of the series line.

### INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
  function load(args) {
    args.smithchart.series[0].fill = '#009933';
    args.smithchart.series[0].visibility = 'visible';
    args.smithchart.series[0].opacity = 0.75;
    args.smithchart.series[0].width = 2.5;
    args.smithchart.series[0].marker = { dataLabel: { visible: true } };
  }
  return (<SmithchartComponent id='smithchart' load={load.bind(this)}>
    <SmithchartSeriesCollectionDirective>
      <SmithchartSeriesDirective points=[
        { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
      },
        { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
      2 },
        { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
      reactance: 1.3 },
        { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
      1 },
        { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
      0.4 },
        { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
      0.15 },
      ]} name='Transmission1'>
        </SmithchartSeriesDirective>
      </SmithchartSeriesCollectionDirective>
    </SmithchartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));
```

### INDEX.TSX

```
{% raw %}
import * as React from "react";
```

```

import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
    function load(args: ISmithchartLoadedEventArgs): void {
        args.smithchart.series[0].fill = '#009933';
        args.smithchart.series[0].visibility = 'visible';
        args.smithchart.series[0].opacity = 0.75;
        args.smithchart.series[0].width = 2.5;
        args.smithchart.series[0].marker= { dataLabel: { visible: true }};
    }
    return ( <SmithchartComponent id='smithchart' load={load.bind(this)}>
        <SmithchartSeriesCollectionDirective>
            <SmithchartSeriesDirective
                points=[
                    { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
                    { resistance: 6, reactance: 4.5 }, {
resistance: 4.5, reactance: 2 },
                    { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
                    { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },
                    { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
                    { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
                ]} name='Transmission1'
            >
        </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
    </SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));
{% enddraw %}

```

## Smithchart print in React Smithchart component

### Print

The rendered Smith chart can be printed directly from browser by calling the public method print. The ID of the Smith chart's div element must be passed as an argument to the public method.

### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
    let smithchartInstance;
    return (<SmithchartComponent id='smithchart'>
        <SmithchartSeriesCollectionDirective>
            <SmithchartSeriesDirective points=[
                { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
},

```



```

        { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },
        { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },
        { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },
        { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
        { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },
    ]] name='Transmission1'>
        </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
</SmithchartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));
document.getElementById('print').onclick = () => {
    App.smithchartInstance.print();
};

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
    let smithchartInstance: SmithchartComponent;
    return ( <SmithchartComponent id='smithchart'>
        <SmithchartSeriesCollectionDirective>
            <SmithchartSeriesDirective
                points=[
                    { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
                    { resistance: 6, reactance: 4.5 }, {
resistance: 4.5, reactance: 2 },
                    { resistance: 3.5, reactance: 1.6 }, {
resistance: 2.5, reactance: 1.3 },
                    { resistance: 2, reactance: 1.2 }, {
resistance: 1.5, reactance: 1 },
                    { resistance: 1, reactance: 0.8 }, {
resistance: 0.5, reactance: 0.4 },
                    { resistance: 0.3, reactance: 0.2 }, {
resistance: 0, reactance: 0.15 },
                ]] name='Transmission1'
            >
        </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
</SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));
document.getElementById('print').onclick = () => {
    App.smithchartInstance.print();
};

```

```
};
```

## Export

The rendered Smith chart can be exported to JPEG, PNG, SVG, or PDF format using the export method. The input parameters for this method are Export Type for format and fileName for result.

## INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
  let smithchartInstance;
  return (<SmithchartComponent id='smithchart'>
    <SmithchartSeriesCollectionDirective>
      <SmithchartSeriesDirective points=[
        { resistance: 10, reactance: 25 }, { resistance: 8, reactance: 6
      },
        { resistance: 6, reactance: 4.5 }, { resistance: 4.5, reactance:
2 },
        { resistance: 3.5, reactance: 1.6 }, { resistance: 2.5,
reactance: 1.3 },
        { resistance: 2, reactance: 1.2 }, { resistance: 1.5, reactance:
1 },
        { resistance: 1, reactance: 0.8 }, { resistance: 0.5, reactance:
0.4 },
        { resistance: 0.3, reactance: 0.2 }, { resistance: 0, reactance:
0.15 },
      ] name='Transmission1'>
      </SmithchartSeriesDirective>
    </SmithchartSeriesCollectionDirective>
  </SmithchartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));
document.getElementById('print').onclick = () => {
  App.smithchartInstance.export();
};
```

## INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { SmithchartComponent, SmithchartSeriesCollectionDirective,
SmithchartSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
  let smithchartInstance: SmithchartComponent;
  return ( <SmithchartComponent id='smithchart'>
    <SmithchartSeriesCollectionDirective>
      <SmithchartSeriesDirective
        points=[
          { resistance: 10, reactance: 25 }, {
resistance: 8, reactance: 6 },
```

```

resistance: 4.5, reactance: 2 }, { resistance: 6, reactance: 4.5 }, {
resistance: 2.5, reactance: 1.3 }, { resistance: 3.5, reactance: 1.6 }, {
resistance: 1.5, reactance: 1 }, { resistance: 2, reactance: 1.2 }, {
resistance: 0.5, reactance: 0.4 }, { resistance: 1, reactance: 0.8 }, {
resistance: 0, reactance: 0.15 }, { resistance: 0.3, reactance: 0.2 }, {
    ]} name='Transmission1'
    >
    </SmithchartSeriesDirective>
  </SmithchartSeriesCollectionDirective>
</SmithchartComponent> );
};
export default App;
ReactDOM.render(<App />, document.getElementById('smithchart'));
document.getElementById('print').onclick = () => {
  App.smithchartInstance.export();
};

```

### Accessibility in React Smithchart component

The Smith chart component followed the accessibility guidelines and standards, including [ADA, Section 508](#), [WCAG 2.2](#) standards, and [WCAG roles](#) that are commonly used to evaluate accessibility.

The accessibility compliance for the Smith chart component is outlined below.

| Accessibility Criteria | Compatibility |

| -- | -- |

| [WCAG 2.2](#) Support |  |

| [Section 508](#) Support |  |

| Screen Reader Support |  |

| Color Contrast |  |

| Mobile Device Support |  |

| Keyboard Navigation Support |  |

| [Accessibility Checker](#) Validation |  |

| [Axe-core](#) Accessibility Validation |  |

<style>

```
.post .post-content img {
display: inline-block;
margin: 0.5em 0;
}
</style>
<div> - All
features of the component meet the requirement.</div>
<div> - Some features of the component do not meet the requirement.</div>
<div> - The component does not meet the requirement.</div>
```

### WAI-ARIA attributes

The Smith chart component followed the [WAI-ARIA](#) patterns to meet the accessibility. The following ARIA attributes are used in the Smith chart component:

- `img (role)`
- `region (role)`
- `aria-label (attribute)`
- `aria-hidden (attribute)`

### Keyboard interaction

The Smith chart component followed the [keyboard interaction](#) guideline, making it easy for people who use assistive technologies (AT) and those who completely rely on keyboard navigation. The following keyboard shortcuts are supported by the Smith chart component.

| **Press** | **To do this** |

| --- | --- |

| **Tab** | Moves the focus to the next element in the Smith chart. |

| **Shift + Tab** | Moves the focus to the previous element in the Smith chart. |

| **Ctrl + P** | Prints the Smith chart. |

### Ensuring accessibility

The Smith chart component's accessibility levels are ensured through an [accessibility-checker](#) and [axe-core](#) software tools during automated testing.

The accessibility compliance of the Smith chart component is shown in the following sample. Open the [sample](#) in a new window to evaluate the accessibility of the Smith chart component with accessibility tools.

See also

- [Accessibility in Syncfusion React components](#)

## Sparkline

### Getting Started

This section explains you the steps required to create a sparkline and demonstrate its basic usage.

### Dependencies

The following list of minimum dependencies are required to use the sparkline:

```
`ts
|-- @syncfusion/ej2-react-charts
|-- @syncfusion/ej2-base
|-- @syncfusion/ej2-data
|-- @syncfusion/ej2-charts
|-- @syncfusion/ej2-svg-base
|-- @syncfusion/ej2-react-base
|-- @syncfusion/ej2-pdf-export
|-- @syncfusion/ej2-file-utils
|-- @syncfusion/ej2-compression
`,`
```

### Installation and Configuration

You can use [create-react-app](#) to setup the applications.

To install `create-react-app` run the following command.

```
`
npm install -g create-react-app
`,`
```

- To set-up a React application in TypeScript environment, run the following command.

```
`
create-react-app quickstart --template typescript
cd quickstart
npm start
`,`
```

- To set-up a React application in JavaScript environment, run the following command.

```
`
create-react-app quickstart
cd quickstart
`,`
```

npm start

,

- Install Syncfusion packages using below command.

,

npm install @syncfusion/ej2-react-charts --save

,

### Add Sparkline to the Project

Now, you can start adding Sparkline component in the application. For getting started, add the Sparkline component in `src/index.tsx` file using following code.

```
`ts
```

```
import * as React from 'react';
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
class App extends React.Component {
  render() {
    return ( <SparklineComponent></SparklineComponent> );
  }
}
export default App;
```

,

Now run the `npm start` command in the console, it will run your application and open the browser window.

,

npm start

,

Since the data source has not been specified to the sparkline, no shape will be rendered. Only an empty SVG element is appended to the sparkline container.

### Module Injection

The sparkline component is segregated into individual feature-wise modules. To use a particular feature, inject its feature module using the `<Inject services={} />` method. The module available in sparkline and its descriptions is as follows.

- SparklineTooltip - Inject this provider to use tooltip series.

In the current application, the above basic sparkline is modified to visualize the types of sparkline.

In this application, tooltip feature of the sparkline is used. Now, import the SparklineTooltip module from the sparkline package, and inject it into the sparkline control using the `Sparkline.Inject` method.

```
`ts
import * as React from 'react';
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
class App extends React.Component {
  render() {
    return ( <SparklineComponent><Inject services={[SparklineTooltip]} /></SparklineComponent> );
  }
}
export default App;
```

### Bind data source to Sparkline

The `dataSource` property is used for binding data source to the sparkline. This property takes the collection value as input. For example, the list of objects can be provided as input.

#### INDEX.JSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return ( <SparklineComponent id='sparkline' height='100px'
width='70%' dataSource=[
      { x: 0, xval: '2005', yval: 20090440 },
      { x: 1, xval: '2006', yval: 20264080 },
      { x: 2, xval: '2007', yval: 20434180 },
      { x: 3, xval: '2008', yval: 21007310 },
      { x: 4, xval: '2009', yval: 21262640 },
      { x: 5, xval: '2010', yval: 21515750 },
      { x: 6, xval: '2011', yval: 21766710 },
      { x: 7, xval: '2012', yval: 22015580 },
      { x: 8, xval: '2013', yval: 22262500 },
      { x: 9, xval: '2014', yval: 22507620 },
    ] xName='xval' yName='yval'>
    </SparklineComponent> );
  }
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

#### INDEX.TSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SparklineComponent id='sparkline'
height='100px' width='70%'
dataSource=[
```

```

    { x: 0, xval: '2005', yval: 20090440 },
    { x: 1, xval: '2006', yval: 20264080 },
    { x: 2, xval: '2007', yval: 20434180 },
    { x: 3, xval: '2008', yval: 21007310 },
    { x: 4, xval: '2009', yval: 21262640 },
    { x: 5, xval: '2010', yval: 21515750 },
    { x: 6, xval: '2011', yval: 21766710 },
    { x: 7, xval: '2012', yval: 22015580 },
    { x: 8, xval: '2013', yval: 22262500 },
    { x: 9, xval: '2014', yval: 22507620 },
  ]}
  xName='xval' yName='yval'>
</SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));

```

### Change the type of Sparkline

You can change the sparkline type by setting the [type] property to [Line], [Column], [WinLoss], [Pie], or [Area]. Here, the sparkline type has been set to [area].

#### INDEX.JSX

```

import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return (
      <SparklineComponent id='sparkline' height='100px'
        width='70%' dataSource=[
          { x: 0, xval: '2005', yval: 20090440 },
          { x: 1, xval: '2006', yval: 20264080 },
          { x: 2, xval: '2007', yval: 20434180 },
          { x: 3, xval: '2008', yval: 21007310 },
          { x: 4, xval: '2009', yval: 21262640 },
          { x: 5, xval: '2010', yval: 21515750 },
          { x: 6, xval: '2011', yval: 21766710 },
          { x: 7, xval: '2012', yval: 22015580 },
          { x: 8, xval: '2013', yval: 22262500 },
          { x: 9, xval: '2014', yval: 22507620 },
        ] xName='xval' yName='yval' type='Area'>
      </SparklineComponent>);
  }
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));

```

#### INDEX.TSX

```

import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
  return (
    <SparklineComponent id='sparkline'
      height='100px' width='70%'

```



```

    dataSource=[
      { x: 0, xval: '2005', yval: 20090440 },
      { x: 1, xval: '2006', yval: 20264080 },
      { x: 2, xval: '2007', yval: 20434180 },
      { x: 3, xval: '2008', yval: 21007310 },
      { x: 4, xval: '2009', yval: 21262640 },
      { x: 5, xval: '2010', yval: 21515750 },
      { x: 6, xval: '2011', yval: 21766710 },
      { x: 7, xval: '2012', yval: 22015580 },
      { x: 8, xval: '2013', yval: 22262500 },
      { x: 9, xval: '2014', yval: 22507620 },
    ]
    xName='xval' yName='yval' type='Area'>
  </SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));

```

### Enable tooltip for Sparkline

The sparkline displays additional information through tooltip when the mouse is hovered over the sparkline. You can enable tooltip by setting the `[visible]` property to true in `[tooltipSettings]` and injecting `SparklineTooltip` module using the `Sparkline.Inject(SparklineTooltip)` method.

#### INDEX.JSX

```

{% raw %}
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { SparklineComponent, Inject, SparklineTooltip } from
 '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return (
      <SparklineComponent id='sparkline' height='100px'
        width='70%' tooltipSettings={{
          visible: true, format: '${xval} : ${yval}',
        }} dataSource=[
          { x: 0, xval: '2005', yval: 20090440 },
          { x: 1, xval: '2006', yval: 20264080 },
          { x: 2, xval: '2007', yval: 20434180 },
          { x: 3, xval: '2008', yval: 21007310 },
          { x: 4, xval: '2009', yval: 21262640 },
          { x: 5, xval: '2010', yval: 21515750 },
          { x: 6, xval: '2011', yval: 21766710 },
          { x: 7, xval: '2012', yval: 22015580 },
          { x: 8, xval: '2013', yval: 22262500 },
          { x: 9, xval: '2014', yval: 22507620 },
        ] xName='xval' yName='yval' type='Area'>
        <Inject services={[SparklineTooltip]} />
      </SparklineComponent>);
  }
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));
{% endraw %}

```

**INDEX.TSX**

```
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent, Inject, SparklineTooltip }
  from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SparklineComponent id='sparkline'
    height='100px' width='70%'
    tooltipSettings={ {
      visible: true, format: '${xval} : ${yval}',
    } }
    dataSource=[
      { x: 0, xval: '2005', yval: 20090440 },
      { x: 1, xval: '2006', yval: 20264080 },
      { x: 2, xval: '2007', yval: 20434180 },
      { x: 3, xval: '2008', yval: 21007310 },
      { x: 4, xval: '2009', yval: 21262640 },
      { x: 5, xval: '2010', yval: 21515750 },
      { x: 6, xval: '2011', yval: 21766710 },
      { x: 7, xval: '2012', yval: 22015580 },
      { x: 8, xval: '2013', yval: 22262500 },
      { x: 9, xval: '2014', yval: 22507620 },
    ]
    xName='xval' yName='yval' type='Area'>
    <Inject services={ [SparklineTooltip] } />
  </SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

## Sparkline dimensions in React Sparkline component

## Size for container

Sparkline can be rendered to its container size. You can set the size through inline or CSS as shown in the following code.

```
,
<div id='container'>
  <div id='element' style="width:650px; height:350px;"></div>
</div>
,
```

**INDEX.JSX**

```
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return ( <SparklineComponent id='sparkline' height='350px'
      width='650px' dataSource=[
        { x: 0, xval: '2005', yval: 20090440 },
        { x: 1, xval: '2006', yval: 20264080 },

```

```

        { x: 2, xval: '2007', yval: 20434180 },
        { x: 3, xval: '2008', yval: 21007310 },
        { x: 4, xval: '2009', yval: 21262640 },
        { x: 5, xval: '2010', yval: 21515750 },
        { x: 6, xval: '2011', yval: 21766710 },
        { x: 7, xval: '2012', yval: 22015580 },
        { x: 8, xval: '2013', yval: 22262500 },
        { x: 9, xval: '2014', yval: 22507620 },
      ]}
      // Assign the dataSource values to series of sparkline 'xName and
      yName'
      xName='xval' yName='yval'>
    </SparklineComponent>;
  }
}
ReactDOM.render(<App />, document.getElementById('sparkline'));

```

**INDEX.TSX**

```

import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SparklineComponent id='sparkline'
    height='350px' width='650px'
    dataSource= { [
      { x: 0, xval: '2005', yval: 20090440 },
      { x: 1, xval: '2006', yval: 20264080 },
      { x: 2, xval: '2007', yval: 20434180 },
      { x: 3, xval: '2008', yval: 21007310 },
      { x: 4, xval: '2009', yval: 21262640 },
      { x: 5, xval: '2010', yval: 21515750 },
      { x: 6, xval: '2011', yval: 21766710 },
      { x: 7, xval: '2012', yval: 22015580 },
      { x: 8, xval: '2013', yval: 22262500 },
      { x: 9, xval: '2014', yval: 22507620 },
    ] }
    // Assign the dataSource values to series of sparkline 'xName and yName'
    xName= 'xval' yName= 'yval'>
  </SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));

```

<!-- markdownlint-disable MD036 -->

[Size for sparkline](#)

<!-- markdownlint-disable MD036 -->

You can also set the size for sparkline directly using the [width](#) and [height](#) properties.

**In pixel**

You can set the size for sparkline in pixel as demonstrated in the following code.

**INDEX.JSX**

```
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return (<SparklineComponent id='sparkline' height='150' width='350'
    dataSource=[
      { x: 0, xval: '2005', yval: 20090440 },
      { x: 1, xval: '2006', yval: 20264080 },
      { x: 2, xval: '2007', yval: 20434180 },
      { x: 3, xval: '2008', yval: 21007310 },
      { x: 4, xval: '2009', yval: 21262640 },
      { x: 5, xval: '2010', yval: 21515750 },
      { x: 6, xval: '2011', yval: 21766710 },
      { x: 7, xval: '2012', yval: 22015580 },
      { x: 8, xval: '2013', yval: 22262500 },
      { x: 9, xval: '2014', yval: 22507620 },
    ]
    // Assign the dataSource values to series of sparkline 'xName and
    yName'
    xName='xval' yName='yval'>
    </SparklineComponent>);
  }
}
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

## INDEX.TSX

```
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SparklineComponent id='sparkline'
  height='150' width='350'
  dataSource= { [
    { x: 0, xval: '2005', yval: 20090440 },
    { x: 1, xval: '2006', yval: 20264080 },
    { x: 2, xval: '2007', yval: 20434180 },
    { x: 3, xval: '2008', yval: 21007310 },
    { x: 4, xval: '2009', yval: 21262640 },
    { x: 5, xval: '2010', yval: 21515750 },
    { x: 6, xval: '2011', yval: 21766710 },
    { x: 7, xval: '2012', yval: 22015580 },
    { x: 8, xval: '2013', yval: 22262500 },
    { x: 9, xval: '2014', yval: 22507620 },
  ] }
  // Assign the dataSource values to series of sparkline 'xName and yName'
  xName= 'xval' yName= 'yval'>
  </SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

## In percentage

By setting values in percentage, sparkline gets its dimension with respect to its container. For example, when the height is set to '50%', sparkline is rendered to half of its container height.

### INDEX.JSX

```
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return (<SparklineComponent id='sparkline' height='50%' width='350%'
    dataSource=[
      { x: 0, xval: '2005', yval: 20090440 },
      { x: 1, xval: '2006', yval: 20264080 },
      { x: 2, xval: '2007', yval: 20434180 },
      { x: 3, xval: '2008', yval: 21007310 },
      { x: 4, xval: '2009', yval: 21262640 },
      { x: 5, xval: '2010', yval: 21515750 },
      { x: 6, xval: '2011', yval: 21766710 },
      { x: 7, xval: '2012', yval: 22015580 },
      { x: 8, xval: '2013', yval: 22262500 },
      { x: 9, xval: '2014', yval: 22507620 },
    ]
    // Assign the dataSource values to series of sparkline 'xName and
    yName'
    xName='xval' yName='yval'>
    </SparklineComponent>);
  }
}
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

### INDEX.TSX

```
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SparklineComponent id='sparkline'
  height='50%' width='350%'
  dataSource= { [
    { x: 0, xval: '2005', yval: 20090440 },
    { x: 1, xval: '2006', yval: 20264080 },
    { x: 2, xval: '2007', yval: 20434180 },
    { x: 3, xval: '2008', yval: 21007310 },
    { x: 4, xval: '2009', yval: 21262640 },
    { x: 5, xval: '2010', yval: 21515750 },
    { x: 6, xval: '2011', yval: 21766710 },
    { x: 7, xval: '2012', yval: 22015580 },
    { x: 8, xval: '2013', yval: 22262500 },
    { x: 9, xval: '2014', yval: 22507620 },
  ] }
  // Assign the dataSource values to series of sparkline 'xName and yName'
  xName= 'xval' yName= 'yval'>
  </SparklineComponent> );
}
export default App;
```

```
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

## Sparkline types in React Sparkline component

Different types of shapes can be used to represent the sparkline. You can change the sparkline type by setting the type property. Sparkline supports the following types:

- Line
- Column
- Win-Loss
- Pie
- Area

The following code sample shows different types of sparklines.

<!-- markdownlint-disable MD036 -->

### Line

The [Line] type is used to render the sparkline series as line.

#### INDEX.JSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return (<SparklineComponent id='sparkline' height='100px'
width='70%' dataSource=[
      { x: 0, xval: '2005', yval: 20090440 },
      { x: 1, xval: '2006', yval: 20264080 },
      { x: 2, xval: '2007', yval: 20434180 },
      { x: 3, xval: '2008', yval: 21007310 },
      { x: 4, xval: '2009', yval: 21262640 },
      { x: 5, xval: '2010', yval: 21515750 },
      { x: 6, xval: '2011', yval: 21766710 },
      { x: 7, xval: '2012', yval: 22015580 },
      { x: 8, xval: '2013', yval: 22262500 },
      { x: 9, xval: '2014', yval: 22507620 },
    ]
    // Assign the dataSource values to series of sparkline 'xName and
yName'
    xName='xval' yName='yval'
    // Assign 'Line' as type of the sparkline
    type='Line'>
    </SparklineComponent>);
  }
}
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

#### INDEX.TSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
```

```
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SparklineComponent id='sparkline'
    height='100px' width='70%'
    dataSource= { [
      { x: 0, xval: '2005', yval: 20090440 },
      { x: 1, xval: '2006', yval: 20264080 },
      { x: 2, xval: '2007', yval: 20434180 },
      { x: 3, xval: '2008', yval: 21007310 },
      { x: 4, xval: '2009', yval: 21262640 },
      { x: 5, xval: '2010', yval: 21515750 },
      { x: 6, xval: '2011', yval: 21766710 },
      { x: 7, xval: '2012', yval: 22015580 },
      { x: 8, xval: '2013', yval: 22262500 },
      { x: 9, xval: '2014', yval: 22507620 },
    ] }
    // Assign the dataSource values to series of sparkline 'xName and yName'
    xName= 'xval' yName= 'yval'
    // Assign 'Line' as type of the sparkline
    type= 'Line' >
  </SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

## Column

The [Column] type is used to render the sparkline series as column.

### INDEX.JSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return ( <SparklineComponent id='sparkline' height='150px'
    width='200px' dataSource=[
      { x: 0, xval: '2005', yval: 20090440 },
      { x: 1, xval: '2006', yval: 20264080 },
      { x: 2, xval: '2007', yval: 20434180 },
      { x: 3, xval: '2008', yval: 21007310 },
      { x: 4, xval: '2009', yval: 21262640 },
      { x: 5, xval: '2010', yval: 21515750 },
      { x: 6, xval: '2011', yval: 21766710 },
      { x: 7, xval: '2012', yval: 22015580 },
      { x: 8, xval: '2013', yval: 22262500 },
      { x: 9, xval: '2014', yval: 22507620 },
    ] }
    // Assign the dataSource values to series of sparkline 'xName and yName'
    xName='xval' yName='yval'
    // Assign 'Column' as type of the sparkline
    type='Column'>
  </SparklineComponent>);
}
}
```

```
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

### INDEX.TSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SparklineComponent id='sparkline'
    height='150px' width='200px'
    dataSource= { [
      { x: 0, xval: '2005', yval: 20090440 },
      { x: 1, xval: '2006', yval: 20264080 },
      { x: 2, xval: '2007', yval: 20434180 },
      { x: 3, xval: '2008', yval: 21007310 },
      { x: 4, xval: '2009', yval: 21262640 },
      { x: 5, xval: '2010', yval: 21515750 },
      { x: 6, xval: '2011', yval: 21766710 },
      { x: 7, xval: '2012', yval: 22015580 },
      { x: 8, xval: '2013', yval: 22262500 },
      { x: 9, xval: '2014', yval: 22507620 },
    ] }
    // Assign the dataSource values to series of sparkline 'xName and yName'
    xName= 'xval' yName= 'yval'
    // Assign 'Column' as type of the sparkline
    type= 'Column' >
  </SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

### Pie

The [Pie] type is used to render the sparkline series as pie.

### INDEX.JSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return ( <SparklineComponent id='sparkline' height='200px'
    width='70%' dataSource={ [
      { x: 0, xval: '2005', yval: 20090440 },
      { x: 1, xval: '2006', yval: 20264080 },
      { x: 2, xval: '2007', yval: 20434180 },
      { x: 3, xval: '2008', yval: 21007310 },
      { x: 4, xval: '2009', yval: 21262640 },
      { x: 5, xval: '2010', yval: 21515750 },
      { x: 6, xval: '2011', yval: 21766710 },
      { x: 7, xval: '2012', yval: 22015580 },
      { x: 8, xval: '2013', yval: 22262500 },
      { x: 9, xval: '2014', yval: 22507620 },
    ] }
    type= 'Pie' >
  </SparklineComponent> );
  }
}
```



```
// Assign the dataSource values to series of sparkline 'xName and
yName'
xName='xval' yName='yval'
// Assign 'Pie' as type of the sparkline
type='Pie'>
</SparklineComponent>;
}
}
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

**INDEX.TSX**

```
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SparklineComponent id='sparkline'
    height='200px' width='70%'
    dataSource= { [
      { x: 0, xval: '2005', yval: 20090440 },
      { x: 1, xval: '2006', yval: 20264080 },
      { x: 2, xval: '2007', yval: 20434180 },
      { x: 3, xval: '2008', yval: 21007310 },
      { x: 4, xval: '2009', yval: 21262640 },
      { x: 5, xval: '2010', yval: 21515750 },
      { x: 6, xval: '2011', yval: 21766710 },
      { x: 7, xval: '2012', yval: 22015580 },
      { x: 8, xval: '2013', yval: 22262500 },
      { x: 9, xval: '2014', yval: 22507620 },
    ] }
    // Assign the dataSource values to series of sparkline 'xName and yName'
    xName= 'xval' yName= 'yval'
    // Assign 'Pie' as type of the sparkline
    type= 'Pie' >
  </SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

**Win Loss**

The [WinLoss] type is used to render the sparkline series as Win Loss.

**INDEX.JSX**

```
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return ( <SparklineComponent id='sparkline' height='100px'
    width='70%' dataSource={ [
      { x: 0, xval: '2005', yval: 20090440 },
      { x: 1, xval: '2006', yval: 20264080 },
      { x: 2, xval: '2007', yval: -20434180 },
      { x: 3, xval: '2008', yval: 21007310 },
    ] }
  >
```

```

        { x: 4, xval: '2009', yval: 21262640 },
        { x: 5, xval: '2010', yval: -21515750 },
        { x: 6, xval: '2011', yval: 21766710 },
        { x: 7, xval: '2012', yval: 22015580 },
        { x: 8, xval: '2013', yval: -22262500 },
        { x: 9, xval: '2014', yval: 22507620 },
      ]}
      // Assign the dataSource values to series of sparkline 'xName and
yName'
      xName='xval' yName='yval'
      // Assign 'WinLoss' as type of the sparkline
      type='WinLoss'>
    </SparklineComponent>;
  }
}
ReactDOM.render(<App />, document.getElementById('sparkline'));

```

**INDEX.TSX**

```

import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SparklineComponent id='sparkline'
    height='100px' width='70%'
    dataSource= { [
      { x: 0, xval: '2005', yval: 20090440 },
      { x: 1, xval: '2006', yval: 20264080 },
      { x: 2, xval: '2007', yval: -20434180 },
      { x: 3, xval: '2008', yval: 21007310 },
      { x: 4, xval: '2009', yval: 21262640 },
      { x: 5, xval: '2010', yval: -21515750 },
      { x: 6, xval: '2011', yval: 21766710 },
      { x: 7, xval: '2012', yval: 22015580 },
      { x: 8, xval: '2013', yval: -22262500 },
      { x: 9, xval: '2014', yval: 22507620 },
    ] }
    // Assign the dataSource values to series of sparkline 'xName and yName'
    xName= 'xval' yName= 'yval'
    // Assign 'WinLoss' as type of the sparkline
    type= 'WinLoss' >
  </SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));

```

**Area**

The [Area] type is used to render the sparkline series as area.

**INDEX.JSX**

```

import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {

```

```

render() {
    return (<SparklineComponent id='sparkline' height='100px'
width='70%' dataSource=[
        { x: 0, xval: '2005', yval: 20090440 },
        { x: 1, xval: '2006', yval: 20264080 },
        { x: 2, xval: '2007', yval: 20434180 },
        { x: 3, xval: '2008', yval: 21007310 },
        { x: 4, xval: '2009', yval: 21262640 },
        { x: 5, xval: '2010', yval: 21515750 },
        { x: 6, xval: '2011', yval: 21766710 },
        { x: 7, xval: '2012', yval: 22015580 },
        { x: 8, xval: '2013', yval: 22262500 },
        { x: 9, xval: '2014', yval: 22507620 },
    ]
    // Assign the dataSource values to series of sparkline 'xName and
yName'
    xName='xval' yName='yval'
    // Assign 'Area' as type of the sparkline
    type='Area'>
    </SparklineComponent>);
}
}
ReactDOM.render(<App />, document.getElementById('sparkline'));

```

**INDEX.TSX**

```

import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
    return ( <SparklineComponent id='sparkline'
height='100px' width='70%'
dataSource= { [
        { x: 0, xval: '2005', yval: 20090440 },
        { x: 1, xval: '2006', yval: 20264080 },
        { x: 2, xval: '2007', yval: 20434180 },
        { x: 3, xval: '2008', yval: 21007310 },
        { x: 4, xval: '2009', yval: 21262640 },
        { x: 5, xval: '2010', yval: 21515750 },
        { x: 6, xval: '2011', yval: 21766710 },
        { x: 7, xval: '2012', yval: 22015580 },
        { x: 8, xval: '2013', yval: 22262500 },
        { x: 9, xval: '2014', yval: 22507620 },
    ] }
    // Assign the dataSource values to series of sparkline 'xName and yName'
    xName= 'xval' yName= 'yval'
    // Assign 'Area' as type of the sparkline
    type= 'Area' >
    </SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));

```

**Axis customization in React Sparkline component**

You can customize axis value types and min and max values of the sparkline.

### Change value type of the sparkline

You can change the sparkline value type by setting the `valueType` property to `Numeric`, `Category`, or `DateTime`.

<!-- markdownlint-disable MD036 -->

### DateTime

You can assign date-time values to the sparkline by setting the `valueType` property to `DateTime`.

### INDEX.JSX

```
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return (
      <SparklineComponent id='sparkline' height='150px'
width='130px' dataSource=[
        { xDate: new Date(2018, 0, 1), x: 0, yval: 4 },
        { xDate: new Date(2018, 0, 2), x: 1, yval: 4.5 },
        { xDate: new Date(2018, 0, 3), x: 2, yval: 8 },
        { xDate: new Date(2018, 0, 4), x: 3, yval: 7 },
        { xDate: new Date(2018, 0, 5), x: 4, yval: 6 },
        { xDate: new Date(2018, 0, 8), x: 5, yval: 8 },
        { xDate: new Date(2018, 0, 9), x: 6, yval: 8 },
        { xDate: new Date(2018, 0, 10), x: 7, yval: 6.5 },
        { xDate: new Date(2018, 0, 11), x: 8, yval: 4 },
        { xDate: new Date(2018, 0, 12), x: 9, yval: 5.5 }
      ]
      // Assign the dataSource values to series of sparkline 'xName and
yName'
      xName='xDate' yName='yval'
      // Assign the "Column" as type of the sparkline
      type='Column'
      // Assign the "DateTime" as value type to the sparkline
      valueType='DateTime'>
    </SparklineComponent>);
  }
}
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

### INDEX.TSX

```
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
  return (
    <SparklineComponent id='sparkline'
height='150px' width='130px'
dataSource= { [
      { xDate: new Date(2018, 0, 1), x: 0, yval: 4 },
      { xDate: new Date(2018, 0, 2), x: 1, yval: 4.5 },
      { xDate: new Date(2018, 0, 3), x: 2, yval: 8 },
      { xDate: new Date(2018, 0, 4), x: 3, yval: 7 },
      { xDate: new Date(2018, 0, 5), x: 4, yval: 6 },
      { xDate: new Date(2018, 0, 8), x: 5, yval: 8 },
    ]
    }
  >
  </SparklineComponent>
  );
}
```

```

    { xDate: new Date(2018, 0, 9), x: 6, yval: 8 },
    { xDate: new Date(2018, 0, 10), x: 7, yval: 6.5 },
    { xDate: new Date(2018, 0, 11), x: 8, yval: 4 },
    { xDate: new Date(2018, 0, 12), x: 9, yval: 5.5 }
  ] }
  // Assign the dataSource values to series of sparkline 'xName and yName'
  xName= 'xDate' yName= 'yval'
  // Assign the "Column" as type of the sparkline
  type= 'Column'
  // Assign the "DateTime" as value type to the sparkline
  valueType= 'DateTime'>
</SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));

```

<!-- markdownlint-disable MD036 -->

### Category

You can assign category values to the sparkline by setting [valueType] to [Category].

#### INDEX.JSX

```

import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return (<SparklineComponent id='sparkline' height='150px'
width='130px' dataSource=[
      { x: 0, xval: 'Robert', yval: 60 },
      { x: 1, xval: 'Andrew', yval: 65 },
      { x: 2, xval: 'Suyama', yval: 70 },
      { x: 3, xval: 'Michael', yval: 80 },
      { x: 4, xval: 'Janet', yval: 55 },
      { x: 5, xval: 'Davolio', yval: 90 },
      { x: 6, xval: 'Fuller', yval: 75 },
      { x: 7, xval: 'Nancy', yval: 85 }
    ]>
      // Assign the dataSource values to series of sparkline 'xName and
yName'
      xName='xval' yName='yval'
      // Assign the "Column" as type of the sparkline
      type='Column'
      // Assign the "Category" as value type to the sparkline
      valueType='Category'>
    </SparklineComponent>);
  }
}
ReactDOM.render(<App />, document.getElementById('sparkline'));

```

#### INDEX.TSX

```

import * as React from 'react';
import * as ReactDOM from "react-dom";

```

```
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SparklineComponent id='sparkline'
    height='150px' width='130px'
    dataSource= { [
      { x: 0, xval: 'Robert', yval: 60 },
      { x: 1, xval: 'Andrew', yval: 65 },
      { x: 2, xval: 'Suyama', yval: 70 },
      { x: 3, xval: 'Michael', yval: 80 },
      { x: 4, xval: 'Janet', yval: 55 },
      { x: 5, xval: 'Davolio', yval: 90 },
      { x: 6, xval: 'Fuller', yval: 75 },
      { x: 7, xval: 'Nancy', yval: 85 }
    ] }
    // Assign the dataSource values to series of sparkline 'xName and yName'
    xName= 'xval' yName= 'yval'
    // Assign the "Column" as type of the sparkline
    type= 'Column'
    // Assign the "Category" as value type to the sparkline
    valueType= 'Category'>
  </SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

### Numeric

You can assign numeric values to the sparkline by setting `[valueType]` to `[Numeric]`.

### INDEX.JSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return ( <SparklineComponent id='sparkline' height='150px'
    width='130px' dataSource=[
      { x: 1, xval: 2010, yval: 190 },
      { x: 2, xval: 2011, yval: 165 },
      { x: 3, xval: 2012, yval: 158 },
      { x: 4, xval: 2013, yval: 175 },
      { x: 5, xval: 2014, yval: 200 },
      { x: 6, xval: 2015, yval: 180 },
      { x: 7, xval: 2016, yval: 210 }
    ]}
    // Assign the dataSource values to series of sparkline 'xName and yName'
    xName='xval' yName='yval'
    // Assign the "Column" as type of the sparkline
    type='Column'
    // Assign the "Numeric" as value type to the sparkline
    valueType='Numeric'>
  </SparklineComponent>);
}
}
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

**INDEX.TSX**

```
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SparklineComponent id='sparkline'
    height='150px' width='130px'
    dataSource= { [
      { x: 1, xval: 2010, yval: 190 },
      { x: 2, xval: 2011, yval: 165 },
      { x: 3, xval: 2012, yval: 158 },
      { x: 4, xval: 2013, yval: 175 },
      { x: 5, xval: 2014, yval: 200 },
      { x: 6, xval: 2015, yval: 180 },
      { x: 7, xval: 2016, yval: 210 }
    ] }
    // Assign the dataSource values to series of sparkline 'xName and yName'
    xName= 'xval' yName= 'yval'
    // Assign the "Column" as type of the sparkline
    type= 'Column'
    // Assign the "Numeric" as value type to the sparkline
    valueType= 'Numeric'>
    </SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

<!-- markdownlint-disable MD036 -->

**Change min and max values of axis**

You can change the min and max values of x-axis by setting the [minX] and [maxX] values to the [axisSettings] property. You can also change the min and max values of y-axis by setting the [minY] and [maxY] values to the [axisSettings] property.

**INDEX.JSX**

```
{% raw %}
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return ( <SparklineComponent id='sparkline' height='150px'
    width='130px' dataSource={ [
      { x: 0, yval: 50 },
      { x: 1, yval: 30 },
      { x: 2, yval: 20 },
      { x: 3, yval: 30 },
      { x: 4, yval: 50 },
      { x: 5, yval: 40 },
      { x: 6, yval: 20 },
      { x: 7, yval: 10 },
      { x: 8, yval: 30 },
    ] }>
```

```

        { x: 9, yval: 10 },
        { x: 10, yval: 40 }
    ]}
    // Assign the dataSource values to series of sparkline 'xName and
yName'
    xName='x' yName='yval'
    // Set min and max values to the y-axis of sparkline
    axisSettings={ {
        minY: 0, maxY: 150
    } }
    // Assign the "Column" as type of the sparkline
    type='Column'
    // Assign the "Numeric" as value type to the sparkline
    valueType='Numeric'>
</SparklineComponent>;
    }
}
ReactDOM.render(<App />, document.getElementById('sparkline'));
{% endraw %}

```

## INDEX.TSX

```

import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
    return ( <SparklineComponent id='sparkline'
        height='150px' width='130px'
        dataSource= { [
            { x: 0, yval: 50 },
            { x: 1, yval: 30 },
            { x: 2, yval: 20 },
            { x: 3, yval: 30 },
            { x: 4, yval: 50 },
            { x: 5, yval: 40 },
            { x: 6, yval: 20 },
            { x: 7, yval: 10 },
            { x: 8, yval: 30 },
            { x: 9, yval: 10 },
            { x: 10, yval: 40 }
        ] }
        // Assign the dataSource values to series of sparkline 'xName and yName'
        xName= 'x' yName= 'yval'
        // Set min and max values to the y-axis of sparkline
        axisSettings= { {
            minY: 0, maxY: 150
        } }
        // Assign the "Column" as type of the sparkline
        type= 'Column'
        // Assign the "Numeric" as value type to the sparkline
        valueType= 'Numeric'>
    </SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));

```



### Change value of axis

You can set horizontal axis line value of the sparkline by setting `[value]` to the `[axisSettings]` property. The following code example shows this.

#### INDEX.JSX

```
{% raw %}
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return (<SparklineComponent id='sparkline' height='150px'
width='130px' dataSource=[
      { x: 0, yval: 50 },
      { x: 1, yval: 30 },
      { x: 2, yval: 20 },
      { x: 3, yval: 30 },
      { x: 4, yval: 50 },
      { x: 5, yval: 40 },
      { x: 6, yval: 20 },
      { x: 7, yval: 10 },
      { x: 8, yval: 30 },
      { x: 9, yval: 10 },
      { x: 10, yval: 40 }
    ]
    // Assign the dataSource values to series of sparkline 'xName and
yName '
    xName='x' yName='yval'
    // Set min and max values to the y-axis of sparkline
    axisSettings={{
      value: 25
    }}
    // Assign the "Column" as type of the sparkline
    type='Column'
    // Assign the "Numeric" as value type to the sparkline
    valueType='Numeric'>
    </SparklineComponent>);
  }
}
ReactDOM.render(<App />, document.getElementById('sparkline'));
{% endraw %}
```

#### INDEX.TSX

```
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SparklineComponent id='sparkline'
    height='150px' width='130px'
    dataSource= { [
      { x: 0, yval: 50 },
      { x: 1, yval: 30 },
      { x: 2, yval: 20 },
      { x: 3, yval: 30 },
    ]
  }
  >
  </SparklineComponent>);
}
```

```

        { x: 4, yval: 50 },
        { x: 5, yval: 40 },
        { x: 6, yval: 20 },
        { x: 7, yval: 10 },
        { x: 8, yval: 30 },
        { x: 9, yval: 10 },
        { x: 10, yval: 40 }
    ] }
    // Assign the dataSource values to series of sparkline 'xName and yName'
    xName= 'x' yName= 'yval'
    // Set min and max values to the y-axis of sparkline
    axisSettings= { {
        value: 25
    } }
    // Assign the "Column" as type of the sparkline
    type= 'Column'
    // Assign the "Numeric" as value type to the sparkline
    valueType= 'Numeric'>
</SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));

```

### Axis line customization

Axis of the sparkline can be collapsed using the `visible` property in `lineSettings`; this is not applicable for win-loss. You can customize the `color`, `width`, `opacity`, and `dashArray` of axis line.

### INDEX.JSX

```

{% raw %}
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
    render() {
        return (<SparklineComponent id='sparkline' height='150px'
width='130px' dataSource=[
            { x: 0, yval: 50 },
            { x: 1, yval: 30 },
            { x: 2, yval: 20 },
            { x: 3, yval: 30 },
            { x: 4, yval: 50 },
            { x: 5, yval: 40 },
            { x: 6, yval: 20 },
            { x: 7, yval: 10 },
            { x: 8, yval: 30 },
            { x: 9, yval: 10 },
            { x: 10, yval: 40 }
        ]>
            // Assign the dataSource values to series of sparkline 'xName and
yName'
            xName='x' yName='yval'
            // Set min and max values to the y-axis of sparkline
            axisSettings={
                // To configure axis line settings
                lineSettings: {

```

```

        visible: true,
        color: "#ff14ae",
        dashArray: 5
      }
    })
    // Assign the "Numeric" as value type to the sparkline
    valueType='Numeric'>
  </SparklineComponent>;
}
}
ReactDOM.render(<App />, document.getElementById('sparkline'));
{% endraw %}

```

## INDEX.TSX

```

import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SparklineComponent id='sparkline'
    height='150px' width='130px'
    dataSource= { [
      { x: 0, yval: 50 },
      { x: 1, yval: 30 },
      { x: 2, yval: 20 },
      { x: 3, yval: 30 },
      { x: 4, yval: 50 },
      { x: 5, yval: 40 },
      { x: 6, yval: 20 },
      { x: 7, yval: 10 },
      { x: 8, yval: 30 },
      { x: 9, yval: 10 },
      { x: 10, yval: 40 }
    ] }
    // Assign the dataSource values to series of sparkline 'xName and yName'
    xName= 'x' yName= 'yval'
    // Set min and max values to the y-axis of sparkline
    axisSettings= { {
      // To configure axis line settings
      lineSettings: {
        visible: true,
        color: "#ff14ae",
        dashArray:5
      }
    } }
    // Assign the "Numeric" as value type to the sparkline
    valueType= 'Numeric'>
  </SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));

```

### Special points customization in React Sparkline component

You can customize the points by initializing the point colors. The customization options allows to differentiate the [start], [end], [positive], [negative], and [low] points. This customization is only applicable for line, column, and area type sparklines.

<!-- markdownlint-disable MD036 -->

#### INDEX.JSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return (
      <SparklineComponent id='sparkline' height='150px'
width='130px' dataSource=[
      { x: 0, xval: 'AUDI', yval: 1 },
      { x: 1, xval: 'BMW', yval: 5 },
      { x: 2, xval: 'BUICK', yval: -1 },
      { x: 3, xval: 'CETROEN', yval: -6 },
      { x: 4, xval: 'CHEVROLET', yval: 0 },
      { x: 5, xval: 'FIAT', yval: 1 },
      { x: 6, xval: 'FORD', yval: -2 },
      { x: 7, xval: 'HONDA', yval: 7 },
      { x: 8, xval: 'HYUNDAI', yval: -9 },
      { x: 9, xval: 'JEEP', yval: 0 },
      { x: 10, xval: 'KIA', yval: -10 },
      { x: 11, xval: 'MAZDA', yval: 3 },
    ]
    // Assign the dataSource values to series of sparkline 'xName and
yName'
    xName='xval' yName='yval'
    // Assign 'Column' as type of the sparkline
    type='Column'
    // Assign "Category" as the value type of the sparkline
    valueType='Category'
    // To configure sparkline series highest y value point color.
    highPointColor='blue'
    // To configure sparkline series lowest y value point color.
    lowPointColor='orange'
    // To configure sparkline series first x value point color.
    startPointColor='green'
    // To configure sparkline series last x value point color.
    endPointColor='green'
    // To configure sparkline series negative y value point color.
    negativePointColor='red'
    </SparklineComponent>
  )
}
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

#### INDEX.TSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
```

```
function App() {
  return ( <SparklineComponent id='sparkline'
    height='150px' width='130px'
    dataSource= { [
      { x: 0, xval: 'AUDI', yval: 1 },
      { x: 1, xval: 'BMW', yval: 5 },
      { x: 2, xval: 'BUICK', yval: -1 },
      { x: 3, xval: 'CETROEN', yval: -6 },
      { x: 4, xval: 'CHEVROLET', yval: 0 },
      { x: 5, xval: 'FIAT', yval: 1 },
      { x: 6, xval: 'FORD', yval: -2 },
      { x: 7, xval: 'HONDA', yval: 7 },
      { x: 8, xval: 'HYUNDAI', yval: -9 },
      { x: 9, xval: 'JEEP', yval: 0 },
      { x: 10, xval: 'KIA', yval: -10 },
      { x: 11, xval: 'MAZDA', yval: 3 },
    ] }
    // Assign the dataSource values to series of sparkline 'xName and yName'
    xName= 'xval' yName= 'yval'
    // Assign 'Column' as type of the sparkline
    type= 'Column'
    // Assign "Category" as the value type of the sparkline
    valueType= 'Category'
    // To configure sparkline series highest y value point color.
    highPointColor= 'blue'
    // To configure sparkline series lowest y value point color.
    lowPointColor= 'orange'
    // To configure sparkline series first x value point color.
    startPointColor= 'green'
    // To configure sparkline series last x value point color.
    endPointColor= 'green'
    // To configure sparkline series negative y value point color.
    negativePointColor= 'red' >
  </SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

### Tie point color

Tie point color is used to configure the win-loss series type sparkline's y-value point color. The following code sample shows the tie point color of sparkline series.

#### INDEX.JSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return ( <SparklineComponent id='sparkline' height='150px'
width='130px' dataSource=[12, 15, -10, 13, 15, 6, -12, 17, 13, 0, 8, -10]
    // Assign the 'WinLoss' as type of Sparkline
    type='WinLoss'
    // Assign "Numeric" as the value type of the sparkline
    valueType='Numeric' tiePointColor='blue'>
```

```

        </SparklineComponent>;
    }
}
ReactDOM.render(<App />, document.getElementById('sparkline'));

```

### INDEX.TSX

```

import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
    return ( <SparklineComponent id='sparkline'
        height='150px' width='130px'
        dataSource= { [12, 15, -10, 13, 15, 6, -12, 17, 13, 0, 8, -10] }
        // Assign the 'WinLoss' as type of Sparkline
        type= 'WinLoss'
        // Assign "Numeric" as the value type of the sparkline
        valueType= 'Numeric'
        tiePointColor= 'blue' >
    </SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));

```

## Range band in React Sparkline component

This section explains how to customize the sparkline with multiple range bands.

### Range band customization

The range band feature is used to highlight a particular range along with the y-axis using the [startRange] and [endRange] properties. You can also customize the [color] and [opacity] of the range band.

### INDEX.JSX

```

import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
    render() {
        return ( <SparklineComponent id='sparkline' height='200px'
width='150px' lineWidth={2} fill='#0d3c9b' dataSource=[0, 6, 4, 1, 3, 2,
5]]
        // To configure range band settings
        rangeBandSettings=[
            {
                startRange: 1,
                endRange: 3,
                color: '#bfd4fc',
                opacity: 0.4
            }
        ]>
        </SparklineComponent>;
    }
}
ReactDOM.render(<App />, document.getElementById('sparkline'));

```

**INDEX.TSX**

```
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SparklineComponent id='sparkline'
    height='200px' width='150px'
    lineWidth= { 2 }
    fill= '#0d3c9b'
    dataSource= { [0, 6, 4, 1, 3, 2, 5] }
    // To configure range band settings
    rangeBandSettings= { [
      {
        startRange: 1,
        endRange: 3,
        color: '#bfd4fc',
        opacity:0.4
      }
    ] }>
</SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

**Multiple range band customization**

You can define multiple range bands to a sparkline as shown in the following code sample.

**INDEX.JSX**

```
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return ( <SparklineComponent id='sparkline' height='200px'
width='150px' lineWidth={2} fill='#0d3c9b' dataSource=[0, 6, 4, 1, 3, 2,
5]
    // To configure range band settings
    rangeBandSettings=[
      {
        startRange: 1,
        endRange: 3,
        color: '#bfd4fc',
        opacity: 0.4
      },
      {
        startRange: 4,
        endRange: 5,
        color: 'red',
        opacity: 0.4
      }
    ]>
</SparklineComponent>);
  }
}
```

```
}
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

## INDEX.TSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SparklineComponent id='sparkline'
    height='200px' width='150px'
    lineWidth= { 2 }
    fill= '#0d3c9b'
    dataSource= { [0, 6, 4, 1, 3, 2, 5] }
    // To configure range band settings
    rangeBandSettings= { [
      {
        startRange: 1,
        endRange: 3,
        color: '#bfd4fc',
        opacity:0.4
      },
      {
        startRange: 4,
        endRange: 5,
        color: 'red',
        opacity:0.4
      }
    ] }>
    </SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

## Marker in React Sparkline component

This section explains how to add markers to the sparkline.

### Adding marker to the sparkline

To add marker to the sparkline, specify the **visible** of **markerSettings** as following values. The **visible** will accept multiple values too.

- All - Enables markers for all points.
- Start - Enables marker for the start point.
- End - Enables marker for the end point.
- High - Enables marker for the high point.
- Low - Enables marker for the low point.
- Negative - Enables markers for the negative points.

The following code example shows enabling markers for all points.

## INDEX.JSX

```
{% raw %}
```



```
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return (<SparklineComponent id='sparkline' height='200px'
width='350px' axisSettings={{
      minX: -1, maxX: 7, maxY: 7, minY: -1
    }} dataSource=[0, 6, 4, 1, 3, 2, 5]}
// To enable markers for all points
markerSettings={{
  visible: ['All']
}}>
    </SparklineComponent>);
  }
}
ReactDOM.render(<App />, document.getElementById('sparkline'));
{% endraw %}
```

### INDEX.TSX

```
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SparklineComponent id='sparkline'
height='200px' width='350px'
axisSettings= { {
  minX: -1, maxX: 7, maxY: 7, minY: -1
} }
dataSource= { [0, 6, 4, 1, 3, 2, 5] }
// To enable markers for all points
markerSettings= { {
  visible: ['All']
} }>
</SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

### Adding marker to special point

In sparkline, markers can be enabled for particular points such as the start, end, low, high, or negative. The following code examples shows enabling markers for the high and low points.

### INDEX.JSX

```
{% raw %}
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return (<SparklineComponent id='sparkline' height='200px'
width='350px' axisSettings={{
      minX: -1, maxX: 7, maxY: 7, minY: -1
```

```

    }} dataSource=[3, 6, 4, 1, 3, 2, 5]}
    // To enable marker for high and low points with color customization
    highPointColor='blue' lowPointColor='red' markerSettings={{
      visible: ['High', 'Low']
    }}>
  </SparklineComponent>;
}
}
ReactDOM.render(<App />, document.getElementById('sparkline'));
{% endraw %}

```

## INDEX.TSX

```

import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SparklineComponent id='sparkline'
    height='200px' width='350px'
    axisSettings= { {
      minX: -1, maxX: 7, maxY: 7, minY: -1
    } }
    dataSource= { [3, 6, 4, 1, 3, 2, 5] }
    // To enable marker for high and low points with color customization
    highPointColor= 'blue'
    lowPointColor= 'red'
    markerSettings= { {
      visible: ['High', 'Low']
    } }>
  </SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));

```

## Customizing markers

Sparkline markers can be customized in terms of fill color, border color, width, opacity, and size. The following code example shows customizing marker's fill, border, and size.

## INDEX.JSX

```

{% raw %}
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return ( <SparklineComponent id='sparkline' height='200px'
width='350px' axisSettings={{
  minX: -1, maxX: 7, maxY: 7, minY: -1
}} dataSource=[3, 6, 4, 1, 3, 2, 5]}
    // To enable marker for high and low points with color customization
    fill='blue' markerSettings={{
      visible: ['All'],
      size: 5, fill: 'white', border: { color: 'blue', width: 2 }
    }}>

```

```

        </SparklineComponent>;
    }
}
ReactDOM.render(<App />, document.getElementById('sparkline'));
{% endraw %}

```

### INDEX.TSX

```

import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
    return ( <SparklineComponent id='sparkline'
        height='200px' width='350px'
        axisSettings= { {
            minX: -1, maxX: 7, maxY: 7, minY: -1
        } }
        dataSource= { [3, 6, 4, 1, 3, 2, 5] }
        // To enable marker for high and low points with color customization
        fill= 'blue'
        markerSettings= { {
            visible: ['All'],
            size: 5, fill: 'white', border: { color: 'blue', width: 2}
        } }>
    </SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));

```

## Data labels in React Sparkline component

Data labels are used to display values of data points to improve the readability.

### Enable data label

To enable data label for sparkline series, provide **visible** of the **dataLabelSettings** as following possible values:

- All - Enables data label of all points.
- Start - Enables data label of the start point.
- End - Enables data label of the end point.
- High - Enables data label of the high point.
- Low - Enables data label of the low point.
- Negative - Enables data labels of the negative points.

The following example shows enabling sparkline data label for all points.

### INDEX.JSX

```

{% raw %}
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
    render() {

```

```

    return (<SparklineComponent id='sparkline' height='200px'
width='350px' axisSettings={{
    minX: -1, maxX: 7, maxY: 7, minY: -1
  }} fill='blue' dataSource=[0, 6, 4, 1, 3, 2, 5]]
  // To enable data label for all points
  dataLabelSettings={{
    visible: ['All']
  }}>
    </SparklineComponent>);
  }
}
ReactDOM.render(<App />, document.getElementById('sparkline'));
{% endraw %}

```

### INDEX.TSX

```

import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
  return (<SparklineComponent id='sparkline'
    height='200px' width='350px'
    axisSettings= { {
      minX: -1, maxX: 7, maxY: 7, minY: -1
    } }
    fill= 'blue'
    dataSource= { [ 0, 6, 4, 1, 3, 2, 5 ] }
    // To enable data label for all points
    dataLabelSettings= { {
      visible: ['All']
    } }>
    </SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));

```

### Customize data label

Data labels can be customized using the fill, border, opacity, and textStyle. The following code example shows customizing data label border, text color, and fill color.

### INDEX.JSX

```

{% raw %}
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return (<SparklineComponent id='sparkline' height='200px'
width='350px' axisSettings={{
    minX: -1, maxX: 7, maxY: 7, minY: -1
  }} fill='blue' dataSource=[0, 6, 4, 1, 3, 2, 5]]
  // To customize data label fill, border, and text color
  dataLabelSettings={{
    visible: ['All'],

```

```

        border: { color: 'blue', width: 1 },
        fill: 'blue', opacity: 0.4,
        textStyle: {
            color: 'white'
        }
    }
}
</SparklineComponent>);
}
}
ReactDOM.render(<App />, document.getElementById('sparkline'));
{% endraw %}

```

### INDEX.TSX

```

import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
    return ( <SparklineComponent id='sparkline'
        height='200px' width='350px'
        axisSettings= { {
            minX: -1, maxX: 7, maxY: 7, minY: -1
        } }
        fill= 'blue'
        dataSource= { [ 0, 6, 4, 1, 3, 2, 5 ] }
        // To customize data label fill, border, and text color
        dataLabelSettings= { {
            visible: ['All'],
            border: { color: 'blue', width: 1},
            fill: 'blue', opacity: 0.4,
            textStyle: {
                color: 'white'
            }
        } }
    </SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));

```

### Format data label text

The text of data labels can be formatted using the `format` API in the sparkline `dataLabelSettings`. By default, data label shows the y-value of point. The following code example shows how to display x and y-values for points.

### INDEX.JSX

```

{% raw %}
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
    render() {
        return ( <SparklineComponent id='sparkline' height='200px'
            width='500px' axisSettings={{
                minX: -1, maxX: 7, maxY: 7, minY: -1
            }}

```

```

    }} fill='blue' valueType='Category' xName='x' yName='y'
dataSource=[
    { x: 'Mon', y: 3 },
    { x: 'Tue', y: 5 },
    { x: 'Wed', y: 2 },
    { x: 'Thu', y: 4 },
    { x: 'Fri', y: 6 },
  ]
  // To enable data label for all points
  dataLabelSettings={
    format: '${x} : ${y}',
    visible: ['All'],
    border: { color: 'blue', width: 1 },
    fill: 'blue', opacity: 0.4,
    textStyle: {
      color: 'white'
    }
  }
}>
</SparklineComponent>;
}
}
ReactDOM.render(<App />, document.getElementById('sparkline'));
{% enddraw %}

```

**INDEX.TSX**

```

import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
  return (
    <SparklineComponent id='sparkline'
      height='200px' width='500px'
      axisSettings= { {
        minX: -1, maxX: 7, maxY: 7, minY: -1
      } }
      fill= 'blue'
      valueType= 'Category'
      xName= 'x' yName= 'y'
      dataSource= { [
        {x: 'Mon', y: 3 },
        {x: 'Tue', y: 5 },
        {x: 'Wed', y: 2 },
        {x: 'Thu', y: 4 },
        {x: 'Fri', y: 6 },
      ] }
      // To enable data label for all points
      dataLabelSettings= { {
        format: '${x} : ${y}',
        visible: ['All'],
        border: { color: 'blue', width: 1},
        fill: 'blue', opacity: 0.4,
        textStyle: {
          color: 'white'
        }
      } }
    >
    </SparklineComponent> );
}

```

```

}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));

```

## User interaction in React Sparkline component

Sparkline has two user interaction features: tooltip and tracker line.

### Tooltip

The sparkline provides options to display details about values of data points through tooltips when hovering the mouse over data point. To use tooltip in sparkline, inject the `SparklineTooltip` module to sparkline using the inject method.

The following code example shows enabling tooltip for sparkline with format.

#### INDEX.JSX

```

{% raw %}
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent, Inject, SparklineTooltip } from
 '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return (<SparklineComponent id='sparkline' height='200px'
width='500px' axisSettings={{
  minX: -1, maxX: 7, maxY: 8, minY: -1
}} fill='blue' valueType='Category' xName='x' yName='y'
dataSource=[
  { x: 'Mon', y: 3 },
  { x: 'Tue', y: 5 },
  { x: 'Wed', y: 2 },
  { x: 'Thu', y: 4 },
  { x: 'Fri', y: 6 },
]
// To enable tooltip for sparkline
tooltipSettings={{
  visible: true,
  // formatting tooltip text
  format: '${x} : ${y}'
}}>
    <Inject services={[SparklineTooltip]} />
    </SparklineComponent>;
  }
}
ReactDOM.render(<App />, document.getElementById('sparkline'));
{% endraw %}

```

#### INDEX.TSX

```

import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent, Inject, SparklineTooltip
} from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SparklineComponent id='sparkline'

```

```

height='200px' width='500px'
axisSettings= { {
  minX: -1, maxX: 7, maxY: 8, minY: -1
} }
fill= 'blue'
valueType= 'Category'
xName= 'x' yName= 'y'
dataSource= { [
  {x: 'Mon', y: 3 },
  {x: 'Tue', y: 5 },
  {x: 'Wed', y: 2 },
  {x: 'Thu', y: 4 },
  {x: 'Fri', y: 6 },
] }
// To enable tooltip for sparkline
tooltipSettings= { {
  visible: true,
  // formatting tooltip text
  format: '${x} : ${y}'
} }>
<Inject services={[SparklineTooltip]} />
</SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));

```

### Tooltip customization

The fill color, text styles, format, and border of the tooltip can be customized. The following code example shows customization of tooltip's fill color and text style.

### INDEX.JSX

```

{% raw %}
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent, Inject, SparklineTooltip } from
 '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return (<SparklineComponent id='sparkline' height='200px'
width='500px' axisSettings={{
  minX: -1, maxX: 7, maxY: 8, minY: -1
}} fill='#033e96' valueType='Category' xName='x' yName='y'
dataSource=[
  { x: 'Mon', y: 3 },
  { x: 'Tue', y: 5 },
  { x: 'Wed', y: 2 },
  { x: 'Thu', y: 4 },
  { x: 'Fri', y: 6 },
] }
// To enable tooltip for sparkline with fill color customization
tooltipSettings={{
  visible: true,
  format: '${x} : ${y}',
  fill: '#033e96',
  textStyle: {

```



```

        color: 'white'
      },
    } }>
    <Inject services={[SparklineTooltip]} />
  </SparklineComponent>;
}
}
ReactDOM.render(<App />, document.getElementById('sparkline'));
{% endraw %}

```

## INDEX.TSX

```

import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent, Inject, SparklineTooltip } from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SparklineComponent id='sparkline'
    height='200px' width='500px'
    axisSettings= { {
      minX: -1, maxX: 7, maxY: 8, minY: -1
    } }
    fill= '#033e96'
    valueType= 'Category'
    xName= 'x' yName= 'y'
    dataSource= { [
      {x: 'Mon', y: 3 },
      {x: 'Tue', y: 5 },
      {x: 'Wed', y: 2 },
      {x: 'Thu', y: 4 },
      {x: 'Fri', y: 6 },
    ] }
    // To enable tooltip for sparkline with fill color customization
    tooltipSettings= { {
      visible: true,
      format: '${x} : ${y}',
      fill: '#033e96',
      textStyle: {
        color: 'white'
      },
    },
  } }>
    <Inject services={[SparklineTooltip]} />
  </SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));

```

### Tooltip template

Sparkline tooltip has template support. By using tooltip template, you can customize tooltips. The following code example shows more customization options provided to `sparktooltip` css class that is used in tooltip template div. Using this template, images also can be added to tooltip.

## INDEX.JSX

```
{% raw %}
```

```

import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent, Inject, SparklineTooltip } from
'@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return (<SparklineComponent id='sparkline' height='200px'
width='500px' axisSettings={{
      minX: -1, maxX: 7, maxY: 8, minY: -1
    }} fill='#033e96' valueType='Category' xName='x' yName='y'
dataSource=[
      { x: 'Mon', y: 3 },
      { x: 'Tue', y: 5 },
      { x: 'Wed', y: 2 },
      { x: 'Thu', y: 4 },
      { x: 'Fri', y: 6 },
    ]
    // To enable tooltip template for sparkline with fill color, border
radius and padding customization
    tooltipSettings={{
      visible: true,
      template: '<div id="sparktooltip" style="border-radius:
5px;background: #008cff;" +
        'color: #FFFFFF !important;font-size: 16px;font-style:
italic;padding: 8px;">' +
        '${x} : ${y} </div>'
    }}>
    <Inject services={[SparklineTooltip]}>
    </SparklineComponent>);
  }
}
ReactDOM.render(<App />, document.getElementById('sparkline'));
{% endraw %}

```

## INDEX.TSX

```

import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent, Inject, SparklineTooltip
} from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SparklineComponent id='sparkline'
height='200px' width='500px'
axisSettings= { {
  minX: -1, maxX: 7, maxY: 8, minY: -1
} }
fill= '#033e96'
valueType= 'Category'
xName= 'x' yName= 'y'
dataSource= { [
  {x: 'Mon', y: 3 },
  {x: 'Tue', y: 5 },
  {x: 'Wed', y: 2 },
  {x: 'Thu', y: 4 },
  {x: 'Fri', y: 6 },
] }

```

```
// To enable tooltip template for sparkline with fill color, border
radius and padding customization
tooltipSettings= { {
  visible: true,
  template: '<div id="sparktooltip" style="border-radius:
5px;background: #008cff;" +
  'color: #FFFFFF !important;font-size: 16px;font-style:
italic;padding: 8px;">' +
  '${x} : ${y} </div>'
} }>
<Inject services={[SparklineTooltip]} />
</SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

### Track line

The track line tracks data points that are closer to the mouse position or touch contact.

To enable track lines for sparkline, specify the `visible` option of `trackLineSettings` to true. Based on theme, tracker color will be changed. The default value of tracker color is black.

To use track line in sparkline, inject the `SparklineTooltip` module to sparkline using the inject method.

### INDEX.JSX

```
{% raw %}
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent, Inject, SparklineTooltip } from
'@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return (<SparklineComponent id='sparkline' height='200px'
width='500px' axisSettings={{
      minX: -1, maxX: 46, maxY: 10, minY: -1
    }} fill='#033e96' dataSource=[5, 3, 4, 6, 8, 7, 9, 1, 3, 5, 3,
4, 6, 8, 7, 9, 1, 3, 5, 2, 4, 6, 7, 9, 5, 8, 3, 6, 1, 7, 4, 2, 5, 2, 4, 6,
7, 9, 5, 8, 3, 6, 1, 7, 4, 2]]
      // To enable tooltip template for sparkline with fill color, border
radius and padding customization
      tooltipSettings={{
        trackLineSettings: {
          visible: true,
          color: '#033e96',
          width: 1
        }
      }}>
      <Inject services={[SparklineTooltip]} />
      </SparklineComponent>);
  }
}
ReactDOM.render(<App />, document.getElementById('sparkline'));
{% endraw %}
```

**INDEX.TSX**

```
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent, Inject, SparklineTooltip }
  from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SparklineComponent id='sparkline'
    height='200px' width='500px'
    axisSettings= { {
      minX: -1, maxX: 46, maxY: 10, minY: -1
    } }
    fill= '#033e96'
    dataSource= [5, 3, 4, 6, 8, 7, 9, 1, 3, 5, 3, 4, 6, 8, 7, 9, 1, 3, 5,
2, 4, 6, 7, 9, 5, 8, 3, 6, 1, 7, 4, 2, 5, 2, 4, 6, 7, 9, 5, 8, 3, 6, 1, 7,
4, 2] ]
    // To enable tooltip template for sparkline with fill color, border
    radius and padding customization
    tooltipSettings= { {
      trackLineSettings: {
        visible: true,
        color: '#033e96',
        width: 1
      }
    } } >
    <Inject services={[SparklineTooltip]} />
  </SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

**Appearance in React Sparkline component**

The appearance of the sparkline can be customized using margin, container Area border, and container Area background.

**Sparkline border**

The **containerArea border** of the sparkline is used to render border to cover sparkline area.

The following code example shows the sparkline with overall border.

**INDEX.JSX**

```
{% raw %}
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return ( <SparklineComponent id='sparkline' height='200px'
width='350px' containerArea={ {
      border: { color: '#033e96', width: 2 }
    } } border= { {
      color: '#033e96', width: 1
    } } type='Area' fill='b2cfff' dataSource= [3, 6, 4, 1, 3, 2,
5] ] >
    </SparklineComponent> );
  }
}
```

```

    }
  }
  ReactDOM.render(<App />, document.getElementById('sparkline'));
  {% endraw %}

```

### INDEX.TSX

```

import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SparklineComponent id='sparkline'
    height='200px' width='350px'
    containerArea= { {
      border: { color: '#033e96', width: 2 }
    }}
    border= { {
      color: '#033e96', width: 1
    }}
    type= 'Area'
    fill= '#b2cfff'
    dataSource=[ [ 3, 6, 4, 1, 3, 2, 5]]>
    </SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));

```

### Sparkline padding

Padding is used to specify padding value between container and sparkline. By default, padding value of the sparkline is 5. Sparkline padding values are specified by the left, right, top, and bottom.

The following code example shows the sparkline with overall padding is set to 20.

### INDEX.JSX

```

{% raw %}
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return ( <SparklineComponent id='sparkline' height='200px'
    width='350px' containerArea={{
      border: { color: '#033e96', width: 2 }
    }}
    // To render sparkline with padding
    padding={{ left: 20, right: 20, bottom: 20, top: 20 }} border={{
      color: '#033e96', width: 1
    }} type='Area' fill='#b2cfff' dataSource=[ [3, 6, 4, 1, 3, 2,
5]]>
    </SparklineComponent> );
  }
}
ReactDOM.render(<App />, document.getElementById('sparkline'));
{% endraw %}

```

**INDEX.TSX**

```
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SparklineComponent id='sparkline'
    height='200px' width='350px'
    containerArea= { {
      border: { color: '#033e96', width: 2 }
    }}
    // To render sparkline with padding
    padding= { { left: 20, right: 20, bottom: 20, top: 20 } }
    border= { {
      color: '#033e96', width: 1
    }}
    type= 'Area'
    fill= '#b2cfff'
    dataSource=[ [ 3, 6, 4, 1, 3, 2, 5 ] ]>
</SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

## Sparkline area customization

The background color of the sparkline area can be customized using the `containerArea` background color. By default, the sparkline background color is transparent.

**INDEX.JSX**

```
{% raw %}
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return ( <SparklineComponent id='sparkline' height='200px'
width='350px' containerArea={{
      border: { color: '#033e96', width: 2 },
      // To render sparkline with background
      background: '#eff1f4',
    }} padding={{ left: 20, right: 20, bottom: 20, top: 20 }}
border={{
      color: '#033e96', width: 1
    }} type='Area' fill='#b2cfff' dataSource=[ [ 3, 6, 4, 1, 3, 2,
5 ] ]>
</SparklineComponent>);
  }
}
ReactDOM.render(<App />, document.getElementById('sparkline'));
{% endraw %}
```

**INDEX.TSX**

```
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent } from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SparklineComponent id='sparkline'
    height='200px' width='350px'
    containerArea= { {
      border: { color: '#033e96', width: 2 },
      // To render sparkline with background
      background: '#eff1f4',
    }}
    padding= { { left: 20, right: 20, bottom: 20, top: 20} }
    border= { {
      color: '#033e96', width: 1
    }}
    type= 'Area'
    fill= '#b2cfff'
    dataSource=[ [ 3, 6, 4, 1, 3, 2, 5] ]>
</SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

### Sparkline theme

Datalabel and track line colors of the sparkline will be changed based on theme. For example, for dark theme, the color of datalabel and track line should be white; for light theme, their value should be black. The possible values for sparkline theme are **Material**, **Fabric**, **Bootstrap**, and **Highcontrast**.

The following code example shows the color for datalabel and track line is set to white for dark theme.

### INDEX.JSX

```
{% raw %}
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent, Inject, SparklineTooltip } from
 '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return ( <SparklineComponent id='sparkline' height='200px'
width='350px'
  // To specify theme
  theme='Highcontrast' dataLabelSettings={{ visible: ['All'] }}
  tooltipSettings={{
    trackLineSettings: { visible: true }
  }} axisSettings={{
    minX: -1, maxX: 7
  }} lineWidth={3} border={{
    color: 'transparent', width: 2
  }} type='Line' fill='#007dd1' dataSource=[ [3, 6, 4, 1, 3, 2,
5] ]>
    <Inject services=[ [SparklineTooltip] ]/>
    </SparklineComponent> );
  }
}
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

```
{% endraw %}
```

### INDEX.TSX

```
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent, Inject, SparklineTooltip }
  from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SparklineComponent id='sparkline'
    height='200px' width='350px'
    // To specify theme
    theme= 'Highcontrast'
    dataLabelSettings= { { visible: ['All'] } }
    tooltipSettings= { {
      trackLineSettings: { visible: true }
    } }
    axisSettings= { {
      minX: -1, maxX: 7
    } }
    lineWidth= { 3 }
    border= { {
      color: 'transparent', width: 2
    } }
    type= 'Line'
    fill= '#007dd1'
    dataSource=[ [ 3, 6, 4, 1, 3, 2, 5 ] ]>
    <Inject services={[SparklineTooltip]} />
  </SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));
```

### Localization in React Sparkline component

The sparkline control supports localization. The default culture for localization is **en-US**. You can change the culture using the **setCulture** method.

#### Tooltip format

Sparkline tooltip supports localization. The following code sample shows tooltip text with currency format based on culture.

### INDEX.JSX

```
{% raw %}
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { SparklineComponent, Inject, SparklineTooltip } from
  '@syncfusion/ej2-react-charts';
export class App extends React.Component {
  render() {
    return ( <SparklineComponent id='sparkline' height='200px'
width='350px' containerArea={{
      border: { color: '#033e96', width: 2 }
    }} tooltipSettings={{
      visible: true
    }} /> );
  }
}
```



```

    }}
    // To specify currency format
    format={"c0"} useGroupingSeparator={true} lineWidth={3} padding={{
left: 20, right: 20, bottom: 20, top: 20 }} border={{ color: '#033e96',
width: 2 }} type='Area' fill='#b2cfff' dataSource=[30000, 60000, 40000,
10000, 30000, 20000, 50000]]>
    <Inject services={[SparklineTooltip]}/>
    </SparklineComponent>;
  }
}
ReactDOM.render(<App />, document.getElementById('sparkline'));
{% endraw %}

```

## INDEX.TSX

```

import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { SparklineComponent, Inject, SparklineTooltip }
  from '@syncfusion/ej2-react-charts';
function App() {
  return ( <SparklineComponent id='sparkline'
    height='200px' width='350px'
    containerArea= { {
      border: { color: '#033e96', width: 2 }
    } }
    tooltipSettings= { {
      visible: true
    } }
    // To specify currency format
    format= { "c0" }
    useGroupingSeparator= { true }
    lineWidth= { 3 }
    padding= { { left: 20, right: 20, bottom: 20, top: 20 } }
    border= { { color: '#033e96', width: 2 } }
    type= 'Area'
    fill= '#b2cfff'
    dataSource= { [30000, 60000, 40000, 10000, 30000, 20000, 50000] }>
    <Inject services={[SparklineTooltip]}/>
  </SparklineComponent> );
}
export default App;
ReactDOM.render(<App />, document.getElementById('sparkline'));

```

## Accessibility in React Sparkline component

The Sparkline component followed the accessibility guidelines and standards, including [ADA, Section 508](#), [WCAG 2.2](#) standards, and [WCAG roles](#) that are commonly used to evaluate accessibility.

The accessibility compliance for the Sparkline component is outlined below.

| Accessibility Criteria | Compatibility |

| -- | -- |

| [WCAG 2.2](#) Support |  |

```

| Section 508 Support |  |

| Screen Reader Support |  |

| Right-To-Left Support |  |

| Color Contrast |  |

| Mobile Device Support |  |

| Keyboard Navigation Support |  |

| Accessibility Checker Validation |  |

| Axe-core Accessibility Validation |  |

<style>

.post .post-content img {
display: inline-block;
margin: 0.5em 0;
}

</style>

<div> - All
features of the component meet the requirement.</div>

<div> - Some features of the component do not meet the requirement.</div>

<div> - The component does not meet the requirement.</div>

```

### WAI-ARIA attributes

The Sparkline component followed the [WAI-ARIA](#) patterns to meet the accessibility. The following ARIA attributes are used in the Sparkline component:

- `img` (role)
- `aria-label` (attribute)
- `aria-hidden` (attribute)

### Keyboard interaction

The Sparkline component followed the [keyboard interaction](#) guideline, making it easy for people who use assistive technologies (AT) and those who completely rely on keyboard navigation. The following keyboard shortcuts are supported by the Sparkline component.

| **Press** | **To do this** |

| --- | --- |

| **Ctrl + P** | **Prints the Sparkline.** |

### Ensuring accessibility

The Sparkline component's accessibility levels are ensured through an [accessibility-checker](#) and [axe-core](#) software tools during automated testing.

The accessibility compliance of the Sparkline component is shown in the following sample. Open the [sample](#) in a new window to evaluate the accessibility of the Sparkline component with accessibility tools.

See also

- [Accessibility in Syncfusion React components](#)

### Ej1 api migration in React Sparkline component

This article describes the API migration process of Accordion component from Essential JS 1 to Essential JS 2.

### Sparkline Types

{% raw %}

| **Behavior** | **API in Essential JS 1** | **API in Essential JS 2** |

| --- | --- | --- |

|Type| **Property:** `type`<br/><br/> <EJ.Sparkline id="sparkline" type="line"> </EJ.Sparkline>, document.getElementById('sparkline') | **Property:** `type`<br/><br/> <SparklineComponent id='sparkline' type='Line'></SparklineComponent> ><br/><br/>document.getElementById('sparkline'); |

### Databinding

| **Behavior** | **API in Essential JS 1** | **API in Essential JS 2** |

| --- | --- | --- |

|Datasource| **Property:** `dataSource`<br/><br/> <EJ.Sparkline id="sparkline" dataSource={dataSource1}> </EJ.Sparkline>, document.getElementById('sparkline') | **Property:** `dataSource`<br/><br/> <SparklineComponent id='sparkline' dataSource={{ {x:0,yval:50},{x:1,yval:30}}}></SparklineComponent> ><br/><br/>document.getElementById('sparkline'); |

|Binding X values with datasource| **Property:** `xName`<br/><br/> <EJ.Sparkline id="sparkline" xName="x"> </EJ.Sparkline>, document.getElementById('sparkline') | **Property:** `xName`<br/><br/> <SparklineComponent id='sparkline' xName='xValue'></SparklineComponent> ><br/><br/>document.getElementById('sparkline'); |

|Binding Y values with datasource| **Property:** `yName`<br/><br/> <EJ.Sparkline id="sparkline" yName="yval"> </EJ.Sparkline>, document.getElementById('sparkline') | **Property:**

```
yName<br/><br/> <SparklineComponent id='sparkline' yName='yValue'></SparklineComponent>
>,<br><br>document.getElementById('sparkline');|
```

```
<!-- markdownlint-disable MD038 -->
```

## Markers

| **Behavior** | **API in Essential JS 1** | **API in Essential JS 2** |

| --- | --- | --- |

| Enable markers | **Property:** *markerSettings.visible*<br/><br/> var markerSettings = { visible:true }  
<br/> <EJ.Sparkline id="sparkline" markerSettings={markerSettings}> </EJ.Sparkline>,  
document.getElementById('sparkline') | **Property:** *markerSettings.visible*<br/><br/>  
<SparklineComponent id='sparkline' markerSettings={ visible: ['All']}></SparklineComponent>  
>,<br><br>document.getElementById('sparkline');|

| Color | **Property:** *markerSettings.fill*<br/><br/> var markerSettings = { fill:"green" } <br/>  
<EJ.Sparkline id="sparkline" markerSettings={markerSettings}> </EJ.Sparkline>,  
document.getElementById('sparkline') | **Property:** *markerSettings.fill*<br/><br/>  
<SparklineComponent id='sparkline' markerSettings={ fill:'green'}></SparklineComponent>  
>,<br><br>document.getElementById('sparkline');|

| Size | **Property:** *markerSettings.width*<br/><br/> var markerSettings = { width:2 } <br/>  
<EJ.Sparkline id="sparkline" markerSettings={markerSettings}> </EJ.Sparkline>,  
document.getElementById('sparkline') | **Property:** *markerSettings.size*<br/><br/>  
<SparklineComponent id='sparkline' markerSettings={ size:10}></SparklineComponent>  
>,<br><br>document.getElementById('sparkline');|

| Opacity | **Property:** *markerSettings.opacity*<br/><br/> var markerSettings = { opacity:0.5 } <br/>  
<EJ.Sparkline id="sparkline" markerSettings={markerSettings}> </EJ.Sparkline>,  
document.getElementById('sparkline') | **Property:** *markerSettings.opacity*<br/><br/>  
<SparklineComponent id='sparkline' markerSettings={ opacity:0.5}></SparklineComponent>  
>,<br><br>document.getElementById('sparkline');|

| Border color | **Property:** *markerSettings.border.color*<br/><br/> var markerSettings = { border:{  
color: "red" } } <br/> <EJ.Sparkline id="sparkline" markerSettings={markerSettings}>  
</EJ.Sparkline> , document.getElementById('sparkline') | **Property:**  
*markerSettings.border.color*<br/><br/> <SparklineComponent id='sparkline' markerSettings={  
border:{color:'green'}}></SparklineComponent>  
>,<br><br>document.getElementById('sparkline');|

| Border width | **Property:** *markerSettings.border.width*<br/><br/> var markerSettings = {  
border:{width:2 } } <br/> <EJ.Sparkline id="sparkline" markerSettings={markerSettings}>  
</EJ.Sparkline> , document.getElementById('sparkline') | **Property:**  
*markerSettings.border.width*<br/><br/> <SparklineComponent id='sparkline' markerSettings={  
border:{width:2}}></SparklineComponent> >,<br><br>document.getElementById('sparkline');|

| Border opacity | **Property:** *markerSettings.border.opacity*<br/><br/> var markerSettings = {  
border:{opacity:0.5 } } <br/> <EJ.Sparkline id="sparkline" markerSettings={markerSettings}>  
</EJ.Sparkline> , document.getElementById('sparkline') | Not applicable |

## Data labels

| **Behavior** | **API in Essential JS 1** | **API in Essential JS 2** |

| --- | --- | --- |

| Enable data labels | Not applicable | **Property:** *dataLabelSettings.visible*  
`<SparklineComponent id='sparkline' dataLabelSettings={ } visible:['All'] ></SparklineComponent >`  
`>, <br><br>document.getElementById('sparkline');`

| Color | Not applicable | **Property:** *dataLabelSettings.fill*  
`<SparklineComponent id='sparkline' dataLabelSettings={ } color:'green'></SparklineComponent >`  
`>, <br><br>document.getElementById('sparkline');`

| Opacity | Not applicable | **Property:** *dataLabelSettings.opacity*  
`<SparklineComponent id='sparkline' dataLabelSettings={ } opacity:0.5></SparklineComponent >`  
`>, <br><br>document.getElementById('sparkline');`

| Border color | Not applicable | **Property:** *dataLabelSettings.border.color*  
`<SparklineComponent id='sparkline' dataLabelSettings={ } border:{color:'green'}></SparklineComponent >`  
`>, <br><br>document.getElementById('sparkline');`

| Border width | Not applicable | **Property:** *dataLabelSettings.border.width*  
`<SparklineComponent id='sparkline' dataLabelSettings={ } border:{width:2}></SparklineComponent >`  
`>, <br><br>document.getElementById('sparkline');`

| Format | Not applicable | **Property:** *dataLabelSettings.format*  
`<SparklineComponent id='sparkline' dataLabelSettings={ } format:'${xval}: ${yval}'></SparklineComponent >`  
`>, <br><br>document.getElementById('sparkline');`

| Horizontal Offset | Not applicable | **Property:** *dataLabelSettings.offset.x*  
`<SparklineComponent id='sparkline' dataLabelSettings={ } offset:{x:10}></SparklineComponent >`  
`>, <br><br>document.getElementById('sparkline');`

| Vertical Offset | Not applicable | **Property:** *dataLabelSettings.offset.y*  
`<SparklineComponent id='sparkline' dataLabelSettings={ } offset:{y:10}></SparklineComponent >`  
`>, <br><br>document.getElementById('sparkline');`

| Font color | Not applicable | **Property:** *dataLabelSettings.textStyle.color*  
`<SparklineComponent id='sparkline' dataLabelSettings={ } textStyle:{color:'red'}></SparklineComponent >`  
`>, <br><br>document.getElementById('sparkline');`

| Font family | Not applicable | **Property:** *dataLabelSettings.textStyle.fontFamily*  
`<SparklineComponent id='sparkline' dataLabelSettings={ } textStyle:{fontFamily:'Arial'}></SparklineComponent >`  
`>, <br><br>document.getElementById('sparkline');`

| Font style | Not applicable | **Property:** *dataLabelSettings.textStyle.fontStyle*  
`<SparklineComponent id='sparkline' dataLabelSettings={ } textStyle:{fontStyle:'Normal'}></SparklineComponent >`  
`>, <br><br>document.getElementById('sparkline');`

|Font weight| Not applicable | **Property:** *dataLabelSettings.textStyle.fontWeight*<br/><br/>  
 <SparklineComponent id='sparkline' dataLabelSettings={  
 textStyle:{fontWeight:'Bold'}}></SparklineComponent  
 >,<br><br>document.getElementById('sparkline');|

|Font opacity| Not applicable | **Property:** *dataLabelSettings.textStyle.opacity*<br/><br/>  
 <SparklineComponent id='sparkline' dataLabelSettings={  
 textStyle:{opacity:0.5}}></SparklineComponent  
 >,<br><br>document.getElementById('sparkline');|

|Font size| Not applicable | **Property:** *dataLabelSettings.textStyle.fontSize*<br/><br/>  
 <SparklineComponent id='sparkline' dataLabelSettings={  
 textStyle:{fontSize:'10px'}}></SparklineComponent  
 >,<br><br>document.getElementById('sparkline');|

### Range band

| **Behavior** | **API in Essential JS 1** | **API in Essential JS 2** |

| --- | --- | --- |

|Color| **Property:** *rangeBandSettings.color*<br/><br/>var rangeBandSettings = { color:"green" }  
 <br/> <EJ.Sparkline id="sparkline" rangeBandSettings={rangeBandSettings}> </EJ.Sparkline>,  
 document.getElementById('sparkline') | **Property:** *rangeBandSettings.color*<br/><br/>  
 <SparklineComponent id='sparkline' rangeBandSettings={  
 color:'green'}></SparklineComponent >,<br><br>document.getElementById('sparkline');|

|Opacity| **Property:** *rangeBandSettings.opacity*<br/><br/> var rangeBandSettings = { opacity:0.5 }  
 <br/> <EJ.Sparkline id="sparkline" rangeBandSettings={rangeBandSettings}> </EJ.Sparkline>,  
 document.getElementById('sparkline') | **Property:** *rangeBandSettings.opacity*<br/><br/>  
 <SparklineComponent id='sparkline' rangeBandSettings={ opacity:0.5}></SparklineComponent  
 >,<br><br>document.getElementById('sparkline');|

|Start range| **Property:** *rangeBandSettings.startRange*<br/><br/> var rangeBandSettings = {  
 startRange: 4 } <br/> <EJ.Sparkline id="sparkline" rangeBandSettings={rangeBandSettings}>  
 </EJ.Sparkline>, document.getElementById('sparkline') | **Property:**  
*rangeBandSettings.startRange*<br/><br/> <SparklineComponent id='sparkline'  
 rangeBandSettings={ startRange:4}></SparklineComponent  
 >,<br><br>document.getElementById('sparkline');|

|End range| **Property:** *rangeBandSettings.endRange*<br/><br/> var rangeBandSettings = {  
 endRange:30 } <br/> <EJ.Sparkline id="sparkline" rangeBandSettings={rangeBandSettings}>  
 </EJ.Sparkline>, document.getElementById('sparkline') | **Property:**  
*rangeBandSettings.endRange*<br/><br/> <SparklineComponent id='sparkline' rangeBandSettings={  
 endRange:30}></SparklineComponent >,<br><br>document.getElementById('sparkline');|

### Special points customization

| **Behavior** | **API in Essential JS 1** | **API in Essential JS 2** |

| --- | --- | --- |

| High point color | **Property:** *highPointColor*  
`highPointColor="green"> </EJ.Sparkline>, document.getElementById('sparkline') | Property:  
highPointColor  
highPointColor='green'></SparklineComponent  
>,<br><br>document.getElementById('sparkline');;`

| Low point color | **Property:** *lowPointColor*  
`lowPointColor="green"> </EJ.Sparkline>, document.getElementById('sparkline') | Property:  
lowPointColor  
lowPointColor='green'></SparklineComponent  
>,<br><br>document.getElementById('sparkline');;`

| Negative point color | **Property:** *negativePointColor*  
`negativePointColor="green"> </EJ.Sparkline>, document.getElementById('sparkline') | Property:  
negativePointColor  
negativePointColor='green'></SparklineComponent  
>,<br><br>document.getElementById('sparkline');;`

| Start point color | **Property:** *startPointColor*  
`startPointColor="green"> </EJ.Sparkline>, document.getElementById('sparkline') | Property:  
startPointColor  
startPointColor='green'></SparklineComponent  
>,<br><br>document.getElementById('sparkline');;`

| End point color | **Property:** *endPointColor*  
`endPointColor="green"> </EJ.Sparkline>, document.getElementById('sparkline') | Property:  
endPointColor  
endPointColor='green'></SparklineComponent  
>,<br><br>document.getElementById('sparkline');;`

| Tie point color | **Property:** *tiePointColor*  
`<SparklineComponent id='sparkline' tiePointColor='green'></SparklineComponent`  
`>,<br><br>document.getElementById('sparkline');`;

#### Axis customization

| **Behavior** | **API in Essential JS 1** | **API in Essential JS 2** |

| --- | --- | --- |

| Show axis line | **Property:** *axisSettings.visible*  
`var axisLineSettings = { visible : true};<br><EJ.Sparkline id="sparkline" axisLineSttings={axisLineSettings}> </EJ.Sparkline>,<br>  
document.getElementById('sparkline') | Property: axisSettings.lineSettings.visible  
<SparklineComponent id='sparkline' axisSettings={} lineSettings:{ visible: true }></SparklineComponent  
>,<br><br>document.getElementById('sparkline');;`

| Line color | **Property:** *axisSettings.color*  
`var axisLineSettings = { color : "red"};<br><EJ.Sparkline id="sparkline" axisLineSttings={axisLineSettings}> </EJ.Sparkline>,<br>  
document.getElementById('sparkline') | Property: axisSettings.lineSettings.color  
<br><br>`



```
<SparklineComponent id='sparkline' axisSettings={} lineSettings:{ color: 'green' }
></SparklineComponent >,<br><br>document.getElementById('sparkline');|
```

| Line width | **Property:** *axisSettings.width*<br><br> var axisLineSettings = { width : 2};<br><EJ.Sparkline id="sparkline" axisLineSttings={axisLineSettings}> </EJ.Sparkline>,<br>document.getElementById('sparkline') | **Property:** *axisSettings.lineSettings.width*<br><br><SparklineComponent id='sparkline' axisSettings={} lineSettings:{ width: 2 }></SparklineComponent >,<br><br>document.getElementById('sparkline');|

| Dash array | **Property:** *axisSettings.dashArray*<br><br> var axisLineSettings = { dashArray : 1,2};<br><EJ.Sparkline id="sparkline" axisLineSttings={axisLineSettings}> </EJ.Sparkline>,<br>document.getElementById('sparkline') | **Property:** *axisSettings.lineSettings.dashArray*<br><br><SparklineComponent id='sparkline' axisSettings={} lineSettings:{ dashArray: 1,2 }></SparklineComponent >,<br><br>document.getElementById('sparkline');|

| X axis minimum value | Not applicable | **Property:** *axisSettings.minX*<br><br> <SparklineComponent id='sparkline' axisSettings={} minX: 2 ></SparklineComponent >,<br><br>document.getElementById('sparkline');|

| X axis maximum value | Not applicable | **Property:** *axisSettings.maxX*<br><br><SparklineComponent id='sparkline' axisSettings={} maxX:10 ></SparklineComponent >,<br><br>document.getElementById('sparkline');|

| Y axis minimum value | Not applicable | **Property:** *axisSettings.minY*<br><br> <SparklineComponent id='sparkline' axisSettings={ {minY: 5 }></SparklineComponent >,<br><br>document.getElementById('sparkline');|

| Y axis maximum value | Not applicable | **Property:** *axisSettings.maxY*<br><br><SparklineComponent id='sparkline' axisSettings={} maxY: 10 ></SparklineComponent >,<br><br>document.getElementById('sparkline');|

| Horizontal axis line position | Not applicable | **Property:** *axisSettings.value*<br><br><SparklineComponent id='sparkline' axisSettings={} value: 10 ></SparklineComponent >,<br><br>document.getElementById('sparkline');|

#### Appearance customization

| **Behavior** | **API in Essential JS 1** | **API in Essential JS 2** |

| --- | --- | --- |

| Background color | **Property:** *background*<br><br> <EJ.Sparkline id="sparkline" background="grey"> </EJ.Sparkline>,<br>document.getElementById('sparkline') | **Property:** *containerArea.background*<br><br> <SparklineComponent id='sparkline' background='grey'></SparklineComponent >,<br><br>document.getElementById('sparkline');|

| Border color | Not applicable | **Property:** *containerArea.border.color*<br><br><SparklineComponent id='sparkline' containerArea={} border: {color:'green'}></SparklineComponent >,<br><br>document.getElementById('sparkline');|



|Border width| Not applicable | **Property:** *containerArea.border.width*<br/><br/>  
 <SparklineComponent id='sparkline' containerArea={} border: {width:2}  
 ></SparklineComponent >,<br><br>document.getElementById('sparkline');|

|Series color| **Property:** *fill*<br/><br/> <EJ.Sparkline id="sparkline" fill="green"> </EJ.Sparkline>,  
 document.getElementById('sparkline') | **Property:** *fill*<br/><br/> <SparklineComponent  
 id='sparkline' fill='grey'></SparklineComponent  
 >,<br><br>document.getElementById('sparkline');|

|Series opacity| **Property:** *opacity*<br/><br/> <EJ.Sparkline id="sparkline" opacity={0.5}>  
 </EJ.Sparkline>, document.getElementById('sparkline') | **Property:** *opacity*<br/><br/>  
 <SparklineComponent id='sparkline' opacity={0.5}></SparklineComponent  
 >,<br><br>document.getElementById('sparkline');|

|Line series width| **Property:** *width*<br/><br/> <EJ.Sparkline id="sparkline" width={2}>  
 </EJ.Sparkline>, document.getElementById('sparkline') | **Property:** *lineWidth*<br/><br/>  
 <SparklineComponent id='sparkline' lineWidth={ 2 }></SparklineComponent  
 >,<br><br>document.getElementById('sparkline');|

|Series border color| **Property:** *border.color*<br/><br/> var border =  
 {color:"green"};<br/><EJ.Sparkline id="sparkline" border={border}> </EJ.Sparkline>,  
 document.getElementById('sparkline') | **Property:** *border.color*<br/><br/> <SparklineComponent  
 id='sparkline' border: }color:'green' }></SparklineComponent  
 >,<br><br>document.getElementById('sparkline');|

|Series border width| **Property:** *border.width*<br/><br/> var border = {width:2};<br/><EJ.Sparkline  
 id="sparkline" border={border}> </EJ.Sparkline>, document.getElementById('sparkline')  
 | **Property:** *border.width*<br/><br/> <SparklineComponent id='sparkline' border= }width:2}  
 ></SparklineComponent >,<br><br>document.getElementById('sparkline');|

|Series palette| **Property:** *palette*<br/><br/> <EJ.Sparkline id="sparkline" palettes={palettes}>  
 </EJ.Sparkline>, document.getElementById('sparkline')<br/><br/>var palettes = { "red", "green",  
 "orange", "blue" }; | **Property:** *palette*<br/><br/> <SparklineComponent id='sparkline'  
 palette=""></SparklineComponent >,<br><br>document.getElementById('sparkline');|

|Theme| **Property:** *theme*<br/><br/> <EJ.Sparkline id="sparkline" theme="flatdark">  
 </EJ.Sparkline>, document.getElementById('sparkline') | **Property:** *theme*<br/><br/>  
 <SparklineComponent id='sparkline' theme='flatdark'></SparklineComponent  
 >,<br><br>document.getElementById('sparkline');|

|Width| **Property:** *size.width*<br/><br/> var size = { width:200};<br/><EJ.Sparkline id="sparkline"  
 size={size}> </EJ.Sparkline>, document.getElementById('sparkline') | **Property:** *width*<br/><br/>  
 <SparklineComponent id='sparkline' size=} width: '300px' }></SparklineComponent  
 >,<br><br>document.getElementById('sparkline');|

|Height| **Property:** *size.height*<br/><br/> var size = { height:200};<br/><EJ.Sparkline id="sparkline"  
 size={size}> </EJ.Sparkline>, document.getElementById('sparkline') | **Property:** *height*<br/><br/>  
 <SparklineComponent id='sparkline' size=} height:'300px' }></SparklineComponent >,  
 document.getElementById('sparkline');|

## Tooltip

| **Behavior** | **API in Essential JS 1** | **API in Essential JS 2** |

| --- | --- | --- |

| Show tooltip | **Property:** *tooltip.visible*  
`<EJ.Sparkline id="sparkline" tooltip={tooltip}> </EJ.Sparkline>, document.getElementById('sparkline')`  
**Property:** *tooltipSettings.visible*  
`<SparklineComponent id='sparkline' tooltipSettings={visible:true}> <Inject services=[[SparklineTooltip]] /> </SparklineComponent>`  
`>, <br><br>document.getElementById('sparkline');`

| Background | **Property:** *tooltip.fill*  
`<EJ.Sparkline id="sparkline" tooltip={tooltip}> </EJ.Sparkline>, document.getElementById('sparkline')`  
**Property:** *tooltipSettings.fill*  
`<SparklineComponent id='sparkline' tooltipSettings={fill:'green'}> <Inject services=[[SparklineTooltip]] /> </SparklineComponent>`  
`>, <br><br>document.getElementById('sparkline');`

| Format | Not applicable | **Property:** *tooltipSettings.format*  
`<SparklineComponent id='sparkline' tooltipSettings={format:'${x} : ${yval}'}> <Inject services=[[SparklineTooltip]] /> </SparklineComponent>`  
`>, <br><br>document.getElementById('sparkline');`

| Template | **Property:** *tooltip.template*  
`<EJ.Sparkline id="sparkline" tooltip={tooltip}> </EJ.Sparkline>, document.getElementById('sparkline')`  
`<div id="item"> //.. </div> <br>`  
**Property:** *tooltipSettings.template*  
`<SparklineComponent id='sparkline' tooltipSettings={template:'<div> //... </div>'}> <Inject services=[[SparklineTooltip]] /> </SparklineComponent>`  
`>, <br><br>document.getElementById('sparkline');`

| Font color | **Property:** *tooltip.font.color*  
`<EJ.Sparkline id="sparkline" tooltip={tooltip}> </EJ.Sparkline>, document.getElementById('sparkline')`  
**Property:** *tooltipSettings.textStyle.color*  
`<SparklineComponent id='sparkline' tooltipSettings={textStyle:{color:'green'}}> <Inject services=[[SparklineTooltip]] /> </SparklineComponent>`  
`>, <br><br>document.getElementById('sparkline');`

| Font opacity | **Property:** *tooltip.font.opacity*  
`<EJ.Sparkline id="sparkline" tooltip={tooltip}> </EJ.Sparkline>, document.getElementById('sparkline')`  
**Property:** *tooltipSettings.textStyle.opacity*  
`<SparklineComponent id='sparkline' tooltipSettings={textStyle:{opacity:0.5}}> <Inject services=[[SparklineTooltip]] /> </SparklineComponent>`  
`>, <br><br>document.getElementById('sparkline');`

| Font size | **Property:** *tooltip.font.size*  
`<EJ.Sparkline id="sparkline" tooltip={tooltip}> </EJ.Sparkline>, document.getElementById('sparkline')`  
**Property:** *tooltipSettings.textStyle.size*  
`<SparklineComponent id='sparkline' tooltipSettings={textStyle:{size:'12px'}}> <Inject services=[[SparklineTooltip]] /> </SparklineComponent>`  
`>, <br><br>document.getElementById('sparkline');`

| Font family| **Property:** *tooltip.font.fontFamily*<br/><br/> var tooltip = { font:{  
fontFamily:"Arial"};<br/><EJ.Sparkline id="sparkline" tooltip={tooltip}> </EJ.Sparkline>,  
document.getElementById('sparkline') | **Property:** *tooltipSettings.textStyle.fontFamily*<br/><br/>  
<SparklineComponent id='sparkline' tooltipSettings={} textStyle:{fontFamily:'Arial'} >> <Inject  
services={[SparklineTooltip]} /> </SparklineComponent  
><br/><br/>document.getElementById('sparkline');|

| Font style| **Property:** *tooltip.font.fontStyle*<br/><br/> var tooltip = { font:{  
fontStyle:"Normal"};<br/><EJ.Sparkline id="sparkline" tooltip={tooltip}> </EJ.Sparkline>,  
document.getElementById('sparkline') | **Property:** *tooltipSettings.textStyle.fontStyle*<br/><br/>  
<SparklineComponent id='sparkline' tooltipSettings={} textStyle:{fontStyle:'Normal'} >> <Inject  
services={[SparklineTooltip]} /> </SparklineComponent  
><br/><br/>document.getElementById('sparkline');|

| Font weight| **Property:** *tooltip.font.fontWeight*<br/><br/> var tooltip = { font:{  
fontWeight:"Bold"};<br/><EJ.Sparkline id="sparkline" tooltip={tooltip}> </EJ.Sparkline>,  
document.getElementById('sparkline') | **Property:** *tooltipSettings.textStyle.fontWeight*<br/><br/>  
<SparklineComponent id='sparkline' tooltipSettings={} textStyle:{fontWeight:'Bold'} >> <Inject  
services={[SparklineTooltip]} /> </SparklineComponent  
><br/><br/>document.getElementById('sparkline');|

| Enable track line| Not applicable | **Property:** *tooltipSettings.trackLineSettings.visible*<br/><br/>  
<SparklineComponent id='sparkline' size={} height:'300px' >></SparklineComponent  
><br/><br/>document.getElementById('sparkline');|<SparklineComponent id='sparkline'  
tooltipSettings={} trackLineSettings:{visible:true} >> <Inject services={[SparklineTooltip]} />  
</SparklineComponent ><br/><br/>document.getElementById('sparkline');|

| Track line width| Not applicable | **Property:** *tooltipSettings.trackLineSettings.width*<br/><br/>  
<SparklineComponent id='sparkline' tooltipSettings={} trackLineSettings:{width:2} >> <Inject  
services={[SparklineTooltip]} /> </SparklineComponent  
><br/><br/>document.getElementById('sparkline');|

## Rendering

| **Behavior** | **API in Essential JS 1** | **API in Essential JS 2** |

| --- | --- | --- |

| Enable canvas rendering| **Property:** *enableCanvasRendering*<br/><br/> <EJ.Sparkline id="sparkline"  
enableCanvasRendering={true}> </EJ.Sparkline>, document.getElementById('sparkline') | Not  
applicable |

## Localization

| **Behavior** | **API in Essential JS 1** | **API in Essential JS 2** |

| --- | --- | --- |

| Localization| **Property:** *locale*<br/><br/> <EJ.Sparkline id="sparkline" locale="en-US">  
</EJ.Sparkline>, document.getElementById('sparkline') | **Property:** *type*<br/><br/>  
<SparklineComponent id='sparkline' locale='en-US' >> </SparklineComponent  
><br/><br/>document.getElementById('sparkline');|

## Methods

| **Behavior** | **API in Essential JS 1** | **API in Essential JS 2** |

| --- | --- | --- |

| Dynamically updating sparkline | **Method:** *redraw*  
`<EJ.Sparkline id="sparkline">  
 </EJ.Sparkline>, document.getElementById('sparkline') <br/> function sparkMethod() { var  
 sparObj = $("#sparkline").data("ejSparkline"); sparkObj.redraw(); } | Method: refresh  
 export class Spark extends SampleBase<{}, {}> { <br/>private sparkInstance:  
 SparklineComponent; <br/> private change(): void{<br/>this.sparkInstance.refresh();<br/>} |`

## Events

| **Behavior** | **API in Essential JS 1** | **API in Essential JS 2** |

| --- | --- | --- |

| Load | **Event:** *load*  
`<EJ.Sparkline id="sparkline"  
 load={load}></EJ.Sparkline>, <br/>document.getElementById('maps')<br/>function load(args) {}  
 | Event: load  
 load={this.load.bind(this)}></SparklineComponent>, <br/>document.getElementById('sparkline')  
 <br/> public load(args: ILoadEventArgs): void {} |`

| Load completed | **Event:** *loaded*  
`<EJ.Sparkline id="sparkline"  
 loaded={loaded}></EJ.Sparkline>, <br/>document.getElementById('maps')<br/>function  
 loaded(args) {} | Event: loaded  
 loaded={this.loaded.bind(this)}></SparklineComponent>, <br/>document.getElementById('spark  
 line');<br/> public loaded(args: ILoadedEventArgs): void {} |`

| Initialize tooltip | **Event:** *tooltipInitialize*  
`<EJ.Sparkline id="sparkline"  
 tooltipInitialize={tooltipInitialize}></EJ.Sparkline>, <br/>document.getElementById('maps')<br/>f  
 unction tooltipInitialize(args) {} | Event: tooltipInitialize  
 id='sparkline'  
 tooltipInitialize={this.tooltipInitialize.bind(this)}></SparklineComponent>, <br/>document.getEle  
 mentById('sparkline');<br/> public tooltipInitialize(args: ITooltipInitializeEventArgs): void {} |`

| Series rendering | **Event:** *seriesRendering*  
`<EJ.Sparkline id="sparkline"  
 seriesRendering={seriesRendering}></EJ.Sparkline>, <br/>document.getElementById('maps')<br/>  
 >function seriesRendering(args) {} | Event: seriesRendering  
 id='sparkline'  
 seriesRendering={this.seriesRendering.bind(this)}></SparklineComponent>, <br/>document.getE  
 lementById('sparkline');<br/> public seriesRendering(args: ISeriesRenderingEventArgs): void {} |`

| Region mouse move | **Event:** *pointRegionMouseMove*  
`<EJ.Sparkline id="sparkline"  
 pointRegionMouseMove={pointRegionMouseMove}></EJ.Sparkline>, <br/>document.getEleme  
 ntById('maps')<br/>function pointRegionMouseMove(args) {} | Event:  
pointRegionMouseMove  
 pointRegionMouseMove={this.pointRegionMouseMove.bind(this)}></SparklineComponent>, <b  
 r/>document.getElementById('sparkline');<br/> public pointRegionMouseMove(args:  
 IPointRegionMouseMoveEventArgs): void {} |`

|Region click| **Event:** *pointRegionMouseClick*<br/><br/> <EJ.Sparkline id="sparkline"  
 pointRegionMouseClick={pointRegionMouseClick}></EJ.Sparkline>,<br/>document.getElementById('maps')<br/>function pointRegionMouseClick(args) {} | **Event:**  
*pointRegionMouseClick*<br/><br/> <SparklineComponent id='sparkline'  
 pointRegionMouseClick={this.pointRegionMouseClick.bind(this)}></SparklineComponent>,<br/>  
 document.getElementById('sparkline');<br/> public pointRegionMouseClick(args:  
 IPointRegionMouseEventArgs): void {} |

|Mouse move| **Event:** *sparklineMouseMove*<br/><br/> <EJ.Sparkline id="sparkline"  
 sparklineMouseMove={sparklineMouseMove}></EJ.Sparkline>,<br/>document.getElementById('maps')<br/>function sparklineMouseMove(args) {} | **Event:** *sparklineMouseMove*<br/><br/>  
 <SparklineComponent id='sparkline'  
 sparklineMouseMove={this.sparklineMouseMove.bind(this)}></SparklineComponent>,<br/>doc  
 ument.getElementById('sparkline');<br/> public sparklineMouseMove(args:  
 ISparklineMouseMoveEventArgs): void {} |

|Mouse leave| **Event:** *sparklineMouseLeave*<br/><br/> <EJ.Sparkline id="sparkline"  
 sparklineMouseLeave={sparklineMouseLeave}></EJ.Sparkline>,<br/>document.getElementById('maps')<br/>function sparklineMouseLeave(args) {} | Not applicable |

|Click| **Event:** *click*<br/><br/> <EJ.Sparkline id="sparkline"  
 click={click}></EJ.Sparkline>,<br/>document.getElementById('maps')<br/>function click(args) {}  
 | **Event:** *sparklineMouseClick*<br/><br/> <SparklineComponent id='sparkline'  
 sparklineMouseClick={this.sparklineMouseClick.bind(this)}></SparklineComponent>,<br/>docu  
 ment.getElementById('sparkline');<br/> public sparklineMouseClick(args:  
 ISparklineMouseClickEventArgs): void {} |

|doubleClick| **Event:** *doubleClick*<br/><br/><EJ.Sparkline id="sparkline"  
 doubleClick={doubleClick}></EJ.Sparkline>,<br/>document.getElementById('maps')<br/>functio  
 n doubleClick(args) {} | Not applicable |

|rightClick| **Event:** *rightClick*<br/><br/> <EJ.Sparkline id="sparkline"  
 rightClick={rightClick}></EJ.Sparkline>,<br/>document.getElementById('maps')<br/>function  
 rightClick(args) {} | Not applicable |

|axisRendering| Not applicable | **Event:** *axisRendering*<br/><br/> <SparklineComponent  
 id='sparkline'  
 axisRendering={this.axisRendering.bind(this)}></SparklineComponent>,<br/>document.getElem  
 entById('sparkline');<br/> public axisRendering(args: IAxisRenderingEventArgs): void {} |

|dataLabelRendering| Not applicable | **Event:** *dataLabelRendering*<br/><br/> <SparklineComponent  
 id='sparkline'  
 dataLabelRendering={this.dataLabelRendering.bind(this)}></SparklineComponent>,<br/>docum  
 ent.getElementById('sparkline');<br/> public dataLabelRendering(args:  
 IDataLabelRenderingEventArgs): void {} |

|markerRendering| Not applicable | **Event:** *markerRendering*<br/><br/> <SparklineComponent  
 id='sparkline'

```

markerRendering={this.markerRendering.bind(this)}></SparklineComponent>,<br/>document.g
etElementById('sparkline');<br/> public markerRendering(args: IMarkerRenderingEventArgs):
void {} |

|pointRendering| Not applicable | Event: pointRendering<br/><br/> <SparklineComponent
id='sparkline'
pointRendering={this.pointRendering.bind(this)}></SparklineComponent>,<br/>document.getEl
ementById('sparkline');<br/> public pointRendering(args: IPointRenderingEventArgs): void {} |

|resize| Not applicable | Event: resize<br/><br/> <SparklineComponent id='sparkline'
resize={this.resize.bind(this)}></SparklineComponent>,<br/>document.getElementById('sparkli
ne');<br/> public resize(args: IResizeEventArgs): void {} |

{% endraw %}

```

## SpeedDial

### Getting Started with React Speed Dial Control

This section explains how to create a simple Speed Dial and demonstrate the basic usage of the Speed Dial component in an React environment.

#### Dependencies

The list of dependencies required to use the Speed Dial component in your application is given below:

```

`js
|-- @syncfusion/ej2-react-buttons
|-- @syncfusion/ej2-react-base
|-- @syncfusion/ej2-base
|-- @syncfusion/ej2-buttons
,

```

#### Installation and Configuration

You can use [Create-react-app](#) to setup the applications. To install `create-react-app` run the following command.

```

`bash
npm install -g create-react-app
,

```

Start a new project using create-react-app command as follows

```

,
create-react-app quickstart --scripts-version=react-scripts-ts
cd quickstart
,
create-react-app quickstart

```

cd quickstart

,

'react-scripts-ts' is used for creating React app with typescript.

### Adding Syncfusion packages

All the available Essential JS 2 packages are published in [npmjs.com](https://www.npmjs.com) public registry.

To install Speed Dial component, use the following command

```
`bash
```

```
npm install @syncfusion/ej2-react-buttons --save
```

,

### Adding Speed Dial component to the Application

To include the Speed Dial component in your application import the `SpeedDialComponent` from `ej2-react-buttons` package in `App.tsx`.

Add the Speed Dial component in application as shown in below code example.

```
`ts
```

```
{ / Import the Speed Dial. / }
```

```
import { SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
```

```
import * as React from 'react';
```

```
import './App.css';
```

```
{ / To render Speed Dial. / }
```

```
function App() {
```

```
  return (
```

```
<SpeedDialComponent id='speeddial'></SpeedDialComponent>
```

```
);
```

```
}
```

```
export default App;
```

,

### Adding CSS Reference

Import the Speed Dial component's required CSS references as follows in `src/App.css`.

```
`css
```

```
@import "../node_modules/@syncfusion/ej2-base/styles/material.css";
```

```
@import "../node_modules/@syncfusion/ej2-buttons/styles/material.css";
```

,

### Running the application

Run the application in the browser using the following command:

npm start

The following example shows a basic Speed Dial component.

### APP.JSX

```
{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial.*/ }
function App() {
    const items = [
        { text: 'Cut' },
        { text: 'Copy' },
        { text: 'Paste' }
    ];
    return (<div id="targetElement" style={{ position: 'relative',
minHeight: '350px', border: '1px solid' }}>
        { /* Initialize the SpeedDial component. */ }
        <SpeedDialComponent id='speeddial' items={items} content='Edit'
target="#targetElement"></SpeedDialComponent>
    </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}
```

### APP.TSX

```
{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent, SpeedDialItemModel } from '@syncfusion/ej2-
react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial.*/ }
function App() {
    const items: SpeedDialItemModel[] = [
        { text: 'Cut' },
        { text: 'Copy' },
        { text: 'Paste' }
    ];
    return (
        <div id="targetElement" style={{position:'relative', minHeight:'350px',
border:'1px solid'}}>
            { /* Initialize the SpeedDial component. */ }
            <SpeedDialComponent id='speeddial' items={items} content='Edit'
target="#targetElement"></SpeedDialComponent>
        </div>
    );
}
export default App;
```



```
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}
```

## Items in React Speed dial component

The React Speed Dial action items can be added by using [items](#) property.

Fields	Type	Description
<a href="#">text</a>	string	Defines the text content of SpeedDialItem.
<a href="#">iconCss</a>	string	Defines one or more CSS classes to include an icon or image in Speed Dial item.
<a href="#">disabled</a>	boolean	Defines whether to enable or disable the SpeedDialItem.
<a href="#">id</a>	string	Defines a unique value for the SpeedDialItem which can be used to identify the item in event args.
<a href="#">title</a>	string	Defines the title of SpeedDialItem to display tooltip.

## Icons in Speed Dial items

You can customize the icon and text of Speed Dial action items using [iconCss](#) and [text](#) properties.

### Icon only

You can show icon only in Speed Dial items by setting [iconCss](#) property. You can show tooltip on hover to show additional details to end-user by setting [title](#) property.

## APP.JSX

```
{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial.*/ }
function App() {
  const items = [
    { iconCss: 'e-icons e-cut' },
    { iconCss: 'e-icons e-copy' },
    { iconCss: 'e-icons e-paste' }
  ];
  return (<div>
    <div id="targetElement" style={{ position: 'relative', minHeight:
'350px', border: '1px solid' }}></div>
    <SpeedDialComponent id='speeddial' openIconCss='e-icons e-edit'
closeIconCss='e-icons e-close' items={items}
target="#targetElement"></SpeedDialComponent>
    </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}
```

## APP.TSX

```
{% raw %}
```

```

{ /* Import the Speed Dial. */ }
import { SpeedDialComponent, SpeedDialItemModel } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial. */ }
function App() {
  const items: SpeedDialItemModel[] = [
    { iconCss: 'e-icons e-cut' },
    { iconCss: 'e-icons e-copy' },
    { iconCss: 'e-icons e-paste' }
  ];
  return (
    <div>
      <div id="targetElement" style={{position:'relative',
minHeight:'350px', border:'1px solid'}}></div>
      <SpeedDialComponent id='speeddial' openIconCss='e-icons e-edit'
closeIconCss='e-icons e-close' items={items}
target="#targetElement"></SpeedDialComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}

```

### Text only

You can show only text in Speed Dial items by setting [text](#) property.

### APP.JSX

```

{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial. */ }
function App() {
  const items = [
    { text: 'Cut' },
    { text: 'Copy' },
    { text: 'Paste' }
  ];
  return (<div>
    <div id="targetElement" style={{ position: 'relative', minHeight:
'350px', border: '1px solid' }}></div>
    <SpeedDialComponent id='speeddial' content='Edit' items={items}
target="#targetElement"></SpeedDialComponent>
  </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}

```

### APP.TSX

```
{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent, SpeedDialItemModel } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial. */ }
function App() {
  const items: SpeedDialItemModel[] = [
    { text: 'Cut' },
    { text: 'Copy' },
    { text: 'Paste' }
  ];
  return (
    <div>
      <div id="targetElement" style={{position:'relative',
minHeight:'350px', border:'1px solid'}}></div>
      <SpeedDialComponent id='speeddial' content='Edit' items={items}
target="#targetElement"></SpeedDialComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}
```

### Icon with text

You can show icon along with text in Speed Dial items by setting [iconCss](#) and [text](#) properties.

### APP.JSX

```
{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial. */ }
function App() {
  const items = [
    { text: 'Cut', iconCss: 'e-icons e-cut' },
    { text: 'Copy', iconCss: 'e-icons e-copy' },
    { text: 'Paste', iconCss: 'e-icons e-paste' }
  ];
  return (<div>
    <div id="targetElement" style={{ position: 'relative', minHeight:
'350px', border: '1px solid' }}></div>
    <SpeedDialComponent id='speeddial' openIconCss='e-icons e-edit'
closeIconCss='e-icons e-close' content='Edit' items={items}
target="#targetElement"></SpeedDialComponent>
  </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}
```

**APP.TSX**

```
{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent, SpeedDialItemModel } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial. */ }
function App() {
  const items: SpeedDialItemModel[] = [
    { text: 'Cut', iconCss: 'e-icons e-cut' },
    { text: 'Copy', iconCss: 'e-icons e-copy' },
    { text: 'Paste', iconCss: 'e-icons e-paste' }
  ];
  return (
    <div>
      <div id="targetElement" style={{position:'relative',
minHeight:'350px', border:'1px solid'}}></div>
      <SpeedDialComponent id='speeddial' openIconCss='e-icons e-edit'
closeIconCss='e-icons e-close' content='Edit' items={items}
target="#targetElement"></SpeedDialComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}
```

*Disabled*

You can disable Speed Dial items by setting [disabled](#) property as `true`.

**APP.JSX**

```
{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial. */ }
function App() {
  const items = [
    { text: 'Cut', disabled: true },
    { text: 'Copy' },
    { text: 'Paste' }
  ];
  return (<div>
    <div id="targetElement" style={{ position: 'relative', minHeight:
'350px', border: '1px solid' }}></div>
    <SpeedDialComponent id='speeddial' content='Edit' items={items}
target="#targetElement"></SpeedDialComponent>
  </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}
```

**APP.TSX**

```
{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent, SpeedDialItemModel } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial. */ }
function App() {
    const items: SpeedDialItemModel[] = [
        { text: 'Cut', disabled: true },
        { text: 'Copy' },
        { text: 'Paste' }
    ];
    return (
        <div>
            <div id="targetElement" style={{position:'relative',
minHeight:'350px', border:'1px solid'}}></div>
            <SpeedDialComponent id='speeddial' content='Edit' items={items}
target="#targetElement"></SpeedDialComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}
```

**Animation**

The Speed Dial items can be animated during the opening and closing of the popup action items. You can customize the animation's effect, delay, and duration by setting [animation](#) property. By default, Speed Dial animates with a [fade](#) effect and supports all [speeddialanimation](#) effects.

Below example demonstrates the Speed Dial items with applied Zoom effect.

**APP.JSX**

```
{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial. */ }
function App() {
    const items = [
        { text: 'Cut', iconCss: 'e-icons e-cut' },
        { text: 'Copy', iconCss: 'e-icons e-copy' },
        { text: 'Paste', iconCss: 'e-icons e-paste' }
    ];
    const animation = { effect: 'Zoom' };
    return <div>
        <div id="targetElement" style={{ position: 'relative', minHeight:
'350px', border: '1px solid' }}></div>
    </div>

```

```

    <SpeedDialComponent id='speeddial' openIconCss='e-icons e-edit'
    closeIconCss='e-icons e-close' content='Edit' items={items}
    animation={animation} target="#targetElement"></SpeedDialComponent>
    </div>;
  }
  export default App;
  ReactDOM.render(<App />, document.getElementById('button'));
  {% endraw %}

```

## APP.TSX

```

{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent, SpeedDialItemModel,
SpeedDialAnimationSettingsModel } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial. */ }
function App() {
  const items: SpeedDialItemModel[] = [
    { text: 'Cut', iconCss: 'e-icons e-cut' },
    { text: 'Copy', iconCss: 'e-icons e-copy' },
    { text: 'Paste', iconCss: 'e-icons e-paste' }
  ];
  const animation: SpeedDialAnimationSettingsModel =
    { effect: 'Zoom' };
  return (
    <div>
      <div id="targetElement" style={{position:'relative',
minHeight:'350px', border:'1px solid'}}></div>
      <SpeedDialComponent id='speeddial' openIconCss='e-icons e-edit'
closeIconCss='e-icons e-close' content='Edit' items={items}
animation={animation} target="#targetElement"></SpeedDialComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}

```

## Template

The Speed Dial supports to customize the action items and entire pop-up container by setting [itemTemplate](#) and [popupTemplate](#) properties. For more details about templates, check out the link [here](#).

## Positions in React Speed dial component

The Speed dial control can be positioned anywhere on the [target](#) using the [position](#) property. If the [target](#) is not defined, then Speed Dial is positioned based on the browser viewport.

The position values of Speed Dial are as follows:

- TopLeft
- TopCenter
- TopRight

- MiddleLeft
- MiddleCenter
- MiddleRight
- BottomLeft
- BottomCenter
- BottomRight

**APP.JSX**

```
{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial.*/ }
function App() {
    return (<div>
        <div id="targetElement" style={{ position: 'relative', minHeight:
'350px', border: '1px solid' }}></div>
        <SpeedDialComponent id='speeddial' content='Add' position='BottomLeft'
target="#targetElement"></SpeedDialComponent>
    </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}
```

**APP.TSX**

```
{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial.*/ }
function App() {
    return (
        <div>
            <div id="targetElement" style={{position:'relative',
minHeight:'350px', border:'1px solid'}}></div>
            <SpeedDialComponent id='speeddial' content='Add' position='BottomLeft'
target="#targetElement"></SpeedDialComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}
```

**INDEX.CSS**

```
/* Represents the styles for loader */
#loader {
    color: #008cff;
    height: 40px;
```

```

left: 45%;
position: absolute;
top: 45%;
width: 30%;
}

```

### Opens items on hover

You can open the Speed Dial action items on mouse hover by setting the [opensOnHover](#) property.

`ts

*{/ Import the Speed Dial. /}*

```
import { SpeedDialComponent, SpeedDialItemModel } from '@syncfusion/ej2-react-buttons';
```

```
import * as React from 'react';
```

*{/ To render Speed Dial./}*

```
function App() {
```

```
  const items: SpeedDialItemModel[] = [
```

```
    { iconCss: 'e-icons e-cut' },
```

```
    { iconCss: 'e-icons e-copy' },
```

```
    { iconCss: 'e-icons e-paste' }
```

```
  ];
```

```
  return (
```

*{/ Initialize the SpeedDial component. /}*

```
<SpeedDialComponent id='speeddial' openIconCss='e-icons e-edit' closeIconCss='e-icons e-close'
  items={items} opensOnHover= {true} target="#targetElement"></SpeedDialComponent>
```

```
);
```

```
}
```

```
export default App;
```

```
`
```

### Programmatically show/hide Speed Dial items

You can open/close the Speed Dial action items programmatically using [show](#) and [hide](#) methods.

Below example demonstrates open/close action items on button click.

#### **APP.JSX**

```

{% raw %}
{ /* Import the Speed Dial. */ }
import { ButtonComponent, SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial.*/ }
function App() {

```



```

let speeddialObj;
const items = [
  { iconCss: 'e-icons e-cut' },
  { iconCss: 'e-icons e-copy' },
  { iconCss: 'e-icons e-paste' }
];
function showClick() {
  speeddialObj.show();
}
function hideClick() {
  speeddialObj.hide();
}
return (<div>
  <div id="targetElement" style={{ position: 'relative', minHeight:
'340px', border: '1px solid', padding: '10px' }}>
    <ButtonComponent id="show" cssClass="e-primary" onClick={showClick}>
Show </ButtonComponent>
    <ButtonComponent id="hide" cssClass="e-primary" onClick={hideClick}>
Hide </ButtonComponent></div>
    <SpeedDialComponent id='speeddial' openIconCss='e-icons e-edit'
closeIconCss='e-icons e-close' items={items} opensOnHover={true}
target="#targetElement" ref={(scope) => { speeddialObj = scope;
}}></SpeedDialComponent>
  </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}

```

### APP.TSX

```

{% raw %}
{/* Import the Speed Dial. */}
import { ButtonComponent, SpeedDialComponent, SpeedDialItemModel } from
'@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{/* To render Speed Dial.*/}
function App() {
  let speeddialObj: SpeedDialComponent;
  const items: SpeedDialItemModel[] = [
    { iconCss: 'e-icons e-cut' },
    { iconCss: 'e-icons e-copy' },
    { iconCss: 'e-icons e-paste' }
  ];
  function showClick() :void {
    speeddialObj.show();
  }
  function hideClick() :void {
    speeddialObj.hide();
  }
  return (
    <div>
      <div id="targetElement" style={{position:'relative',
minHeight:'340px', border:'1px solid', padding:'10px'}}>

```

```

    <ButtonComponent id="show" cssClass="e-primary" onClick={ showClick
  }> Show </ButtonComponent>
    <ButtonComponent id="hide" cssClass="e-primary" onClick={ hideClick
  }> Hide </ButtonComponent></div>
    <SpeedDialComponent id='speeddial' openIconCss='e-icons e-edit'
  closeIconCss='e-icons e-close' items={items} opensOnHover= {true}
  target="#targetElement" ref={(scope) => { speeddialObj = scope;
  }}></SpeedDialComponent>
  </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}

```

### INDEX.CSS

```

/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
/* Represents the styles for button */
#hide{
  float: right;
}

```

### Programmatically refresh the position

You can refresh the position of the Speed Dial using [refreshPosition](#) method when the **target** position is changed.

The following sample demonstrates the above functionalities of a Speed Dial to refresh the position of speed dial.

### APP.JSX

```

{% raw %}
{ /* Import the Speed Dial. */ }
import { ButtonComponent, SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial.*/ }
function App() {
  let speeddialObj;
  const items = [
    { iconCss: 'e-icons e-cut' },
    { iconCss: 'e-icons e-copy' },
    { iconCss: 'e-icons e-paste' }
  ];
  function refresh() {

```

```

        document.getElementById("targetElement").style.minHeight = "300px";
        speeddialObj.refreshPosition();
    }
    return (<div>
        <div id="targetElement" style={{ position: 'relative', minHeight:
'340px', border: '1px solid', padding: '10px' }}>
            <ButtonComponent id="refresh" cssClass="e-primary" onClick={refresh}>
Refresh </ButtonComponent></div>
            <SpeedDialComponent id='speeddial' openIconCss='e-icons e-edit'
closeIconCss='e-icons e-close' items={items} position='MiddleRight'
target="#targetElement" ref={scope => { speeddialObj = scope;
}}></SpeedDialComponent>
        </div>);
    }
    export default App;
    ReactDOM.render(<App />, document.getElementById('button'));
    {% endraw %}

```

### APP.TSX

```

{% raw %}
/* Import the Speed Dial. */
import { ButtonComponent, SpeedDialComponent, SpeedDialItemModel } from
'@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
/* To render Speed Dial.*/
function App() {
    let speeddialObj: SpeedDialComponent;
    const items: SpeedDialItemModel[] = [
        { iconCss: 'e-icons e-cut' },
        { iconCss: 'e-icons e-copy' },
        { iconCss: 'e-icons e-paste' }
    ];
    function refresh():void {
        document.getElementById("targetElement").style.minHeight = "300px";
        speeddialObj.refreshPosition();
    }
    return (
        <div>
            <div id="targetElement" style={{position:'relative',
minHeight:'340px', border:'1px solid', padding:'10px'}}>
                <ButtonComponent id="refresh" cssClass="e-primary" onClick={refresh}>
Refresh </ButtonComponent></div>
                <SpeedDialComponent id='speeddial' openIconCss='e-icons e-edit'
closeIconCss='e-icons e-close' items={items} position='MiddleRight'
target="#targetElement" ref={scope =>{ speeddialObj = scope
}}></SpeedDialComponent>
            </div>
        );
    }
    export default App;
    ReactDOM.render(<App />, document.getElementById('button'));
    {% endraw %}

```

**INDEX.CSS**

```

/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}

```

## Display modes in React Speed dial component

The action items in React Speed Dial can be displayed in **Linear** and **Radial** display modes by setting [mode](#) property.

## Linear display mode

In **Linear** display mode, Speed Dial action items are displayed in a list-like format either horizontally or vertically. By default, Speed Dial items are displayed in **Linear** mode.

*Direction*

You can open the action items on the top, left, up, and down side of the Speed Dial button by setting [direction](#) property. The default value is **Auto** where the action items are displayed based on the [position](#) of the Speed Dial.

The **Linear** directions of Speed Dial are as follows:

- Left - Action items are displayed on the left side of the button.
- Right - Action items are displayed on the right side of the button.
- Up - Action items are displayed on the top of the button.
- Down - Action items are displayed on the bottom of the button.
- Auto - Action items display direction auto calculated based on **position** of the Speed Dial. If Speed Dial is position at bottom right, then action items displayed at top.

**APP.JSX**

```

{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial.*/ }
function App() {
  const items = [
    { iconCss: 'e-icons e-cut' },
    { iconCss: 'e-icons e-copy' },
    { iconCss: 'e-icons e-paste' }
  ];
  return (<div>
    <div id="targetElement" style={{ position: 'relative', minHeight:
'350px', border: '1px solid' }}></div>

```

```

    <SpeedDialComponent id='speeddial' openIconCss='e-icons e-edit'
    closeIconCss='e-icons e-close' items={items} mode='Linear' direction='Left'
    target="#targetElement"></SpeedDialComponent>
  </div>;
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% enddraw %}

```

### APP.TSX

```

{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent, SpeedDialItemModel } from '@syncfusion/ej2-
react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial. */ }
function App() {
  const items: SpeedDialItemModel[] = [
    { iconCss: 'e-icons e-cut' },
    { iconCss: 'e-icons e-copy' },
    { iconCss: 'e-icons e-paste' }
  ];
  return (
    <div>
      <div id="targetElement" style={{position:'relative',
minHeight:'350px', border:'1px solid'}}></div>
      <SpeedDialComponent id='speeddial' openIconCss='e-icons e-edit'
closeIconCss='e-icons e-close' items={items} mode='Linear' direction=
'Left' target="#targetElement"></SpeedDialComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}

```

### Radial display mode (Radial Menu)

In **Radial** mode, Speed Dial action items are displayed in a circular pattern like a radial menu. For more details about radial mode, check out the link [here](#).

### Radial menu in React Speed dial component

The React Speed Dial action items can be displayed in a circular pattern like a radial menu by setting **mode** property. You can customize the **direction**, **startAngle**, **endAngle** and **offset** by setting **radialSettings** property.

### Radial menu direction

You can open the action items in either clockwise or anticlockwise by setting **direction** property. The default value is **Auto** where the action items are displayed based on the **position** property of the Speed Dial.

`ts

```

*/ Import the Speed Dial. */
import { SpeedDialComponent, SpeedDialItemModel, RadialSettingsModel } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
*/ To render Speed Dial.*/
function App() {
  const items: SpeedDialItemModel[] = [
    { iconCss: 'e-icons e-cut' },
    { iconCss: 'e-icons e-copy' },
    { iconCss: 'e-icons e-paste' },
    { iconCss: 'e-icons e-edit' },
    { iconCss: 'e-icons e-save' }
  ];
  const radialSettings: RadialSettingsModel = { direction: 'AntiClockwise' };
  return (
*/ Initialize the SpeedDial component. */
    <SpeedDialComponent id='speeddial' openIconCss='e-icons e-edit' closeIconCss='e-icons e-close'
      items={items} mode= 'Radial' radialSettings= {radialSettings}
      target="#targetElement"></SpeedDialComponent>
  );
}
export default App;

```

### Radial menu start and end angle

You can modify the start and end angle of action items by setting [startAngle](#) and [endAngle](#) properties. If the angle is not defined, the action items are displayed based on the **position** property of the Speed Dial.

`ts

```

*/ Import the Speed Dial. */
import { SpeedDialComponent, SpeedDialItemModel, RadialSettingsModel } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
*/ To render Speed Dial.*/
function App() {
  const items: SpeedDialItemModel[] = [

```

```

{ iconCss: 'e-icons e-cut' },
{ iconCss: 'e-icons e-copy' },
{ iconCss: 'e-icons e-paste' },
{ iconCss: 'e-icons e-edit' },
{ iconCss: 'e-icons e-save' }
];

const radialSettings: RadialSettingsModel = { direction: 'AntiClockwise', startAngle: 180, endAngle: 360 };
return (
  // Initialize the SpeedDial component. //

  <SpeedDialComponent id='speeddial' openIconCss='e-icons e-edit' closeIconCss='e-icons e-close'
    items={items} mode= 'Radial' radialSettings= {radialSettings} position='MiddleCenter'
    target="#targetElement"></SpeedDialComponent>

);
}

export default App;

ReactDOM.render(<App />,document.getElementById('button'));
`

```

### Offset

You can modify the offset distance between action items and Speed Dial button using [offset](#) property.

```

`ts

// Import the Speed Dial. //

import { SpeedDialComponent, SpeedDialItemModel, RadialSettingsModel } from '@syncfusion/ej2-react-buttons';

import * as React from 'react';

// To render Speed Dial.//

function App() {
  const items: SpeedDialItemModel[] = [
    { iconCss: 'e-icons e-cut' },
    { iconCss: 'e-icons e-copy' },
    { iconCss: 'e-icons e-paste' },
    { iconCss: 'e-icons e-edit' },
    { iconCss: 'e-icons e-save' }
  ];

  const radialSettings: RadialSettingsModel = { offset: '80px' };

```

```

return (
  /* Initialize the SpeedDial component. */
  <SpeedDialComponent id='speeddial' openIconCss='e-icons e-edit' closeIconCss='e-icons e-close'
    items={items} mode='Radial' radialSettings= {radialSettings}
    target="#targetElement"></SpeedDialComponent>
);
}

export default App;

```

Below example demonstrates the radial menu settings of the Speed Dial.

### APP.JSX

```

{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial.*/ }
function App() {
  const items = [
    { iconCss: 'e-icons e-cut' },
    { iconCss: 'e-icons e-copy' },
    { iconCss: 'e-icons e-paste' },
    { iconCss: 'e-icons e-edit' },
    { iconCss: 'e-icons e-save' }
  ];
  const radialSettings = { offset: '80px', direction: 'AntiClockwise',
    startAngle: 90, endAngle: 270 };
  return (<div>
    <div id="targetElement" style={{ position: 'relative', minHeight:
      '350px', border: '1px solid' }}></div>
    <SpeedDialComponent id='speeddial' openIconCss='e-icons e-edit'
      closeIconCss='e-icons e-close' items={items} mode='Radial'
      position='MiddleRight' radialSettings={radialSettings}
      target="#targetElement"></SpeedDialComponent>
    </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}

```

### APP.TSX

```

{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent, SpeedDialItemModel, RadialSettingsModel } from
  '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial.*/ }

```



```
function App() {
  const items: SpeedDialItemModel[] = [
    { iconCss: 'e-icons e-cut' },
    { iconCss: 'e-icons e-copy' },
    { iconCss: 'e-icons e-paste' },
    { iconCss: 'e-icons e-edit' },
    { iconCss: 'e-icons e-save' }
  ];
  const radialSettings: RadialSettingsModel = { offset: '80px', direction:
  'AntiClockwise', startAngle: 90, endAngle: 270 };
  return (
    <div>
      <div id="targetElement" style={{position:'relative',
minHeight:'350px', border:'1px solid'}}></div>
      <SpeedDialComponent id='speeddial' openIconCss='e-icons e-edit'
closeIconCss='e-icons e-close' items={items} mode= 'Radial' position=
'MiddleRight' radialSettings= {radialSettings}
target="#targetElement"></SpeedDialComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}
```

## Template in React Speed dial component

This section explains available templates in SpeedDial component and its usage.

### Item template

You can use the [itemTemplate](#) property to set a template content for the SpeedDial items. The template content is defined as a child content of `itemTemplate` tag directive.

### APP.JSX

```
{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial.*/ }
function App() {
  const items = [
    { iconCss: 'e-icons e-cut', text: 'Cut' },
    { iconCss: 'e-icons e-copy', text: 'Copy' },
    { iconCss: 'e-icons e-paste', text: 'Paste' }
  ];
  function itemTemplate(props) {
    let classname = "icon " + props.properties.iconCss;
    return (<div className="itemlist">
      <span className={classname}></span>
      <span className="text">{props.properties.text}</span>
    </div>);
  }
  return (<div>
    <div id="targetElement" style={{ position: 'relative', minHeight:
'350px', border: '1px solid' }}></div>
```

```

    /* Initialize the SpeedDial component */
    <SpeedDialComponent id='speeddial' openIconCss='e-icons e-edit'
content="Edit" items={items} itemTemplate={itemTemplate}
target="#targetElement"></SpeedDialComponent>
    </div>;
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}

```

## APP.TSX

```

{% raw %}
/* Import the Speed Dial. */
import { SpeedDialComponent, SpeedDialItemModel } from '@syncfusion/ej2-
react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
/* To render Speed Dial.*/
function App() {
    const items: SpeedDialItemModel[] = [
        { iconCss: 'e-icons e-cut', text: 'Cut' },
        { iconCss: 'e-icons e-copy', text: 'Copy' },
        { iconCss: 'e-icons e-paste', text: 'Paste' }
    ];
    function itemTemplate(props: any): JSX.Element {
        let classname: string = "icon " + props.properties.iconCss;
        return (
            <div className="itemlist">
                <span className={classname}></span>
                <span className="text">{props.properties.text}</span>
            </div>);
    }
    return (
        <div>
            <div id="targetElement" style={{position:'relative',
minHeight:'350px', border:'1px solid'}}></div>
            /* Initialize the SpeedDial component */
            <SpeedDialComponent id='speeddial' openIconCss='e-icons e-edit'
content="Edit" items={items} itemTemplate={itemTemplate}
target="#targetElement"></SpeedDialComponent>
            </div>
        );
    }
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}

```

## INDEX.CSS

```

/* Represents the styles for loader */
#loader {
    color: #008cff;
    height: 40px;
    left: 45%;
    position: absolute;

```

```

    top: 45%;
    width: 30%;
  }
  /* Represents the styles for speeddial list items */
  .itemlist .text {
    padding: 0 5px;
  }
  .e-speeddial-li .itemlist {
    display: inherit;
    width: 100%;
    border: 1px solid transparent;
    align-items: center;
    padding: 5px;
    border-radius: 500px;
    background-color: rgba(104, 99, 104, 0.1);
    box-shadow: 0 0 4px grey;
  }
  .e-speeddial-li .itemlist:hover {
    background-color: rgb(224, 224, 224);
  }

```

### Popup template

You can use the [popupTemplate](#) property to set a template content for popup of SpeedDial component. The template content is defined as a child content of `popupTemplate` tag directive.

### APP.JSX

```

{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial.*/ }
function App() {
  function popupTemplate() {
    return (<div>
      <div className="speeddial-form">
        <p>Here you can customize your code.</p>
      </div>
    </div>);
  }
  return (<div>
    <div id="targetElement" style={{ position: 'relative', minHeight:
'350px', border: '1px solid' }}></div>
    { /* Initialize the SpeedDial component */ }
    <SpeedDialComponent id='speeddial' content="Edit"
popupTemplate={popupTemplate} target="#targetElement"></SpeedDialComponent>
    </div>);
  }
  export default App;
  ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}

```

### APP.TSX

```
{% raw %}
{/* Import the Speed Dial. */}
import { SpeedDialComponent, SpeedDialItemModel } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{/* To render Speed Dial.*/}
function App() {
  function popupTemplate(): JSX.Element {
    return (
      <div>
        <div className="speeddial-form">
          <p>Here you can customize your code.</p>
        </div>
      </div>);
  }
  return (
    <div>
      <div id="targetElement" style={{position:'relative',
minHeight:'350px', border:'1px solid'}}></div>
      {/* Initialize the SpeedDial component */}
      <SpeedDialComponent id='speeddial' content="Edit"
popupTemplate={popupTemplate} target="#targetElement"></SpeedDialComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}
```

## INDEX.CSS

```
/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
/* Represents the styles for speeddial-form */
.speeddial-form {
  width: 200px;
  height: 80px;
  text-align: center;
  border-radius: 15px;
  box-shadow: rgb(0 0 0 / 10%) 0px 10px 15px -3px, rgb(0 0 0 / 5%) 0px 4px
6px -2px;
  background: #f5f5f5;
  padding: 15px;
}
```

## Styles in React Speed dial component

This section briefs different ways to style SpeedDial component.

### SpeedDial button

You can customize the icon and text of React Speed Dial button using [openIconCss](#), [closeIconCss](#) and [content](#) properties.

#### Icon only

You can use the [openIconCss](#) and [closeIconCss](#) properties to show icons in speed dial button. You can also show tooltip on hover to show additional details to end-user by setting `title` attribute.

#### APP.JSX

```
{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial.*/ }
function App() {
  return (<div>
    <div id="targetElement" style={{ position: 'relative', minHeight:
'350px', border: '1px solid' }}></div>
    { /* Initialize the SpeedDial component with only icon */}
    <SpeedDialComponent id='speeddial' openIconCss='e-icons e-edit'
target="#targetElement"></SpeedDialComponent>
    </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}
```

#### APP.TSX

```
{% raw %}
{ /* Import the Speed Dial. */}
import { SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial.*/}
function App() {
  return (
    <div>
      <div id="targetElement" style={{position:'relative',
minHeight:'350px', border:'1px solid'}}></div>
      { /* Initialize the SpeedDial component with only icon */}
      <SpeedDialComponent id='speeddial' openIconCss='e-icons e-edit'
target="#targetElement"></SpeedDialComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />,document.getElementById('button'));
{% endraw %}
```

#### Text only

You can show only text in Speed Dial button by setting [content](#) property without setting icon properties..

**APP.JSX**

```
{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial.*/ }
function App() {
    return (<div>
        <div id="targetElement" style={{ position: 'relative', minHeight:
'350px', border: '1px solid' }}></div>
        { /* Initialize the SpeedDial component with only text */ }
        <SpeedDialComponent id='speeddial' content='Edit'
target="#targetElement"></SpeedDialComponent>
        </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}
```

**APP.TSX**

```
{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial.*/ }
function App() {
    return (
        <div>
            <div id="targetElement" style={{position:'relative',
minHeight:'350px', border:'1px solid'}}></div>
            { /* Initialize the SpeedDial component with only text */ }
            <SpeedDialComponent id='speeddial' content='Edit'
target="#targetElement"></SpeedDialComponent>
            </div>
        );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}
```

*Icon with text*

You show icon and text in SpeedDial button using [openIconCss](#), [closeIconCss](#) and [content](#) properties together.

**APP.JSX**

```
{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial.*/ }
```

```
function App() {
  return (<div>
    <div id="targetElement" style={{ position: 'relative', minHeight:
'350px', border: '1px solid' }}></div>
    {/* Initialize the SpeedDial component with icon and text */}
    <SpeedDialComponent id='speeddial' content='Edit' openIconCss='e-icons
e-edit' target="#targetElement"></SpeedDialComponent>
  </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}
```

### APP.TSX

```
{% raw %}
{/* Import the Speed Dial. */}
import { SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{/* To render Speed Dial.*/}
function App() {
  return (
    <div>
      <div id="targetElement" style={{position:'relative',
minHeight:'350px', border:'1px solid'}}></div>
      {/* Initialize the SpeedDial component with icon and text */}
      <SpeedDialComponent id='speeddial' content='Edit' openIconCss='e-icons
e-edit' target="#targetElement"></SpeedDialComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}
```

### Disabled

You can enable or disable the SpeedDial component using [disabled](#) property.

`ts

*{/ Import the Speed Dial. /}*

import { SpeedDialComponent } from '@syncfusion/ej2-react-buttons';

import \* as React from 'react';

*{/ To render Speed Dial. /}*

function App() {

return (

*{/ Initialize the SpeedDial component in disabled state /}*

<SpeedDialComponent id='speeddial' content='Edit' disabled={true}  
target="#targetElement"></SpeedDialComponent>

```
);
}

export default App;
,
```

### cssClass

The React Speed Dial supports the following predefined styles that can be defined using the [cssClass](#) property. You can customize by setting the `cssClass` property with the below defined class.

cssClass	Description
-----	-----
e-primary	Used to represent a primary action.
e-outline	Used to represent an appearance of button with outline.
e-info	Used to represent an informative action.
e-success	Used to represent a positive action.
e-warning	Used to represent an action with caution.
e-danger	Used to represent a negative action.

### APP.JSX

```
{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial.*/ }
function App() {
  return (
    <div id="targetElement" style={{ position: 'relative', minHeight:
'350px', border: '1px solid' }}></div>
    { /* Initialize the SpeedDial component with applied warning style */ }
    <SpeedDialComponent id='speeddial' content='Edit' cssClass='e-warning'
target="#targetElement"></SpeedDialComponent>
    </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}
```

### APP.TSX

```
{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial.*/ }
function App() {
  return (
    <div>
```



```

    <div id="targetElement" style={{position:'relative',
minHeight:'350px', border:'1px solid'}}></div>
    {/* Initialize the SpeedDial component with applied warning style */}
    <SpeedDialComponent id='speeddial' content='Edit' cssClass='e-warning'
target="#targetElement"></SpeedDialComponent>
  </div>
);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}

```

### Visible

You can set the Speed Dial button to visible/hidden state using [visible](#) property.

`ts

```

{/ Import the Speed Dial. /}
import { SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
{/ To render Speed Dial./}
function App() {
  return (
    {/ Initialize the SpeedDial component in hidden state /}
    <SpeedDialComponent id='speeddial' content='Edit' visible={false}
target="#targetElement"></SpeedDialComponent>
  );
}
export default App;
`

```

### Tooltip

You can show tooltip on hover to show additional details to end-user by setting [title](#) to Speed Dial button.

### APP.JSX

```

{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial.*/ }
function App() {
  const items = [
    { iconCss: 'e-icons e-cut', title: 'Cut' },
    { iconCss: 'e-icons e-copy', title: 'Copy' },
    { iconCss: 'e-icons e-paste', title: 'Paste' }
  ]

```

```

    ];
    return (<div>
      <div id="targetElement" style={{ position: 'relative', minHeight:
'350px', border: '1px solid' }}></div>
      { /* Initialize the SpeedDial component */ }
      <SpeedDialComponent id='speeddial' items={items} openIconCss='e-icons
e-edit' target='#targetElement'></SpeedDialComponent>
    </div>);
  }
  export default App;
  ReactDOM.render(<App />, document.getElementById('button'));
  {% endraw %}

```

### APP.TSX

```

{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent, SpeedDialItemModel } from '@syncfusion/ej2-
react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial.*/ }
function App() {
  const items: SpeedDialItemModel[] = [
    { iconCss:'e-icons e-cut', title:'Cut' },
    { iconCss:'e-icons e-copy', title:'Copy' },
    { iconCss:'e-icons e-paste', title:'Paste' }
  ];
  return (
    <div>
      <div id="targetElement" style={{position:'relative',
minHeight:'350px', border:'1px solid'}}></div>
      { /* Initialize the SpeedDial component */ }
      <SpeedDialComponent id='speeddial' items={items} openIconCss='e-icons
e-edit' target='#targetElement'></SpeedDialComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}

```

### Opens on hover

You can use [opensOnHover](#) property to open actions items on hover itself. By default action items displayed only when clicking the speed dial button.

### APP.JSX

```

{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial.*/ }
function App() {

```

```

const items = [
  { iconCss: 'e-icons e-cut' },
  { iconCss: 'e-icons e-copy' },
  { iconCss: 'e-icons e-paste' }
];
return (<div>
  <div id="targetElement" style={{ position: 'relative', minHeight:
'350px', border: '1px solid' }}></div>
  /* Initialize the SpeedDial component */
  <SpeedDialComponent id='speeddial' openIconCss='e-icons e-edit'
closeIconCss='e-icons e-close' items={items} opensOnHover={true}
target="#targetElement"></SpeedDialComponent>
</div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}

```

### APP.TSX

```

{% raw %}
/* Import the Speed Dial. */
import { SpeedDialComponent, SpeedDialItemModel } from '@syncfusion/ej2-
react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
/* To render Speed Dial.*/
function App() {
  const items: SpeedDialItemModel[] = [
    { iconCss: 'e-icons e-cut' },
    { iconCss: 'e-icons e-copy' },
    { iconCss: 'e-icons e-paste' }
  ];
  return (
    <div>
      <div id="targetElement" style={{position:'relative',
minHeight:'350px', border:'1px solid'}}></div>
      /* Initialize the SpeedDial component */
      <SpeedDialComponent id='speeddial' openIconCss='e-icons e-edit'
closeIconCss='e-icons e-close' items={items} opensOnHover= {true}
target="#targetElement"></SpeedDialComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}

```

### Modal in React Speed dial component

You can use the [modal](#) property to set the Speed Dial as modal which adds an overlay to prevent the background interaction.

### APP.JSX

```
{% raw %}
```

```

{ /* Import the Speed Dial. */ }
import { SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial.*/ }
function App() {
    const items = [
        { iconCss: 'e-icons e-cut' },
        { iconCss: 'e-icons e-copy' },
        { iconCss: 'e-icons e-paste' }
    ];
    return (<div>
        <div id="targetElement" style={{ position: 'relative', minHeight:
'350px', border: '1px solid' }}></div>
        { /* Initialize the SpeedDial component with modal */ }
        <SpeedDialComponent id='speeddial' items={items} openIconCss='e-icons
e-edit' modal={true} target='#targetElement'></SpeedDialComponent>
        </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}

```

### APP.TSX

```

{% raw %}
{ /* Import the Speed Dial. */ }
import { SpeedDialComponent, SpeedDialItemModel } from '@syncfusion/ej2-
react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* To render Speed Dial.*/ }
function App() {
    const items: SpeedDialItemModel[] = [
        { iconCss:'e-icons e-cut'},
        { iconCss:'e-icons e-copy'},
        { iconCss:'e-icons e-paste'}
    ];
    return (
        <div>
            <div id="targetElement" style={{position:'relative',
minHeight:'350px', border:'1px solid'}}></div>
            { /* Initialize the SpeedDial component with modal */ }
            <SpeedDialComponent id='speeddial' items={items} openIconCss='e-icons
e-edit' modal={true} target='#targetElement'></SpeedDialComponent>
            </div>
        );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}

```

### Event in React Speed dial component

This section explains the Speed Dial events that will be triggered when appropriate actions are performed.

### clicked

The SpeedDial component triggers the [clicked](#) event with [SpeedDialItemEventArgs](#) argument when an action item is clicked. You can use this event to perform the required action.

```
`ts
```

```
import { SpeedDialComponent, SpeedDialItemModel, SpeedDialItemEventArgs } from '@syncfusion/ej2-react-buttons';
```

```
import * as React from 'react';
```

```
{/ Initialize action items. /}
```

```
function App() {
```

```
  const items: SpeedDialItemModel[] = [
```

```
    { text: 'Cut' },
```

```
    { text: 'Copy' },
```

```
    { text: 'Paste' } 
```

```
  ];
```

```
  function itemClick(args:SpeedDialItemEventArgs){
```

```
    //Your required action here
```

```
  }
```

```
  return (
```

```
    {/ Initialize the SpeedDial component /}
```

```
    <SpeedDialComponent id='speeddial' items={items} content='Edit' clicked={ itemClick }></SpeedDialComponent>
```

```
  );
```

```
}
```

```
export default App;
```

```
`
```

### created

The Speed Dial component triggers the [created](#) event when SpeedDial component rendering is completed.

```
`ts
```

```
import { SpeedDialComponent, SpeedDialItemModel } from '@syncfusion/ej2-react-buttons';
```

```
import * as React from 'react';
```

```
{/ Initialize action items. /}
```

```
function App() {
```

```
  const items: SpeedDialItemModel[] = [
```

```
    { text: 'Cut' },
```

```

{ text: 'Copy' },
{ text: 'Paste' }
];
function created(){
//Your required action here
}
return (
  /* Initialize the SpeedDial component */
  <SpeedDialComponent id='speeddial' items={items} content='Edit' created={ created }></SpeedDialComponent>
);
}
export default App;
`

```

### beforeOpen

The SpeedDial component triggers the [beforeOpen](#) event with [SpeedDialBeforeOpenCloseEventArgs](#) argument before the SpeedDial popup is opened.

```

`ts
import { SpeedDialComponent, SpeedDialItemModel, SpeedDialBeforeOpenCloseEventArgs } from
 '@syncfusion/ej2-react-buttons';
import * as React from 'react';
/* Initialize action items. */
function App() {
  const items: SpeedDialItemModel[] = [
    { text: 'Cut' },
    { text: 'Copy' },
    { text: 'Paste' }
  ];
  function beforeOpen(args:SpeedDialBeforeOpenCloseEventArgs){
    //Your required action here
  }
  return (
    /* Initialize the SpeedDial component */
    <SpeedDialComponent id='speeddial' items={items} content='Edit' beforeOpen={ beforeOpen }></SpeedDialComponent>
  );
}

```

```
);
}
export default App;
`
```

### onOpen

The SpeedDial component triggers the [onOpen](#) event with [SpeedDialOpenCloseEventArgs](#) argument when SpeedDial popup is opened.

```
`ts
import { SpeedDialComponent, SpeedDialItemModel, SpeedDialOpenCloseEventArgs } from
 '@syncfusion/ej2-react-buttons';
import * as React from 'react';
*/ Initialize action items. */
function App() {
  const items: SpeedDialItemModel[] = [
    { text: 'Cut' },
    { text: 'Copy' },
    { text: 'Paste' }
  ];
  function onOpen(args:SpeedDialOpenCloseEventArgs){
    //Your required action here
  }
  return (
    */ Initialize the SpeedDial component */
    <SpeedDialComponent id='speeddial' items={items} content='Edit' onOpen={ onOpen
    }></SpeedDialComponent>
  );
}
export default App;
`
```

### beforeClose

The SpeedDial component triggers the [beforeClose](#) event with [SpeedDialBeforeOpenCloseEventArgs](#) argument before the SpeedDial popup is closed.

```
`ts
import { SpeedDialComponent, SpeedDialItemModel, SpeedDialBeforeOpenCloseEventArgs } from
 '@syncfusion/ej2-react-buttons';
```

```

import * as React from 'react';
// Initialize action items. //
function App() {
  const items: SpeedDialItemModel[] = [
    { text: 'Cut' },
    { text: 'Copy' },
    { text: 'Paste' }
  ];
  function beforeClose(args:SpeedDialBeforeOpenCloseEventArgs){
    //Your required action here
  }
  return (
    // Initialize the SpeedDial component //
    <SpeedDialComponent id='speeddial' items={items} content='Edit' beforeClose={ beforeClose }></SpeedDialComponent>
  );
}
export default App;
`

```

### onClose

The SpeedDial component triggers the [onClose](#) event with [SpeedDialOpenCloseEventArgs](#) argument when SpeedDial popup is closed.

`ts

```

import { SpeedDialComponent, SpeedDialItemModel, SpeedDialOpenCloseEventArgs } from
 '@syncfusion/ej2-react-buttons';
import * as React from 'react';
// Initialize action items. //
function App() {
  const items: SpeedDialItemModel[] = [
    { text: 'Cut' },
    { text: 'Copy' },
    { text: 'Paste' }
  ];
  function onClose(args:SpeedDialOpenCloseEventArgs){

```



```
//Your required action here
}
return (
  {/ Initialize the SpeedDial component /}
  <SpeedDialComponent id='speeddial' items={items} content='Edit' onClose={ onClose
}></SpeedDialComponent>
);
}
export default App;
`
```

### [beforeItemRender](#)

The SpeedDial component triggers the [beforeItemRender](#) event with [SpeedDialItemEventArgs](#) argument for each **SpeedDialItem** once it is rendered.

```
`ts
import { SpeedDialComponent, SpeedDialItemModel, SpeedDialItemEventArgs } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
{/ Initialize action items. /}
function App() {
  const items: SpeedDialItemModel[] = [
    { text: 'Cut' },
    { text: 'Copy' },
    { text: 'Paste' }
  ];
  function beforeItemRender(args:SpeedDialItemEventArgs){
    //Your required action here
  }
  return (
    {/ Initialize the SpeedDial component /}
    <SpeedDialComponent id='speeddial' items={items} content='Edit' beforeItemRender={
      this.beforeItemRender=this.beforeItemRender.bind(this) }></SpeedDialComponent>
  );
}
export default App;
`
```

Below example demonstrates the clicked event of the Speed Dial component.

### APP.JSX

```
{% raw %}
import { SpeedDialComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* Initialize action items. */ }
function App() {
    const items = [
        { text: 'Cut' },
        { text: 'Copy' },
        { text: 'Paste' }
    ];
    function itemClick(args) {
        alert(args.item.text + " is clicked");
    }
    return (<div>
        <div id="targetElement" style={{ position: 'relative', minHeight:
'350px', border: '1px solid' }}></div>
        { /* Initialize the SpeedDial component */ }
        <SpeedDialComponent id='speeddial' items={items} content='Edit'
target='#targetElement' clicked={itemClick}></SpeedDialComponent>
        </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}
```

### APP.TSX

```
{% raw %}
import { SpeedDialComponent, SpeedDialItemModel, SpeedDialItemEventArgs }
from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{ /* Initialize action items. */ }
function App() {
    const items: SpeedDialItemModel[] = [
        { text: 'Cut' },
        { text: 'Copy' },
        { text: 'Paste' }
    ];
    function itemClick(args: SpeedDialItemEventArgs) {
        alert(args.item.text + " is clicked")
    }
    return (
        <div>
            <div id="targetElement" style={{position:'relative',
minHeight:'350px', border:'1px solid'}}></div>
            { /* Initialize the SpeedDial component */ }
            <SpeedDialComponent id='speeddial' items={items} content='Edit'
target='#targetElement' clicked={ itemClick }></SpeedDialComponent>
            </div>
        );
}
}
```

```
export default App;
ReactDOM.render(<App />, document.getElementById('button')) ;
{% endraw %}
```

## Accessibility in React Speed dial component

The Speed Dial component followed the accessibility guidelines and standards, including [ADA](#), [Section 508](#), [WCAG 2.2](#) standards, and [WCAG roles](#) that are commonly used to evaluate accessibility.

The accessibility compliance for the Speed Dial component is outlined below.

| Accessibility Criteria | Compatibility |

| -- | -- |

| [WCAG 2.2](#) Support |  |

| [Section 508](#) Support |  |

| Screen Reader Support |  |

| Right-To-Left Support |  |

| Color Contrast |  |

| Mobile Device Support |  |

| Keyboard Navigation Support |  |

| [Accessibility Checker](#) Validation |  |

| [Axe-core](#) Accessibility Validation |  |

<style>

.post .post-content img {

display: inline-block;

margin: 0.5em 0;

}

</style>

<div> - All features of the component meet the requirement.</div>

<div> - Some features of the component do not meet the requirement.</div>

<div> - The component does not meet the requirement.</div>

### WAI-ARIA attributes

The Speed Dial component followed the [WAI-ARIA](#) patterns to meet the accessibility. The following ARIA attributes are used in the Speed Dial component:

Attributes	Purpose
-----	-----
<code>role=menu</code>	Specifies that the SpeedDial item has a submenu.
<code>role=menuitem</code>	Indicates an actionable item within the SpeedDial submenu.
<code>aria-label</code>	Indicates the SpeedDial Popup item text.
<code>aria-expanded</code>	It indicates whether the SpeedDial current state is expanded or collapsed.
<code>aria-haspopup</code>	It indicates whether the SpeedDial has popup items or not.
<code>aria-controls</code>	Attribute is set to the SpeedDial button and it points to the corresponding content.
<code>aria-disabled</code>	Indicates the state of menu item whether it is disabled.

### Keyboard interaction

The Speed Dial component followed the [keyboard interaction](#) guideline, making it easy for people who use assistive technologies (AT) and those who completely rely on keyboard navigation. The following keyboard shortcuts are supported by the Speed Dial component.

Press	To do this
-----	-----
<code>Enter</code>	Open/close the menu.
<code>Up-arrow</code>	Focuses the next menu item.
<code>Left-Arrow</code>	Focuses the previous menu item.
<code>Down-Arrow</code>	Focuses the previous menu item.
<code>Right-Arrow</code>	Focuses the next menu item.
<code>Home</code>	Focuses the first menu item.
<code>End</code>	Focuses the last menu item.
<code>Esc</code>	Closes the menu.

### Ensuring accessibility

The Speed Dial component's accessibility levels are ensured through an [accessibility-checker](#) and [axe-core](#) software tools during automated testing.

The accessibility compliance of the Speed Dial component is shown in the following sample. Open the [sample](#) in a new window to evaluate the accessibility of the Speed Dial component with accessibility tools.

See also

- [Accessibility in Syncfusion React components](#)

## Spinner

### Getting Started

Initialize the Spinner using `createSpinner` method.

You can show/hide Spinner by using `showSpinner` and `hideSpinner` methods accordingly. You need to set target to render based on specific target.

The following steps explains you on how to create and how to show/hide your Spinner.

\*Import the `createSpinner` method from `ej2-popups` library into your file as shown in below.

,

```
import { createSpinner } from '@syncfusion/ej2-popups';
```

,

\*Show and hide this spinner by using `showSpinner` and `hideSpinner` methods for loading in your page and import them in your file as shown in below.

,

```
import { showSpinner, hideSpinner } from '@syncfusion/ej2-popups';
```

,

### Create the Spinner globally

The Spinner can be render globally in a page using public exported functions of the `ej2-popups` package.

[Class-component]

### APP.JSX

```
import { createSpinner, showSpinner } from '@syncfusion/ej2-popups';
import * as React from 'react';
export default class App extends React.Component {
  componentDidMount(prevProps) {
    //createSpinner() method is used to create spinner
    createSpinner({
      // Specify the target for the spinner to show
      target: document.getElementById('container'),
    });
    // showSpinner() will make the spinner visible
    showSpinner(document.getElementById('container'));
  }
  render() {
    return (
      <div className="control-pane">
        <div id="container" className="control-section col-lg-12 spinner-
target"></div>
      </div>);
  }
}
```

**APP.TSX**

```
import { createSpinner, showSpinner, hideSpinner } from '@syncfusion/ej2-
popups';
import * as React from 'react';
import * as ReactDOM from "react-dom";
export default class App extends React.Component<{}>, {}> {
  componentDidMount(prevProps) {
    //createSpinner() method is used to create spinner
    createSpinner({
      // Specify the target for the spinner to show
      target: document.getElementById('container'),
    });
    // showSpinner() will make the spinner visible
    showSpinner(document.getElementById('container'));
  }
  render() {
    return (
      <div className="control-pane">
        <div id="container"
          className="control-section col-lg-12 spinner-target"
        ></div>
      </div>
    );
  }
}
```

**[Functional-component]****APP.JSX**

```
import { createSpinner, showSpinner } from '@syncfusion/ej2-popups';
import * as React from 'react';
import { useEffect } from "react";
function App() {
  useEffect(() => {
    componentDidMount();
  });
  function componentDidMount(prevProps) {
    //createSpinner() method is used to create spinner
    createSpinner({
      // Specify the target for the spinner to show
      target: document.getElementById('container'),
    });
    // showSpinner() will make the spinner visible
    showSpinner(document.getElementById('container'));
  }
  return (<div className="control-pane">
    <div id="container" className="control-section col-lg-12 spinner-
target"></div>
  </div>);
}
```

**APP.TSX**

```
import { createSpinner, showSpinner, hideSpinner } from '@syncfusion/ej2-
popups';
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { useEffect } from "react";
function App() {
  useEffect(() => {
    //createSpinner() method is used to create spinner
    createSpinner({
      // Specify the target for the spinner to show
      target: document.getElementById('container'),
    });
    // showSpinner() will make the spinner visible
    showSpinner(document.getElementById('container'));
  })
  return (
    <div className="control-pane">
      <div id="container"
        className="control-section col-lg-12 spinner-target"
      ></div>
    </div>
  );
}
export default App;
```

### Template in React Spinner component

You can use custom templates on the Spinner instead of the default Spinner by specifying the template in the `setSpinner` method.

The following steps explain you on how to define template for Spinner.

\*Import the `setSpinner` method from `ej2-react-popups` library into your `index.tsx` as shown in below.

```
import { setSpinner } from '@syncfusion/ej2-react-popups';
```

\*Pass your custom template to the `setSpinner` method like as below.

```
`ts
```

```
// Specify the template content to be displayed in the Spinner
```

```
setSpinner({ template: '<div style="width:100%;height:100%" class="custom-rolling"><div></div></div>'});
```

You should set the template to the Spinner before creating the respective Essential JS 2 component.

Also, until we replace `setSpinner` template, the further Essential JS 2 component rendering is created with given template only.

- Now, render the Essential JS 2 component. It's render the Spinner with the template specified in the `setSpinner` method.

In the below sample, we have rendered the Grid component with custom Spinner using `setSpinner` method.

You have to define the styles for the template in `index.html`.

[Class-component]

#### APP.JSX

```
import * as React from 'react';
import { GridComponent, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-grids';
import { setSpinner } from '@syncfusion/ej2-react-popups';
import { data } from './datasource';
export default class App extends React.Component {
  gridData = data.slice(0, 7);
  onFirstGridCreated() {
    this.hideSpinner = () => true;
    // Specify the template content to be displayed in the Spinner
    setSpinner({ template: '<div style="width:100%;height:100%"
class="custom-rolling"><div></div></div>' });
  }
  onSecondGridCreated() {
    this.hideSpinner = () => true;
  }
  render() {
    return (<div>
      <GridComponent id='grid1' dataSource={this.gridData}
created={this.onFirstGridCreated}>
        <ColumnsDirective>
          <ColumnDirective field='OrderID' headerText='Order
ID' width='120' textAlign='right' type='number'></ColumnDirective>
          <ColumnDirective field='CustomerID'
headerText='Customer ID' width='140' type='string'></ColumnDirective>
          <ColumnDirective field='Freight'
headerText='Freight' width='120' format='C' textAlign='right' />
          <ColumnDirective field='OrderDate' headerText='Order
Date' width='140' format='yMd' textAlign='right' />
        </ColumnsDirective>
      </GridComponent>
      <br /><br /><br />
      <GridComponent id='grid2' dataSource={this.gridData}
created={this.onSecondGridCreated}>
        <ColumnsDirective id='grid2' dataSource={this.gridData}>
          <ColumnDirective field='OrderID' headerText='Order
ID' width='120' textAlign='right' type='number'></ColumnDirective>
          <ColumnDirective field='CustomerID'
headerText='Customer ID' width='140' type='string'></ColumnDirective>
          <ColumnDirective field='Freight'
headerText='Freight' width='120' format='C' textAlign='right' />
          <ColumnDirective field='OrderDate' headerText='Order
Date' width='140' format='yMd' textAlign='right' />
        </ColumnsDirective>
      </GridComponent>
    </div>);
  }
}
```



**APP.TSX**

```

import * as React from 'react';
import * as ReactDOM from "react-dom";
import { GridComponent, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-grids';
import { setSpinner } from '@syncfusion/ej2-react-popups';
import { data } from './datasource';
export default class App extends React.Component<{}, {}> {
    private gridData: Object[] = data.slice(0, 7);
    public onFirstGridCreated(): void {
        this.hideSpinner = () => true;
        // Specify the template content to be displayed in the Spinner
        setSpinner({ template: '<div style="width:100%;height:100%"
class="custom-rolling"><div></div></div>' });
    }
    public onSecondGridCreated(): void {
        this.hideSpinner = () => true;
    }
    render() {
        return (
            <div>
                <GridComponent id='grid1' dataSource={this.gridData}
created={this.onFirstGridCreated}>
                    <ColumnsDirective>
                        <ColumnDirective field='OrderID' headerText='Order
ID' width='120' textAlign='right' type='number'></ColumnDirective>
                        <ColumnDirective field='CustomerID'
headerText='Customer ID' width='140' type='string'></ColumnDirective>
                        <ColumnDirective field='Freight'
headerText='Freight' width='120' format='C' textAlign='right' />
                        <ColumnDirective field='OrderDate' headerText='Order
Date' width='140' format='yMd' textAlign='right' />
                    </ColumnsDirective>
                </GridComponent>
                <br/><br/>
                <GridComponent id='grid2' dataSource={this.gridData}
created={this.onSecondGridCreated}>
                    <ColumnsDirective id='grid2' dataSource={this.gridData}>
                        <ColumnDirective field='OrderID' headerText='Order
ID' width='120' textAlign='right' type='number'></ColumnDirective>
                        <ColumnDirective field='CustomerID'
headerText='Customer ID' width='140' type='string'></ColumnDirective>
                        <ColumnDirective field='Freight'
headerText='Freight' width='120' format='C' textAlign='right' />
                        <ColumnDirective field='OrderDate' headerText='Order
Date' width='140' format='yMd' textAlign='right' />
                    </ColumnsDirective>
                </GridComponent>
            </div>
        );
    }
}

```

**[Functional-component]****APP.JSX**

```

import * as React from 'react';
import { GridComponent, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-grids';
import { setSpinner } from '@syncfusion/ej2-react-popups';
import { data } from './datasource';
function App() {
    let gridData = data.slice(0, 7);
    const [status, setStatus] = React.useState({ hideSpinner: true });
    function onFirstGridCreated() {
        setStatus({ hideSpinner: true });
        // Specify the template content to be displayed in the Spinner
        setSpinner({ template: '<div style="width:100%;height:100%"
class="custom-rolling"><div></div></div>' });
    }
    function onSecondGridCreated() {
        setStatus({ hideSpinner: true });
    }
    return (<div>
        <GridComponent id='grid1' dataSource={gridData}
created={onFirstGridCreated}>
            <ColumnsDirective>
                <ColumnDirective field='OrderID' headerText='Order
ID' width='120' textAlign='right' type='number'></ColumnDirective>
                <ColumnDirective field='CustomerID'
headerText='Customer ID' width='140' type='string'></ColumnDirective>
                <ColumnDirective field='Freight'
headerText='Freight' width='120' format='C' textAlign='right' />
                <ColumnDirective field='OrderDate' headerText='Order
Date' width='140' format='yMd' textAlign='right' />
            </ColumnsDirective>
        </GridComponent>
        <br /><br /><br />
        <GridComponent id='grid2' dataSource={gridData}
created={onSecondGridCreated}>
            <ColumnsDirective id='grid2' dataSource={gridData}>
                <ColumnDirective field='OrderID' headerText='Order
ID' width='120' textAlign='right' type='number'></ColumnDirective>
                <ColumnDirective field='CustomerID'
headerText='Customer ID' width='140' type='string'></ColumnDirective>
                <ColumnDirective field='Freight'
headerText='Freight' width='120' format='C' textAlign='right' />
                <ColumnDirective field='OrderDate' headerText='Order
Date' width='140' format='yMd' textAlign='right' />
            </ColumnsDirective>
        </GridComponent>
    </div>);
}

```

**APP.TSX**

```

import * as React from 'react';
import * as ReactDOM from "react-dom";

```

```

import { GridComponent, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-grids';
import { setSpinner } from '@syncfusion/ej2-react-popups';
import { data } from './datasource';
function App() {
    let gridData: Object[] = data.slice(0, 7);
    const [status, setStatus] = React.useState({ hideSpinner: true });
    function onFirstGridCreated(): void {
        setStatus({ hideSpinner: true });
        // Specify the template content to be displayed in the Spinner
        setSpinner({ template: '<div style="width:100%;height:100%"
class="custom-rolling"><div></div></div>' });
    }
    function onSecondGridCreated(): void {
        setStatus({ hideSpinner: true });
    }
    return (
        <div>
            <GridComponent id='grid1' dataSource={gridData}
created={onFirstGridCreated}>
                <ColumnsDirective>
                    <ColumnDirective field='OrderID' headerText='Order
ID' width='120' textAlign='right' type='number'></ColumnDirective>
                    <ColumnDirective field='CustomerID'
headerText='Customer ID' width='140' type='string'></ColumnDirective>
                    <ColumnDirective field='Freight'
headerText='Freight' width='120' format='C' textAlign='right' />
                    <ColumnDirective field='OrderDate' headerText='Order
Date' width='140' format='yMd' textAlign='right' />
                </ColumnsDirective>
            </GridComponent>
            <br/><br/><br/>
            <GridComponent id='grid2' dataSource={gridData}
created={onSecondGridCreated}>
                <ColumnsDirective id='grid2' dataSource={gridData}>
                    <ColumnDirective field='OrderID' headerText='Order
ID' width='120' textAlign='right' type='number'></ColumnDirective>
                    <ColumnDirective field='CustomerID'
headerText='Customer ID' width='140' type='string'></ColumnDirective>
                    <ColumnDirective field='Freight'
headerText='Freight' width='120' format='C' textAlign='right' />
                    <ColumnDirective field='OrderDate' headerText='Order
Date' width='140' format='yMd' textAlign='right' />
                </ColumnsDirective>
            </GridComponent>
        </div>
    );
}
export default App;

```

### Types in React Spinner component

By default, the Spinner is loaded in the applicable Essential JS 2 component based on the theme imported into the page. Based on the theme, the type is set to the Spinner.

The available types are:

- Material
- Fabric
- Bootstrap

You can change the Essential JS 2 component spinner type by passing the type of the spinner as parameter to the `setSpinner` method like as below.

```
`ts
// Specify the type of the Spinner to be displayed
setSpinner({ type: 'Bootstrap'});
`
```

After Essential JS 2 component creation only, you can change the Essential JS 2 component spinner type.

[Class-component]

#### APP.JSX

```
import * as React from 'react';
import { GridComponent, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-grids';
import { setSpinner } from '@syncfusion/ej2-react-popups';
import { data } from './datasource';
// By default, Spinner is rendered with specified theme inside the Grid
component.
export default class App extends React.Component {
  gridData = data.slice(0, 7);
  onGridCreated() {
    this.hideSpinner = () => true;
    // Specify the type of the Spinner to be displayed in the Grid.
    setSpinner({ type: 'Bootstrap' });
  }
  render() {
    return (
      <div>
        <GridComponent id='grid' dataSource={this.gridData}
        created={this.onGridCreated}>
          <ColumnsDirective>
            <ColumnDirective field='OrderID' headerText='Order
            ID' width='120' textAlign='right' type='number'></ColumnDirective>
            <ColumnDirective field='CustomerID'
            headerText='Customer ID' width='140' type='string'></ColumnDirective>
            <ColumnDirective field='Freight'
            headerText='Freight' width='120' format='C'
            textAlign='right'></ColumnDirective>
            <ColumnDirective field='OrderDate' headerText='Order
            Date' width='140' format='yMd' textAlign='right'></ColumnDirective>
          </ColumnsDirective>
        </GridComponent>
      </div>
    );
  }
}
```

#### APP.TSX

```

import * as React from 'react';
import * as ReactDOM from "react-dom";
import { GridComponent, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-grids';
import { setSpinner } from '@syncfusion/ej2-react-popups';
import { data } from './datasource';
// By default, Spinner is rendered with specified theme inside the Grid
component.
export default class App extends React.Component<{}, {}> {
  private gridData: Object[] = data.slice(0, 7);
  public onGridCreated(): void {
    this.hideSpinner = () => true;
    // Specify the type of the Spinner to be displayed in the Grid.
    setSpinner({ type: 'Bootstrap' });
  }
  render() {
    return (
      <div>
        <GridComponent id='grid' dataSource={this.gridData}
created={this.onGridCreated}>
          <ColumnsDirective>
            <ColumnDirective field='OrderID' headerText='Order
ID' width='120' textAlign='right' type='number'></ColumnDirective>
            <ColumnDirective field='CustomerID'
headerText='Customer ID' width='140' type='string'></ColumnDirective>
            <ColumnDirective field='Freight'
headerText='Freight' width='120' format='C'
textAlign='right'></ColumnDirective>
            <ColumnDirective field='OrderDate' headerText='Order
Date' width='140' format='yMd' textAlign='right'></ColumnDirective>
          </ColumnsDirective>
        </GridComponent>
      </div>
    );
  }
}

```

### [Functional-component]

#### APP.JSX

```

import * as React from 'react';
import { GridComponent, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-grids';
import { setSpinner } from '@syncfusion/ej2-react-popups';
import { data } from './datasource';
// By default, Spinner is rendered with specified theme inside the Grid
component.
function App() {
  let gridData = data.slice(0, 7);
  const [status, setStatus] = React.useState({ hideSpinner: true });
  function onGridCreated() {
    setStatus({ hideSpinner: true });
    // Specify the type of the Spinner to be displayed in the Grid.
    setSpinner({ type: 'Bootstrap' });
  }
}

```

```

    return (<div>
      <GridComponent id='grid' dataSource={gridData}
created={onGridCreated}>
        <ColumnsDirective>
          <ColumnDirective field='OrderID' headerText='Order
ID' width='120' textAlign='right' type='number'></ColumnDirective>
          <ColumnDirective field='CustomerID'
headerText='Customer ID' width='140' type='string'></ColumnDirective>
          <ColumnDirective field='Freight'
headerText='Freight' width='120' format='C'
textAlign='right'></ColumnDirective>
          <ColumnDirective field='OrderDate' headerText='Order
Date' width='140' format='yMd' textAlign='right'></ColumnDirective>
        </ColumnsDirective>
      </GridComponent>
    </div>);
  }
}

```

### APP.TSX

```

import * as React from 'react';
import * as ReactDOM from "react-dom";
import { GridComponent, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-grids';
import { setSpinner } from '@syncfusion/ej2-react-popups';
import { data } from './datasource';
// By default, Spinner is rendered with specified theme inside the Grid
component.
function App() {
  let gridData: Object[] = data.slice(0, 7);
  const [status, setStatus] = React.useState({ hideSpinner: true });
  function onGridCreated(): void {
    setStatus({ hideSpinner: true });
    // Specify the type of the Spinner to be displayed in the Grid.
    setSpinner({ type: 'Bootstrap' });
  }

  return (
    <div>
      <GridComponent id='grid' dataSource={gridData}
created={onGridCreated}>
        <ColumnsDirective>
          <ColumnDirective field='OrderID' headerText='Order
ID' width='120' textAlign='right' type='number'></ColumnDirective>
          <ColumnDirective field='CustomerID'
headerText='Customer ID' width='140' type='string'></ColumnDirective>
          <ColumnDirective field='Freight'
headerText='Freight' width='120' format='C'
textAlign='right'></ColumnDirective>
          <ColumnDirective field='OrderDate' headerText='Order
Date' width='140' format='yMd' textAlign='right'></ColumnDirective>
        </ColumnsDirective>
      </GridComponent>
    </div>
  );
}
export default App;

```

### Style in React Spinner component

The following content provides the exact CSS structure that can be used to modify the control's appearance based on the user preference.

#### Customizing the spinner

Use the following CSS to customize the spinner stroke color.

##### *Material theme*

```
`css
.e-spinner-pane .e-spinner-inner .e-spin-material {
stroke: green;
}
`
```

##### *Fabric theme*

```
`css
.e-spinner-pane .e-spinner-inner .e-spin-fabric {
stroke: green;
}
`
```

##### *Bootstrap theme*

```
`css
.e-spinner-pane .e-spinner-inner .e-spin-bootstrap {
fill: green;
stroke: green;
}
`
```

##### *Bootstrap4 theme*

```
`css
.e-spinner-pane .e-spinner-inner .e-spin-bootstrap4 {
stroke: green;
}
`
```

##### *High Contrast theme*

```
`css
.e-spinner-pane .e-spinner-inner .e-spin-high-contrast .e-path-arc {
stroke: green;
}
```

```
}  
,
```

## SplitButton

### Getting Started

This section briefly explains how to create a simple **SplitButton** component and configure its available functionalities in React.

### Dependencies

The following list of dependencies are required to use the **SplitButton** component in your application.

```
`js  
|-- @syncfusion/ej2-react-splitbuttons  
|-- @syncfusion/ej2-react-base  
|-- @syncfusion/ej2-base  
|-- @syncfusion/ej2-splitbuttons  
|-- @syncfusion/ej2-popups  
|-- @syncfusion/ej2-buttons  
,
```

### Installation and configuration

You can use [Create-react-app](#) to setup the applications.

To install **create-react-app** run the following command.

```
`bash  
npm install -g create-react-app  
,
```

To set-up a React application in TypeScript environment, run the following command.

```
`bash  
npx create-react-app my-app --template typescript  
cd my-app  
npm start  
,
```

To set-up a React application in JavaScript environment, run the following command.

```
`bash  
npx create-react-app my-app  
cd my-app  
npm start  
,
```



### Adding syncfusion packages

All the available Essential JS 2 packages are published in [npmjs.com](https://www.npmjs.com) public registry. You can choose the component that you want to install.

To install SplitButton component, use the following command

```
`bash
npm install @syncfusion/ej2-react-splitbuttons --save
`
```

### Adding CSS reference

Import the SplitButton component required CSS references as follows in `src/App.css`.

```
`css
/ import the SplitButton dependency styles /
@import "../node_modules/@syncfusion/ej2-base/styles/material.css";
@import "../node_modules/@syncfusion/ej2-buttons/styles/material.css";
@import "../node_modules/@syncfusion/ej2-popups/styles/material.css";
@import "../node_modules/@syncfusion/ej2-splitbuttons/styles/material.css";
`
```

### Adding SplitButton component

Now, you can start adding SplitButton component in the application. For getting started, add the SplitButton component in `src/App.tsx` file using following code.

Add the below code in the `src/App.tsx` to initialize the SplitButton.

```
`ts
import { enableRipple } from '@syncfusion/ej2-base';
import { SplitButtonComponent } from '@syncfusion/ej2-react-splitbuttons';
import * as React from 'react';
import './App.css';
enableRipple(true);
// To render SplitButton.
function App() {
  return (
    <div style={{marginTop: '150px'}}>
      <SplitButtonComponent id="element"/>
    </div>
  );
}
```

```
export default App;  
`
```

### Binding data source

After initialization, populate the SplitButton with data using the [items](#) property. Here, an array of string values is passed to the SplitButton component.

```
`ts  
import { enableRipple } from '@syncfusion/ej2-base';  
import { ItemModel, SplitButtonComponent } from '@syncfusion/ej2-react-splitbuttons';  
import * as React from 'react';  
import './App.css';  
enableRipple(true);  
// To render SplitButton.  
function App() {  
  let items: ItemModel[] = [  
    {  
      text: 'Cut',  
    },  
    {  
      text: 'Copy',  
    },  
    {  
      text: 'Paste',  
    }  
  ];  
  return (  
    <div style={{marginTop: '150px'}}><SplitButtonComponent id="element" items = {items}> Paste  
    </SplitButtonComponent>  
    </div>  
  );  
}  
export default App;  
`
```

### Run the application

After completing the configuration required to render a basic SplitButton, run the following command to display the output in your default browser.

`

npm start

`

The following example shows a basic SplitButton component.

#### **APP.JSX**

```
import { enableRipple } from '@syncfusion/ej2-base';
import { SplitButtonComponent } from '@syncfusion/ej2-react-splitbuttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render SplitButton.
function App() {
    let items = [
        {
            text: 'Cut',
        },
        {
            text: 'Copy',
        },
        {
            text: 'Paste',
        }
    ];
    return (<div>
        <SplitButtonComponent id="element" items={items}> Paste
    </SplitButtonComponent>
    </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
```

#### **APP.TSX**

```
import { enableRipple } from '@syncfusion/ej2-base';
import { ItemModel, SplitButtonComponent } from '@syncfusion/ej2-react-splitbuttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render SplitButton.
function App() {
    let items: ItemModel[] = [
        {
            text: 'Cut',
        },
        {
            text: 'Copy',
        },
        {
            text: 'Paste',
        }
    ];
    return (
        <div>
```

```

    <SplitButtonComponent id="element" items = {items}> Paste
  </SplitButtonComponent>
</div>
);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

### See Also

- [SplitButton with icons](#)

### Icons and separator in React Split button component

#### SplitButton icons

SplitButton can have an icon to provide the visual representation of the action. To place the icon on a SplitButton, set the [iconCss](#) property to **e-icons** with the required icon CSS. By default, the icon is positioned to the left side of the SplitButton. You can customize the icon's position by using the [iconPosition](#) property

The following example illustrates how to place icon in SplitButton component.

#### APP.JSX

```

import { enableRipple } from '@syncfusion/ej2-base';
import { SplitButtonComponent } from '@syncfusion/ej2-react-splitbuttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render SplitButton.
function App() {
  let items = [
    {
      text: 'Cut',
    },
    {
      text: 'Copy',
    },
    {
      text: 'Paste',
    }
  ];
  return (
    <div>
      <SplitButtonComponent items={items} iconCss='e-sb-icons e-paste'>Paste</SplitButtonComponent>
      <SplitButtonComponent items={items} iconPosition="Top" iconCss='e-sb-icons e-paste'>Paste</SplitButtonComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

#### APP.TSX

```

import { enableRipple } from '@syncfusion/ej2-base';

```

```
import { ItemModel, SplitButtonComponent } from '@syncfusion/ej2-react-splitbuttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render SplitButton.
function App() {
  let items: ItemModel[] = [
    {
      text: 'Cut',
    },
    {
      text: 'Copy',
    },
    {
      text: 'Paste',
    }
  ];
  return (
    <div>
      <SplitButtonComponent items = {items} iconCss= 'e-sb-icons e-paste'>Paste</SplitButtonComponent>
      <SplitButtonComponent items = {items} iconPosition= "Top" iconCss= 'e-sb-icons e-paste'>Paste</SplitButtonComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
```

The Essential JS 2 provides a set of icons that can be loaded by applying **e-icons** class name to the element.

You can also use third party icons on the SplitButton using the [iconCss](#) property.

### Vertical button

Vertical Button in SplitButton can be achieved by adding **e-vertical** class using [cssClass](#) property.

The following example illustrates how to vertical support in SplitButton component.

### APP.JSX

```
import { enableRipple } from '@syncfusion/ej2-base';
import { SplitButtonComponent } from '@syncfusion/ej2-react-splitbuttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render SplitButton.
function App() {
  let items = [
    {
      text: 'Cut',
    },
    {
      text: 'Copy',
    },
    {
      text: 'Paste',
    }
  ];
}
```

```

    }
    ];
    return (<div>
      <SplitButtonComponent cssClass='e-vertical' items={items} iconCss='e-
sb-icons e-paste' iconPosition='Top'>Paste</SplitButtonComponent>
    </div>);
  }
  export default App;
  ReactDOM.render(<App />, document.getElementById('button'));

```

### APP.TSX

```

import { enableRipple } from '@syncfusion/ej2-base';
import { ItemModel, SplitButtonComponent } from '@syncfusion/ej2-react-
splitbuttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render SplitButton.
function App() {
  let items: ItemModel[] = [
    {
      text: 'Cut',
    },
    {
      text: 'Copy',
    },
    {
      text: 'Paste',
    }
  ];
  return (
    <div>
      <SplitButtonComponent cssClass= 'e-vertical' items = {items} iconCss=
'e-sb-icons e-paste' iconPosition='Top'>Paste</SplitButtonComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

The Essential JS 2 provides a set of icons that can be loaded by applying **e-icons** class name to the element.

You can also use third party icons on the SplitButton using the [iconCss](#) property.

### Separator

SplitButton component has Separator support. This can be achieved by setting **separator** as **true**.

The following example illustrates how to enable separator support in SplitButton component.

### APP.JSX

```

import { enableRipple } from '@syncfusion/ej2-base';
import { SplitButtonComponent } from '@syncfusion/ej2-react-splitbuttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';

```

```

enableRipple(true);
// To render SplitButton.
function App() {
  let items = [
    {
      iconCss: 'e-sb-icons e-cut',
      text: 'Cut'
    },
    {
      iconCss: 'e-icons e-copy',
      text: 'Copy'
    },
    {
      iconCss: 'e-sb-icons e-paste',
      text: 'Paste'
    },
    {
      separator: true
    },
    {
      iconCss: 'e-sb-icons e-font',
      text: 'Font'
    },
    {
      iconCss: 'e-sb-icons e-paragraph',
      text: 'Paragraph'
    }
  ];
  return (<div>
    <SplitButtonComponent items={items} iconCss='e-sb-icons e-
paste'>Paste</SplitButtonComponent>
    </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

**APP.TSX**

```

import { enableRipple } from '@syncfusion/ej2-base';
import { ItemModel, SplitButtonComponent } from '@syncfusion/ej2-react-
splitbuttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render SplitButton.
function App() {
  let items: ItemModel[] = [
    {
      iconCss: 'e-sb-icons e-cut',
      text: 'Cut'
    },
    {
      iconCss: 'e-icons e-copy',
      text: 'Copy'
    },
    {

```

```

        iconCss: 'e-sb-icons e-paste',
        text: 'Paste'
    },
    {
        separator: true
    },
    {
        iconCss: 'e-sb-icons e-font',
        text: 'Font'
    },
    {
        iconCss: 'e-sb-icons e-paragraph',
        text: 'Paragraph'
    }
  ];
  return (
    <div>
      <SplitButtonComponent items = {items} iconCss= 'e-sb-icons e-paste'>Paste</SplitButtonComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

### See Also

- [SplitButton popup with icons](#)

## Popup items in React Split button component

### Icons

The Popup action item have an icon or image to provide visual representation of the action. To place the icon on a popup item, set the [iconCss](#) property to `e-icons` with the required icon CSS. By default, the icon is positioned to the left side of the popup action item.

In the following sample, the icons for Cut, Copy, and Paste menu items are added using the `iconCss` property

### APP.JSX

```

import { enableRipple } from '@syncfusion/ej2-base';
import { SplitButtonComponent } from '@syncfusion/ej2-react-splitbuttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render SplitButton.
function App() {
  let items = [
    {
      iconCss: 'e-sb-icons e-cut',
      text: 'Cut'
    },
    {
      iconCss: 'e-icons e-copy',
      text: 'Copy'
    }
  ];
}

```



```

    },
    {
      iconCss: 'e-sb-icons e-paste',
      text: 'Paste'
    }
  ];
  return (<div>
    <SplitButtonComponent items={items} iconCss='e-sb-icons e-
paste'>Paste</SplitButtonComponent>
  </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

**APP.TSX**

```

import { enableRipple } from '@syncfusion/ej2-base';
import { ItemModel, SplitButtonComponent } from '@syncfusion/ej2-react-
splitbuttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render SplitButton.
function App() {
  let items: ItemModel[] = [
    {
      iconCss: 'e-sb-icons e-cut',
      text: 'Cut'
    },
    {
      iconCss: 'e-icons e-copy',
      text: 'Copy'
    },
    {
      iconCss: 'e-sb-icons e-paste',
      text: 'Paste'
    }
  ];
  return (
    <div>
      <SplitButtonComponent items = {items} iconCss= 'e-sb-icons e-
paste'>Paste</SplitButtonComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

**Template***Item Templating*

Popup items can be customized by using the [beforeItemRender](#) event. The item render event triggers while rendering each Popup action item. The event argument will be used to identify the action item and customize it based on the requirement.

**APP.JSX**

```

import { createElement, enableRipple } from '@syncfusion/ej2-base';
import { SplitButtonComponent } from '@syncfusion/ej2-react-splitbuttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render SplitButton.
function App() {
    let items = [
        {
            text: 'Cut',
        },
        {
            text: 'Copy',
        },
        {
            text: 'Paste',
        }
    ];
    function itemBeforeEvent(args) {
        const shortCutSpan = createElement('span');
        const text = args.item.text;
        args.element.appendChild(shortCutSpan);
        shortCutSpan.setAttribute('class', 'shortcut');
        const clsName = (text === 'Copy') ? 'e-icons' : 'e-sb-icons';
        shortCutSpan.classList.add(clsName);
        (text === 'Cut') ? shortCutSpan.classList.add('e-cut') : (text ===
        'Paste') ? shortCutSpan.classList.add('e-paste') :
        shortCutSpan.classList.add('e-copy');
    }
    return (<div>
        <SplitButtonComponent items={items} beforeItemRender={itemBeforeEvent}
        iconCss='e-sb-icons e-paste' />
    </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

**APP.TSX**

```

import { createElement, enableRipple } from '@syncfusion/ej2-base';
import { ItemModel, MenuEventArgs, SplitButtonComponent } from
 '@syncfusion/ej2-react-splitbuttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render SplitButton.
function App() {
    let items: ItemModel[] = [
        {
            text: 'Cut',
        },
        {
            text: 'Copy',
        },
        {
            text: 'Paste',
        }
    ];

```

```

    }];
    function itemBeforeEvent(args: MenuEventArgs): void {
        const shortCutSpan = createElement('span');
        const text = args.item.text;
        args.element.appendChild(shortCutSpan);
        shortCutSpan.setAttribute('class', 'shortcut');
        const clsName = (text === 'Copy') ? 'e-icons' : 'e-sb-icons';
        shortCutSpan.classList.add(clsName);
        (text === 'Cut') ? shortCutSpan.classList.add('e-cut') : (text ===
        'Paste') ? shortCutSpan.classList.add('e-paste') :
        shortCutSpan.classList.add('e-copy');
    }
    return (
        <div>
            <SplitButtonComponent items = {items} beforeItemRender=
            {itemBeforeEvent} iconCss= 'e-sb-icons e-paste' />
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

### Popup Templating

The whole popup can be customized as per the requirement. In the following example, the popup can be customized by handling it in [target](#) property.

### APP.JSX

```

import { enableRipple } from '@syncfusion/ej2-base';
import { SplitButtonComponent } from '@syncfusion/ej2-react-splitbuttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render SplitButton.
function App() {
    return <div>
        <div id='dropdowntarget'>
            <div id='first'>
                <div id='black' />
                <div id='red' />
                <div id='green' />
                <div id='gray' />
                <div id='blue' />
                <div id='violet' />
                <div id='brown' />
                <div id='darkgoldenrod' />
                <div id='aquamarine' />
            </div>
        </div>
        <SplitButtonComponent target='#dropdowntarget' iconCss='e-sb-icons
        e-color' />
    </div>;
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

**APP.TSX**

```
import { enableRipple } from '@syncfusion/ej2-base';
import { SplitButtonComponent } from '@syncfusion/ej2-react-splitbuttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render SplitButton.
function App() {
  return (
    <div>
      <div id='dropdowntarget'>
        <div id= "first">
          <div id='black' />
          <div id='red' />
          <div id='green' />
          <div id='gray' />
          <div id='blue' />
          <div id='violet' />
          <div id='brown' />
          <div id='darkgoldenrod' />
          <div id='aquamarine' />
        </div>
      </div>
      <SplitButtonComponent target='#dropdowntarget' iconCss= 'e-sb-icons
e-color' />
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
```

See Also

- [Popup items grouping](#)
- [SplitButton popup with separator](#)

### Accessibility in React SplitButton component

The SplitButton component followed the accessibility guidelines and standards, including [ADA](#), [Section 508](#), [WCAG 2.2](#) standards, and [WCAG roles](#) that are commonly used to evaluate accessibility.

The accessibility compliance for the SplitButton component is outlined below.

| Accessibility Criteria | Compatibility |

| -- | -- |

| [WCAG 2.2](#) Support |  |

| [Section 508](#) Support |  |

| Screen Reader Support |  |

| Right-To-Left Support |  |

| Color Contrast |  |

| Mobile Device Support |  |

| Keyboard Navigation Support |  |

| [Accessibility Checker](#) Validation |  |

| [Axe-core](#) Accessibility Validation |  |

```
<style>
.post .post-content img {
display: inline-block;
margin: 0.5em 0;
}
</style>

<div> - All
features of the component meet the requirement.</div>

<div> - Some features of the component do not meet the requirement.</div>

<div> - The component does not meet the requirement.</div>
```

#### WAI-ARIA attributes

The SplitButton component followed the [WAI-ARIA] patterns to meet the accessibility. The following ARIA attributes are used in the SplitButton component:

| Attributes | Purpose |

| --- | --- |

| **role** | Indicates the SplitButton component as **button**, SplitButton popup as **menu**, and the SplitButton popup action items as **menuitem**. |

| **aria-haspopup** | Indicates the availability and type of interactive SplitButton popup element. |

| **aria-expanded** | Indicates whether the SplitButton popup can be expanded or collapsed, as well as indicates whether its current state is expanded or collapsed. |

| **aria-owns** | Identifies an elements in order to define a visual, functional, or contextual parent/child relationship between DOM elements where the DOM hierarchy cannot be used to represent the relationship. |

| `aria-disabled` | Indicates that the element is perceivable but disabled, so it is not editable or otherwise operable. |

#### Keyboard interaction

The SplitButton component followed the [keyboard interaction] guideline, making it easy for people who use assistive technologies (AT) and those who completely rely on keyboard navigation. The following keyboard shortcuts are supported by the SplitButton component.

| **Press** | **To do this** |

| --- | --- |

| **Esc** | Closes the opened popup. |

| **Enter** | Opens the popup, or activates the highlighted item and closes the popup. |

| **Spacer** | Opens the popup. |

| **Up** | Navigates up or to the previous action item. |

| **Down** | Navigates down or to the next action item. |

| **Alt + Up Arrow** | Closes the popup. |

| **Alt + Down Arrow** | Opens the popup. |

#### Ensuring accessibility

The SplitButton component's accessibility levels are ensured through an [accessibility-checker](#) and [axe-core](#) software tools during automated testing.

The accessibility compliance of the SplitButton component is shown in the following sample. Open the [sample](#) in a new window to evaluate the accessibility of the SplitButton component with accessibility tools.

See also

- [Accessibility in Syncfusion React components](#)

#### Style and appearance in React Split button component

To modify the SplitButton appearance, you need to override the default CSS of SplitButton component. Please find the list of CSS classes and its corresponding section in SplitButton. Also, you have an option to create your own custom theme for the controls using our [Theme Studio](#).

CSS Class | Purpose of Class

| `.e-dropdown-btn` | To customize the button part in splitbutton

| `.e-split-btn` | To customize the dropdown part in split button

| `.e-dropdown-popup ul .e-item` | To customize the progress button list items

| `.e-dropdown-popup ul .e-item .e-menu-icon` | To customize the progress button list items icon

#### How To

##### Create right to left splitbutton in React Split button component

SplitButton component has RTL support. This can be achieved by setting `enableRtl` as `true`.

The following example illustrates how to enable right-to-left support in SplitButton component.

### **APP.JSX**

```
import { enableRipple } from '@syncfusion/ej2-base';
import { SplitButtonComponent } from '@syncfusion/ej2-react-splitbuttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render SplitButton.
function App() {
    let items = [
        {
            text: 'Autosum'
        },
        {
            text: 'Average'
        },
        {
            text: 'Count numbers'
        },
        {
            text: 'Min'
        },
        {
            text: 'Max'
        }
    ];
    return (<div>
        <SplitButtonComponent items={items} enableRtl={true} iconCss='e-sb e-sigma'>Autosum</SplitButtonComponent>
    </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
```

### **APP.TSX**

```
import { enableRipple } from '@syncfusion/ej2-base';
import { ItemModel, SplitButtonComponent } from '@syncfusion/ej2-react-splitbuttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render SplitButton.
function App() {
    let items: ItemModel[] = [
        {
            text: 'Autosum'
        },
        {
            text: 'Average'
        },
        {
            text: 'Count numbers'
        },
        {

```

```

        text: 'Min'
      },
      {
        text: 'Max'
      }
    ];
    return (
      <div>
        <SplitButtonComponent items = {items} enableRtl = {true} iconCss= 'e-sb
e-sigma'>Autosum</SplitButtonComponent>
      </div>
    );
  }
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

### Group items in popup in React Split button component

Items in popup can be grouped in SplitButton by templating entire popup with ListView. To achieve grouping in ListView, check [ListView grouping](#) documentation. To template ListView in popup, create ListView with id `listview` and provide it as `target` for SplitButton.

The following example illustrates how to group items in popup using ListView component.

#### APP.JSX

```

import { enableRipple } from '@syncfusion/ej2-base';
import { SplitButtonComponent } from '@syncfusion/ej2-react-splitbuttons';
import { ListViewComponent } from '@syncfusion/ej2-react-lists';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render SplitButton.
function App() {
  // Datasource for listview.
  let listItems = [
    {
      'category': 'Basic',
      'text': 'Cut'
    },
    {
      'category': 'Basic',
      'text': 'Copy',
    },
    {
      'category': 'Basic',
      'text': 'Paste'
    },
    {
      'category': 'Advanced',
      'text': 'Paste as Formula'
    },
    {
      'category': 'Advanced',
      'text': 'Paste as Hyperlink'
    },
  ];
  let field = { groupBy: 'category' };

```



```

    return (<div>
      <ListViewComponent id="listview" dataSource={listItems}
fields={field} sortOrder="Descending"/>
      <SplitButtonComponent
target="#listview">ClipBoard</SplitButtonComponent>
    </div>);
  }
  export default App;
  ReactDOM.render(<App />, document.getElementById('button'));

```

### APP.TSX

```

import { enableRipple } from '@syncfusion/ej2-base';
import { SplitButtonComponent } from '@syncfusion/ej2-react-splitbuttons';
import { ListViewComponent } from '@syncfusion/ej2-react-lists';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render SplitButton.
function App() {
  // Datasource for listview.
  let listItems: Array<{ [key: string]: string; }> = [
    {
      'category': 'Basic',
      'text': 'Cut'
    },
    {
      'category': 'Basic',
      'text': 'Copy',
    },
    {
      'category': 'Basic',
      'text': 'Paste'
    },
    {
      'category': 'Advanced',
      'text': 'Paste as Formula'
    },
    {
      'category': 'Advanced',
      'text': 'Paste as Hyperlink'
    },
  ];
  let field: any = { groupBy: 'category' };
  return (
    <div>
      <ListViewComponent id="listview" dataSource={listItems}
fields={field} sortOrder="Descending"/>
      <SplitButtonComponent
target="#listview">ClipBoard</SplitButtonComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

### Open a dialog on popup item click in React Split button component

This section explains about how to open a dialog on SplitButton popup item click. This can be achieved by handling dialog open in [select](#) event of the SplitButton.

In the following example, Dialog will open while selecting **Update...** item.

#### APP.JSX

```
import { enableRipple } from '@syncfusion/ej2-base';
import { DialogComponent } from '@syncfusion/ej2-react-popups';
import { SplitButtonComponent } from '@syncfusion/ej2-react-splitbuttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render SplitButton.
function App() {
  let dialogRef;
  let items = [
    {
      text: 'Help'
    },
    {
      text: 'About'
    },
    {
      text: 'Update...'
    }
  ];
  let alertDlgButtons = [{
    buttonModel: {
      content: 'Ok',
      isPrimary: true
    },
    'click': () => {
      hide();
    }
  },
  {
    buttonModel: {
      content: 'Cancel',
      isPrimary: true
    },
    'click': () => {
      hide();
    }
  }
  ];
  function hide() {
    dialogRef.hide();
  }
  function select(args) {
    if (args.item.text === 'Update...') {
      dialogRef.show();
    }
  }
  return (<div>
```

```

    <DialogComponent ref={dialog => dialogRef = dialog} width='250px'
    id='dialog' content='Are you sure want to update?' header='Software Update'
    buttons={alertDlgButtons} visible={false}/>
    <SplitButtonComponent items={items}
    select={select}>Help</SplitButtonComponent>
  </div>;
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

### APP.TSX

```

import { enableRipple } from '@syncfusion/ej2-base';
import { DialogComponent } from '@syncfusion/ej2-react-popups';
import { ItemModel, MenuEventArgs, SplitButtonComponent } from
 '@syncfusion/ej2-react-splitbuttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render SplitButton.
function App() {
  let dialogRef: DialogComponent;
  let items: ItemModel[] = [
    {
      text: 'Help'
    },
    {
      text: 'About'
    },
    {
      text: 'Update...'
    }
  ];
  let alertDlgButtons: any [] = [{
    buttonModel: {
      content: 'Ok',
      isPrimary: true
    },
    'click': () => {
      hide();
    }
  },
  {
    buttonModel: {
      content: 'Cancel',
      isPrimary: true
    },
    'click': () => {
      hide();
    }
  }
  ];
  function hide() {
    dialogRef.hide();
  }
  function select (args: MenuEventArgs) {
    if (args.item.text === 'Update...') {
      dialogRef.show();
    }
  }
}

```

```

    }
  }
  return (
    <div>
      <DialogComponent ref={dialog => dialogRef = dialog as
DialogComponent} width='250px' id='dialog' content='Are you sure want to
update?' header='Software Update' buttons={alertDlgButtons}
visible={false}/>
      <SplitButtonComponent items = {items}
select={select}>Help</SplitButtonComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

### Set the disabled state in React Split button component

SplitButton component can be enabled or disabled using [disabled](#) property. To disable SplitButton component, set the disabled property as true.

The following example illustrates how to set the disable state in SplitButton component.

#### APP.JSX

```

import { enableRipple } from '@syncfusion/ej2-base';
import { SplitButtonComponent } from '@syncfusion/ej2-react-splitbuttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render SplitButton.
function App() {
  let items = [
    {
      text: 'Autosum'
    },
    {
      text: 'Average'
    },
    {
      text: 'Count numbers',
    },
    {
      text: 'Min'
    },
    {
      text: 'Max'
    }
  ];
  return (<div>
    <SplitButtonComponent items={items} disabled={true} iconCss='e-sb e-
sigma'>Autosum</SplitButtonComponent>
    </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

**APP.TSX**

```
import { enableRipple } from '@syncfusion/ej2-base';
import { SplitButtonComponent, ItemModel } from '@syncfusion/ej2-react-splitbuttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render SplitButton.
function App() {
    let items: ItemModel[] = [
        {
            text: 'Autosum'
        },
        {
            text: 'Average'
        },
        {
            text: 'Count numbers',
        },
        {
            text: 'Min'
        },
        {
            text: 'Max'
        }
    ];
    return (
        <div>
            <SplitButtonComponent items = {items} disabled = {true} iconCss= 'e-sb
e-sigma'>Autosum</SplitButtonComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
```

Underline a character in a text in React Split button component

Underline a particular character in a text can be handled in [beforeItemRender](#) event by adding `<u>` tag in between the text and given as innerHTML in `li` rendering.

In the following example, **C** is underlined in the text **Copy**.

**APP.JSX**

```
import { enableRipple } from '@syncfusion/ej2-base';
import { SplitButtonComponent } from '@syncfusion/ej2-react-splitbuttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render SplitButton.
function App() {
    let items = [
        {
            text: 'Cut'
        },
        {
            text: 'Copy'
        },
        {
            text: 'Paste'
        }
    ];
    return (
        <div>
            <SplitButtonComponent items = {items} disabled = {true} iconCss= 'e-sb
e-sigma'>Autosum</SplitButtonComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
```

```

        text: 'Copy'
    },
    {
        text: 'Paste'
    }
];
function itemRender(args) {
    if (args.item.text === 'Copy') {
        // To underline a particular text.
        args.element.innerHTML = '<u>C</u>opy';
    }
    return (<div>
        <SplitButtonComponent items={items}
beforeItemRender={itemRender}>Paste</SplitButtonComponent>
        </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

### APP.TSX

```

import { enableRipple } from '@syncfusion/ej2-base';
import { SplitButtonComponent, ItemModel, MenuEventArgs } from
 '@syncfusion/ej2-react-splitbuttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render SplitButton.
function App() {
    let items: ItemModel[] = [
        {
            text: 'Cut'
        },
        {
            text: 'Copy'
        },
        {
            text: 'Paste'
        }
    ];
    function itemRender(args: MenuEventArgs) {
        if (args.item.text === 'Copy') {
            // To underline a particular text.
            args.element.innerHTML = '<u>C</u>opy';
        }
    }
    return (
        <div>
            <SplitButtonComponent items = {items}
beforeItemRender={itemRender}>Paste</SplitButtonComponent>
            </div>
        );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

## Ej1 api migration in React Split button component

This article describes the API migration process of SplitButton component from Essential JS 1 to Essential JS 2.

### Properties

{% raw %}

| Behavior | API in Essential JS 1 | API in Essential JS 2 |

| --- | --- | --- |

| Specifies the text of the splitbutton | **Property:** *text* <br/><br/> <EJ.SplitButton id="splitbutton" text="login"></EJ.SplitButton> | **Property:** *content* <br/><br/> <SplitButtonComponent id="splitbutton" content="Paste"></SplitButtonComponent> |

| Popup content | **Property:** *target* <br/><br/> <EJ.SplitButton id="splitbutton" text="SplitButton" target="#target"></EJ.SplitButton> | **Property:** *target* <br/><br/> <SplitButtonComponent id="splitbutton" target="#target" content="SplitButton" ></SplitButtonComponent> |

| Popup items | Not applicable | **Property:** *items* <br/><br/> <SplitButtonComponent id="splitbutton" content="Paste" items={this.items}></SplitButtonComponent> <br/> constructor(props: {}) { <br/> &#160; this.items = [ <br/> &#160; { <br/> &#160; &#160; text: 'Cut' <br/> &#160; }, <br/> &#160; { <br/> &#160; &#160; text: 'Copy' <br/> &#160; }, <br/> &#160; { <br/> &#160; &#160; text: 'Paste' <br/> &#160; }]; <br/> } |

| Arrow position | **Property:** *arrowPosition* <br/><br/> <EJ.SplitButton id="splitbutton" text="login" target="#target" arrowPosition="left"></EJ.SplitButton> | Not applicable |

| Button mode | **Property:** *buttonMode* <br/><br/> <EJ.SplitButton id="splitbutton" text="login" target="#target" buttonMode="dropdown"></EJ.SplitButton> | Not applicable |

| Content type | **Property:** *contentType* <br/><br/> <EJ.SplitButton id="splitbutton" text="login" contentType="textonly" target="#target"></EJ.SplitButton> | Not applicable |

| Icons | **Property:** *prefixIcon* <br/><br/> <EJ.SplitButton id="splitbutton" text="login" contentType="textandimage" prefixIcon="e-icon e-handup" target="#target"></EJ.SplitButton> | **Property:** *iconCss* <br/><br/> <SplitButtonComponent id="splitbutton" content="Paste" items={this.items} iconCss="e-icons e-paste"></SplitButtonComponent> |

| Icon position | **Property:** *imagePosition* <br/><br/> <EJ.SplitButton id="splitbutton" text="login" contentType="textandimage" prefixIcon="e-icon e-handup" target="#target" imagePosition="imagetop"></EJ.SplitButton> | **Property:** *iconPosition* <br/><br/> <SplitButtonComponent id="splitbutton" content="Paste" items={this.items} iconCss="e-icons e-paste" iconPosition="Top"></SplitButtonComponent> |

| Secondary icon | **Property:** *suffixIcon* <br/><br/> <EJ.SplitButton id="splitbutton" text="login" contentType="imagetextimage" prefixIcon="e-icon e-handup" suffixIcon="e-icon e-palette" target="#target"></EJ.SplitButton> | Not applicable |

| Adding custom class | **Property:** *cssClass* <br/><br/> <EJ.SplitButton id="splitbutton" text="SplitButton" target="#target" cssClass="custom-class"></EJ.SplitButton> | **Property:**

```
cssClass <br/><br/> <SplitButtonComponent id="splitbutton" items={this.items}
cssClass="custom-class"></SplitButtonComponent> |
```

```
| Disabled state | Property: enabled <br/><br/> <EJ.SplitButton id="splitbutton" text="login"
target="#target" enabled={false}></EJ.SplitButton> | Property: disabled <br/><br/>
<SplitButtonComponent id="splitbutton" items={this.items}
disabled={true}></SplitButtonComponent> |
```

```
| RTL | Property: enableRTL <br/><br/> <EJ.SplitButton id="splitbutton" text="login"
target="#target" contentType="textandimage" prefixIcon="e-icon e-handup"
enableRTL={true}></EJ.SplitButton> | Property: enableRtl <br/><br/> <SplitButtonComponent
id="splitbutton" items={this.items} content="Paste" enableRtl={true} iconCss="e-icons e-
paste"></SplitButtonComponent> |
```

```
| Height | Property: height <br/><br/> <EJ.SplitButton id="splitbutton" text="login"
target="#target" height="30"></EJ.SplitButton> | Not applicable |
```

```
| Width | Property: width <br/><br/> <EJ.SplitButton id="splitbutton" text="login"
target="#target" width="100"></EJ.SplitButton> | Not applicable |
```

```
| HTML attributes | Property: htmlAttributes <br/><br/> var attributes = { disabled:"disabled" }; <br/>
<EJ.SplitButton id="splitbutton" text="login" target="#target"
htmlAttributes="attributes"></EJ.SplitButton> | Not applicable |
```

```
| Show rounded corner | Property: showRoundedCorner <br/><br/> <EJ.SplitButton id="splitbutton"
text="login" target="#target" showRoundedCorner={true}></EJ.SplitButton> | Not applicable |
```

```
| Size | Property: size <br/><br/> <EJ.SplitButton id="splitbutton" size="small" text="Small"
target="#target"></EJ.SplitButton> | Property: cssClass <br/><br/> <SplitButtonComponent
id="splitbutton" items={this.items} content="Small" cssClass="e-
small"></SplitButtonComponent> |
```

## Methods

| Behavior | API in Essential JS 1 | API in Essential JS 2 |

| --- | --- | --- |

```
| Destroy method | Method: destroy <br/><br/> <EJ.SplitButton id="splitbutton"
text="SplitButton" target="#target"></EJ.SplitButton><br/>var splitButton =
$('#splitbutton').ejSplitButton('instance');<br/> splitButton.destroy(); | Method: destroy <br/><br/>
<SplitButtonComponent id="splitbutton" items={this.items} content="SplitButton" ref={{scope}}
=> {this.splitButton = scope}></SplitButtonComponent><br/> constructor(props: {}) { <br/> &#160;
this.splitButton.destroy(); <br/> } |
```

```
| Disable method | Method: disable <br/><br/> <EJ.SplitButton id="splitbutton" text="SplitButton"
target="#target"></EJ.SplitButton><br/>var splitButton =
$('#splitbutton').ejSplitButton('instance');<br/> splitButton.disable(); | Not applicable |
```

```
| Enable method | Method: enable <br/><br/> <EJ.SplitButton id="splitbutton" text="SplitButton"
target="#target"></EJ.SplitButton><br/>var splitButton =
$('#splitbutton').ejSplitButton('instance');<br/> splitButton.enable(); | Not applicable |
```



| Hide popup | **Method:** *hide* <br/><br/> <EJ.SplitButton id="splitbutton" text="SplitButton" target="#target"></EJ.SplitButton><br/>var splitButton = \$('#splitbutton').ejSplitButton('instance');<br/> splitButton.hide(); | **Method:** *toggle* <br/><br/> <SplitButtonComponent id="splitbutton" items={this.items} content="SplitButton" ref={{scope}}>=> {this.splitButton = scope}></SplitButtonComponent><br/> constructor(props: {}) { <br/> &#160; this.splitButton.toggle(); <br/> } |

| Show popup | **Method:** *show* <br/><br/> <EJ.SplitButton id="splitbutton" text="SplitButton" target="#target"></EJ.SplitButton><br/>var splitButton = \$('#splitbutton').ejSplitButton('instance');<br/> splitButton.show(); | **Method:** *toggle* <br/><br/> <SplitButtonComponent id="splitbutton" items={this.items} content="SplitButton" ref={{scope}}>=> {this.splitButton = scope}></SplitButtonComponent><br/> constructor(props: {}) { <br/> &#160; this.splitButton.toggle(); <br/> } |

## Events

| Behavior | API in Essential JS 1 | API in Essential JS 2 |

| --- | --- | --- |

| BeforeOpen event | **Event:** *beforeOpen* <br/><br/> <EJ.SplitButton id="splitbutton" text="SplitButton" target="#target" beforeOpen={beforeOpen}></EJ.SplitButton><br/>function beforeOpen(args) {<br/> &#160;&#160;&#160;&#160; / code block / <br/>} | **Event:** *beforeOpen* <br/><br/> <SplitButtonComponent id="splitbutton" items={this.items} content="SplitButton" beforeOpen={this.beforeOpen.bind(this)}></SplitButtonComponent><br/> beforeOpen(args) {<br/> &#160;&#160;&#160;&#160; / code block / <br/>} |

| Click event | **Event:** *click* <br/><br/> <EJ.SplitButton id="splitbutton" text="SplitButton" target="#target" click={click}></EJ.SplitButton> <br/>function click(args) {<br/> &#160;&#160;&#160;&#160; / code block / <br/>} | **Event:** *click* <br/><br/> <SplitButtonComponent id="splitbutton" items={this.items} content="SplitButton" click={this.click.bind(this)}></SplitButtonComponent><br/> click(args) {<br/> &#160;&#160;&#160;&#160; / code block / <br/>} |

| Close event | **Event:** *close* <br/><br/> <EJ.SplitButton id="splitbutton" text="SplitButton" target="#target" close={close}></EJ.SplitButton> <br/>function close(args) {<br/> &#160;&#160;&#160;&#160; / code block / <br/>} | **Event:** *close* <br/><br/> <SplitButtonComponent id="splitbutton" items={this.items} content="SplitButton" close={this.close.bind(this)}></SplitButtonComponent><br/> close(args) {<br/> &#160;&#160;&#160;&#160; / code block / <br/>} |

| Create event | **Event:** *create* <br/><br/> <EJ.SplitButton id="splitbutton" text="SplitButton" target="#target" create={create}></EJ.SplitButton> <br/>function create(args) {<br/> &#160;&#160;&#160;&#160; / code block / <br/>} | **Event:** *created* <br/><br/> <SplitButtonComponent id="splitbutton" items={this.items} content="SplitButton" created={this.created.bind(this)}></SplitButtonComponent><br/> created() {<br/> &#160;&#160;&#160;&#160; / code block / <br/>} |

| Destroy event | **Event:** *destroy* <br/><br/> <EJ.SplitButton id="splitbutton" text="SplitButton" target="#target" destroy={destroy}></EJ.SplitButton> <br/>function destroy(args) {<br/> &#160;&#160;&#160;&#160; / code block \*/ <br/>} | Not applicable |

| ItemMouseOut event | **Event:** *itemMouseOut* <br/><br/> <EJ.SplitButton id="splitbutton" text="SplitButton" target="#target" itemMouseOut={itemMouseOut}></EJ.SplitButton>  
<br/>function itemMouseOut(args) {<br/> &#160;&#160;&#160;&#160; / code block \*/ <br/>} | Not applicable |

| ItemMouseOver event | **Event:** *itemMouseOver* <br/><br/> <EJ.SplitButton id="splitbutton" text="SplitButton" target="#target" itemMouseOver={itemMouseOver}></EJ.SplitButton>  
<br/>function itemMouseOver(args) {<br/> &#160;&#160;&#160;&#160; / code block \*/ <br/>} | Not applicable |

| Item select event | **Event:** *itemSelected* <br/><br/> <EJ.SplitButton id="splitbutton" text="SplitButton" target="#target" itemSelected={itemSelected}></EJ.SplitButton>  
<br/>function itemSelected(args) {<br/> &#160;&#160;&#160;&#160; / code block / <br/>} | **Event:** *select* <br/><br/> <SplitButtonComponent id="splitbutton" items={this.items} content="SplitButton" select={this.select.bind(this)}></SplitButtonComponent> <br/> select(args) {<br/> &#160;&#160;&#160;&#160; / code block / <br/>} |

| Open event | **Event:** *open* <br/><br/> <EJ.SplitButton id="splitbutton" text="SplitButton" target="#target" open={open}></EJ.SplitButton> <br/>function open(args) {<br/> &#160;&#160;&#160;&#160; / code block / <br/>} | **Event:** *open* <br/><br/> <SplitButtonComponent id="splitbutton" items={this.items} content="SplitButton" open={this.open.bind(this)}></SplitButtonComponent> <br/>function open(args) {<br/> &#160;&#160;&#160;&#160; / code block / <br/>} |

| BeforeClose event | Not applicable | **Event:** *beforeClose* <br/><br/> <SplitButtonComponent id="splitbutton" items={this.items} content="SplitButton" beforeClose={this.beforeClose.bind(this)}></SplitButtonComponent> <br/> beforeClose(args) {<br/> &#160;&#160;&#160;&#160; / code block \*/ <br/>} |

| BeforeItemRender event | Not applicable | **Event:** *beforeItemRender* <br/><br/> <SplitButtonComponent id="splitbutton" items={this.items} content="SplitButton" beforeItemRender={this.beforeItemRender.bind(this)}></SplitButtonComponent> <br/> beforeItemRender(args) {<br/> &#160;&#160;&#160;&#160; / code block \*/ <br/>} |

{% endraw %}

## Splitter

### Getting Started

The following section explains the steps required to build the Splitter component with step-by-step procedure.

### Dependencies

The following list of dependencies required to use the Splitter component in your application:

```
`js
|-- @syncfusion/ej2-layouts
|-- @syncfusion/ej2-base
|-- @syncfusion/ej2-react-layouts
```

```
|-- @syncfusion/ej2-react-base  
,
```

### Installation and configuration

You can use [Create-react-app](#) to setup the applications.

To install `create-react-app` run the following command.

```
`bash  
npm install -g create-react-app  
,
```

Start a new project using create-react-app command as follows

```
<div class='tsx'>  
,
```

```
create-react-app quickstart --scripts-version=react-scripts-ts  
cd quickstart  
,
```

```
</div>  
<div class='jsx'>  
,
```

```
create-react-app quickstart  
cd quickstart  
,
```

```
</div>
```

Install the below required dependency package in order to use the `Splitter` component in your application.

```
npm install @syncfusion/ej2-react-layouts --save  
,
```

- The Splitter CSS files are available in the `ej2-layouts` package folder.

This can be referred in your application using the following code.

```
[src/App.css]
```

```
`css  
@import '../..../node_modules/@syncfusion/ej2-base/styles/material.css';  
@import '../..../node_modules/@syncfusion/ej2-layouts/styles/material.css';  
,
```

### Adding Splitter to the project

The Splitter can be initialized through `<SplitterComponent>` tag-directive with `<PanelsDirective>` and `<PaneDirective>` as child elements respectively.

Please refer the below code snippet,

```
`ts
import { PaneDirective, PanelsDirective, SplitterComponent } from '@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  public render() {
    return (
      <div className="App">
        <SplitterComponent id="splitter" height="250px" width="600px">
          <PanelsDirective>
            <PaneDirective/>
            <PaneDirective/>
            <PaneDirective/>
          </PanelsDirective>
        </SplitterComponent>
      </div>
    );
  }
}
export default App;
`
```

After completing the configurations to render the Splitter, use the following command to display the output in your default browser.

```
npm start
```

Output will be as follows:

### **APP.JSX**

```
import { PaneDirective, PanelsDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  render() {
```

```

    return (<div className="App">
      <SplitterComponent id="splitter" height="250px" width="600px">
        <PanelsDirective>
          <PanelDirective />
          <PanelDirective />
          <PanelDirective />
        </PanelsDirective>
      </SplitterComponent>
    </div>);
  }
}
export default App;

```

### APP.TSX

```

import { PanelDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  public render() {
    return (
      <div className="App">
        <SplitterComponent id="splitter" height="250px" width="600px">
          <PanelsDirective>
            <PanelDirective/>
            <PanelDirective/>
            <PanelDirective/>
          </PanelsDirective>
        </SplitterComponent>
      </div>
    );
  }
}
export default App;

```

### Orientation

Splitter supports both **Horizontal** and **Vertical** orientation for the panes. By default, it will be rendered in **Horizontal** orientation. You can change it by using [orientation](#) property.

### APP.JSX

```

import { PanelDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  render() {
    return (<div className="App">
      <SplitterComponent id="splitter-vertical" height="250px"
width="50%" orientation={ 'Vertical' }>
        <PanelsDirective>
          <PanelDirective />
          <PanelDirective />
        </PanelsDirective>
      </SplitterComponent>
    </div>);
  }
}

```

```

    }
  }
  export default App;

```

### APP.TSX

```

import { PaneDirective, PanesDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  public render() {
    return (<div className="App">
      <SplitterComponent id="splitter-vertical" height="250px"
width="50%" orientation = {'Vertical'}>
        <PanesDirective>
          <PaneDirective/>
          <PaneDirective/>
        </PanesDirective>
      </SplitterComponent>
    </div>
    );
  }
}
export default App;

```

### Load content to the pane

You can load the pane contents in either HTML element or string types using [content](#) property.

For detailed information, refer to the [Pane Content](#) section

### APP.JSX

```

import { PaneDirective, PanesDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  firstPane() {
    return (<div>
      <div className="content">
        <h3 className="h3">HTML</h3>
        <div className="code-preview">
          &#60;<span>!DOCTYPE html&#62;</span></div>
          <div>&#60;<span>html&#62;</span></div>
          <div>&#60;<span>body&#62;</span></div>
          &#60;<span>div</span> id="custom-image"&#62;
          <div>&#60;<span>img</span> src="src/albert.png"&#62;</div>
          <div>&#60;<span>div</span>&#62;</div>
          <div>&#60;<span>/body&#62;</span></div>
          <div>&#60;<span>/html&#62;</span></div>
        </div>
      </div>
    </div>);
  }
  secondPane() {
    const openBrace = '{';
    const closeBrace = '}';

```

```

        return (<div>
<div className="content">
  <h3 className="h3">CSS</h3>
  <div className="code-preview">
    <span>img {openBrace} </span>
    <div id="code-text">margin:<span>0 auto;</span></div>
    <div id="code-text">display:<span>flex;</span></div>
    <div id="code-text">height:<span>70px;</span></div>
    <span>{closeBrace}</span>
  </div>
</div>
</div>);
}
thirdPane() {
  const openBrace = '{';
  const closeBrace = '}';
  return (<div>
<div className="content">
  <h3 className="h3">JavaScript</h3>
  <div className="code-preview">
    <span>var</span> image = document.getElementById("custom-image");
    <div>image.addEventListener("click", function() {openBrace}</div>
      <div> // Code block for click action</div>
      // Code block for click action</div>
    <span> {closeBrace}) </span>
  </div>
</div>
</div>);
}
render() {
  return (<div className="App">
<SplitterComponent id="splitter" height="250px" width="550px">
<PanelsDirective>
<PanelDirective content={this.firstPane}/>
<PanelDirective content={this.secondPane}/>
<PanelDirective content={this.thirdPane}/>
</PanelsDirective>
</SplitterComponent>
</div>);
}
}
export default App;

```

## APP.TSX

```

import { PanelDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  public firstPane() {
    return (
      <div>
        <div className="content">
          <h3 className="h3">HTML</h3>
          <div className="code-preview">
            &#60;<span>!DOCTYPE html&#62;</span>

```

```

        <div>&#60;<span>html&#62;</span></div>
        <div>&#60;<span>body&#62;</span></div>
        &#60;<span>div</span> id="custom-image"&#62;
        <div>&#60;<span>img</span> src="src/albert.png"&#62;</div>
        <div>&#60;<span>div</span>&#62;</div>
        <div>&#60;<span>/body&#62;</span></div>
        <div>&#60;<span>/html&#62;</span></div>
    </div>
</div>
</div>
);
}
public secondPane() {
    const openBrace: string = '{';
    const closeBrace: string = '}';
    return (
        <div>
        <div className="content">
            <h3 className="h3">CSS</h3>
            <div className="code-preview">
                <span>img {openBrace} </span>
                <div id="code-text">margin:<span>0 auto;</span></div>
                <div id="code-text">display:<span>flex;</span></div>
                <div id="code-text">height:<span>70px;</span></div>
                <span>{closeBrace}</span>
            </div>
        </div>
        </div>
    );
}
public thirdPane() {
    const openBrace: string = '{';
    const closeBrace: string = '}';
    return (
        <div>
        <div className="content">
            <h3 className="h3">JavaScript</h3>
            <div className="code-preview">
                <span>var</span> image = document.getElementById("custom-image");
                <div>image.addEventListener("click", function() { openBrace }</div>
                <div>// Code block for click action</div>
                <span> {closeBrace }</span>
            </div>
        </div>
        </div>
    );
}
}
public render() {
    return <div className="App">
        <SplitterComponent id="splitter" height="250px" width="550px">
        <PanelsDirective>
        <PanelDirective content = {this.firstPane}/>
        <PanelDirective content = {this.secondPane}/>
        <PanelDirective content = {this.thirdPane}/>
        </PanelsDirective>
        </SplitterComponent>
    </div>);
}

```



```
}  
}  
export default App;
```

### See Also

- [Construct different layouts using Splitter](#)

### Getting Started

The following section explains the steps required to build the Splitter component with step-by-step procedure.

### Dependencies

The following list of dependencies required to use the Splitter component in your application:

```
`js  
|-- @syncfusion/ej2-layouts  
|-- @syncfusion/ej2-base  
|-- @syncfusion/ej2-react-layouts  
|-- @syncfusion/ej2-react-base  
,
```

### Installation and configuration

You can use [Create-react-app](#) to setup the applications.

To install `create-react-app` run the following command.

```
,  
  
npm install -g create-react-app  
,
```

Start a new project using create-react-app command as follows

```
<div class='tsx'>  
,  
  
create-react-app quickstart --scripts-version=react-scripts-ts  
cd quickstart  
,  
  
</div>  
<div class='jsx'>  
,  
  
create-react-app quickstart  
cd quickstart
```

```
`
```

```
</div>
```

Install the below required dependency package in order to use the `Splitter` component in your application.

```
`bash
```

```
npm install @syncfusion/ej2-react-layouts --save
```

```
`
```

- The Splitter CSS files are available in the `ej2-layouts` package folder.

This can be referred in your application using the following code.

```
[src/App.css]
```

```
`css
```

```
@import '../node_modules/@syncfusion/ej2-base/styles/material.css';
```

```
@import '../node_modules/@syncfusion/ej2-layouts/styles/material.css';
```

```
`
```

### [Adding Splitter to the project](#)

The Splitter can be initialized through `<SplitterComponent>` tag-directive with `<PanesDirective>` and `<PaneDirective>` as child elements respectively.

Please refer the below code snippet,

```
`ts
```

```
import { PaneDirective, PanesDirective, SplitterComponent } from '@syncfusion/ej2-react-layouts';
```

```
import * as React from "react";
```

```
function App() {
```

```
  return (
```

```
    <div className="App">
```

```
      <SplitterComponent id="splitter" height="250px" width="600px">
```

```
        <PanesDirective>
```

```
          <PaneDirective/>
```

```
          <PaneDirective/>
```

```
          <PaneDirective/>
```

```
        </PanesDirective>
```

```
      </SplitterComponent>
```

```
    </div>
```

```
  );
```

```
}  
export default App;  
`
```

After completing the configurations to render the Splitter, use the following command to display the output in your default browser.

```
`  
npm start  
`
```

Output will be as follows:

#### **APP.JSX**

```
import { PaneDirective, PanesDirective, SplitterComponent } from  
'@syncfusion/ej2-react-layouts';  
import * as React from "react";  
function App() {  
    return (<div className="App">  
        <SplitterComponent id="splitter" height="250px" width="600px">  
            <PanesDirective>  
                <PaneDirective />  
                <PaneDirective />  
                <PaneDirective />  
            </PanesDirective>  
        </SplitterComponent>  
    </div>);  
}  
export default App;
```

#### **APP.TSX**

```
import { PaneDirective, PanesDirective, SplitterComponent } from  
'@syncfusion/ej2-react-layouts';  
import * as React from "react";  
function App() {  
    return (  
        <div className="App">  
            <SplitterComponent id="splitter" height="250px" width="600px">  
                <PanesDirective>  
                    <PaneDirective/>  
                    <PaneDirective/>  
                    <PaneDirective/>  
                </PanesDirective>  
            </SplitterComponent>  
        </div>  
    );  
}  
export default App;
```

### Orientation

Splitter supports both **Horizontal** and **Vertical** orientation for the panes. By default, it will be rendered in **Horizontal** orientation. You can change it by using [orientation](#) property.

#### APP.JSX

```
import { PaneDirective, PanesDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
function App() {
    return (<div className="App">
        <SplitterComponent id="splitter-vertical" height="250px" width="50%"
orientation={ 'Vertical' }>
            <PanesDirective>
                <PaneDirective />
                <PaneDirective />
            </PanesDirective>
        </SplitterComponent>
    </div>);
}
export default App;
```

#### APP.TSX

```
import { PaneDirective, PanesDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
function App() {
    return (
        <div className="App">
            <SplitterComponent id="splitter-vertical" height="250px" width="50%"
orientation = { 'Vertical' }>
                <PanesDirective>
                    <PaneDirective/>
                    <PaneDirective/>
                </PanesDirective>
            </SplitterComponent>
        </div>
    );
}
export default App;
```

### Load content to the pane

You can load the pane contents in either HTML element or string types using [content](#) property.

For detailed information, refer to the [Pane Content](#) section

#### APP.JSX

```
import { PaneDirective, PanesDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
import { useState } from 'react';
function App() {
    const firstPaneContent = `<div>
        <div className="content">
```

```

    <h3 className="h3">HTML</h3>
    <div className="code-preview">
      &#60;<span>!DOCTYPE html&#62;</span>
      <div>&#60;<span>html&#62;</span></div>
      <div>&#60;<span>body&#62;</span></div>
      &#60;<span>div</span> id="custom-image"&#62;
      <div>&#60;<span>img</span> src="src/albert.png"&#62;</div>
      <div>&#60;<span>div</span>&#62;</div>
      <div>&#60;<span>/body&#62;</span></div>
      <div>&#60;<span>/html&#62;</span></div>
    </div>
  </div>
  </div>`;
  const secondPaneContent = `<div>
    <div className="content">
      <h3 className="h3">CSS</h3>
      <div className="code-preview">
        <span>img { </span>
        <div id="code-text">margin:<span>0 auto;</span></div>
        <div id="code-text">display:<span>flex;</span></div>
        <div id="code-text">height:<span>70px;</span></div>
        <span> }</span>
      </div>
    </div>
  </div>`;
  const thirdPaneContent = `<div>
    <div className="content">
      <h3 className="h3">JavaScript</h3>
      <div className="code-preview">
        <span>var</span> image = document.getElementById("custom-image");
        <div>image.addEventListener("click", function() { </div>
          <div>// Code block for click action</div>
        <span> }); </span>
      </div>
    </div>
  </div>`;
  const [panelContent] = useState(firstPaneContent);
  const [pane2Content] = useState(secondPaneContent);
  const [pane3Content] = useState(thirdPaneContent);
  return (<div className="App">
    <SplitterComponent id="splitter" height="250px" width="550px">
      <PanelsDirective>
        <PaneDirective content={panelContent}/>
        <PaneDirective content={pane2Content}/>
        <PaneDirective content={pane3Content}/>
      </PanelsDirective>
    </SplitterComponent>
  </div>);
}
export default App;

```

## APP.TSX

```

import { PaneDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";

```

```

import { useState } from 'react';
function App() {
  const firstPaneContent = `<div>
    <div className="content">
      <h3 className="h3">HTML</h3>
      <div className="code-preview">
        &#60;<span>!DOCTYPE html&#62;</span>
        <div>&#60;<span>html&#62;</span></div>
        <div>&#60;<span>body&#62;</span></div>
        &#60;<span>div</span> id="custom-image"&#62;
        <div>&#60;<span>img</span> src="src/albert.png"&#62;</div>
        <div>&#60;<span>div</span>&#62;</div>
        <div>&#60;<span>/body&#62;</span></div>
        <div>&#60;<span>/html&#62;</span></div>
      </div>
    </div>
  </div>`;
  const secondPaneContent = `<div>
    <div className="content">
      <h3 className="h3">CSS</h3>
      <div className="code-preview">
        <span>img { </span>
        <div id="code-text">margin:<span>0 auto;</span></div>
        <div id="code-text">display:<span>flex;</span></div>
        <div id="code-text">height:<span>70px;</span></div>
        <span> }</span>
      </div>
    </div>
  </div>`;
  const thirdPaneContent = `<div>
    <div className="content">
      <h3 className="h3">JavaScript</h3>
      <div className="code-preview">
        <span>var</span> image = document.getElementById("custom-image");
        <div>image.addEventListener("click", function() { </div>
          <div>// Code block for click action</div>
        <span> }); </span>
      </div>
    </div>
  </div>`;
  const [panelContent] = useState(firstPaneContent);
  const [pane2Content] = useState(secondPaneContent);
  const [pane3Content] = useState(thirdPaneContent);
  return (
    <div className="App">
      <SplitterComponent id="splitter" height="250px" width="550px">
        <PanelsDirective>
          <PanelDirective content={panelContent} />
          <PanelDirective content={pane2Content} />
          <PanelDirective content={pane3Content} />
        </PanelsDirective>
      </SplitterComponent>
    </div>
  );
}
export default App;

```

See Also

- [Construct different layouts using Splitter](#)

## Split panes in React Splitter component

This section explain about split panes behaviors.

### Horizontal layout

By default, splitter will render in horizontal orientation. Splitter container will be split as panes in horizontal flow direction with vertical separator.

[Class-component]

#### APP.JSX

```
import { PaneDirective, PanesDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  leftContent() {
    return (<div>
      <div className="content">
        <h3>Grid </h3>
        The ASP.NET DataGrid control, or DataTable is a feature-rich
        control used to display data in a tabular format.
      </div>
    </div>);
  }
  middleContent() {
    return (<div>
      <div className="content">
        <h3>Schedule </h3>
        The ASP.NET Scheduler, a.k.a. event calendar, facilitates almost
        all calendar features, thus allowing users to manage their time efficiently.
      </div>
    </div>);
  }
  rightContent() {
    return (<div>
      <div className="content">
        <h3>Chart </h3>
        ASP.NET charts, a well-crafted easy-to-use charting package, is
        used to add beautiful charts in web and mobile applications.
      </div>
    </div>);
  }
  render() {
    return (<div className="App">
      <SplitterComponent id="horizontal" height="250px" width='600px'>
        <PanesDirective>
          <PaneDirective size='200px' content={this.leftContent}/>
          <PaneDirective size='200px' content={this.middleContent}/>
          <PaneDirective size='200px' content={this.rightContent}/>
        </PanesDirective>
      </SplitterComponent>
    </div>);
  }
}
```

```

        </SplitterComponent>
    </div>);
    }
}
export default App;

```

### APP.TSX

```

import { PaneDirective, PanesDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
    public leftContent(): JSX.Element {
        return (
            <div>
                <div className="content">
                    <h3>Grid </h3>
                    The ASP.NET DataGrid control, or DataTable is a feature-rich
                    control used to display data in a tabular format.
                </div>
            </div>
        );
    }
    public middleContent(): JSX.Element {
        return (
            <div>
                <div className="content">
                    <h3>Schedule </h3>
                    The ASP.NET Scheduler, a.k.a. event calendar, facilitates almost
                    all calendar features, thus allowing users to manage their time efficiently.
                </div>
            </div>
        );
    }
    public rightContent(): JSX.Element {
        return (
            <div>
                <div className="content">
                    <h3>Chart </h3>
                    ASP.NET charts, a well-crafted easy-to-use charting package, is
                    used to add beautiful charts in web and mobile applications.
                </div>
            </div>
        );
    }
    public render() {
        return (<div className="App">
            <SplitterComponent id="horizontal" height="250px" width='600px'>
                <PanesDirective>
                    <PaneDirective size='200px' content = {this.leftContent}/>
                    <PaneDirective size='200px' content = {this.middleContent}/>
                    <PaneDirective size='200px' content = {this.rightContent}/>
                </PanesDirective>
            </SplitterComponent>
        </div>);
    }
}

```



```
}
export default App;
```

## [Functional-component]

### APP.JSX

```
import { PaneDirective, PanesDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
function App() {
  function leftContent() {
    return (<div>
      <div className="content">
        <h3>Grid </h3>
        The ASP.NET DataGrid control, or DataTable is a feature-rich
        control used to display data in a tabular format.
      </div>
    </div>);
  }
  function middleContent() {
    return (<div>
      <div className="content">
        <h3>Schedule </h3>
        The ASP.NET Scheduler, a.k.a. event calendar, facilitates almost
        all calendar features, thus allowing users to manage their time efficiently.
      </div>
    </div>);
  }
  function rightContent() {
    return (<div>
      <div className="content">
        <h3>Chart </h3>
        ASP.NET charts, a well-crafted easy-to-use charting package, is
        used to add beautiful charts in web and mobile applications.
      </div>
    </div>);
  }
  return (<div className="App">
    <SplitterComponent id="horizontal" height="250px" width='600px'>
      <PanesDirective>
        <PaneDirective size='200px' content={leftContent}/>
        <PaneDirective size='200px' content={middleContent}/>
        <PaneDirective size='200px' content={rightContent}/>
      </PanesDirective>
    </SplitterComponent>
  </div>);
}
export default App;
```

### APP.TSX

```
import { PaneDirective, PanesDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
function App () {
```

```

function leftContent(): JSX.Element {
  return (
    <div>
      <div className="content">
        <h3>Grid </h3>
        The ASP.NET DataGrid control, or DataTable is a feature-rich
        control used to display data in a tabular format.
      </div>
    </div>
  );
}
function middleContent(): JSX.Element {
  return (
    <div>
      <div className="content">
        <h3>Schedule </h3>
        The ASP.NET Scheduler, a.k.a. event calendar, facilitates almost
        all calendar features, thus allowing users to manage their time efficiently.
      </div>
    </div>
  );
}
function rightContent(): JSX.Element {
  return (
    <div>
      <div className="content">
        <h3>Chart </h3>
        ASP.NET charts, a well-crafted easy-to-use charting package, is
        used to add beautiful charts in web and mobile applications.
      </div>
    </div>
  );
}
return (<div className="App">
  <SplitterComponent id="horizontal" height="250px" width='600px'>
    <PanelsDirective>
      <PanelDirective size='200px' content = {leftContent}/>
      <PanelDirective size='200px' content = {middleContent}/>
      <PanelDirective size='200px' content = {rightContent}/>
    </PanelsDirective>
  </SplitterComponent>
</div>);
}
export default App;

```

### Vertical layout

By setting [orientation](#) property as **Vertical**, splitter will render in vertical orientation. Splitter container will be split as panes in vertical flow direction with horizontal separator.

### [Class-component]

#### APP.JSX

```

import { PanelDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";

```

```

class App extends React.Component {
  leftContent() {
    return (<div>
      <div className="content">
        <h3>Grid </h3>
        The ASP.NET DataGrid control, or DataTable is a feature-rich
        control used to display data in a tabular format.
      </div>
    </div>);
  }
  middleContent() {
    return (<div>
      <div className="content">
        <h3>Schedule </h3>
        The ASP.NET Scheduler, a.k.a. event calendar, facilitates almost
        all calendar features, thus allowing users to manage their time efficiently.
      </div>
    </div>);
  }
  rightContent() {
    return (<div>
      <div className="content">
        <h3>Chart </h3>
        ASP.NET charts, a well-crafted easy-to-use charting package, is
        used to add beautiful charts in web and mobile applications.
      </div>
    </div>);
  }
  render() {
    return (<div className="App">
      <SplitterComponent id="vertical" height="300px" width='600px'
        orientation='Vertical'>
        <PanelsDirective>
          <PanelDirective size='100px' content={this.leftContent}/>
          <PanelDirective size='100px' content={this.middleContent}/>
          <PanelDirective size='100px' content={this.rightContent}/>
        </PanelsDirective>
      </SplitterComponent>
    </div>);
  }
}
export default App;

```

### APP.TSX

```

import { PanelDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  public leftContent(): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>Grid </h3>
          The ASP.NET DataGrid control, or DataTable is a feature-rich
          control used to display data in a tabular format.

```

```

        </div>
      </div>
    );
  }
  public middleContent(): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>Schedule </h3>
          The ASP.NET Scheduler, a.k.a. event calendar, facilitates almost
          all calendar features, thus allowing users to manage their time efficiently.
        </div>
      </div>
    );
  }
  public rightContent(): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>Chart </h3>
          ASP.NET charts, a well-crafted easy-to-use charting package, is
          used to add beautiful charts in web and mobile applications.
        </div>
      </div>
    );
  }
  public render() {
    return (<div className="App">
      <SplitterComponent id="vertical" height="300px" width='600px'
        orientation='Vertical'>
        <PanelsDirective>
          <PanelDirective size='100px' content = {this.leftContent}/>
          <PanelDirective size='100px' content = {this.middleContent}/>
          <PanelDirective size='100px' content = {this.rightContent}/>
        </PanelsDirective>
      </SplitterComponent>
    </div>);
  }
}
export default App;

```

## [Functional-component]

### APP.JSX

```

import { PanelDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
function App() {
  function leftContent() {
    return (<div>
      <div className="content">
        <h3>Grid </h3>
        The ASP.NET DataGrid control, or DataTable is a feature-rich
        control used to display data in a tabular format.
      </div>
    </div>
  )
}

```

```

        </div>);
    }
    function middleContent() {
        return (<div>
            <div className="content">
                <h3>Schedule </h3>
                The ASP.NET Scheduler, a.k.a. event calendar, facilitates almost
                all calendar features, thus allowing users to manage their time efficiently.
            </div>
        </div>);
    }
    function rightContent() {
        return (<div>
            <div className="content">
                <h3>Chart </h3>
                ASP.NET charts, a well-crafted easy-to-use charting package, is
                used to add beautiful charts in web and mobile applications.
            </div>
        </div>);
    }
    return (<div className="App">
        <SplitterComponent id="vertical" height="300px" width='600px'
        orientation='Vertical'>
            <PanelsDirective>
                <PanelDirective size='100px' content={leftContent}/>
                <PanelDirective size='100px' content={middleContent}/>
                <PanelDirective size='100px' content={rightContent}/>
            </PanelsDirective>
        </SplitterComponent>
    </div>);
}
export default App;

```

## APP.TSX

```

import { PanelDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
function App () {
    function leftContent(): JSX.Element {
        return (
            <div>
                <div className="content">
                    <h3>Grid </h3>
                    The ASP.NET DataGrid control, or DataTable is a feature-rich
                    control used to display data in a tabular format.
                </div>
            </div>
        );
    }
    function middleContent(): JSX.Element {
        return (
            <div>
                <div className="content">
                    <h3>Schedule </h3>

```

```

        The ASP.NET Scheduler, a.k.a. event calendar, facilitates almost
        all calendar features, thus allowing users to manage their time efficiently.
    </div>
</div>
);
}
function rightContent(): JSX.Element {
    return (
        <div>
            <div className="content">
                <h3>Chart </h3>
                ASP.NET charts, a well-crafted easy-to-use charting package, is
                used to add beautiful charts in web and mobile applications.
            </div>
        </div>
    );
}
return (<div className="App">
    <SplitterComponent id="vertical" height="300px" width='600px'
orientation='Vertical'>
    <PanelsDirective>
        <PanelDirective size='100px' content = {leftContent}/>
        <PanelDirective size='100px' content = {middleContent}/>
        <PanelDirective size='100px' content = {rightContent}/>
    </PanelsDirective>
    </SplitterComponent>
</div>);
}
export default App;

```

## Multiple panes

You can render the multiple panes with both **Horizontal** and **Vertical** orientations.

### [Class-component]

#### APP.JSX

```

import { PanelDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
    panelContent() {
        return (<div>
            <div className="content">
                <h3>PARIS </h3>
                Paris, the city of lights and love - this short guide is full of
                ideas for how to make the most of the romanticism...
            </div>
        </div>);
    }
    pane2Content() {
        return (<div>
            <div className="content">
                <h3>CAMEMBERT </h3>
                The village in the Orne département of Normandy where the famous
                French cheese is originated from.
            </div>
        </div>);
    }
}

```

```

        </div>
      </div>);
    }
    pane3Content() {
      return <div>
        <div className="content">
          <h3>GRENOBLE </h3>
          The capital city of the French Alps and a major scientific center
          surrounded by many ski resorts, host of the Winter Olympics in 1968.
        </div>
      </div>);
    }
    pane4Content() {
      return <div>
        <div className="content">
          <h3>AUSTRALIA </h3>
          Australia is a country and continent surrounded by the Indian and
          Pacific oceans. Its major cities - Sydney, Brisbane, Melbourne, Perth,
          Adelaide - are coastal. Its capital, Canberra, is inland.
        </div>
      </div>);
    }
    render() {
      return <div className="App">
        <SplitterComponent id="multiple" height="250px" width='600px'>
          <PanelsDirective>
            <PanelDirective size='150px' content={this.panelContent}/>
            <PanelDirective size='150px' content={this.pane2Content}/>
            <PanelDirective size='150px' content={this.pane3Content}/>
            <PanelDirective size='150px' content={this.pane4Content}/>
          </PanelsDirective>
        </SplitterComponent>
      </div>);
    }
  }
  export default App;

```

### APP.TSX

```

import { PanelDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  public panelContent(): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>PARIS </h3>
          Paris, the city of lights and love - this short guide is full of
          ideas for how to make the most of the romanticism...
        </div>
      </div>
    );
  }
  public pane2Content(): JSX.Element {
    return (

```

```

        <div>
            <div className="content">
                <h3>CAMEMBERT </h3>
                The village in the Orne département of Normandy where the famous
                French cheese is originated from.
            </div>
        </div>
    );
}
public pane3Content(): JSX.Element {
    return (
        <div>
            <div className="content">
                <h3>GRENOBLE </h3>
                The capital city of the French Alps and a major scientific center
                surrounded by many ski resorts, host of the Winter Olympics in 1968.
            </div>
        </div>
    );
}
public pane4Content(): JSX.Element {
    return (
        <div>
            <div className="content">
                <h3>AUSTRALIA </h3>
                Australia is a country and continent surrounded by the Indian and
                Pacific oceans. Its major cities - Sydney, Brisbane, Melbourne, Perth,
                Adelaide - are coastal. Its capital, Canberra, is inland.
            </div>
        </div>
    );
}
public render() {
    return (<div className="App">
        <SplitterComponent id="multiple" height="250px" width='600px'>
            <PanelsDirective>
                <PanelDirective size='150px' content = {this.panelContent}/>
                <PanelDirective size='150px' content = {this.pane2Content}/>
                <PanelDirective size='150px' content = {this.pane3Content}/>
                <PanelDirective size='150px' content = {this.pane4Content}/>
            </PanelsDirective>
        </SplitterComponent>
    </div>);
}
}
export default App;

```

### [Functional-component]

#### APP.JSX

```

import { PanelDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
function App() {
    function panelContent() {

```



```

    return (<div>
      <div className="content">
        <h3>PARIS </h3>
        Paris, the city of lights and love - this short guide is full of
        ideas for how to make the most of the romanticism...
      </div>
    </div>);
  }
  function pane2Content() {
    return (<div>
      <div className="content">
        <h3>CAMEMBERT </h3>
        The village in the Orne département of Normandy where the famous
        French cheese is originated from.
      </div>
    </div>);
  }
  function pane3Content() {
    return (<div>
      <div className="content">
        <h3>GRENOBLE </h3>
        The capital city of the French Alps and a major scientific center
        surrounded by many ski resorts, host of the Winter Olympics in 1968.
      </div>
    </div>);
  }
  function pane4Content() {
    return (<div>
      <div className="content">
        <h3>AUSTRALIA </h3>
        Australia is a country and continent surrounded by the Indian and
        Pacific oceans. Its major cities - Sydney, Brisbane, Melbourne, Perth,
        Adelaide - are coastal. Its capital, Canberra, is inland.
      </div>
    </div>);
  }
  return (<div className="App">
    <SplitterComponent id="multiple" height="250px" width='600px'>
      <PanelsDirective>
        <PanelDirective size='150px' content={panelContent}/>
        <PanelDirective size='150px' content={pane2Content}/>
        <PanelDirective size='150px' content={pane3Content}/>
        <PanelDirective size='150px' content={pane4Content}/>
      </PanelsDirective>
    </SplitterComponent>
  </div>);
}
export default App;

```

## APP.TSX

```

import { PanelDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
function App () {
  function panelContent(): JSX.Element {

```

```

    return (
      <div>
        <div className="content">
          <h3>PARIS </h3>
          Paris, the city of lights and love - this short guide is full of
          ideas for how to make the most of the romanticism...
        </div>
      </div>
    );
  }
  function pane2Content(): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>CAMEMBERT </h3>
          The village in the Orne département of Normandy where the famous
          French cheese is originated from.
        </div>
      </div>
    );
  }
  function pane3Content(): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>GRENOBLE </h3>
          The capital city of the French Alps and a major scientific center
          surrounded by many ski resorts, host of the Winter Olympics in 1968.
        </div>
      </div>
    );
  }
  function pane4Content(): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>AUSTRALIA </h3>
          Australia is a country and continent surrounded by the Indian and
          Pacific oceans. Its major cities - Sydney, Brisbane, Melbourne, Perth,
          Adelaide - are coastal. Its capital, Canberra, is inland.
        </div>
      </div>
    );
  }
  return (<div className="App">
    <SplitterComponent id="multiple" height="250px" width='600px'>
      <PanelsDirective>
        <PaneDirective size='150px' content = {panelContent}/>
        <PaneDirective size='150px' content = {pane2Content}/>
        <PaneDirective size='150px' content = {pane3Content}/>
        <PaneDirective size='150px' content = {pane4Content}/>
      </PanelsDirective>
    </SplitterComponent>
  </div>);
}
export default App;

```

### Separator

By default, pane separator will be render with **1px** width/height. You can customize the separator size by using [separatorSize](#) property.

For Horizontal orientation, it will be considered as separator width.

For Vertical orientation, it will be considered as separator height.

### [Class-component]

#### APP.JSX

```
import { PaneDirective, PanesDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  leftContent() {
    return (<div>
      <div className="content">
        <h3>Grid </h3>
        The ASP.NET DataGrid control, or DataTable is a feature-rich
        control used to display data in a tabular format.
      </div>
    </div>);
  }
  middleContent() {
    return (<div>
      <div className="content">
        <h3>Schedule </h3>
        The ASP.NET Scheduler, a.k.a. event calendar, facilitates almost
        all calendar features, thus allowing users to manage their time efficiently.
      </div>
    </div>);
  }
  rightContent() {
    return (<div>
      <div className="content">
        <h3>Chart </h3>
        ASP.NET charts, a well-crafted easy-to-use charting package, is
        used to add beautiful charts in web and mobile applications.
      </div>
    </div>);
  }
  render() {
    return (<div className="App">
      <SplitterComponent id="separator" height="250px" width='600px'
        separatorSize={5}>
        <PanesDirective>
          <PaneDirective size='200px' content={this.leftContent}/>
          <PaneDirective size='200px' content={this.middleContent}/>
          <PaneDirective size='200px' content={this.rightContent}/>
        </PanesDirective>
      </SplitterComponent>
    </div>);
  }
}
```

```
export default App;
```

### APP.TSX

```
import { PaneDirective, PanesDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  public leftContent(): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>Grid </h3>
          The ASP.NET DataGrid control, or DataTable is a feature-rich
          control used to display data in a tabular format.
        </div>
      </div>
    );
  }
  public middleContent(): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>Schedule </h3>
          The ASP.NET Scheduler, a.k.a. event calendar, facilitates almost
          all calendar features, thus allowing users to manage their time efficiently.
        </div>
      </div>
    );
  }
  public rightContent(): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>Chart </h3>
          ASP.NET charts, a well-crafted easy-to-use charting package, is
          used to add beautiful charts in web and mobile applications.
        </div>
      </div>
    );
  }
  public render() {
    return (<div className="App">
      <SplitterComponent id="separator" height="250px" width='600px'
        separatorSize={5}>
        <PanesDirective>
          <PaneDirective size='200px' content = {this.leftContent as any}/>
          <PaneDirective size='200px' content = {this.middleContent as any}/>
          <PaneDirective size='200px' content = {this.rightContent as any}/>
        </PanesDirective>
      </SplitterComponent>
    </div>);
  }
}
export default App;
```

**[Functional-component]****APP.JSX**

```

import { PaneDirective, PanesDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
function App() {
    function leftContent() {
        return (<div>
            <div className="content">
                <h3>Grid </h3>
                The ASP.NET DataGrid control, or DataTable is a feature-rich
control used to display data in a tabular format.
            </div>
        </div>);
    }
    function middleContent() {
        return (<div>
            <div className="content">
                <h3>Schedule </h3>
                The ASP.NET Scheduler, a.k.a. event calendar, facilitates almost
all calendar features, thus allowing users to manage their time efficiently.
            </div>
        </div>);
    }
    function rightContent() {
        return (<div>
            <div className="content">
                <h3>Chart </h3>
                ASP.NET charts, a well-crafted easy-to-use charting package, is
used to add beautiful charts in web and mobile applications.
            </div>
        </div>);
    }
    return (<div className="App">
        <SplitterComponent id="separator" height="250px" width='600px'
separatorSize={5}>
            <PanelsDirective>
                <PaneDirective size='200px' content={leftContent}/>
                <PaneDirective size='200px' content={middleContent}/>
                <PaneDirective size='200px' content={rightContent}/>
            </PanelsDirective>
        </SplitterComponent>
    </div>);
}
export default App;

```

**APP.TSX**

```

import { PaneDirective, PanesDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
function App () {
    function leftContent(): JSX.Element {
        return (
            <div>

```

```

        <div className="content">
            <h3>Grid </h3>
            The ASP.NET DataGrid control, or DataTable is a feature-rich
            control used to display data in a tabular format.
        </div>
    </div>
    );
}
function middleContent(): JSX.Element {
    return (
        <div>
            <div className="content">
                <h3>Schedule </h3>
                The ASP.NET Scheduler, a.k.a. event calendar, facilitates almost
                all calendar features, thus allowing users to manage their time efficiently.
            </div>
        </div>
    );
}
function rightContent(): JSX.Element {
    return (
        <div>
            <div className="content">
                <h3>Chart </h3>
                ASP.NET charts, a well-crafted easy-to-use charting package, is
                used to add beautiful charts in web and mobile applications.
            </div>
        </div>
    );
}
return (
    <div className="App">
        <SplitterComponent id="separator" height="250px" width='600px'
        separatorSize={5}>
            <PanelsDirective>
                <PanelDirective size='200px' content = {leftContent as any}/>
                <PanelDirective size='200px' content = {middleContent as any}/>
                <PanelDirective size='200px' content = {rightContent as any}/>
            </PanelsDirective>
        </SplitterComponent>
    </div>
    );
}
export default App;

```

### Nested Splitter

Splitter provides support to render the nested pane to achieve the complex layouts. You can use the same `<div>` element for splitter pane and nested splitter.

Also you can render the nested splitter using direct child of the splitter pane. For this, nested splitter should have `100%` width and height to match with the parent pane dimensions.

[Class-component]

### APP.JSX

```
{% raw %}
import { PaneDirective, PanesDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
    innerSplitterInstance;
    splitterInstance;
    leftPaneContent() {
        return (<div>
            <div className="content">
                <a href="https://www.syncfusion.com/ebooks/neuralnetworks"
target="_blank">Neural Networks Using C# Succinctly</a>
                <p>Neural networks are an exciting field of software
development used to calculate outputs from input data.
                While the idea seems simple enough, the implications of such
networks are staggering—think optical character recognition,
                speech recognition, and regression analysis. With Neural
Networks Using C# Succinctly by James McCaffrey, you'll learn
                how to create your own neural network to solve classification
problems, or problems where the outcomes can only be one of
                several values. <br /><br />Learn about encoding and
normalizing data, activation functions and how to choose the right one, and
ultimately
                how to train a neural network to find weights and bias values
that provide accurate predictions.
                An artificial neural network (sometimes abbreviated ANN, or
shortened to just "neural network" when the context is clear) is
                a software system that loosely models biological neurons and
synapses. Before explaining exactly how neural networks work, it is
                useful to understand what types of problems they can solve.
            </p>
            </div>
        </div>);
    }
    rightPaneContent1() {
        return (<div>
            <div className="content">
                <a
href="https://www.syncfusion.com/ebooks/data_capture_and_extraction_with_c_s
harp_succinctly" target="_blank"> Data Capture and Extraction with C#
Succinctly </a>
                <p>Capturing and extracting information is one of the most important
tasks a developer can perform, and making this task more
                engaging without relying entirely on specialized tools is an
efficient way to improve productivity.
                In Data Capture and Extraction with C# Succinctly, author Ed Freitas
guides readers toward getting more out of C# in minimal time.
                Email has become a pillar of our modern and connected society, and
it now serves as a primary means of communication. Because each email
                is filled with valuable information, data extraction has emerged as
a worthwhile skill set for developers in today's business world.
            </p>
            </div>
        </div>);
    }
    rightPaneContent2() {
        return (<div>
```

```

        <div className="content">
            <a href="https://www.syncfusion.com/ebooks/spark"
target="_blank">Spark Succinctly</a>
            <p>Mastering big data requires an aptitude at every step of
information processing.
                Post-processing, one of the most important steps, is where you
find Apache Spark frequently employed.
                Spark Succinctly, by Marko Svaljek, addresses Spark's use in
the ultimate step in handling big data. This e-book, the
                third installment in Svaljek's IoT series, teaches the basics
of using Spark and explores how to work with RDDs, Scala and
                Python tasks, JSON files, and Cassandra. Many of the leading
companies in the world today face the problem of big data.
            </p>
        </div>
    </div>;
}
onCreate() {
    // Initialize Splitter component
    const cont =
this.innerSplitterInstance.element.querySelector(".e-pane")[1];
    cont.appendChild(this.splitterInstance.element);
}
render() {
    return <div id="target" className="control-section splitter-
expand">
        <SplitterComponent id="splitter1" separatorSize={3} height="350px"
width="100%" ref={(splitter) => { this.innerSplitterInstance = splitter; }}>
            <PanelsDirective>
                <PaneDirective size='48%' content={this.leftPaneContent}/>
                <PaneDirective />
            </PanelsDirective>
        </SplitterComponent>
        <SplitterComponent id="splitter2" separatorSize={3}
orientation="Vertical" ref={(splitter) => { this.splitterInstance =
splitter; }} created={this.onCreate = this.onCreate.bind(this)}>
            <PanelsDirective>
                <PaneDirective size='50%' content={this.rightPaneContent1}/>
                <PaneDirective content={this.rightPaneContent2}/>
            </PanelsDirective>
        </SplitterComponent>
    </div>;
}
}
export default App;
{% endraw %}

```

### APP.TSX

```

{% raw %}
import { PaneDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
    private innerSplitterInstance: any;
    private splitterInstance: any;

```



```

public leftPaneContent(): JSX.Element {
    return (
        <div>
            <div className="content">
                <a href="https://www.syncfusion.com/ebooks/neuralnetworks"
target="_blank">Neural Networks Using C# Succinctly</a>
                <p>Neural networks are an exciting field of software
development used to calculate outputs from input data.
                While the idea seems simple enough, the implications of such
networks are staggering—think optical character recognition,
                speech recognition, and regression analysis. With Neural
Networks Using C# Succinctly by James McCaffrey, you'll learn
                how to create your own neural network to solve classification
problems, or problems where the outcomes can only be one of
                several values. <br/><br/>Learn about encoding and normalizing
data, activation functions and how to choose the right one, and ultimately
                how to train a neural network to find weights and bias values
that provide accurate predictions.
                An artificial neural network (sometimes abbreviated ANN, or
shortened to just "neural network" when the context is clear) is
                a software system that loosely models biological neurons and
synapses. Before explaining exactly how neural networks work, it is
                useful to understand what types of problems they can solve.
            </p>
            </div>
        </div>
    );
}

public rightPaneContent1(): JSX.Element {
    return (
        <div>
            <div className="content">
                <a href =
"https://www.syncfusion.com/ebooks/data_capture_and_extraction_with_c_sharp_
succinctly" target = "_blank" > Data Capture and Extraction with C#
Succinctly </a>
                <p>Capturing and extracting information is one of the most important
tasks a developer can perform, and making this task more
                engaging without relying entirely on specialized tools is an
efficient way to improve productivity.
                In Data Capture and Extraction with C# Succinctly, author Ed Freitas
guides readers toward getting more out of C# in minimal time.
                Email has become a pillar of our modern and connected society, and
it now serves as a primary means of communication. Because each email
                is filled with valuable information, data extraction has emerged as
a worthwhile skill set for developers in today's business world.
            </p>
            </div >
        </div>
    );
}

public rightPaneContent2(): JSX.Element {
    return (
        <div>
            <div className="content">
                <a href="https://www.syncfusion.com/ebooks/spark"
target="_blank">Spark Succinctly</a>

```

```

        <p>Mastering big data requires an aptitude at every step of
information processing.
        Post-processing, one of the most important steps, is where you
find Apache Spark frequently employed.
        Spark Succinctly, by Marko Svaljek, addresses Spark's use in
the ultimate step in handling big data. This e-book, the
        third installment in Svaljek's IoT series, teaches the basics
of using Spark and explores how to work with RDDs, Scala and
        Python tasks, JSON files, and Cassandra. Many of the leading
companies in the world today face the problem of big data.
    </p>
    </div>
</div>
);
}
public onCreate(): void {
    // Initialize Splitter component
    const cont = this.innerSplitterInstance.element.querySelector(".e-
pane")[1];
    cont.appendChild(this.splitterInstance.element);
}
public render(): JSX.Element {
    return (
        <div id="target" className="control-section splitter-expand" >
        <SplitterComponent id="splitter1" separatorSize={3}
height="350px" width="100%" ref={(splitter) => { this.innerSplitterInstance
= splitter! }}>
            <PanelsDirective>
                <PaneDirective size='48%' content={this.leftPaneContent} />
                <PaneDirective />
            </PanelsDirective>
        </SplitterComponent>
        <SplitterComponent id="splitter2" separatorSize={3}
orientation="Vertical" ref={(splitter) => { this.splitterInstance =
splitter; }} created={this.onCreate = this.onCreate.bind(this)}>
            <PanelsDirective>
                <PaneDirective size='50%' content={this.rightPaneContent1}
/>
                <PaneDirective content = {this.rightPaneContent2} />
            </PanelsDirective>
        </SplitterComponent>
        </div>
    );
}
}
export default App;
{% endraw %}

```

### [Functional-component]

#### APP.JSX

```

{% raw %}
import { PaneDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";

```

```
function App() {
  let innerSplitterInstance;
  let splitterInstance;
  function leftPaneContent() {
    return (<div>
      <div className="content">
        <a href="https://www.syncfusion.com/ebooks/neuralnetworks"
target="_blank">Neural Networks Using C# Succinctly</a>
        <p>Neural networks are an exciting field of software
development used to calculate outputs from input data.
        While the idea seems simple enough, the implications of such
networks are staggering—think optical character recognition,
speech recognition, and regression analysis. With Neural
Networks Using C# Succinctly by James McCaffrey, you'll learn
        how to create your own neural network to solve classification
problems, or problems where the outcomes can only be one of
several values. <br /><br />Learn about encoding and
normalizing data, activation functions and how to choose the right one, and
ultimately
        how to train a neural network to find weights and bias values
that provide accurate predictions.
        An artificial neural network (sometimes abbreviated ANN, or
shortened to just "neural network" when the context is clear) is
        a software system that loosely models biological neurons and
synapses. Before explaining exactly how neural networks work, it is
        useful to understand what types of problems they can solve.
      </p>
    </div>
  </div>);
}
function rightPaneContent1() {
  return (<div>
    <div className="content">
      <a
href="https://www.syncfusion.com/ebooks/data_capture_and_extraction_with_c_s
harp_succinctly" target="_blank"> Data Capture and Extraction with C#
Succinctly </a>
      <p>Capturing and extracting information is one of the most important
tasks a developer can perform, and making this task more
engaging without relying entirely on specialized tools is an
efficient way to improve productivity.
      In Data Capture and Extraction with C# Succinctly, author Ed Freitas
guides readers toward getting more out of C# in minimal time.
      Email has become a pillar of our modern and connected society, and
it now serves as a primary means of communication. Because each email
is filled with valuable information, data extraction has emerged as
a worthwhile skill set for developers in today's business world.
    </p>
  </div>
  </div>);
}
function rightPaneContent2() {
  return (<div>
    <div className="content">
      <a href="https://www.syncfusion.com/ebooks/spark"
target="_blank">Spark Succinctly</a>
```

```

    <p>Mastering big data requires an aptitude at every step of
    information processing.
    Post-processing, one of the most important steps, is where you
    find Apache Spark frequently employed.
    Spark Succinctly, by Marko Svaljek, addresses Spark's use in
    the ultimate step in handling big data. This e-book, the
    third installment in Svaljek's IoT series, teaches the basics
    of using Spark and explores how to work with RDDs, Scala and
    Python tasks, JSON files, and Cassandra. Many of the leading
    companies in the world today face the problem of big data.
    </p>
  </div>
</div>;
}
function onCreate() {
  // Initialize Splitter component
  const cont = innerSplitterInstance.element.querySelectorAll(".e-
pane")[1];
  cont.appendChild(splitterInstance.element);
}
return (<div id="target" className="control-section splitter-expand">
  <SplitterComponent id="splitter1" separatorSize={3} height="350px"
width="100%" ref={(splitter) => { innerSplitterInstance = splitter; }}>
    <PanelsDirective>
      <PanelDirective size='48%' content={leftPaneContent}/>
      <PanelDirective />
    </PanelsDirective>
  </SplitterComponent>
  <SplitterComponent id="splitter2" separatorSize={3}
orientation="Vertical" ref={(splitter) => { splitterInstance = splitter; }}
created={onCreate} = onCreate.bind(this)>
    <PanelsDirective>
      <PanelDirective size='50%' content={rightPaneContent1}/>
      <PanelDirective content={rightPaneContent2}/>
    </PanelsDirective>
  </SplitterComponent>
</div>);
}
export default App;
{% endraw %}

```

## APP.TSX

```

{% raw %}
import { PanelDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
function App () {
  let innerSplitterInstance: any;
  let splitterInstance: any;
  function leftPaneContent(): JSX.Element {
    return (
      <div>
        <div className="content">
          <a href="https://www.syncfusion.com/ebooks/neuralnetworks"
target="_blank">Neural Networks Using C# Succinctly</a>

```

```

        <p>Neural networks are an exciting field of software
development used to calculate outputs from input data.
        While the idea seems simple enough, the implications of such
networks are staggering—think optical character recognition,
        speech recognition, and regression analysis. With Neural
Networks Using C# Succinctly by James McCaffrey, you'll learn
        how to create your own neural network to solve classification
problems, or problems where the outcomes can only be one of
        several values. <br/><br/>Learn about encoding and normalizing
data, activation functions and how to choose the right one, and ultimately
        how to train a neural network to find weights and bias values
that provide accurate predictions.
        An artificial neural network (sometimes abbreviated ANN, or
shortened to just "neural network" when the context is clear) is
        a software system that loosely models biological neurons and
synapses. Before explaining exactly how neural networks work, it is
        useful to understand what types of problems they can solve.
    </p>
</div>
</div>
);
}
function rightPaneContent1(): JSX.Element {
    return (
        <div>
            <div className="content">
                <a href =
"https://www.syncfusion.com/ebooks/data_capture_and_extraction_with_c_sharp_
succinctly" target = "_blank" > Data Capture and Extraction with C#
Succinctly </a>
                <p>Capturing and extracting information is one of the most important
tasks a developer can perform, and making this task more
                engaging without relying entirely on specialized tools is an
efficient way to improve productivity.
                In Data Capture and Extraction with C# Succinctly, author Ed Freitas
guides readers toward getting more out of C# in minimal time.
                Email has become a pillar of our modern and connected society, and
it now serves as a primary means of communication. Because each email
                is filled with valuable information, data extraction has emerged as
a worthwhile skill set for developers in today's business world.
            </p>
        </div >
    </div>
);
}
function rightPaneContent2(): JSX.Element {
    return (
        <div>
            <div className="content">
                <a href="https://www.syncfusion.com/ebooks/spark"
target="_blank">Spark Succinctly</a>
                <p>Mastering big data requires an aptitude at every step of
information processing.
                Post-processing, one of the most important steps, is where you
find Apache Spark frequently employed.
                Spark Succinctly, by Marko Svaljek, addresses Spark's use in
the ultimate step in handling big data. This e-book, the

```

```

        third installment in Svaljek's IoT series, teaches the basics
of using Spark and explores how to work with RDDs, Scala and
        Python tasks, JSON files, and Cassandra.Many of the leading
companies in the world today face the problem of big data.
    </p>
    </div>
</div>
);
}
function onCreate(): void {
    // Initialize Splitter component
    const cont = innerSplitterInstance.element.querySelectorAll(".e-
pane")[1];
    cont.appendChild(splitterInstance.element);
}

return (
    <div id="target" className="control-section splitter-expand" >
    <SplitterComponent id="splitter1" separatorSize={3}
height="350px" width="100%" ref={(splitter) => { innerSplitterInstance =
splitter! }}>
        <PanelsDirective>
            <PanelDirective size='48%' content={leftPaneContent} />
            <PanelDirective />
        </PanelsDirective>
    </SplitterComponent>
    <SplitterComponent id="splitter2" separatorSize={3}
orientation="Vertical" ref={(splitter) => { splitterInstance = splitter; }}
created={onCreate = onCreate.bind(this)}>
        <PanelsDirective>
            <PanelDirective size='50%' content={rightPaneContent1} />
            <PanelDirective content = {rightPaneContent2} />
        </PanelsDirective>
    </SplitterComponent>
    </div>
);
}
export default App;
{% endraw %}

```

### Add or remove pane

You can add or remove panes programmatically to the splitter. By using [addPane](#) and [removePane](#) methods.

#### Add pane

You can add the panes dynamically in the splitter by passing [pane properties](#) along with index to the addPane method.

[Class-component]

### APP.JSX

```

{% raw %}
import { PanelDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {

```

```

splitterInstance;
paneDetails = {
  content: 'New Pane',
  max: '250px',
  min: '30px',
  size: '150px',
};
firstPane() {
  return (<div>
    <div className="content">Panel</div>
  </div>);
}
secondPane() {
  return (<div>
    <div className="content">Pane2</div>
  </div>);
}
btnClick() {
  if (this.splitterInstance.allPanels.length > 1) {
    this.splitterInstance.addPane(this.paneDetails, 1);
  }
}
render() {
  return (<div className="App">
    <SplitterComponent id="separator" height="200px" width='580px'
    ref={ (splitter) => { this.splitterInstance = splitter; } }>
    <PanelsDirective>
      <PaneDirective size='200px' content={this.firstPane}/>
      <PaneDirective size='200px' content={this.secondPane}/>
    </PanelsDirective>
    </SplitterComponent>
    <div id='addButton'>
      <button className='e-control e-btn' onClick={this.btnClick =
this.btnClick.bind(this)}>Add Pane</button>
    </div>
  </div>);
}
}
export default App;
{% endraw %}

```

### APP.TSX

```

{% raw %}
import { PaneDirective, PanePropertiesModel, PanelsDirective,
SplitterComponent } from '@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  public splitterInstance: SplitterComponent;
  public paneDetails: PanePropertiesModel = {
    content: 'New Pane',
    max: '250px',
    min: '30px',
    size: '150px',
  }
  public firstPane(): JSX.Element {

```

```

    return (
      <div>
        <div className="content">Panel</div>
      </div>
    );
  }
  public secondPane(): JSX.Element {
    return (
      <div>
        <div className="content">Pane2</div>
      </div>
    );
  }
  public btnClick(): void {
    if ((this.splitterInstance as any).allPanes.length > 1) {
      this.splitterInstance.addPane(this.paneDetails, 1);
    }
  }
  public render() {
    return (<div className="App">
      <SplitterComponent id="separator" height="200px" width='580px'
ref={ (splitter) => { this.splitterInstance = splitter! } }>
        <PanelsDirective>
          <PaneDirective size='200px' content = {this.firstPane}/>
          <PaneDirective size='200px' content = {this.secondPane}/>
        </PanelsDirective>
      </SplitterComponent>
      <div id='addButton'>
        <button className='e-control e-btn' onClick={this.btnClick =
this.btnClick.bind(this) }>Add Pane</button>
      </div>
    </div>);
  }
}
export default App;
{% enddraw %}

```

### [Functional-component]

#### APP.JSX

```

{% raw %}
import { PaneDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
function App() {
  let splitterInstance;
  let paneDetails = {
    content: 'New Pane',
    max: '250px',
    min: '30px',
    size: '150px',
  };
  function firstPane() {
    return (<div>
      <div className="content">Panel</div>

```



```

    </div>);
  }
  function secondPane() {
    return (<div>
      <div className="content">Pane2</div>
    </div>);
  }
  function btnClick() {
    if (splitterInstance.allPanes.length > 1) {
      splitterInstance.addPane(paneDetails, 1);
    }
  }
  return (<div className="App">
    <SplitterComponent id="separator" height="200px" width='580px'
    ref={ (splitter) => { splitterInstance = splitter; } }>
    <PanelsDirective>
      <PaneDirective size='200px' content={firstPane}/>
      <PaneDirective size='200px' content={secondPane}/>
    </PanelsDirective>
    </SplitterComponent>
    <div id='addButton'>
      <button className='e-control e-btn' onClick={btnClick =
      btnClick.bind(this)}>Add Pane</button>
    </div>
  </div>);
}
export default App;
{% endraw %}

```

## APP.TSX

```

{% raw %}
import { PaneDirective, PanePropertiesModel, PanelsDirective,
SplitterComponent } from '@syncfusion/ej2-react-layouts';
import * as React from "react";
function App () {
  let splitterInstance: SplitterComponent;
  let paneDetails: PanePropertiesModel = {
    content: 'New Pane',
    max: '250px',
    min: '30px',
    size: '150px',
  }
  function firstPane(): JSX.Element {
    return (
      <div>
        <div className="content">Panel</div>
      </div>
    );
  }
  function secondPane(): JSX.Element {
    return (
      <div>
        <div className="content">Pane2</div>
      </div>
    );
  }
}

```

```

    }
    function btnClick(): void {
        if ((splitterInstance as any).allPanes.length > 1) {
            splitterInstance.addPane(paneDetails, 1);
        }
    }
    return (<div className="App">
        <SplitterComponent id="separator" height="200px" width='580px'
ref={splitter} => { splitterInstance = splitter! }>>
        <PanelsDirective>
            <PaneDirective size='200px' content = {firstPane}/>
            <PaneDirective size='200px' content = {secondPane}/>
        </PanelsDirective>
        </SplitterComponent>
        <div id='addButton'>
            <button className='e-control e-btn' onClick={btnClick =
btnClick.bind(this) }>Add Pane</button>
        </div>
    </div>);
}
export default App;
{% enddraw %}

```

#### Remove pane

You can remove the split panes dynamically by passing the pane index to [removePane](#) method.

[Class-component]

#### APP.JSX

```

{% raw %}
import { PaneDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
    splitterInstance;
    firstPane() {
        return (<div>
            <div className="content">Panel1</div>
        </div>);
    }
    secondPane() {
        return (<div>
            <div className="content">Pane2</div>
        </div>);
    }
    thirdPane() {
        return (<div>
            <div className="content">Pane3</div>
        </div>);
    }
    btnClick() {
        if (this.splitterInstance.allPanes.length > 1) {
            this.splitterInstance.removePane(1);
        }
    }
}

```

```

render() {
    return (<div className="App">
        <SplitterComponent id="separator" height="200px" width='600px'
ref={(splitter) => { this.splitterInstance = splitter; }}>
            <PanelsDirective>
                <PaneDirective size='200px' content={this.firstPane}/>
                <PaneDirective size='200px' content={this.secondPane}/>
                <PaneDirective size='200px' content={this.thirdPane}/>
            </PanelsDirective>
        </SplitterComponent>
        <div id='removeButton'>
            <button className='e-control e-btn' onClick={this.btnClick =
this.btnClick.bind(this)}>Remove Pane</button>
        </div>
    </div>);
}
}
export default App;
{% endraw %}

```

### APP.TSX

```

{% raw %}
import { PaneDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
    public splitterInstance: SplitterComponent;
    public firstPane(): JSX.Element {
        return (
            <div>
                <div className="content">Panel</div>
            </div>
        );
    }
    public secondPane(): JSX.Element {
        return (
            <div>
                <div className="content">Pane2</div>
            </div>
        );
    }
    public thirdPane(): JSX.Element {
        return (
            <div>
                <div className="content">Pane3</div>
            </div>
        );
    }
    public btnClick(): void {
        if ((this.splitterInstance as any).allPanels.length > 1) {
            this.splitterInstance.removePane(1);
        }
    }
    public render() {
        return (

```

```

    <div className="App">
      <SplitterComponent id="separator" height="200px" width='600px'
ref={(splitter) => { this.splitterInstance = splitter! }}>
        <PanelsDirective>
          <PanelDirective size='200px' content = {this.firstPane}/>
          <PanelDirective size='200px' content = {this.secondPane}/>
          <PanelDirective size='200px' content = {this.thirdPane}/>
        </PanelsDirective>
      </SplitterComponent>
      <div id='removeButton'>
        <button className='e-control e-btn' onClick={this.btnClick =
this.btnClick.bind(this) }>Remove Pane</button>
      </div>
    </div>
  );
}
}
export default App;
{% endraw %}

```

## [Functional-component]

### APP.JSX

```

{% raw %}
import { PanelDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
function App() {
  let splitterInstance;
  function firstPane() {
    return <div>
      <div className="content">Panel1</div>
    </div>;
  }
  function secondPane() {
    return <div>
      <div className="content">Panel2</div>
    </div>;
  }
  function thirdPane() {
    return <div>
      <div className="content">Panel3</div>
    </div>;
  }
  function btnClick() {
    if (splitterInstance.allPanels.length > 1) {
      splitterInstance.removePanel(1);
    }
  }
  return <div className="App">
    <SplitterComponent id="separator" height="200px" width='600px'
ref={(splitter) => { splitterInstance = splitter; }}>
      <PanelsDirective>
        <PanelDirective size='200px' content={firstPane}/>
        <PanelDirective size='200px' content={secondPane}/>

```

```

        <PaneDirective size='200px' content={thirdPane}/>
      </PanesModule>
    </SplitterComponent>
    <div id='removeButton'>
      <button className='e-control e-btn' onClick={btnClick =
btnClick.bind(this)}>Remove Pane</button>
    </div>
  </div>);
}
export default App;
{% endraw %}

```

### APP.TSX

```

{% raw %}
import { PaneDirective, PanesModule, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
function App () {
  let splitterInstance: SplitterComponent;
  function firstPane(): JSX.Element {
    return (
      <div>
        <div className="content">Panel</div>
      </div>
    );
  }
  function secondPane(): JSX.Element {
    return (
      <div>
        <div className="content">Pane2</div>
      </div>
    );
  }
  function thirdPane(): JSX.Element {
    return (
      <div>
        <div className="content">Pane3</div>
      </div>
    );
  }
  function btnClick(): void {
    if ((splitterInstance as any).allPanesModule.length > 1) {
      splitterInstance.removePane(1);
    }
  }
  return (
    <div className="App">
      <SplitterComponent id="separator" height="200px" width='600px'
ref={splitter} => { splitterInstance = splitter! }>
        <PanesModule>
          <PaneDirective size='200px' content = {firstPane}/>
          <PaneDirective size='200px' content = {secondPane}/>
          <PaneDirective size='200px' content = {thirdPane}/>
        </PanesModule>
      </SplitterComponent>
    </div>
  );
}

```

```

        <div id='removeButton'>
            <button className='e-control e-btn' onClick={btnClick =
btnClick.bind(this) }>Remove Pane</button>
        </div>
    </div>
    );
}
export default App;
{% enddraw %}

```

See Also

- [Resizable split panes](#)
- [Collapsible panes](#)
- [Define size to a panes](#)
- [Specify content to a panes](#)

## Expand collapse in React Splitter component

### Collapsible panes

The Splitter panes can be configured with built-in expand and collapse functionalities. By default, the collapsible behavior is disabled. Enable the [collapsible](#) behavior in the paneSettings property to show or hide the expand or collapse icons in the panes. You can dynamically expand and collapse the panes by the corresponding icons.

The following code shows how to enable collapsible behavior.

[Class-component]

### APP.JSX

```

import { PaneDirective, PanesDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
    firstPaneContent() {
        return <div>
            <div className='content'>
                <h3>PARIS </h3>Paris, the city of lights and love - this short guide
is full of ideas for how to make the most of the romanticism...
            </div>
        </div>;
    }
    ;
    secondPaneContent() {
        return <div>
            <div className='content'>
                <h3>CAMEMBERT </h3>The village in the Orne département of Normandy
where the famous French cheese is originated from. </div>
            </div>;
    }
    ;
    thirdPaneContent() {
        return <div>
            <div className='content'>

```

```

        <h3>GRENOBLE </h3>The capital city of the French Alps and a major
        scientific center surrounded by many ski resorts, host of the Winter
        Olympics in 1968.
    </div>
    </div>);
    }
    ;
    render() {
        return (<div className="App">
            <SplitterComponent id="expand-collapse" height="250px" width='600px'>
                <PanelsDirective>
                    <PanelDirective size='200px' content={this.firstPanelContent}
collapsible={true}/>
                    <PanelDirective size='200px' content={this.secondPanelContent}
collapsible={true}/>
                    <PanelDirective size='200px' content={this.thirdPanelContent}
collapsible={true}/>
                </PanelsDirective>
            </SplitterComponent>
        </div>);
    }
}
export default App;

```

## APP.TSX

```

import { PanelDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
    public firstPanelContent(): JSX.Element {
        return(<div>
            <div className='content'>
                <h3>PARIS </h3>Paris, the city of lights and love - this short guide
is full of ideas for how to make the most of the romanticism...
            </div>
        </div>);
    };
    public secondPanelContent(): JSX.Element {
        return(<div>
            <div className='content'>
                <h3>CAMEMBERT </h3>The village in the Orne département of Normandy
where the famous French cheese is originated from. </div>
            </div>);
    };
    public thirdPanelContent(): JSX.Element {
        return(<div>
            <div className='content'>
                <h3>GRENOBLE </h3>The capital city of the French Alps and a major
scientific center surrounded by many ski resorts, host of the Winter
Olympics in 1968.
            </div>
        </div>);
    };
    public render(): JSX.Element {
        return (<div className="App">

```

```

        <SplitterComponent id="expand-collapse" height="250px" width='600px'>
        <PanelsDirective>
            <PanelDirective size='200px' content = {this.firstPanelContent}
collapsible={true}/>
            <PanelDirective size='200px' content = {this.secondPanelContent}
collapsible={true}/>
            <PanelDirective size='200px' content = {this.thirdPanelContent}
collapsible={true}/>
        </PanelsDirective>
        </SplitterComponent>
    </div>;
    }
}
export default App;

```

### [Functional-component]

#### APP.JSX

```

import { PanelDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
function App() {
    function firstPanelContent() {
        return <div>
            <div className='content'>
                <h3>PARIS </h3>Paris, the city of lights and love - this short guide
is full of ideas for how to make the most of the romanticism...
            </div>
        </div>;
    }
    ;
    function secondPanelContent() {
        return <div>
            <div className='content'>
                <h3>CAMEMBERT </h3>The village in the Orne département of Normandy
where the famous French cheese is originated from. </div>
            </div>;
    }
    ;
    function thirdPanelContent() {
        return <div>
            <div className='content'>
                <h3>GRENOBLE </h3>The capital city of the French Alps and a major
scientific center surrounded by many ski resorts, host of the Winter
Olympics in 1968.
            </div>
        </div>;
    }
    ;
    return <div className="App">
        <SplitterComponent id="expand-collapse" height="250px"
width='600px'>
            <PanelsDirective>
                <PanelDirective size='200px' content={firstPanelContent}
collapsible={true}/>

```



```

        <PaneDirective size='200px' content={secondPaneContent}
collapsible={true}/>
        <PaneDirective size='200px' content={thirdPaneContent}
collapsible={true}/>
      </PanelsDirective>
    </SplitterComponent>
  </div>);
}
export default App;

```

## APP.TSX

```

import { PaneDirective, PanelsDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
function App () {
  function firstPaneContent(): JSX.Element {
    return(<div>
      <div className='content'>
        <h3>PARIS </h3>Paris, the city of lights and love - this short guide
is full of ideas for how to make the most of the romanticism...
      </div>
    </div>);
  };
  function secondPaneContent(): JSX.Element {
    return(<div>
      <div className='content'>
        <h3>CAMEMBERT </h3>The village in the Orne département of Normandy
where the famous French cheese is originated from. </div>
      </div>);
  };
  function thirdPaneContent(): JSX.Element {
    return(<div>
      <div className='content'>
        <h3>GRENOBLE </h3>The capital city of the French Alps and a major
scientific center surrounded by many ski resorts, host of the Winter
Olympics in 1968.
      </div>
    </div>);
  };
  return (
    <div className="App">
      <SplitterComponent id="expand-collapse" height="250px"
width='600px'>
        <PanelsDirective>
          <PaneDirective size='200px' content = {firstPaneContent}
collapsible={true}/>
          <PaneDirective size='200px' content = {secondPaneContent}
collapsible={true}/>
          <PaneDirective size='200px' content = {thirdPaneContent}
collapsible={true}/>
        </PanelsDirective>
      </SplitterComponent>
    </div>
  );
}

```

```
export default App;
```

Control the expand and collapse action in programmatic manner

The Splitter provides an option to control the expand and collapse behavior programmatically by using the `expand` and `collapse` methods. Refer to the following code sample.

[Class-component]

#### APP.JSX

```
import { PaneDirective, PanesDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  splitterInstance;
  expandClick() {
    this.splitterInstance.expand(0);
  }
  collapseClick() {
    this.splitterInstance.collapse(0);
  }
  render() {
    return (<div className="App">
      <SplitterComponent id="expand-method" height="250px" width='600px'
        ref={splitter => this.splitterInstance = splitter}>
        <PanesDirective>
          <PaneDirective size='200px' content='Left pane' collapsible={true}/>
          <PaneDirective size='200px' content='Middle pane' collapsible={true}/>
          <PaneDirective size='200px' content='Right pane' collapsible={true}/>
        </PanesDirective>
      </SplitterComponent>
      <div id='btn-container'>
        <button className="e-control e-btn" onClick={this.expandClick =
          this.expandClick.bind(this)} id="expand"> Expand </button>
        <button className="e-control e-btn" onClick={this.collapseClick =
          this.collapseClick.bind(this)} id="collapse"> Collapse </button>
      </div>
    </div>);
  }
}
export default App;
```

#### APP.TSX

```
import { PaneDirective, PanesDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  public splitterInstance: SplitterComponent;
  public expandClick(): void {
    this.splitterInstance.expand(0);
  }
  public collapseClick(): void {
    this.splitterInstance.collapse(0);
  }
}
```

```

public render() {
  return (<div className="App">
    <SplitterComponent id="expand-method" height="250px" width='600px'
    ref={splitter => this.splitterInstance = splitter!}>
      <PanelsDirective>
        <PanelDirective size='200px' content = 'Left pane' collapsible={true}
        />
        <PanelDirective size='200px' content = 'Middle pane'
        collapsible={true}/>
        <PanelDirective size='200px' content = 'Right pane'
        collapsible={true}/>
      </PanelsDirective>
    </SplitterComponent>
    <div id='btn-container'>
      <button className="e-control e-btn"
      onClick={this.expandClick=this.expandClick.bind(this)} id="expand"> Expand
    </button>
      <button className="e-control e-btn"
      onClick={this.collapseClick=this.collapseClick.bind(this)} id="collapse">
      Collapse </button>
    </div>
  </div>);
}
}
export default App;

```

### [Functional-component]

#### APP.JSX

```

import { PanelDirective, PanelsDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
function App() {
  let splitterInstance;
  function expandClick() {
    splitterInstance.expand(0);
  }
  function collapseClick() {
    splitterInstance.collapse(0);
  }
  return (<div className="App">
    <SplitterComponent id="expand-method" height="250px" width='600px'
    ref={splitter => splitterInstance = splitter}>
      <PanelsDirective>
        <PanelDirective size='200px' content='Left pane'
        collapsible={true}/>
        <PanelDirective size='200px' content='Middle pane'
        collapsible={true}/>
        <PanelDirective size='200px' content='Right pane'
        collapsible={true}/>
      </PanelsDirective>
    </SplitterComponent>
    <div id='btn-container'>
      <button className="e-control e-btn" onClick={expandClick =
      expandClick.bind(this)} id="expand"> Expand </button>

```

```

        <button className="e-control e-btn" onClick={collapseClick =
collapseClick.bind(this) } id="collapse"> Collapse </button>
      </div>
    </div>);
  }
  export default App;

```

### APP.TSX

```

import { PaneDirective, PanesDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
function App () {
  let splitterInstance: SplitterComponent;
  function expandClick(): void {
    splitterInstance.expand(0);
  }
  function collapseClick(): void {
    splitterInstance.collapse(0);
  }
  return (
    <div className="App">
      <SplitterComponent id="expand-method" height="250px" width='600px'
ref={splitter =>splitterInstance = splitter!}>
        <PanesDirective>
          <PaneDirective size='200px' content = 'Left pane'
collapsible={true} />
          <PaneDirective size='200px' content = 'Middle pane'
collapsible={true} />
          <PaneDirective size='200px' content = 'Right pane'
collapsible={true} />
        </PanesDirective>
      </SplitterComponent>
      <div id='btn-container'>
        <button className="e-control e-btn"
onClick={expandClick=expandClick.bind(this) } id="expand"> Expand </button>
        <button className="e-control e-btn"
onClick={collapseClick=collapseClick.bind(this) } id="collapse"> Collapse
      </button>
      </div>
    </div>
  );
}
export default App;

```

### Specify initial state to panes

You can render specific panes with collapsed state on page load. Specify a Boolean value to the [collapsed](#) property to control this behavior. The following code explains how to collapse panes on rendering (the second panes renders with collapsed state).

#### [Class-component]

### APP.JSX

```

import { PaneDirective, PanesDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  firstPaneContent() {
    return (<div>
      <div className='content'>
        <h3>PARIS </h3>Paris, the city of lights and love - this short guide
is full of ideas for how to make the most of the romanticism...
      </div>
    </div>);
  }
;
  secondPaneContent() {
    return (<div>
      <div className='content'>
        <h3>CAMEMBERT </h3>The village in the Orne département of Normandy
where the famous French cheese is originated from. </div>
      </div>);
  }
;
  thirdPaneContent() {
    return (<div>
      <div className='content'>
        <h3>GRENOBLE </h3>The capital city of the French Alps and a major
scientific center surrounded by many ski resorts, host of the Winter
Olympics in 1968.
      </div>
    </div>);
  }
;
  render() {
    return (<div className="App">
      <SplitterComponent id="collapsed" height="250px" width='600px'>
        <PanesDirective>
          <PaneDirective size='200px' content={this.firstPaneContent}/>
          <PaneDirective size='200px' content={this.secondPaneContent}
collapsed={true}/>
          <PaneDirective size='200px' content={this.thirdPaneContent}/>
        </PanesDirective>
      </SplitterComponent>
    </div>);
  }
}
export default App;

```

### APP.TSX

```

import { PaneDirective, PanesDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  public firstPaneContent(): JSX.Element {
    return (<div>
      <div className='content'>

```

```

    <h3>PARIS </h3>Paris, the city of lights and love - this short guide
is full of ideas for how to make the most of the romanticism...
    </div>
    </div>);
};
public secondPaneContent(): JSX.Element {
    return (<div>
        <div className='content'>
            <h3>CAMEMBERT </h3>The village in the Orne département of Normandy
where the famous French cheese is originated from. </div>
        </div>);
};
public thirdPaneContent(): JSX.Element {
    return (<div>
        <div className='content'>
            <h3>GRENOBLE </h3>The capital city of the French Alps and a major
scientific center surrounded by many ski resorts, host of the Winter
Olympics in 1968.
        </div>
    </div>);
};
public render() {
    return (<div className="App">
        <SplitterComponent id="collapsed" height="250px" width='600px'>
            <PanelsDirective>
                <PanelDirective size='200px' content = {this.firstPaneContent}/>
                <PanelDirective size='200px' content = {this.secondPaneContent}
collapsed={true}/>
                <PanelDirective size='200px' content = {this.thirdPaneContent}/>
            </PanelsDirective>
        </SplitterComponent>
    </div>);
}
}
export default App;

```

## [Functional-component]

### APP.JSX

```

import { PanelDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
function App() {
    function firstPaneContent() {
        return (<div>
            <div className='content'>
                <h3>PARIS </h3>Paris, the city of lights and love - this short guide
is full of ideas for how to make the most of the romanticism...
            </div>
        </div>);
    }
    ;
    function secondPaneContent() {
        return (<div>
            <div className='content'>

```

```

    <h3>CAMEMBERT </h3>The village in the Orne département of Normandy
where the famous French cheese is originated from. </div>
  </div>);
}
;
function thirdPaneContent() {
  return <div>
    <div className='content'>
      <h3>GRENOBLE </h3>The capital city of the French Alps and a major
scientific center surrounded by many ski resorts, host of the Winter
Olympics in 1968.
    </div>
  </div>);
}
;
return <div className="App">
  <SplitterComponent id="collapsed" height="250px" width='600px'>
    <PanelsDirective>
      <PanelDirective size='200px' content={firstPaneContent}/>
      <PanelDirective size='200px' content={secondPaneContent}
collapsed={true}/>
      <PanelDirective size='200px' content={thirdPaneContent}/>
    </PanelsDirective>
  </SplitterComponent>
</div>);
}
export default App;

```

### APP.TSX

```

import { PanelDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
function App () {
  function firstPaneContent(): JSX.Element {
    return <div>
      <div className='content'>
        <h3>PARIS </h3>Paris, the city of lights and love - this short guide
is full of ideas for how to make the most of the romanticism...
      </div>
    </div>);
  };
  function secondPaneContent(): JSX.Element {
    return <div>
      <div className='content'>
        <h3>CAMEMBERT </h3>The village in the Orne département of Normandy
where the famous French cheese is originated from. </div>
      </div>);
  };
  function thirdPaneContent(): JSX.Element {
    return <div>
      <div className='content'>
        <h3>GRENOBLE </h3>The capital city of the French Alps and a major
scientific center surrounded by many ski resorts, host of the Winter
Olympics in 1968.
      </div>
    </div>
  };
}

```

```

    </div>);
  };
  return (
    <div className="App">
      <SplitterComponent id="collapsed" height="250px" width='600px'>
        <PanelsDirective>
          <PaneDirective size='200px' content = {firstPaneContent}/>
          <PaneDirective size='200px' content = {secondPaneContent}
collapsed={true}/>
          <PaneDirective size='200px' content = {thirdPaneContent}/>
        </PanelsDirective>
      </SplitterComponent>
    </div>
  );
}
export default App;

```

See Also

- [Resizable split panes](#)

### Resize in React Splitter component

By default, resizing will be enable for split panes. Resizing gripper element will be add to the separator to makes the resize easy.

Horizontal splitter will allows to resize in horizontal directions.

Vertical splitter will allows to resize in vertical directions.

While resizing, previous and next panes will be adjust its dimensions automatically.

### Min and Max validation

Splitter allows you to set the minimum and maximum sizes for each pane. Resizing will not be occur over the minimum and maximum values.

### [Class-component]

#### APP.JSX

```

import { PaneDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  leftContent() {
    return <div>
      <div className="content">
        <h3>PARIS </h3>
        Paris, the city of lights and love - this short guide is full of
ideas for how to make the most of the romanticism...
      </div>
    </div>);
  }
  middleContent() {
    return <div>
      <div className="content">
        <h3>CAMEMBERT </h3>

```



```

        The village in the Orne département of Normandy where the famous
        French cheese is originated from.
      </div>
    </div>);
  }
  rightContent() {
    return (<div>
      <div className="content">
        <h3>GRENOBLE </h3>
        The capital city of the French Alps and a major scientific center
        surrounded by many ski resorts, host of the Winter Olympics in 1968.
      </div>
    </div>);
  }
  render() {
    return (<div className="App">
      <SplitterComponent id="percentage" height="250px" width='600px'>
        <PanelsDirective>
          <PanelDirective size='200px' content={this.leftContent} min='20%'
max='40%' />
          <PanelDirective size='200px' content={this.middleContent} min='30%'
max='60%' />
          <PanelDirective size='200px' content={this.rightContent} />
        </PanelsDirective>
      </SplitterComponent>
    </div>);
  }
}
export default App;

```

## APP.TSX

```

import { PanelDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  public leftContent(): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>PARIS </h3>
          Paris, the city of lights and love - this short guide is full of
          ideas for how to make the most of the romanticism...
        </div>
      </div>
    );
  }
  public middleContent(): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>CAMEMBERT </h3>
          The village in the Orne département of Normandy where the famous
          French cheese is originated from.
        </div>
      </div>
    );
  }
}

```

```

    );
  }
  public rightContent(): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>GRENOBLE </h3>
          The capital city of the French Alps and a major scientific center
          surrounded by many ski resorts, host of the Winter Olympics in 1968.
        </div>
      </div>
    );
  }
  public render(): JSX.Element {
    return (<div className="App">
      <SplitterComponent id="percentage" height="250px" width='600px'>
        <PanelsDirective>
          <PanelDirective size='200px' content = {this.leftContent} min='20%'
max='40%' />
          <PanelDirective size='200px' content = {this.middleContent}
min='30%' max='60%' />
          <PanelDirective size='200px' content = {this.rightContent} />
        </PanelsDirective>
      </SplitterComponent>
    </div>);
  }
}
export default App;

```

## [Functional-component]

### APP.JSX

```

import { PanelDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
function App() {
  function leftContent() {
    return (<div>
      <div className="content">
        <h3>PARIS </h3>
        Paris, the city of lights and love - this short guide is full of
        ideas for how to make the most of the romanticism...
      </div>
    </div>);
  }
  function middleContent() {
    return (<div>
      <div className="content">
        <h3>CAMEMBERT </h3>
        The village in the Orne département of Normandy where the famous
        French cheese is originated from.
      </div>
    </div>);
  }
  function rightContent() {

```

```

    return (<div>
      <div className="content">
        <h3>GRENOBLE </h3>
        The capital city of the French Alps and a major scientific center
        surrounded by many ski resorts, host of the Winter Olympics in 1968.
      </div>
    </div>);
  }
  return (<div className="App">
    <SplitterComponent id="percentage" height="250px" width='600px'>
      <PanelsDirective>
        <PanelDirective size='200px' content={leftContent} min='20%'
max='40%' />
        <PanelDirective size='200px' content={middleContent} min='30%'
max='60%' />
        <PanelDirective size='200px' content={rightContent} />
      </PanelsDirective>
    </SplitterComponent>
  </div>);
}
export default App;

```

## APP.TSX

```

import { PanelDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
function App () {
  function leftContent(): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>PARIS </h3>
          Paris, the city of lights and love - this short guide is full of
          ideas for how to make the most of the romanticism...
        </div>
      </div>
    );
  }
  function middleContent(): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>CAMEMBERT </h3>
          The village in the Orne département of Normandy where the famous
          French cheese is originated from.
        </div>
      </div>
    );
  }
  function rightContent(): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>GRENOBLE </h3>

```

```

        The capital city of the French Alps and a major scientific center
        surrounded by many ski resorts, host of the Winter Olympics in 1968.
    </div>
</div>
);
}
return (<div className="App">
    <SplitterComponent id="percentage" height="250px" width='600px'>
        <PanelsDirective>
            <PanelDirective size='200px' content = {leftContent} min='20%'
max='40%' />
            <PanelDirective size='200px' content = {middleContent} min='30%'
max='60%' />
            <PanelDirective size='200px' content = {rightContent} />
        </PanelsDirective>
    </SplitterComponent>
</div>);
}
export default App;

```

### Prevent resizing

You can disable the resizing for the pane by setting `false` to the `resizable` property within `paneSettings`.

### [Class-component]

#### APP.JSX

```

import { PanelDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
    leftContent() {
        return (<div>
            <div className="content">
                <h3>PARIS </h3>
                Paris, the city of lights and love - this short guide is full of
                ideas for how to make the most of the romanticism...
            </div>
        </div>);
    }
    middleContent() {
        return (<div>
            <div className="content">
                <h3>CAMEMBERT </h3>
                The village in the Orne département of Normandy where the famous
                French cheese is originated from.
            </div>
        </div>);
    }
    rightContent() {
        return (<div>
            <div className="content">
                <h3>GRENOBLE </h3>
                The capital city of the French Alps and a major scientific center
                surrounded by many ski resorts, host of the Winter Olympics in 1968.
            </div>
        </div>);
    }
}

```

```

    </div>);
  }
  render() {
    return (<div className="App">
      <SplitterComponent id="percentage" height="250px" width='600px'>
        <PanelsDirective>
          <PanelDirective size='200px' content={this.leftContent}
resizable={false}/>
          <PanelDirective size='200px' content={this.middleContent}/>
          <PanelDirective size='200px' content={this.rightContent}/>
        </PanelsDirective>
      </SplitterComponent>
    </div>);
  }
}
export default App;

```

## APP.TSX

```

import { PanelDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  public leftContent(): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>PARIS </h3>
          Paris, the city of lights and love - this short guide is full of
ideas for how to make the most of the romanticism...
        </div>
      </div>
    );
  }
  public middleContent(): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>CAMEMBERT </h3>
          The village in the Orne département of Normandy where the famous
French cheese is originated from.
        </div>
      </div>
    );
  }
  public rightContent(): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>GRENOBLE </h3>
          The capital city of the French Alps and a major scientific center
surrounded by many ski resorts, host of the Winter Olympics in 1968.
        </div>
      </div>
    );
  }
}

```

```

public render(): JSX.Element {
  return (<div className="App">
    <SplitterComponent id="percentage" height="250px" width='600px'>
      <PanelsDirective>
        <PanelDirective size='200px' content = {this.leftContent}
resizable={false}/>
        <PanelDirective size='200px' content = {this.middleContent} />
        <PanelDirective size='200px' content = {this.rightContent} />
      </PanelsDirective>
    </SplitterComponent>
  </div>);
}
}
export default App;

```

### [Functional-component]

#### APP.JSX

```

import { PanelDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
function App() {
  function leftContent() {
    return (<div>
      <div className="content">
        <h3>PARIS </h3>
        Paris, the city of lights and love - this short guide is full of
ideas for how to make the most of the romanticism...
      </div>
    </div>);
  }
  function middleContent() {
    return (<div>
      <div className="content">
        <h3>CAMEMBERT </h3>
        The village in the Orne département of Normandy where the famous
French cheese is originated from.
      </div>
    </div>);
  }
  function rightContent() {
    return (<div>
      <div className="content">
        <h3>GRENOBLE </h3>
        The capital city of the French Alps and a major scientific center
surrounded by many ski resorts, host of the Winter Olympics in 1968.
      </div>
    </div>);
  }
  return (<div className="App">
    <SplitterComponent id="percentage" height="250px" width='600px'>
      <PanelsDirective>
        <PanelDirective size='200px' content={leftContent}
resizable={false}/>
        <PanelDirective size='200px' content={middleContent}/>

```

```

        <PaneDirective size='200px' content={rightContent}/>
      </PanesModule>
    </SplitterComponent>
  </div>);
}
export default App;

```

### APP.TSX

```

import { PaneDirective, PanesModule, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
function App () {
  function leftContent(): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>PARIS </h3>
          Paris, the city of lights and love - this short guide is full of
          ideas for how to make the most of the romanticism...
        </div>
      </div>
    );
  }
  function middleContent(): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>CAMEMBERT </h3>
          The village in the Orne département of Normandy where the famous
          French cheese is originated from.
        </div>
      </div>
    );
  }
  function rightContent(): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>GRENOBLE </h3>
          The capital city of the French Alps and a major scientific center
          surrounded by many ski resorts, host of the Winter Olympics in 1968.
        </div>
      </div>
    );
  }
  return (<div className="App">
    <SplitterComponent id="percentage" height="250px" width='600px'>
      <PanesModule>
        <PaneDirective size='200px' content = {leftContent}
        resizable={false}/>
        <PaneDirective size='200px' content = {middleContent} />
        <PaneDirective size='200px' content = {rightContent} />
      </PanesModule>
    </SplitterComponent>
  </div>);
}

```

```
}
export default App;
```

Refresh content on resizing

While resizing the panes, you can refresh the pane contents by using either [resizeStart](#), [resizing](#) or [resizeStop](#) events.

Customize the resize grip and cursor

You can customize the resize gripper icon and cursor in CSS level.

[Class-component]

### APP.JSX

```
import { PaneDirective, PanesDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  leftContent() {
    return (<div>
      <div className="content">
        <h3>PARIS </h3>
        Paris, the city of lights and love - this short guide is full of
        ideas for how to make the most of the romanticism...
      </div>
    </div>);
  }
  middleContent() {
    return (<div>
      <div className="content">
        <h3>CAMEMBERT </h3>
        The village in the Orne département of Normandy where the famous
        French cheese is originated from.
      </div>
    </div>);
  }
  rightContent() {
    return (<div>
      <div className="content">
        <h3>GRENOBLE </h3>
        The capital city of the French Alps and a major scientific center
        surrounded by many ski resorts, host of the Winter Olympics in 1968.
      </div>
    </div>);
  }
  render() {
    return (<div className="App">
      <SplitterComponent id="percentage" height="250px" width='600px'>
        <PanesDirective>
          <PaneDirective size='200px' content={this.leftContent}/>
          <PaneDirective size='200px' content={this.middleContent}/>
          <PaneDirective size='200px' content={this.rightContent}/>
        </PanesDirective>
      </SplitterComponent>
    </div>);
  }
}
```



```
}
export default App;
```

### APP.TSX

```
import { PaneDirective, PanesDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  public leftContent(): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>PARIS </h3>
          Paris, the city of lights and love - this short guide is full of
          ideas for how to make the most of the romanticism...
        </div>
      </div>
    );
  }
  public middleContent(): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>CAMEMBERT </h3>
          The village in the Orne département of Normandy where the famous
          French cheese is originated from.
        </div>
      </div>
    );
  }
  public rightContent(): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>GRENOBLE </h3>
          The capital city of the French Alps and a major scientific center
          surrounded by many ski resorts, host of the Winter Olympics in 1968.
        </div>
      </div>
    );
  }
  public render(): JSX.Element {
    return (<div className="App">
      <SplitterComponent id="percentage" height="250px" width='600px'>
        <PanesDirective>
          <PaneDirective size='200px' content = {this.leftContent} />
          <PaneDirective size='200px' content = {this.middleContent} />
          <PaneDirective size='200px' content = {this.rightContent} />
        </PanesDirective>
      </SplitterComponent>
    </div>);
  }
}
export default App;
```

**[Functional-component]****APP.JSX**

```

import { PaneDirective, PanesDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
function App() {
    function leftContent() {
        return (<div>
            <div className="content">
                <h3>PARIS </h3>
                Paris, the city of lights and love - this short guide is full of
                ideas for how to make the most of the romanticism...
            </div>
        </div>);
    }
    function middleContent() {
        return (<div>
            <div className="content">
                <h3>CAMEMBERT </h3>
                The village in the Orne département of Normandy where the famous
                French cheese is originated from.
            </div>
        </div>);
    }
    function rightContent() {
        return (<div>
            <div className="content">
                <h3>GRENOBLE </h3>
                The capital city of the French Alps and a major scientific center
                surrounded by many ski resorts, host of the Winter Olympics in 1968.
            </div>
        </div>);
    }
    return (<div className="App">
        <SplitterComponent id="percentage" height="250px" width='600px'>
            <PanesDirective>
                <PaneDirective size='200px' content={leftContent}/>
                <PaneDirective size='200px' content={middleContent}/>
                <PaneDirective size='200px' content={rightContent}/>
            </PanesDirective>
        </SplitterComponent>
    </div>);
}
export default App;

```

**APP.TSX**

```

import { PaneDirective, PanesDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
function App() {
    function leftContent(): JSX.Element {
        return (
            <div>
                <div className="content">

```

```

        <h3>PARIS </h3>
        Paris, the city of lights and love - this short guide is full of
        ideas for how to make the most of the romanticism...
      </div>
    </div>
  );
}
function middleContent(): JSX.Element {
  return (
    <div>
      <div className="content">
        <h3>CAMEMBERT </h3>
        The village in the Orne département of Normandy where the famous
        French cheese is originated from.
      </div>
    </div>
  );
}
function rightContent(): JSX.Element {
  return (
    <div>
      <div className="content">
        <h3>GRENOBLE </h3>
        The capital city of the French Alps and a major scientific center
        surrounded by many ski resorts, host of the Winter Olympics in 1968.
      </div>
    </div>
  );
}
return (<div className="App">
  <SplitterComponent id="percentage" height="250px" width='600px'>
    <PanelsDirective>
      <PaneDirective size='200px' content = {leftContent} />
      <PaneDirective size='200px' content = {middleContent} />
      <PaneDirective size='200px' content = {rightContent} />
    </PanelsDirective>
  </SplitterComponent>
</div>);
}
export default App;

```

## See Also

- [Collapsible panes](#)

## Pane content in React Splitter component

This section explains how to provide plain text content or HTML markup or integrate other React UI controls as content of splitter.

### Template

You can render the HTML element directly to the splitter pane content using [content](#) template property.

[Class-component]

### APP.JSX

```

import { PaneDirective, PanesDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  firstPaneContent() {
    return (<div>
      <div className='content'>
        <h3>PARIS </h3>Paris, the city of lights and love - this short guide
is full of ideas for how to make the most of the romanticism...
      </div>
    </div>);
  }
;
  secondPaneContent() {
    return (<div>
      <div className='content'>
        <h3>CAMEMBERT </h3>The village in the Orne département of Normandy
where the famous French cheese is originated from. </div>
      </div>);
  }
;
  thirdPaneContent() {
    return (<div>
      <div className='content'>
        <h3>GRENOBLE </h3>The capital city of the French Alps and a major
scientific center surrounded by many ski resorts, host of the Winter
Olympics in 1968.
      </div>
    </div>);
  }
;
  render() {
    return (<div className="App">
      <SplitterComponent id="collapsed" height="250px" width='600px'>
        <PanesDirective>
          <PaneDirective size='200px' content={this.firstPaneContent}/>
          <PaneDirective size='200px' content={this.secondPaneContent}/>
          <PaneDirective size='200px' content={this.thirdPaneContent}/>
        </PanesDirective>
      </SplitterComponent>
    </div>);
  }
}
export default App;

```

**APP.TSX**

```

import { PaneDirective, PanesDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  public firstPaneContent(): JSX.Element {
    return (<div>
      <div className='content'>
        <h3>PARIS </h3>Paris, the city of lights and love - this short guide
is full of ideas for how to make the most of the romanticism...
      </div>
    </div>);
  }
}

```

```

        </div>
      </div>);
    };
    public secondPaneContent(): JSX.Element {
      return <div>
        <div className='content'>
          <h3>CAMEMBERT </h3>The village in the Orne département of Normandy
where the famous French cheese is originated from. </div>
        </div>);
    };
    public thirdPaneContent(): JSX.Element {
      return <div>
        <div className='content'>
          <h3>GRENOBLE </h3>The capital city of the French Alps and a major
scientific center surrounded by many ski resorts, host of the Winter
Olympics in 1968.
        </div>
      </div>);
    };
    public render() {
      return <div className="App">
        <SplitterComponent id="collapsed" height="250px" width='600px'>
          <PanelsDirective>
            <PaneDirective size='200px' content = {this.firstPaneContent}/>
            <PaneDirective size='200px' content = {this.secondPaneContent}/>
            <PaneDirective size='200px' content = {this.thirdPaneContent}/>
          </PanelsDirective>
        </SplitterComponent>
      </div>);
    }
  }
}
export default App;

```

## [Functional-component]

### APP.JSX

```

import { PaneDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
function App() {
  function firstPaneContent() {
    return <div>
      <div className='content'>
        <h3>PARIS </h3>Paris, the city of lights and love - this short guide
is full of ideas for how to make the most of the romanticism...
      </div>
    </div>);
  }
  ;
  function secondPaneContent() {
    return <div>
      <div className='content'>
        <h3>CAMEMBERT </h3>The village in the Orne département of Normandy
where the famous French cheese is originated from. </div>
      </div>);
  }
}

```

```

    }
    ;
    function thirdPaneContent() {
        return (<div>
            <div className='content'>
                <h3>GRENOBLE </h3>The capital city of the French Alps and a major
                scientific center surrounded by many ski resorts, host of the Winter
                Olympics in 1968.
            </div>
        </div>);
    }
    ;
    return (<div className="App">
        <SplitterComponent id="collapsed" height="250px" width='600px'>
            <PanesDirective>
                <PaneDirective size='200px' content={firstPaneContent}/>
                <PaneDirective size='200px' content={secondPaneContent}/>
                <PaneDirective size='200px' content={thirdPaneContent}/>
            </PanesDirective>
        </SplitterComponent>
    </div>);
}
export default App;

```

## APP.TSX

```

import { PaneDirective, PanesDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
function App () {
    function firstPaneContent(): JSX.Element {
        return(<div>
            <div className='content'>
                <h3>PARIS </h3>Paris, the city of lights and love - this short guide
                is full of ideas for how to make the most of the romanticism...
            </div>
        </div>);
    };
    function secondPaneContent(): JSX.Element {
        return(<div>
            <div className='content'>
                <h3>CAMEMBERT </h3>The village in the Orne département of Normandy
                where the famous French cheese is originated from. </div>
            </div>);
    };
    function thirdPaneContent(): JSX.Element {
        return(<div>
            <div className='content'>
                <h3>GRENOBLE </h3>The capital city of the French Alps and a major
                scientific center surrounded by many ski resorts, host of the Winter
                Olympics in 1968.
            </div>
        </div>);
    };
    return (
        <div className="App">

```

```

        <SplitterComponent id="collapsed" height="250px" width='600px'>
        <PanelsDirective>
            <PanelDirective size='200px' content = {firstPaneContent}/>
            <PanelDirective size='200px' content = {secondPaneContent}/>
            <PanelDirective size='200px' content = {thirdPaneContent}/>
        </PanelsDirective>
        </SplitterComponent>
    </div>
    );
}
export default App;

```

## React UI components

You can render any React UI controls along with their native and control events within splitter as pane content.

You can refer [Accordion within splitter](#) and [Listview within splitter](#) samples.

## Plain content

You can add the plain text as a pane contents using either inner HTML or [content](#) API

## [Class-component]

### APP.JSX

```

import { PanelDirective, PanelsDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
    render() {
        return (<div className="App">
            <SplitterComponent id="plain" height="250px" width='600px'>
            <PanelsDirective>
                <PanelDirective size='200px' content='<div class="content"> Left pane
            </div>' />
                <PanelDirective size='200px' content='<div class="content"> Middle pane
            </div>' />
                <PanelDirective size='200px' content='<div class="content"> Right pane
            </div>' />
            </PanelsDirective>
            </SplitterComponent>
            </div>);
    }
}
export default App;

```

### APP.TSX

```

import { PanelDirective, PanelsDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
    public render() {
        return (<div className="App">
            <SplitterComponent id="plain" height="250px" width='600px'>
            <PanelsDirective>

```

```

    <PaneDirective size='200px' content = '<div class="content"> Left pane
</div>' />
    <PaneDirective size='200px' content = '<div class="content"> Middle pane
</div>' />
    <PaneDirective size='200px' content = '<div class="content"> Right pane
</div>' />
  </PanelsDirective>
</SplitterComponent>
</div>);
}
}
export default App;

```

### [Functional-component]

#### APP.JSX

```

import { PaneDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
function App() {
  return (<div className="App">
    <SplitterComponent id="plain" height="250px" width='600px'>
      <PanelsDirective>
        <PaneDirective size='200px' content='<div class="content"> Left pane
</div>' />
        <PaneDirective size='200px' content='<div class="content"> Middle
pane </div>' />
        <PaneDirective size='200px' content='<div class="content"> Right
pane </div>' />
      </PanelsDirective>
    </SplitterComponent>
  </div>);
}
export default App;

```

#### APP.TSX

```

import { PaneDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
function App () {
  return (
    <div className="App">
      <SplitterComponent id="plain" height="250px" width='600px'>
        <PanelsDirective>
          <PaneDirective size='200px' content = '<div class="content"> Left
pane </div>' />
          <PaneDirective size='200px' content = '<div class="content"> Middle
pane </div>' />
          <PaneDirective size='200px' content = '<div class="content"> Right
pane </div>' />
        </PanelsDirective>
      </SplitterComponent>
    </div>
  );
}

```



```
}
export default App;
```

## HTML Markup

Splitter is a layout based container control. You can render the pane contents from existing HTML markup. Converting HTML markup as splitter pane is easy way to add the panes content dynamically.

## [Class-component]

### APP.JSX

```
import { PaneDirective, PanesDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  render() {
    return (<div className="App">
      <SplitterComponent id="plain" height="250px" width='600px'>
        <PanesDirective>
          <PaneDirective size='200px' content='<div class="content"><h3
class="h3">PARIS </h3>Paris, the city of lights and love - this short guide
is full of ideas for how to make the most of the romanticism...</div>'>
            <PaneDirective size='200px' content='<div class="content"><h3
class="h3">CAMEMBERT </h3>The village in the Orne département of Normandy
where the famous French cheese is originated from.</div>'>
              <PaneDirective size='200px' content='<div class="content"><h3
class="h3">GRENOBLE </h3>The capital city of the French Alps and a major
scientific center surrounded by many ski resorts, host of the Winter
Olympics in 1968.</div>'>
            </PaneDirective>
          </PanesDirective>
        </SplitterComponent>
      </div>);
  }
}
export default App;
```

### APP.TSX

```
import { PaneDirective, PanesDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  public render() {
    return (<div className="App">
      <SplitterComponent id="plain" height="250px" width='600px'>
        <PanesDirective>
          <PaneDirective size='200px' content = '<div class="content"><h3
class="h3">PARIS </h3>Paris, the city of lights and love - this short guide
is full of ideas for how to make the most of the romanticism...</div>'>
            <PaneDirective size='200px' content = '<div class="content"><h3
class="h3">CAMEMBERT </h3>The village in the Orne département of Normandy
where the famous French cheese is originated from.</div>'>
              <PaneDirective size='200px' content = '<div class="content"><h3
class="h3">GRENOBLE </h3>The capital city of the French Alps and a major
scientific center surrounded by many ski resorts, host of the Winter
Olympics in 1968.</div>'>
            </PaneDirective>
          </PanesDirective>
        </SplitterComponent>
      </div>);
  }
}
```

```

    </PanesModule>
  </SplitterComponent>
</div>);
}
}
export default App;

```

## [Functional-component]

### APP.JSX

```

import { PaneDirective, PanesDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
function App() {
  return (<div className="App">
    <SplitterComponent id="plain" height="250px" width='600px'>
      <PanesModule>
        <PaneDirective size='200px' content='<div class="content"><h3
class="h3">PARIS </h3>Paris, the city of lights and love - this short guide
is full of ideas for how to make the most of the romanticism...</div>'>
          <PaneDirective size='200px' content='<div class="content"><h3
class="h3">CAMEMBERT </h3>The village in the Orne département of Normandy
where the famous French cheese is originated from.</div>'>
            <PaneDirective size='200px' content='<div class="content"><h3
class="h3">GRENOBLE </h3>The capital city of the French Alps and a major
scientific center surrounded by many ski resorts, host of the Winter
Olympics in 1968.</div>'>
              </PanesModule>
            </SplitterComponent>
          </div>);
        }
      export default App;

```

### APP.TSX

```

import { PaneDirective, PanesDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
function App () {
  return (
    <div className="App">
      <SplitterComponent id="plain" height="250px" width='600px'>
        <PanesModule>
          <PaneDirective size='200px' content = '<div class="content"><h3
class="h3">PARIS </h3>Paris, the city of lights and love - this short guide
is full of ideas for how to make the most of the romanticism...</div>'>
            <PaneDirective size='200px' content = '<div class="content"><h3
class="h3">CAMEMBERT </h3>The village in the Orne département of Normandy
where the famous French cheese is originated from.</div>'>
              <PaneDirective size='200px' content = '<div class="content"><h3
class="h3">GRENOBLE </h3>The capital city of the French Alps and a major
scientific center surrounded by many ski resorts, host of the Winter
Olympics in 1968.</div>'>
                </PanesModule>
              </SplitterComponent>
            }
          }

```

```

    </div>
  );
}
export default App;

```

### Pane content using selector

You can set HTML element as pane [content](#) using the query selectors such as ID or CSS class selectors.

The following code demonstrates how to fetch an element from the document and load it to pane using its ID.

#### [Class-component]

##### APP.JSX

```

{% raw %}
import { PaneDirective, PanesDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  render() {
    return (<div className="App">
      { /* <!-- pane contents --> */ }
      <div id="left-pane-content" style={{ display: "none" }}>
        <div>Left pane<div id='panetext'>size: 25%</div>
        <div id='panetext'>min: 60px</div>
      </div>
      <div id="middle-pane-content" style={{ display: "none" }}>
        <span>Middle pane<div id="panetext">size: 50%</div>
        <div id="panetext">min: 60px</div>
      </span>
      </div>
      <div id="last-pane-content" style={{ display: "none" }}>
        <span>Right pane<div id="panetext">size: 25%</div>
        <div id="panetext">min: 60px</div>
      </span>
      </div>
      <SplitterComponent id="plain" height="250px" width='600px'>
        <PanesDirective>
          <PaneDirective size='200px' min="60px" content='#left-pane-content' />
          <PaneDirective size='200px' min="60px" content='#middle-pane-
content' />
          <PaneDirective size='200px' min="60px" content='#last-pane-content' />
        </PanesDirective>
      </SplitterComponent>
    </div>);
  }
}
export default App;
{% endraw %}

```

##### APP.TSX

```

{% raw %}

```

```

import { PaneDirective, PanesDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
public render() {
return (<div className="App">
{ /* <!-- pane contents --> */}
<div id="left-pane-content" style={{display: "none"}}>
<div>Left pane<div id='panetext'>size: 25%</div>
<div id='panetext'>min: 60px</div>
</div>
</div>
<div id="middle-pane-content" style={{display: "none"}}>
<span>Middle pane<div id="panetext">size: 50%</div>
<div id="panetext">min: 60px</div>
</span>
</div>
<div id="last-pane-content" style={{display: "none"}}>
<span>Right pane<div id="panetext">size: 25%</div>
<div id="panetext">min: 60px</div>
</span>
</div>
<SplitterComponent id="plain" height="250px" width='600px'>
<PanesDirective>
<PaneDirective size='200px' min="60px" content = '#left-pane-content' />
<PaneDirective size='200px' min="60px" content = '#middle-pane-
content' />
<PaneDirective size='200px' min="60px" content = '#last-pane-content' />
</PanesDirective>
</SplitterComponent>
</div>
);
}
}
export default App;
{% endraw %}

```

## [Functional-component]

### APP.JSX

```

{% raw %}
import { PaneDirective, PanesDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
function App() {
return (<div className="App">
{ /* <!-- pane contents --> */}
<div id="left-pane-content" style={{ display: "none" }}>
<div>Left pane<div id='panetext'>size: 25%</div>
<div id='panetext'>min: 60px</div>
</div>
</div>
<div id="middle-pane-content" style={{ display: "none" }}>
<span>Middle pane<div id="panetext">size: 50%</div>
<div id="panetext">min: 60px</div>

```

```

        </span>
      </div>
      <div id="last-pane-content" style={{ display: "none" }}>
        <span>Right pane<div id="panetext">size: 25%</div>
          <div id="panetext">min: 60px</div>
        </span>
      </div>
      <SplitterComponent id="plain" height="250px" width='600px'>
        <PanelsDirective>
          <PanelDirective size='200px' min="60px" content='#left-pane-
content' />
          <PanelDirective size='200px' min="60px" content='#middle-pane-
content' />
          <PanelDirective size='200px' min="60px" content='#last-pane-
content' />
        </PanelsDirective>
      </SplitterComponent>
    </div>);
  }
  export default App;
  {% endraw %}

```

### APP.TSX

```

{% raw %}
import { PanelDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
function App () {
  return (
    <div className="App">
      { /* <!-- pane contents --> */}
      <div id="left-pane-content" style={{display: "none"}}>
        <div>Left pane<div id='panetext'>size: 25%</div>
          <div id='panetext'>min: 60px</div>
        </div>
      </div>
      <div id="middle-pane-content" style={{display: "none"}}>
        <span>Middle pane<div id="panetext">size: 50%</div>
          <div id="panetext">min: 60px</div>
        </span>
      </div>
      <div id="last-pane-content" style={{display: "none"}}>
        <span>Right pane<div id="panetext">size: 25%</div>
          <div id="panetext">min: 60px</div>
        </span>
      </div>
      <SplitterComponent id="plain" height="250px" width='600px'>
        <PanelsDirective>
          <PanelDirective size='200px' min="60px" content = '#left-pane-
content' />
          <PanelDirective size='200px' min="60px" content = '#middle-pane-
content' />
          <PanelDirective size='200px' min="60px" content = '#last-pane-
content' />
        </PanelsDirective>

```

```

        </SplitterComponent>
      </div>
    );
  }
  export default App;
  {% endraw %}

```

## Pane sizing in React Splitter component

Splitter allows you to provide pane sizes either in pixel or percentage formats.

### Pane size in pixel

#### [Class-component]

##### APP.JSX

```

import { PaneDirective, PanesDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  render() {
    return (<div className="App">
      <SplitterComponent id="pixel_size" height="250px" width='600px'>
        <PanesDirective>
          <PaneDirective size='200px' content='Left pane' />
          <PaneDirective size='200px' content='Middle pane' />
          <PaneDirective size='200px' content='Right pane' />
        </PanesDirective>
      </SplitterComponent>
    </div>);
  }
}
export default App;

```

##### APP.TSX

```

import { PaneDirective, PanesDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  public render() {
    return (<div className="App">
      <SplitterComponent id="pixel_size" height="250px" width='600px'>
        <PanesDirective>
          <PaneDirective size='200px' content = 'Left pane' />
          <PaneDirective size='200px' content = 'Middle pane' />
          <PaneDirective size='200px' content = 'Right pane' />
        </PanesDirective>
      </SplitterComponent>
    </div>);
  }
}
export default App;

```

#### [Functional-component]

**APP.JSX**

```
import { PaneDirective, PanesDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
function App() {
    return (<div className="App">
        <SplitterComponent id="pixel_size" height="250px" width='600px'>
            <PanesDirective>
                <PaneDirective size='200px' content='Left pane' />
                <PaneDirective size='200px' content='Middle pane' />
                <PaneDirective size='200px' content='Right pane' />
            </PanesDirective>
        </SplitterComponent>
    </div>);
}
export default App;
```

**APP.TSX**

```
import { PaneDirective, PanesDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
function App () {
    return (
        <div className="App">
            <SplitterComponent id="pixel_size" height="250px" width='600px'>
                <PanesDirective>
                    <PaneDirective size='200px' content = 'Left pane' />
                    <PaneDirective size='200px' content = 'Middle pane' />
                    <PaneDirective size='200px' content = 'Right pane' />
                </PanesDirective>
            </SplitterComponent>
        </div>
    );
}
export default App;
```

Pane size in percentage

[Class-component]

**APP.JSX**

```
import { PaneDirective, PanesDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
    render() {
        return (<div className="App">
            <SplitterComponent id="percentage" height="250px" width='600px'>
                <PanesDirective>
                    <PaneDirective size='30%' content='Left pane' />
                    <PaneDirective size='40%' content='Middle pane' />
                    <PaneDirective size='30%' content='Right pane' />
                </PanesDirective>
            </SplitterComponent>
        </div>);
    }
}
```

```

        </div>);
    }
}
export default App;

```

### APP.TSX

```

import { PaneDirective, PanesDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
public render() {
    return (<div className="App">
        <SplitterComponent id="percentage" height="250px" width='600px'>
        <PanesDirective>
            <PaneDirective size='30%' content = 'Left pane' />
            <PaneDirective size='40%' content = 'Middle pane' />
            <PaneDirective size='30%' content = 'Right pane' />
        </PanesDirective>
        </SplitterComponent>
    </div>);
}
}
export default App;

```

### [Functional-component]

### APP.JSX

```

import { PaneDirective, PanesDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
function App() {
    return (<div className="App">
        <SplitterComponent id="percentage" height="250px" width='600px'>
        <PanesDirective>
            <PaneDirective size='30%' content='Left pane' />
            <PaneDirective size='40%' content='Middle pane' />
            <PaneDirective size='30%' content='Right pane' />
        </PanesDirective>
        </SplitterComponent>
    </div>);
}
export default App;

```

### APP.TSX

```

import { PaneDirective, PanesDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
function App () {
    return (
        <div className="App">
            <SplitterComponent id="percentage" height="250px" width='600px'>
            <PanesDirective>

```



```

        <PaneDirective size='30%' content = 'Left pane' />
        <PaneDirective size='40%' content = 'Middle pane' />
        <PaneDirective size='30%' content = 'Right pane' />
    </PanelsDirective>
</SplitterComponent>
</div>
);
}
export default App;

```

### Auto size panes

You can render the split panes without providing the size values. It will split up the sizes automatically.

### [Class-component]

#### APP.JSX

```

import { PaneDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
    leftContent(data) {
        return (<div>
            <div className="content">
                <h3>Grid </h3>
                The ASP.NET DataGrid control, or DataTable is a feature-rich
control used to display data in a tabular format.
            </div>
        </div>);
    }
    middleContent(data) {
        return (<div>
            <div className="content">
                <h3>Schedule </h3>
                The ASP.NET Scheduler, a.k.a. event calendar, facilitates almost
all calendar features, thus allowing users to manage their time efficiently.
            </div>
        </div>);
    }
    rightContent(data) {
        return (<div>
            <div className="content">
                <h3>Chart </h3>
                ASP.NET charts, a well-crafted easy-to-use charting package, is
used to add beautiful charts in web and mobile applications.
            </div>
        </div>);
    }
    render() {
        return (<div className="App">
            <SplitterComponent id="plain" height="200px" width='600px'>
            <PanelsDirective>
                <PaneDirective content={this.leftContent}/>
                <PaneDirective content={this.middleContent}/>
                <PaneDirective content={this.rightContent}/>
            </PanelsDirective>

```

```

    </SplitterComponent>
      </div>);
    }
  }
}
export default App;

```

## APP.TSX

```

import { PaneDirective, PanesDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  public leftContent(data: any): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>Grid </h3>
          The ASP.NET DataGrid control, or DataTable is a feature-rich
control used to display data in a tabular format.
        </div>
      </div>
    );
  }
  public middleContent(data: any): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>Schedule </h3>
          The ASP.NET Scheduler, a.k.a. event calendar, facilitates almost
all calendar features, thus allowing users to manage their time efficiently.
        </div>
      </div>
    );
  }
  public rightContent(data: any): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>Chart </h3>
          ASP.NET charts, a well-crafted easy-to-use charting package, is
used to add beautiful charts in web and mobile applications.
        </div>
      </div>
    );
  }
  public render() {
    return (<div className="App">
      <SplitterComponent id="plain" height="200px" width='600px'>
        <PanesDirective>
          <PaneDirective content = {this.leftContent}/>
          <PaneDirective content = {this.middleContent}/>
          <PaneDirective content = {this.rightContent}/>
        </PanesDirective>
      </SplitterComponent>
    </div>);
  }
}

```

```
}
export default App;
```

## [Functional-component]

### APP.JSX

```
import { PaneDirective, PanesDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
function App() {
  function leftContent(data) {
    return (<div>
      <div className="content">
        <h3>Grid </h3>
        The ASP.NET DataGrid control, or DataTable is a feature-rich
        control used to display data in a tabular format.
      </div>
    </div>);
  }
  function middleContent(data) {
    return (<div>
      <div className="content">
        <h3>Schedule </h3>
        The ASP.NET Scheduler, a.k.a. event calendar, facilitates almost
        all calendar features, thus allowing users to manage their time efficiently.
      </div>
    </div>);
  }
  function rightContent(data) {
    return (<div>
      <div className="content">
        <h3>Chart </h3>
        ASP.NET charts, a well-crafted easy-to-use charting package, is
        used to add beautiful charts in web and mobile applications.
      </div>
    </div>);
  }
  return (<div className="App">
    <SplitterComponent id="plain" height="200px" width='600px'>
      <PanesDirective>
        <PaneDirective content={leftContent}/>
        <PaneDirective content={middleContent}/>
        <PaneDirective content={rightContent}/>
      </PanesDirective>
    </SplitterComponent>
  </div>);
}
export default App;
```

### APP.TSX

```
import { PaneDirective, PanesDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
function App () {
```

```

function leftContent(data: any): JSX.Element {
  return (
    <div>
      <div className="content">
        <h3>Grid </h3>
        The ASP.NET DataGrid control, or DataTable is a feature-rich
        control used to display data in a tabular format.
      </div>
    </div>
  );
}
function middleContent(data: any): JSX.Element {
  return (
    <div>
      <div className="content">
        <h3>Schedule </h3>
        The ASP.NET Scheduler, a.k.a. event calendar, facilitates almost
        all calendar features, thus allowing users to manage their time efficiently.
      </div>
    </div>
  );
}
function rightContent(data: any): JSX.Element {
  return (
    <div>
      <div className="content">
        <h3>Chart </h3>
        ASP.NET charts, a well-crafted easy-to-use charting package, is
        used to add beautiful charts in web and mobile applications.
      </div>
    </div>
  );
}
return (
  <div className="App">
    <SplitterComponent id="plain" height="200px" width='600px'>
      <PanelsDirective>
        <PanelDirective content = {leftContent}/>
        <PanelDirective content = {middleContent}/>
        <PanelDirective content = {rightContent}/>
      </PanelsDirective>
    </SplitterComponent>
  </div>
);
}
export default App;

```

### Fixed pane

You can render the split panes with fixed sizes. Since last pane is a flexible pane, fixed size will not be applied.

[Class-component]

### APP.JSX

```
import { PaneDirective, PanesDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
export default class App extends React.Component {
  leftContent(data) {
    return (<div>
      <div className="content">
        <h3>Grid </h3>
        The ASP.NET DataGrid control, or DataTable is a feature-rich
        control used to display data in a tabular format.
      </div>
    </div>);
  }
  middleContent(data) {
    return (<div>
      <div className="content">
        <h3>Schedule </h3>
        The ASP.NET Scheduler, a.k.a. event calendar, facilitates almost
        all calendar features, thus allowing users to manage their time efficiently.
      </div>
    </div>);
  }
  rightContent(data) {
    return (<div>
      <div className="content">
        <h3>Chart </h3>
        ASP.NET charts, a well-crafted easy-to-use charting package, is
        used to add beautiful charts in web and mobile applications.
      </div>
    </div>);
  }
  render() {
    return (<div className="App">
      <SplitterComponent id="percentage" height="200px" width='600px'>
        <PanesDirective>
          <PaneDirective size='200px' content={this.leftContent}
resizable={false}/>
          <PaneDirective size='200px' content={this.middleContent}/>
          <PaneDirective size='200px' content={this.rightContent}/>
        </PanesDirective>
      </SplitterComponent>
    </div>);
  }
}
```

## APP.TSX

```
import { PaneDirective, PanesDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
export default class App extends React.Component<{}, {}> {
  public leftContent(data: any): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>Grid </h3>

```

```

        The ASP.NET DataGrid control, or DataTable is a feature-rich
        control used to display data in a tabular format.
    </div>
</div>
);
}
public middleContent(data: any): JSX.Element {
    return (
        <div>
            <div className="content">
                <h3>Schedule </h3>
                The ASP.NET Scheduler, a.k.a. event calendar, facilitates almost
                all calendar features, thus allowing users to manage their time efficiently.
            </div>
        </div>
    );
}
public rightContent(data: any): JSX.Element {
    return (
        <div>
            <div className="content">
                <h3>Chart </h3>
                ASP.NET charts, a well-crafted easy-to-use charting package, is
                used to add beautiful charts in web and mobile applications.
            </div>
        </div>
    );
}
public render() {
    return (<div className="App">
        <SplitterComponent id="percentage" height="200px" width='600px'>
            <PanelsDirective>
                <PaneDirective size='200px' content = {this.leftContent as any}
                resizable={false} />
                <PaneDirective size='200px' content = {this.middleContent as any} />
                <PaneDirective size='200px' content = {this.rightContent as any} />
            </PanelsDirective>
        </SplitterComponent>
    </div>);
}
}

```

## [Functional-component]

### APP.JSX

```

import { PaneDirective, PanelsDirective, SplitterComponent } from
'@syncfusion/ej2-react-layouts';
import * as React from "react";
function App() {
    function leftContent(data) {
        return (<div>
            <div className="content">
                <h3>Grid </h3>
                The ASP.NET DataGrid control, or DataTable is a feature-rich
                control used to display data in a tabular format.
            </div>
        </div>
    );
    }
}

```

```

    </div>
  </div>);
}
function middleContent(data) {
  return (<div>
    <div className="content">
      <h3>Schedule </h3>
      The ASP.NET Scheduler, a.k.a. event calendar, facilitates almost
      all calendar features, thus allowing users to manage their time efficiently.
    </div>
  </div>);
}
function rightContent(data) {
  return (<div>
    <div className="content">
      <h3>Chart </h3>
      ASP.NET charts, a well-crafted easy-to-use charting package, is
      used to add beautiful charts in web and mobile applications.
    </div>
  </div>);
}
return (<div className="App">
  <SplitterComponent id="percentage" height="200px" width='600px'>
    <PanelsDirective>
      <PanelDirective size='200px' content={leftContent}
        resizable={false}/>
      <PanelDirective size='200px' content={middleContent}/>
      <PanelDirective size='200px' content={rightContent}/>
    </PanelsDirective>
  </SplitterComponent>
  </div>);
}
export default App;

```

### APP.TSX

```

import { PanelDirective, PanelsDirective, SplitterComponent } from
  '@syncfusion/ej2-react-layouts';
import * as React from "react";
function App () {
  function leftContent(data: any): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>Grid </h3>
          The ASP.NET DataGrid control, or DataTable is a feature-rich
          control used to display data in a tabular format.
        </div>
      </div>
    );
  }
  function middleContent(data: any): JSX.Element {
    return (
      <div>
        <div className="content">
          <h3>Schedule </h3>

```

```

        The ASP.NET Scheduler, a.k.a. event calendar, facilitates almost
        all calendar features, thus allowing users to manage their time efficiently.
    </div>
</div>
);
}
function rightContent(data: any): JSX.Element {
    return (
        <div>
            <div className="content">
                <h3>Chart </h3>
                ASP.NET charts, a well-crafted easy-to-use charting package, is
                used to add beautiful charts in web and mobile applications.
            </div>
        </div>
    );
}
return (<div className="App">
    <SplitterComponent id="percentage" height="200px" width='600px'>
        <PanelsDirective>
            <PanelDirective size='200px' content = {leftContent as any}
resizable={false} />
            <PanelDirective size='200px' content = {middleContent as any} />
            <PanelDirective size='200px' content = {rightContent as any} />
        </PanelsDirective>
    </SplitterComponent>
</div>);
}
export default App;

```

## Different layouts in React Splitter component

By using splitter control, you can create the different layouts with multiple and nested panes.

### Code editor style layout

Create the element with two child to render the outer splitter and create the inner splitter from first pane of vertical splitter.

#### [Class-component]

```
`javascript
```

```
import { Splitter } from '@syncfusion/ej2-layouts';
```

```
import { PanelDirective, PanelsDirective, SplitterComponent } from '@syncfusion/ej2-react-layouts';
```

```
import * as React from "react";
```

```
class App extends React.Component {
```

```
    public splitterInstance: Splitter;
```

```
    public paneSize1 = "53%";
```

```
    public minimumSize1 = "30%";
```

```
    public content1 = `<div class="splitter-content">
```

```
<h3 class="h3">Preview of sample</h3>
```



```

<div class="splitter-image">
  
</div>
</div>`;

public onCreate(e: any): void {
  (this.splitterInstance as any).element.querySelector('.e-pane').setAttribute('id','splitter1');
  // Initialize Splitter component
  const splitterObj = new Splitter({
    height: '240px',
    paneSettings: [
      { content: `<div class="splitter-content">
        <h3 class="h3">HTML</h3>
        <div class="code-preview">
          <<span>!DOCTYPE html</span>
          <div><<span>html</span></div>
          <div><<span>body</span></div>
          <<span>div</span> id="custom-image">
            <div style="margin-left: 5px"><<span>img</span> src="src/albert.png"></div>
            <div><<span>div</span>></div>
            <div><<span>/body</span></div>
            <div><<span>/html</span></div>
          </div>
        </div>`,
        min: '23%', size: '29%' },
      { content: `<div class="splitter-content">
        <h3 class="h3">CSS</h3>
        <div class="code-preview">
          <span>img {</span>
            <div id="code-text">margin:<span>0 auto;</span></div>
            <div id="code-text">display:<span>flex;</span></div>
            <div id="code-text">height:<span>70px;</span></div>
            <span> }</span>

```

```

</div>
</div>`,min: '15%', size: '20%' },
{ content: `<div class="splitter-content">
<h3 class="h3">JavaScript</h3>
<div class="code-preview">
<span>var</span> image = document.getElementById("custom-image");
<div>image.addEventListener("click", function() {</div>
<div style="padding-left: 20px;"> // Code block for click action</div>
<span> }</span>
</div>
</div>`,
min: '35%', size: '35%' }
],
width: '100%'
});
splitterObj.appendTo('#splitter1');
}
public render(): JSX.Element {
return (
<div id="target" className="control-section code-editor" >
<SplitterComponent id="splitter2" height="400px" orientation="Vertical" ref={{(splitter) => {
this.splitterInstance = splitter! }} created={this.onCreate = this.onCreate.bind(this)}}>
<PanelsDirective>
<PanelDirective />
<PanelDirective size={this.paneSize1} min={this.minimumSize1} content={this.content1}/>
</PanelsDirective>
</SplitterComponent>
</div>
);
}
}
export default App;
`

```

**[Functional-component]**

```

`javascript
import { Splitter } from '@syncfusion/ej2-layouts';
import { PaneDirective, PanesDirective, SplitterComponent } from '@syncfusion/ej2-react-layouts';
import * as React from "react";

function App () {
  let splitterInstance: Splitter;
  let paneSize1 = "53%";
  let minimumSize1 = "30%";
  let content1 = `<div class="splitter-content">
<h3 class="h3">Preview of sample</h3>
<div class="splitter-image">

</div>
</div>`;
  function onCreate(e: any): void {
    (splitterInstance as any).element.querySelector('.e-pane').setAttribute('id','splitter1');
    // Initialize Splitter component
    const splitterObj = new Splitter({
      height: '240px',
      paneSettings: [
        { content:`<div class="splitter-content">
<h3 class="h3">HTML</h3>
<div class="code-preview">
<<span>!DOCTYPE html</span>
<div><<span>html</span></div>
<div><<span>body</span></div>
<<span>div</span> id="custom-image">
<div style="margin-left: 5px"><<span>img</span> src="src/albert.png"></div>
<div><<span>div</span></div>
<div><<span>/body</span></div>
<div><<span>/html</span></div>

```

```

</div>
</div>`,
min: '23%', size: '29%' },
{ content: `<div class="splitter-content">
<h3 class="h3">CSS</h3>
<div class="code-preview">
<span>img {</span>
<div id="code-text">margin:<span>0 auto;</span></div>
<div id="code-text">display:<span>flex;</span></div>
<div id="code-text">height:<span>70px;</span></div>
<span> }</span>
</div>
</div>`,min: '15%', size: '20%' },
{ content: `<div class="splitter-content">
<h3 class="h3">JavaScript</h3>
<div class="code-preview">
<span>var</span> image = document.getElementById("custom-image");
<div>image.addEventListener("click", function() {</div>
<div style="padding-left: 20px;">// Code block for click action</div>
<span> }</span>
</div>
</div>`,
min: '35%', size: '35%' }
],
width: '100%'
});
splitterObj.appendTo('#splitter1');
}
return (
<div id="target" className="control-section code-editor" >
<SplitterComponent id="splitter2" height="400px" orientation="Vertical" ref={{(splitter) => {
splitterInstance = splitter! }} created={onCreate = onCreate.bind(this)}>
<PanelsDirective>

```

```

<PaneDirective />
<PaneDirective size={paneSize1} min={minimumSize1} content={content1}/>
</PanelsDirective>
</SplitterComponent>
</div>
);
}
export default App;
`

```

Once the above configurations has been completed, you will get the output like [this](#)

### Outlook style layout

Create the element with three panes and place the elements within the pane to render **treeview**, **listview** and **RTE**.

### [Class-component]

```

`javascript
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import { PaneDirective, PanelsDirective, SplitterComponent } from '@syncfusion/ej2-react-layouts';
import { ListViewComponent } from '@syncfusion/ej2-react-lists';
import { TreeViewComponent } from '@syncfusion/ej2-react-navigations';
import { HtmlEditor, Image, Inject, Link, QuickToolbar, RichTextEditorComponent, Toolbar, } from
 '@syncfusion/ej2-react-richtexteditor';
import * as React from "react";
class App extends React.Component <{}, {}, any> {
  public splitterInstance: SplitterComponent;
  public listViewObj: ListViewComponent;
  public rteObj: RichTextEditorComponent;
  public treeInstance: TreeViewComponent;
  public textBox1: TextBoxComponent;
  public textBox2: TextBoxComponent;
  public textBox3: TextBoxComponent;
  public dataSource: any = [
    { Name: 'Selma Tally', content: 'Apology marketing email',content2:'Hello Ananya Singleton', id: '1',
    order: 0 },
    { Name: 'Illa Russo', content: 'Annual conference', content2: 'Hi jeani Moresa', id: '4', order: 0 },

```

```

{ Name: 'Camden Macmellon', content: 'Reference request- Camden hester', content2: 'Hello Kerry Best', order: 0 },
{ Name: 'Garth Owen', content: 'Application for job Title', content2: 'Hello Illa Russo', id: '2', order: 0 },
{ Name: 'Ursula Patterson', content: 'Programmaer Position Applicant', content2: 'Hello Kerry best', id: '2', order: 0 }
];

public paneSize1 = "28%";
public minimumSize1 = "27%";
public content1 = "<div class='splitter-content'></div>";
public paneSize2 = "33%";
public minimumSize2 = "23%";
public content2 = "<div class='splitter-content'></div>";
public paneSize3 = "37%";
public minimumSize3 = "30%";
public content3 = "
To...    " + "

Cc...    " + "

Subject

" +

"<div class='forum'><div id='createpostholder'><div id='buttonSection'><button class='e-btn e-primary' id='send'>Send</button>" +

"<button class='e-btn' id='discard'>Discard</button></div></div></div></div></div>";

// Set customized list template
public listTemplate: string = '<div class="settings e-list-wrapper e-list-multi-line e-list-avatar">' +
'<span class="e-list-item-header">${Name}</span>' +
'<span class="e-list-content">${content}</span>' +
'</div>'

// Set customized group-header template
public grouptemplate: string = '<div class="e-list-wrapper"><span class="e-list-item-content"></span></div>';

// Map the appropriate columns to fields property
public listFields: object = { text: 'Name', groupBy: 'order' };
public mailBox: object[] = [

```

```

{ id: 1, name: 'Favorites', hasChild: true},
{ id: 2, pid: 1, name: 'Sales Reports', count: '4' },
{ id: 3, pid: 1, name: 'Sent Items'},
{ id: 4, pid: 1, name: 'Marketing Reports ', count: '6'},
{ id: 5, name: 'Andrew Fuller', hasChild: true, expanded: true },
{ id: 6, pid: 5, name: 'Inbox', selected: true , count: '20'},
{ id: 7, pid: 5, name: 'Drafts', count: '5'},
{ id: 15, pid: 5, name: 'Archive'},
{ id: 8, pid: 5, name: 'Deleted Items'},
{ id: 9, pid: 5, name: 'Sent Items'},
{ id: 10, pid: 5, name: 'Sales Reports' , count: '4'},
{ id: 11, pid: 5, name: 'Marketing Reports', count: '6' },
{ id: 12, pid: 5, name: 'Outbox'},
{ id: 13, pid: 5, name: 'Junk Email'},
{ id: 14, pid: 5, name: 'RSS Feed'},
{ id: 15, pid: 5, name: 'Trash' }
];

private treeFields: object = { dataSource: this.mailBox, id: 'id', parentID: 'pid', text: 'name', hasChildren:
'hasChild' };

public onCreate(e: any): void {

this.splitterInstance.element.querySelectorAll('.splitter-
content')[0].appendChild(this.treeInstance.element);

this.splitterInstance.element.querySelectorAll('.splitter-
content')[1].appendChild(this.listViewObj.element);

(this.splitterInstance as
any).element.querySelector('#createpostholder').insertBefore(this.rteObj.element, (this.splitterInstance
as any).element.querySelector('#createpostholder').childNodes[0]);

(this.splitterInstance as any).element.querySelector('#firstname-
target').appendChild(this.textBox1.element.parentElement);

(this.splitterInstance as any).element.querySelector('#lastname-
target').appendChild(this.textBox2.element.parentElement);

(this.splitterInstance as any).element.querySelector('#subject-
target').appendChild(this.textBox3.element.parentElement);

}

public nodeTemplate(dataSource: any): JSX.Element {

return(<div>

```

```

<div className="treeviewdiv">
  <div className="textcontent">
    <span className="treeName">{dataSource.name}</span>
  </div>
  { dataSource.count &&
    <div className="countcontainer">
      <span className="treeCount e-badge e-badge-primary" >{dataSource.count}</span>
    </div>
  }
</div>
</div>;
};

public onSplitterResize(): void {
  this.rteObj.refresh();
}

public render(): JSX.Element {
  return (
    <div id="target" className="control-section outlook-style" >
      <TextBoxComponent id="firstname" ref={{(textbox) => { this.textBox1 = textbox! }}}/>
      <TextBoxComponent id="lastname" ref={{(textbox) => { this.textBox2 = textbox! }}}/>
      <TextBoxComponent id="subject" ref={{(textbox) => { this.textBox3 = textbox! }}}/>
      <RichTextEditorComponent id="defaultRTE" ref={{(richtexteditor) => { this.rteObj = richtexteditor! }}
        height = "262px">
        <Inject services={{[HtmlEditor, Toolbar, Image, Link, QuickToolbar]}} />
      </RichTextEditorComponent>
      <TreeViewComponent id='splitter-tree' fields= {this.treeFields} nodeTemplate = {this.nodeTemplate}
        ref={{(tree) => { this.treeInstance = tree! }}}/>
      <ListViewComponent id='groupedList-split' fields = { this.listFields } className ='splitter-list' dataSource
        = {this.dataSource}
        cssClass = {'e-list-template'} groupTemplate = {this.grouptemplate} template = {this.listTemplate}
        ref={{(listview) => { this.listViewObj = listview! }}}/>
      <SplitterComponent id="splitter1" height="493px" width="100%" ref={{(splitter) => { this.splitterInstance
        = splitter! }} created={this.onCreate = this.onCreate.bind(this)}
        resizeStop={this.onSplitterResize=this.onSplitterResize.bind(this)}>
        <PanelsDirective>

```



```

<PaneDirective size={this.paneSize1} min={this.minimumSize1} content={this.content1}/>
<PaneDirective size={this.paneSize2} min={this.minimumSize2} content={this.content2}/>
<PaneDirective size={this.paneSize3} min={this.minimumSize3} content={this.content3}/>
</PanelsDirective>
</SplitterComponent>
</div>
);
}
}
export default App;
`

```

### [Functional-component]

```

`javascript
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import { PaneDirective, PanelsDirective, SplitterComponent } from '@syncfusion/ej2-react-layouts';
import { ListViewComponent } from '@syncfusion/ej2-react-lists';
import { TreeViewComponent } from '@syncfusion/ej2-react-navigations';
import { HtmlEditor, Image, Inject, Link, QuickToolbar, RichTextEditorComponent, Toolbar, } from
 '@syncfusion/ej2-react-richtexteditor';
import * as React from "react";
function App () {
let splitterInstance: SplitterComponent;
let listViewObj: ListViewComponent;
let rteObj: RichTextEditorComponent;
let treeInstance: TreeViewComponent;
let textBox1: TextBoxComponent;
let textBox2: TextBoxComponent;
let textBox3: TextBoxComponent;
let dataSource: any = [
{ Name: 'Selma Tally', content: 'Apology marketing email',content2:'Hello Ananya Singleton', id: '1',
order: 0 },
{ Name: 'Illa Russo', content: 'Annual conference', content2: 'Hi jeani Moresa', id: '4', order: 0 },
{ Name: 'Camden Macmellon', content: 'Reference request- Camden hester', content2: 'Hello Kerry
Best', order: 0 },

```

```

{ Name: 'Garth Owen', content: 'Application for job Title', content2:'Hello Illa Russo', id: '2', order: 0 },
{ Name: 'Ursula Patterson', content: 'Programmaer Position Applicant', content2: 'Hello Kerry best', id:
'2', order: 0 }
];
let paneSize1 = "28%";
let minimumSize1 = "27%";
let content1 = "<div class='splitter-content'></div>";
let paneSize2 = "33%";
let minimumSize2 = "23%";
let content2 = "<div class='splitter-content'></div>";
let paneSize3 = "37%";
let minimumSize3 = "30%";
let content3 = "
To...    " + "

Cc...    " + "

Subject

" +
"<div class='forum'><div id='createpostholder'><div id='buttonSection'><button class='e-btn e-primary'
id='send'>Send</button>" +
"<button class='e-btn' id='discard'>Discard</button></div></div></div></div></div>";
// Set customized list template
let listTemplate: string = '<div class="settings e-list-wrapper e-list-multi-line e-list-avatar">' +
'<span class="e-list-item-header">${Name}</span>' +
'<span class="e-list-content">${content}</span>' +
'</div>'
// Set customized group-header template
let grouptemplate: string = '<div class="e-list-wrapper"><span class="e-list-item-
content"></span></div>';
// Map the appropriate columns to fields property
let listFields: object = { text: 'Name', groupBy: 'order' };
let mailBox: object[] = [
{ id: 1, name: 'Favorites', hasChild: true},
{ id: 2, pid: 1, name: 'Sales Reports', count: '4' },

```

```

{ id: 3, pid: 1, name: 'Sent Items'},
{ id: 4, pid: 1, name: 'Marketing Reports ', count: '6'},
{ id: 5, name: 'Andrew Fuller', hasChild: true, expanded: true },
{ id: 6, pid: 5, name: 'Inbox', selected: true , count: '20'},
{ id: 7, pid: 5, name: 'Drafts', count: '5'},
{ id: 15, pid: 5, name: 'Archive'},
{ id: 8, pid: 5, name: 'Deleted Items'},
{ id: 9, pid: 5, name: 'Sent Items'},
{ id: 10, pid: 5, name: 'Sales Reports' , count: '4'},
{ id: 11, pid: 5, name: 'Marketing Reports', count: '6' },
{ id: 12, pid: 5, name: 'Outbox'},
{ id: 13, pid: 5, name: 'Junk Email'},
{ id: 14, pid: 5, name: 'RSS Feed'},
{ id: 15, pid: 5, name: 'Trash' }
];

let treeFields: object = { dataSource: mailBox, id: 'id', parentID: 'pid', text: 'name', hasChildren:
'hasChild' };

function onCreate(e: any): void {
splitterInstance.element.querySelectorAll('.splitter-content')[0].appendChild(treeInstance.element);
splitterInstance.element.querySelectorAll('.splitter-content')[1].appendChild(listViewObj.element);
(splitterInstance as any).element.querySelector('#createpostholder').insertBefore(rteObj.element,
(splitterInstance as any).element.querySelector('#createpostholder').childNodes[0]);

(splitterInstance as any).element.querySelector('#firstname-
target').appendChild(textBox1.element.parentElement);
(splitterInstance as any).element.querySelector('#lastname-
target').appendChild(textBox2.element.parentElement);
(splitterInstance as any).element.querySelector('#subject-
target').appendChild(textBox3.element.parentElement);
}

function nodeTemplate(dataSource: any): JSX.Element {
return(<div>
<div className="treeviewdiv">
<div className="textcontent">
<span className="treeName">{dataSource.name}</span>
</div>

```

```

{ dataSource.count &&
<div className="countcontainer">
<span className="treeCount e-badge e-badge-primary" >{dataSource.count}</span>
</div>
}
</div>
</div>;
};
function onSplitterResize(): void {
rteObj.refresh();
}
return (
<div id="target" className="control-section outlook-style" >
<TextBoxComponent id="firstname" ref={{(textbox) => { textBox1 = textbox! }}}/>
<TextBoxComponent id="lastname" ref={{(textbox) => { textBox2 = textbox! }}}/>
<TextBoxComponent id="subject" ref={{(textbox) => { textBox3 = textbox! }}}/>
<RichTextEditorComponent id="defaultRTE" ref={{(richtexteditor) => { rteObj = richtexteditor! }} height =
"262px">
<Inject services={{[HtmlEditor, Toolbar, Image, Link, QuickToolbar]}} />
</RichTextEditorComponent>
<TreeViewComponent id='splitter-tree' fields= {treeFields} nodeTemplate = {nodeTemplate} ref={{(tree)
=> { treeInstance = tree! }}}/>
<ListViewComponent id='groupedList-split' fields = { listFields } className ='splitter-list' dataSource =
{dataSource}
cssClass = {'e-list-template'} groupTemplate = {grouptemplate} template = {listTemplate} ref={{(listview)
=> { listViewObj = listview! }}}/>
<SplitterComponent id="splitter1" height="493px" width="100%" ref={{(splitter) => { splitterInstance =
splitter! }} created={onCreate = onCreate.bind(this)}
resizeStop={onSplitterResize=onSplitterResize.bind(this)}>
<PanelsDirective>
<PanelDirective size={paneSize1} min={minimumSize1} content={content1}/>
<PanelDirective size={paneSize2} min={minimumSize2} content={content2}/>
<PanelDirective size={paneSize3} min={minimumSize3} content={content3}/>
</PanelsDirective>
</SplitterComponent>

```

```

</div>
);
}
export default App;
`

```

Refer below dependent styles in top of your App.css file.

```

`css
@import '../node_modules/@syncfusion/ej2-base/styles/material.css';
@import '../node_modules/@syncfusion/ej2-layouts/styles/material.css';
@import '../node_modules/@syncfusion/ej2-inputs/styles/material.css';
@import '../node_modules/@syncfusion/ej2-lists/styles/material.css';
@import '../node_modules/@syncfusion/ej2-navigations/styles/material.css';
@import '../node_modules/@syncfusion/ej2-richtexteditor/styles/material.css';
@import '../node_modules/@syncfusion/ej2-buttons/styles/material.css';
`

```

Once the above configurations has been completed, you will get the output like [this](#).

See Also

- [Multiple panes in Splitter](#)

### Style in React Splitter component

The following content provides the exact CSS structure that can be used to modify the control's appearance based on the user preference.

#### Customizing the split bar

Use the following CSS to customize the split bar properties.

##### *Horizontal split bar*

```

`css
/ default split bar color /
.e-splitter .e-split-bar.e-split-bar-horizontal{
background: blue;
}
/ split bar color in hover and active state /
.e-splitter .e-split-bar.e-split-bar-horizontal.e-split-bar-hover,
.e-splitter .e-split-bar.e-split-bar-horizontal.e-split-bar-active {
background: green;
}
`

```

```
}
,
```

#### Vertical split bar

```
`css

/ default split bar color /

.e-splitter .e-split-bar.e-split-bar-vertical {
background: blue;
}

/ split bar color in hover and active state /

.e-splitter .e-split-bar.e-split-bar-vertical.e-split-bar-hover,
.e-splitter .e-split-bar.e-split-bar-vertical.e-split-bar-active {
background: green;
}
,
```

#### Customizing the split bar resize handle

Use the following CSS to customize the split bar resize handle.

#### Horizontal split bar resize handle

```
`css

/ default split bar resize handle color /

.e-splitter .e-split-bar.e-split-bar-horizontal .e-resize-handler {
color: rgba(20, 27, 233, 0.54);
}

/ default split bar resize handle color in hover and active state /

.e-splitter .e-split-bar.e-split-bar-horizontal.e-split-bar-hover .e-resize-handler,
.e-splitter .e-split-bar.e-split-bar-horizontal.e-split-bar-active .e-resize-handler {
color: green;
}
,
```

#### Vertical split bar resize handle

```
`css

/ default split bar resize handle color /

.e-splitter .e-split-bar.e-split-bar-vertical .e-resize-handler {
color: rgba(20, 27, 233, 0.54);
}
,
```

*/ default split bar resize handle color in hover and active state /*

```
.e-splitter .e-split-bar.e-split-bar-vertical.e-split-bar-hover .e-resize-handler,
.e-splitter .e-split-bar.e-split-bar-vertical.e-split-bar-active .e-resize-handler {
color: green;
}
`css
```

### Customizing the split bar arrows

Use the following CSS to customize the split bar arrows.

#### Horizontal split bar resize arrows

``css`

*/ split bar arrows /*

```
.e-splitter .e-split-bar.e-split-bar-horizontal.e-split-bar-hover .e-navigate-arrow.e-arrow-left::before, .e-
splitter .e-split-bar.e-split-bar-horizontal.e-split-bar-active .e-navigate-arrow.e-arrow-left::before, .e-
splitter .e-split-bar.e-split-bar-horizontal.e-split-bar-hover .e-navigate-arrow.e-arrow-left::after, .e-
splitter .e-split-bar.e-split-bar-horizontal.e-split-bar-active .e-navigate-arrow.e-arrow-left::after, .e-
splitter .e-split-bar.e-split-bar-horizontal.e-split-bar-hover .e-navigate-arrow.e-arrow-right::before, .e-
splitter .e-split-bar.e-split-bar-horizontal.e-split-bar-active .e-navigate-arrow.e-arrow-right::before, .e-
splitter .e-split-bar.e-split-bar-horizontal.e-split-bar-hover .e-navigate-arrow.e-arrow-right::after, .e-
splitter .e-split-bar.e-split-bar-horizontal.e-split-bar-active .e-navigate-arrow.e-arrow-right::after {
background-color: green;
}
`css
```

*/ split bar arrows - circular border /*

```
.e-splitter .e-split-bar.e-split-bar-horizontal.e-split-bar-hover .e-navigate-arrow.e-arrow-left, .e-splitter
.e-split-bar.e-split-bar-horizontal.e-split-bar-hover .e-navigate-arrow.e-arrow-right {
border-color: rgba(33, 227, 22, 0.5);
}
`css
```

#### Vertical split bar resize arrows

``css`

*/ split bar arrows /*

```
.e-splitter .e-split-bar.e-split-bar-vertical.e-split-bar-hover .e-navigate-arrow.e-arrow-up::before, .e-
splitter .e-split-bar.e-split-bar-vertical.e-split-bar-active .e-navigate-arrow.e-arrow-up::before, .e-splitter
.e-split-bar.e-split-bar-vertical.e-split-bar-hover .e-navigate-arrow.e-arrow-up::after, .e-splitter .e-split-
bar.e-split-bar-vertical.e-split-bar-active .e-navigate-arrow.e-arrow-up::after, .e-splitter .e-split-bar.e-
split-bar-vertical.e-split-bar-hover .e-navigate-arrow.e-arrow-down::before, .e-splitter .e-split-bar.e-
split-bar-vertical.e-split-bar-active .e-navigate-arrow.e-arrow-down::before, .e-splitter .e-split-bar.e-
split-bar-vertical.e-split-bar-hover .e-navigate-arrow.e-arrow-down::after, .e-splitter .e-split-bar.e-split-
bar-vertical.e-split-bar-active .e-navigate-arrow.e-arrow-down::after {
background-color: green;
}
`css
```

```

}

/ split bar arrows - circular border /

.e-splitter .e-split-bar.e-split-bar-vertical.e-split-bar-hover .e-navigate-arrow.e-arrow-up, .e-splitter .e-
split-bar.e-split-bar-vertical.e-split-bar-hover .e-navigate-arrow.e-arrow-down {
border-color: rgba(33, 227, 22, 0.5);
}
`css

```

### To hide the resize handle in Splitter

Use the following CSS to hide the resize handler in the split bar

#### Hide the horizontal split bar resize arrow

```

`css

.e-splitter .e-split-bar.e-split-bar-horizontal .e-resize-handler {
display: none;
}
`css

```

#### Hide the vertical split bar resize arrow

```

`css

.e-splitter .e-split-bar.e-split-bar-vertical .e-resize-handler {
display: none;
}
`css

```

## Globalization in React Splitter component

### RTL

Specifies the direction of the Splitter component using the enableRtl property. For writing systems that require it like Arabic, Hebrew, etc., the direction can be switched to right-to-left.

The following code shows how to enable RTL behavior.

### [Class-component]

#### APP.JSX

```

import { PaneDirective, PanesDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  render() {
    return (<div className="App">
      <SplitterComponent id="rtl" height="250px" enableRtl="true">
        <PanesDirective>
          <PaneDirective size='200px' content='Left pane' />
          <PaneDirective size='200px' content='Middle pane' />
          <PaneDirective size='200px' content='Right pane' />
        </PanesDirective>
      </SplitterComponent>
    </div>);
  }
}

```



```

    </SplitterComponent>
    </div>);
  }
}
export default App;

```

### APP.TSX

```

import { PaneDirective, PanesDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
class App extends React.Component {
  public render() {
    return (<div className="App">
      <SplitterComponent id="rtl" height="250px" enableRtl="true">
        <PanesDirective>
          <PaneDirective size='200px' content = 'Left pane' />
          <PaneDirective size='200px' content = 'Middle pane' />
          <PaneDirective size='200px' content = 'Right pane' />
        </PanesDirective>
      </SplitterComponent>
    </div>);
  }
}
export default App;

```

### [Functional-component]

### APP.JSX

```

import { PaneDirective, PanesDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
function App() {
  return (<div className="App">
    <SplitterComponent id="rtl" height="250px" enableRtl="true">
      <PanesDirective>
        <PaneDirective size='200px' content='Left pane' />
        <PaneDirective size='200px' content='Middle pane' />
        <PaneDirective size='200px' content='Right pane' />
      </PanesDirective>
    </SplitterComponent>
  </div>);
}
export default App;

```

### APP.TSX

```

import { PaneDirective, PanesDirective, SplitterComponent } from
 '@syncfusion/ej2-react-layouts';
import * as React from "react";
function App () {
  return (
    <div className="App">
      <SplitterComponent id="rtl" height="250px" enableRtl="true">

```

```

    <PanelsDirective>
      <PanelDirective size='200px' content = 'Left pane' />
      <PanelDirective size='200px' content = 'Middle pane' />
      <PanelDirective size='200px' content = 'Right pane' />
    </PanelsDirective>
  </SplitterComponent>
</div>
);
}
export default App;

```

See Also

- [Migration from Essential JS 1](#)

## Accessibility in React Splitter component

The Splitter component followed the accessibility guidelines and standards, including [ADA, Section 508](#), [WCAG 2.2](#) standards, and [WCAG roles](#) that are commonly used to evaluate accessibility.

The accessibility compliance for the Splitter component is outlined below.

| Accessibility Criteria | Compatibility |

| -- | -- |

| [WCAG 2.2 Support](#) |  |

| [Section 508 Support](#) |  |

| [Screen Reader Support](#) |  |

| [Right-To-Left Support](#) |  |

| [Color Contrast](#) |  |

| [Mobile Device Support](#) |  |

| [Keyboard Navigation Support](#) |  |

| [Accessibility Checker Validation](#) |  |

| [Axe-core Accessibility Validation](#) |  |

<style>

.post .post-content img {

display: inline-block;

```
margin: 0.5em 0;
```

```
}
```

```
</style>
```

```
<div> - All
features of the component meet the requirement.</div>
```

```
<div> - Some features of the component do not meet the requirement.</div>
```

```
<div> - The component does not meet the requirement.</div>
```

### Keyboard interaction

You can use the following key shortcuts to access the splitter without interruptions:

| **Keyboard shortcuts** | **Actions** |

| --- | --- |

| Tab | Helps in focusing the splitter on the page and switching between the consecutive splitter bars. |

| Shift + Tab | Helps in focusing the previous splitter bar element on the splitter. |

| Right arrow | Helps in moving the active horizontal orientated splitter bar to its Right side. |

| Left arrow | Helps in moving the active horizontal orientated splitter bar to its Left side. |

| Up arrow | Helps in moving the active vertical orientated splitter bar to its Up side. |

| Down arrow | Helps in moving the active vertical orientated splitter bar to its Down side. |

| Enter | Helps to toggle between expand and collapse actions of the splitter bar when it is active. |

### Ensuring accessibility

The Splitter component's accessibility levels are ensured through an [accessibility-checker](#) and [axe-core](#) software tools during automated testing.

The accessibility compliance of the Splitter component is shown in the following sample. Open the [sample](#) in a new window to evaluate the accessibility of the Splitter component with accessibility tools.

See also

- [Accessibility in Syncfusion React components](#)

### Ej1 api migration in React Splitter component

This article describes the API migration process of Splitter component from Essential JS 1 to Essential JS 2.

### Common

```
{% raw %}
```

| **Behavior** | **API in Essential JS 1** | **API in Essential JS 2** |

| --- | --- | --- |

| Adding custom class | **Property:** *cssClass* `<br /><br /> <EJ.Splitter id="splitter" cssClass="customClass"><EJ.Splitter> | Property: cssClass <br /><SplitterComponent cssClass="customClass"></SplitterComponent> <br /> |`

| Adjusting Height | **Property:** *height* `<br /><br /> <EJ.Splitter id="splitter" height="100%"><EJ.Splitter> | Property: height <br /> <SplitterComponent height="100%"></SplitterComponent> |`

| Adjusting Width | **Property:** *width* `<br /><br /> <EJ.Splitter id="splitter" width="600"><EJ.Splitter> | Property: width <br /><br /> <SplitterComponent width="100%"></SplitterComponent> |`

| Orientation | **Property:** *orientation* `<br /><br /> <EJ.Splitter id="splitter" orientation={ej.Orientation.Vertical}><EJ.Splitter><br /> | Property: orientation <br /> <br /> <SplitterComponent orientation="Vertical"></SplitterComponent> |`

| Separator Size | Not Available | **Property:** *separatorSize* `<br /> <br /> <SplitterComponent separatorSize={4}></SplitterComponent> |`

| Adding HTML attributes | **Property:** *htmlAttributes* `<br /><br /> <EJ.Splitter id="splitter" htmlAttributes={htmlAttributes}><EJ.Splitter><br />var htmlAttributes = { class: "my-class", style: "border: 1px solid red"}<br /> | Not Available |`

| Customize expand/collapse icons | **Property:** `<br /> expanderTemplate <br /><br /> <EJ.Splitter id="splitter" expanderTemplate = ''><EJ.Splitter> | Not Available |`

| Make control flexible for mobile view | **Property:** *isResponsive* `<br /><br /> <EJ.Splitter id="splitter" isResponsive={true}><EJ.Splitter> | By default, Splitter works with mobile mode. |`

| Refresh the Splitter | **Method:** *refresh()* `<br /><br /> <EJ.Splitter id="splitter"><EJ.Splitter><br />$("#splitter").ejSplitter("refresh")<br /> | Method: refresh() <br /><br /> <SplitterComponent id="splitter" ref={splitter=>this.splitterInstance = splitter}></SplitterComponent><br /> constructor(props: {}) { this.splitterInstance.refresh(); } <br /> |`

| Destroy the Control | **Method:** *destroy()* `<br /><br /> <EJ.Splitter id="splitter"><EJ.Splitter><br />$("#splitter").ejSplitter("destroy")<br /> | Method: destroy() <br /> <SplitterComponent id="splitter" ref={splitter=>this.splitterInstance = splitter}></SplitterComponent><br /> constructor(props: {}) { this.splitterInstance.destroy(); } <br /> |`

| Event triggers after the Splitter created successfully | **Event:** *create* `<br /><br /> <EJ.Splitter id="splitter" create={create}><EJ.Splitter><br />function create(args) {}<br /> | Event: created <br /><br /> <SplitterComponent id="splitter" created={this.onCreated.bind(this)}></SplitterComponent><br />private onCreated() {}<br /> |`

| Event triggers when Splitter has been destroyed | **Event:** *destroy* `<br /><br /> <EJ.Splitter id="splitter" destroy={destroy}><EJ.Splitter> <br />function destroy(args) {}<br /> | Not Available |`

### Accessibility and Localization

| **Behavior** | **API in Essential JS 1** | **API in Essential JS 2** |

| --- | --- | --- |

| Keyboard Navigation | **Property:** `allowKeyboardNavigation` `<br /><br />` `<EJ.Splitter id="splitter" allowKeyboardNavigation={true}><EJ.Splitter>` | No separate property for enable/disable keyboard navigation. Its enabled by default. |

| Right to Left | **Property:** `enableRTL` `<br /><br />` `<EJ.Splitter id="splitter" enableRTL={false}><EJ.Splitter>` | **Property:** `enableRtl` `<br /><br />` `<SplitterComponent enableRtl={true}></SplitterComponent>` |

### Control State

| **Behavior** | **API in Essential JS 1** | **API in Essential JS 2** |

| --- | --- | --- |

| Enable/Disable the control | Not Available | **Property:** `enabled` `<br /><br />` `<SplitterComponent enabled={true}></SplitterComponent>` |

### State Maintenance

| **Behavior** | **API in Essential JS 1** | **API in Essential JS 2** |

| --- | --- | --- |

| Save the model value in local storage or cookies | Not Available | **Property:** `enablePersistence` `<br /><br />` `<SplitterComponent enablePersistence={true}></SplitterComponent>` |

### Pane Properties

| **Behavior** | **API in Essential JS 1** | **API in Essential JS 2** |

| --- | --- | --- |

| Default | **Property:** `properties` `<br /><br />` `<EJ.Splitter id="splitter" properties={property}><EJ.Splitter><br />` `var property = []; <br />` | **Property:** `paneSettings` `<br />` `<SplitterComponent>` `<PanelsDirective>` `<PaneDirective />` `</PanelsDirective></SplitterComponent><br />` |

| Pane Content | Not Available | **Property:** `content` `<br />` `<SplitterComponent>` `<PanelsDirective>` `<PaneDirective content="<div>First Pane Content</div>" />` `</PanelsDirective>` `</SplitterComponent><br />` |

| Change the size of the pane | **Property:** `paneSize` `<br />` `<EJ.Splitter id="splitter" properties={property}><EJ.Splitter>` `<br />` `var property = [{paneSize: "30px"}];<br />` | **Property:** `size` `<br />` `<SplitterComponent>` `<PanelsDirective>` `<PaneDirective size="25%" /></PanelsDirective></SplitterComponent><br />` |

| Minimum pane size | **Property:** `minSize` `<br />` `<EJ.Splitter id="splitter" properties={property}><EJ.Splitter><br />` `var property = [{minSize: 30}];<br />` | **Property:** `min` `<br /><br />` `<SplitterComponent>` `<PanelsDirective>` `<PaneDirective min="60px" /></PanelsDirective></SplitterComponent><br />` |

| Maximum pane size | **Property:** `maxSize` `<br /><br />` `<EJ.Splitter id="splitter" properties={property}><EJ.Splitter><br />` `var property = [{maxSize: 30}];<br />` | **Property:** `max` `<br`

```
</><br /><SplitterComponent><PanelsDirective><PanelDirective
max="60px"/></PanelsDirective></SplitterComponent><br /> |
```

| Enable/Disable the Pane Resizable behavior | **Property:** *resizable* <br /><br /><EJ.Splitter id="splitter" properties={property}><EJ.Splitter><br />var property = [{resizable: false }];<br /> | **Property:** *resizable* <br /><br /><SplitterComponent><PanelsDirective><PanelDirective resizable={false}/></PanelsDirective></SplitterComponent><br /> |

| Collapsible | **Property:** *collapsible* <br /><br /><EJ.Splitter id="splitter" properties={property}><EJ.Splitter><br />var property = [{collapsible: true }];<br /> | **Property:** *collapsible* <br /><br /><SplitterComponent><PanelsDirective><PanelDirective collapsible={true}/></PanelsDirective></SplitterComponent><br /> |

| Expandable | **Property:** *expandable* <br /><br /><EJ.Splitter id="splitter" properties={property}><EJ.Splitter> <br />var property = [{expandable: true }];<br /> | Not Available |

| Collapsed | Not Available | **Property:** *collapsed* <br /><br /><SplitterComponent><PanelsDirective><PanelDirective collapsed={true}/></PanelsDirective></SplitterComponent><br /> |

| Add Pane | **Method:** *addItem()* <br /><br /><EJ.Splitter id="splitter"><EJ.Splitter><br />\$("#splitter").ejSplitter("addItem", "New Pane 0", {paneSize:20, minSize:20, maxSize: 100}, 0);<br /> | **Method:** *addPane()* <br /><br /><SplitterComponent id="splitter" ref={splitter=>this.splitterInstance = splitter}/></SplitterComponent><br />constructor(props: {}) {this.splitterInstance. addPane({size: "25%", content: "Pane"}, 0)}<br /> |

| Remove Pane | **Method:** *removeItem()* <br /><br /><EJ.Splitter id="splitter"><EJ.Splitter><br />\$("#splitter").ejSplitter("removeItem", 0);<br /> | **Method:** *removePane()* <br /><br /><SplitterComponent id="splitter" ref={splitter=>this.splitterInstance = splitter}/></SplitterComponent><br />constructor(props: {}) { this.splitterInstance. removePane(0);}<br /> |

| Collapse Pane | **Method:** *collapse()* <br /><br /><EJ.Splitter id="splitter"><EJ.Splitter><br />\$("#splitter").ejSplitter("collapse", 0);<br /> | **Method:** *collapse()* <br /><br /><SplitterComponent id="splitter" ref={splitter=>this.splitterInstance = splitter}/></SplitterComponent><br />constructor(props: {}) {this.splitterInstance.collapse(0);}<br /> |

| Expand Pane | **Method:** *expand()* <br /><br /><EJ.Splitter id="splitter"><EJ.Splitter><br />\$("#splitter").ejSplitter("expand", 0);<br /> | **Method:** *expand()* <br /><br /><SplitterComponent id="splitter" ref={splitter=>this.splitterInstance = splitter}/></SplitterComponent><br />constructor(props: {}) { this.splitterInstance.expand(0);}<br /> |

| Event triggers when before panes get expanded/collapsed | **Event:** *beforeExpandCollapse* <br /><br /><EJ.Splitter id="splitter" beforeExpandCollapse={ beforeExpandCollapse }><EJ.Splitter><br />function beforeExpandCollapse (args) {}<br /> | **Event:** *beforeExpand* <br /><br /><SplitterComponent id="splitter" beforeExpand ={this. onBeforeExpand.bind(this)}/></SplitterComponent><br />private onBeforeExpand () {}<br /> **Event:** *beforeCollapse* <br /><SplitterComponent id="splitter" beforeCollapse ={this. onBeforeCollapse.bind(this)}/></SplitterComponent><br />private onBeforeCollapse() {}<br /> |

| Event triggers when after panes get expanded/collapsed | **Event:** *expandCollapse* `<br /><br /><EJ.Splitter id="splitter" expandCollapse={expandCollapse}><EJ.Splitter><br />function expandCollapse (args) { }<br /> | Event: expand <br /><br /><SplitterComponent id="splitter" expand={this.onExpand.bind(this)}></SplitterComponent><br />private onExpand () { }<br /> Event: collapse <br /><SplitterComponent id="splitter" collapse={this.onCollapse.bind(this)}></SplitterComponent><br />private onCollapse () { }<br /> |`

| Event triggers when Resizing the pane | **Event:** *resize* `<br /><br /><EJ.Splitter id="splitter" resize={resize}><EJ.Splitter><br />function resize (args) { }<br /> | Event: resizing <br /><br /><SplitterComponent id="splitter" resizing={this.onResizing.bind(this)}></SplitterComponent><br />private onResizing() { }<br /> |`

| Event triggers when pane is started to resize | Not Available | **Event:** *resizeStart* `<br /><br /><SplitterComponent id="splitter" resizeStart={this.onResizeStart.bind(this)}></SplitterComponent><br />private onResizeStart () { }<br /> |`

| Event triggers when pane is stopped to resize | Not Available | **Event:** *resizeStop* `<br /><br /><SplitterComponent id="splitter" resizeStop={this.onResizeStop.bind(this)}></SplitterComponent><br />private onResizeStop () { }<br /> |`

| Event triggers when click template icon | **Event:** *clickOnExpander* `<br /><br /><EJ.Splitter id="splitter" clickOnExpander={clickOnExpander}><EJ.Splitter><br />function clickOnExpander (args) { }<br /> | Not Available |`

### Animation

| **Behavior** | **API in Essential JS 1** | **API in Essential JS 2** |

| --- | --- | --- |

| EnableAnimation | **Property:** *enableAnimation* `<br /><br /><EJ.Splitter id="splitter" enableAnimation={true}><EJ.Splitter><br /> &#160; | Not Available |`

| AnimationSpeed | **Property:** *animationSpeed* `<br /><br /><EJ.Splitter id="splitter" animationSpeed={150}><EJ.Splitter> <br />&#160; | Not Available |`

{% endraw %}

## Spreadsheet

### Overview of the React Spreadsheet component

The React Spreadsheet is an user interactive component to organize and analyze data in tabular format with configuration options for customization. It will load data by importing an Excel/CSV file or from local and remote data sources such as JSON, RESTful services, OData services, and more. The populated data can be exported as Excel with xlsx, xls, CSV and PDF formats.

#### Key features

- [Data sources](#): Bind the Spreadsheet component with an array of objects or data from a web service using `DataManager`.
- [Virtualization](#): Provides the option to load large amount of data without performance degradation.

- [Selection](#): Provides the option to select a cell or range of cells.
- [Editing](#): Provides the option to dynamically edit a cell.
- [Formulas](#): Provides built-in calculation library with pre-defined formulas and named range support.
- [Clipboard](#): Provides the option to perform clipboard operations.
- [Cell formatting](#): Provides the option to customize the appearance of cells.
- [Number formatting](#): Provides the option to format the cell value.
- [Open](#): Provides the option to open Excel and CSV files in Spreadsheet.
- [Save](#): Provides the option to save Spreadsheet data as Excel, CSV, and PDF documents.
- [Sorting](#): Helps you to arrange the data to particular order in a selected range of cells.
- [Filtering](#): Helps you to view specific rows in the Spreadsheet by hiding the other rows.
- [Undo Redo](#): Provides the option to perform undo redo operations in Spreadsheet.
- [Collaborative editing](#): Provides the option for real time changes across multiple users in the Spreadsheet.
- [Hyperlink](#): Provides the option to navigate to web link or cell reference within the sheet or to other sheet in Spreadsheet.
- [Resize](#): Allows you to change the row height and column width. Auto fit the rows and columns based on its content.
- [Wrap text](#): Provides the option to display the large content as multiple lines in a single cell.
- [Data validation](#): Provides the option to validate edited values based on data validation rules defined for a cell or range of cells.
- [Find and replace](#): Provides the option to find the data and replace it across all sheets in Spreadsheet.
- [Protect sheet](#): Provides the option to restrict user actions like cell editing, row and column insertion, deletion, and resizing.
- [Borders](#): Provides the option to customize cell gridlines such as color and its style for enhanced UI.
- [Show/hide](#): Provides the option to show/hide rows, columns and sheets.
- [Insert/delete](#): Provides the option to insert/delete rows, columns and sheets.
- [Merge cells](#): Provides the option to combine two or more cells located in the same row or column into a single cell.
- [Conditional formatting](#): Provides the option to format a cell or range of cells based on conditions applied.
- [Autofill](#): Provides the option to fill or copy a series or pattern of values and formats into adjacent cells in any direction.
- [Clear](#): Provides the option to clear the content, formats, and hyperlinks applied to a cell or range of cells in a Spreadsheet.
- [Aggregates](#): Provides the option to check the sum, average, count, and more for the selected cells or range in the sheet.
- [Picture](#): Allows you to view, insert, and modify a picture in a Spreadsheet with customizing options.
- [Chart](#): Transforms your Spreadsheet data to an intuitive overview for better understanding and to make smart business decisions.
- [Freeze panes](#): Allows you to keep the specified rows and columns always visible at the top and left side of the sheet while scrolling through the sheet.
- [Password protection](#): Allows you to protect the workbook with a password.
- [Multi-line editing](#): Allows you to insert a line break between paragraphs of the text within a cell in a Spreadsheet.



- [Calculate range selection](#): Helps you to select a range or multiple ranges when editing a formula in a cell.
- [Right-to-left \(RTL\)](#): Aligns content in the Spreadsheet component from right to left.
- [Templates](#): Templates can be used to create custom user experiences in the Spreadsheet.
- [Globalization](#): Personalize the Spreadsheet component with different languages, as well as culture-specific number, date, and time formatting.
- [AccessibilityLink to the Video](#): Provides with built-in accessibility support which helps to access all the Spreadsheet component features through the keyboard, screen readers, or other assistive technology devices.

## Getting Started

This section explains the steps to create a simple Spreadsheet component in a React application.

To get start quickly with React Spreadsheet, you can check on this video:

### Dependencies

The following list of dependencies are required to use the Spreadsheet component in your application:

```
`js
|-- @syncfusion/ej2-react-spreadsheet
|-- @syncfusion/ej2-react-base
|-- @syncfusion/ej2-spreadsheet
|-- @syncfusion/ej2-base
|-- @syncfusion/ej2-dropdowns
|-- @syncfusion/ej2-navigations
|-- @syncfusion/ej2-grids
`,`
```

### Setup for Local Development

You can use [create-react-app](#) to setup the applications and run the following command to install create-react-app.

```
`
npm install -g create-react-app
`,`
```

Run the following command to setup the basic **React** samples.

```
`
create-react-app quickstart --scripts-version=react-scripts-ts
cd quickstart
npm install
`,`
```

### Adding Syncfusion packages

All the available Essential JS 2 packages are published in [npmjs.com](https://www.npmjs.com) public registry. To install Spreadsheet component use the following command.

```
npm install @syncfusion/ej2-react-spreadsheet --save
```

### Adding CSS reference

Add components style as given below in `src/App.css`.

```
`css
@import '../node_modules/@syncfusion/ej2-base/styles/material.css';
@import '../node_modules/@syncfusion/ej2-inputs/styles/material.css';
@import '../node_modules/@syncfusion/ej2-buttons/styles/material.css';
@import '../node_modules/@syncfusion/ej2-splitbuttons/styles/material.css';
@import '../node_modules/@syncfusion/ej2-lists/styles/material.css';
@import '../node_modules/@syncfusion/ej2-navigations/styles/material.css';
@import '../node_modules/@syncfusion/ej2-popups/styles/material.css';
@import '../node_modules/@syncfusion/ej2-dropdowns/styles/material.css';
@import '../node_modules/@syncfusion/ej2-grids/styles/material.css';
@import '../node_modules/@syncfusion/ej2-react-spreadsheet/styles/material.css';
```

To refer `App.css` in the application then import it in the `src/App.tsx` file.

### Adding Spreadsheet component

Now, you can import the spreadsheet component into your `src/App.tsx` file.

```
`ts
import * as React from 'react';
import { SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
import './App.css';
export default function App() {
  return (<SpreadsheetComponent/>);
}
```

### Run the application

The [create-react-app](https://create-react-app.dev/) will pre-configure the project to compile and run the application in browser. Use the following command to run the application.

npm start

The following example shows a basic spreadsheet component.

#### **APP.JSX**

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
export default function App() {
    return (<SpreadsheetComponent />);
}
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}
```

#### **APP.TSX**

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
export default function App() {
    return (<SpreadsheetComponent />);
}
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}
```

You can refer to our [React Spreadsheet](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Spreadsheet example](#) that shows you how to present and manipulate data.

See Also

- [Data Binding](#)
- [Open and Save](#)

### Creating a Next.js Application Using Syncfusion React Components

This section provides a step-by-step guide for setting up a Next.js application and integrating the Syncfusion React Spreadsheet component.

What is Next.js?

[Next.js](#) is a React framework that makes it easy to build fast, SEO-friendly, and user-friendly web applications. It provides features such as server-side rendering, automatic code splitting, routing, and API routes, making it an excellent choice for building modern web applications.

Prerequisites

Before getting started with the Next.js application, ensure the following prerequisites are met:

- [Node.js 18.17](#) or later.

- The application is compatible with macOS, Windows, and Linux operating systems.

### Create a Next.js application

To create a new **Next.js** application, use one of the commands that are specific to either NPM or Yarn.

#### **NPM**

```
npx create-next-app@latest
```

#### **YARN**

```
yarn create next-app
```

Using one of the above commands will lead you to set up additional configurations for the project as below:

1. Define the project name: Users can specify the name of the project directly. Let's specify the name of the project as **ej2-nextjs-spreadsheet**.

#### **CMD**

```
√ What is your project named? » ej2-nextjs-spreadsheet
```

2. Select the required packages.

#### **CMD**

```
√ What is your project named? ... ej2-nextjs-spreadsheet
√ Would you like to use TypeScript? ... No / `Yes`
√ Would you like to use ESLint? ... No / `Yes`
√ Would you like to use Tailwind CSS? ... `No` / Yes
√ Would you like to use `src/` directory? ... No / `Yes`
√ Would you like to use App Router? (recommended) ... No / `Yes`
√ Would you like to customize the default import alias? ... `No` / Yes
Creating a new Next.js app in D:\ej2-nextjs-spreadsheet.
```

3. Once complete the above mentioned steps to create **ej2-nextjs-spreadsheet**, navigate to the directory using the below command:

#### **CMD**

```
cd ej2-nextjs-spreadsheet
```

The application is ready to run with default settings. Now, let's add Syncfusion components to the project.

### Install Syncfusion React packages

Syncfusion React component packages are available at [npmjs.com](https://www.npmjs.com). To use Syncfusion React components in the project, install the corresponding npm package.

Here, the [React Spreadsheet component](#) is used in the project. To install the React Spreadsheet component, use the following command:

#### **NPM**

```
npm install @syncfusion/ej2-react-spreadsheet --save
```

## YARN

```
yarn add @syncfusion/ej2-react-spreadsheet
```

## Import Syncfusion CSS styles

Syncfusion React components come with [built-in themes](#), which are available in the installed packages. It's easy to adapt the Syncfusion React components to match the style of your application by referring to one of the built-in themes.

Import the **Material** theme into the **src/app/globals.css** file and removed the existing styles in that file, as shown below:

## GLOBALS.CSS

```
@import '../node_modules/@syncfusion/ej2-base/styles/material.css';
@import '../node_modules/@syncfusion/ej2-inputs/styles/material.css';
@import '../node_modules/@syncfusion/ej2-buttons/styles/material.css';
@import '../node_modules/@syncfusion/ej2-splitbuttons/styles/material.css';
@import '../node_modules/@syncfusion/ej2-lists/styles/material.css';
@import '../node_modules/@syncfusion/ej2-navigations/styles/material.css';
@import '../node_modules/@syncfusion/ej2-popups/styles/material.css';
@import '../node_modules/@syncfusion/ej2-dropdowns/styles/material.css';
@import '../node_modules/@syncfusion/ej2-grids/styles/material.css';
@import '../node_modules/@syncfusion/ej2-react-spreadsheet/styles/material.css';
```

To know more about built-in themes and CSS reference for individual components, refer to the [themes](#) section.

## Add Syncfusion React component

Follow the below steps to add the React Spreadsheet component to the Next.js project:

1. Before adding the Spreadsheet component to your markup, create a **datasource.tsx** file within the **src/app/** folder and add the Spreadsheet component data.

## DATASOURCE.TSX

```
export let defaultData: object[] = [
  {
    "Customer Name": "Romona Heaslip",
    "Model": "Taurus",
    "Color": "Aquamarine",
    "Payment Mode": "Debit Card",
    "Delivery Date": "07/11/2015",
    "Amount": "8529.22"
  },
  {
    "Customer Name": "Clare Batterton",
    "Model": "Sparrow",
    "Color": "Pink",
    "Payment Mode": "Cash On Delivery",
  }
]
```

```
"Delivery Date": "7/13/2016",
"Amount": "17866.19"
},
{
  "Customer Name": "Eamon Traise",
  "Model": "Grand Cherokee",
  "Color": "Blue",
  "Payment Mode": "Net Banking",
  "Delivery Date": "09/04/2015",
  "Amount": "13853.09"
},
{
  "Customer Name": "Julius Gorner",
  "Model": "GTO",
  "Color": "Aquamarine",
  "Payment Mode": "Credit Card",
  "Delivery Date": "12/15/2017",
  "Amount": "2338.74"
},
{
  "Customer Name": "Jenna Schoolfield",
  "Model": "LX",
  "Color": "Yellow",
  "Payment Mode": "Credit Card",
  "Delivery Date": "10/08/2014",
  "Amount": "9578.45"
},
{
  "Customer Name": "Marylynne Haring",
  "Model": "Catera",
  "Color": "Green",
  "Payment Mode": "Cash On Delivery",
  "Delivery Date": "7/01/2017",
  "Amount": "19141.62"
},
{
  "Customer Name": "Vilhelmina Leipelt",
  "Model": "7 Series",
  "Color": "Goldenrod",
  "Payment Mode": "Credit Card",
  "Delivery Date": "12/20/2015",
  "Amount": "6543.30"
},
{
  "Customer Name": "Barby Heisler",
  "Model": "Corvette",
  "Color": "Red",
  "Payment Mode": "Credit Card",
  "Delivery Date": "11/24/2014",
  "Amount": "13035.06"
},
{
  "Customer Name": "Karyn Boik",
  "Model": "Regal",
  "Color": "Indigo",
  "Payment Mode": "Debit Card",
  "Delivery Date": "05/12/2014",
```

```
"Amount": "18488.80"
},
{
  "Customer Name": "Jeanette Pamplin",
  "Model": "S4",
  "Color": "Fuscia",
  "Payment Mode": "Net Banking",
  "Delivery Date": "12/30/2014",
  "Amount": "12317.04"
},
{
  "Customer Name": "Cristi Espinos",
  "Model": "TL",
  "Color": "Aquamarine",
  "Payment Mode": "Credit Card",
  "Delivery Date": "12/18/2013",
  "Amount": "6230.13"
},
{
  "Customer Name": "Issy Humm",
  "Model": "Club Wagon",
  "Color": "Pink",
  "Payment Mode": "Cash On Delivery",
  "Delivery Date": "02/02/2015",
  "Amount": "9709.49"
},
{
  "Customer Name": "Tuesday Fautly",
  "Model": "V8 Vantage",
  "Color": "Crimson",
  "Payment Mode": "Debit Card",
  "Delivery Date": "11/19/2014",
  "Amount": "9766.10"
},
{
  "Customer Name": "Rosemaria Thomann",
  "Model": "Caravan",
  "Color": "Violet",
  "Payment Mode": "Net Banking",
  "Delivery Date": "02/08/2014",
  "Amount": "7685.49"
},
{
  "Customer Name": "Lyell Fuentez",
  "Model": "Bravada",
  "Color": "Violet",
  "Payment Mode": "Debit Card",
  "Delivery Date": "08/05/2016",
  "Amount": "18012.45"
},
{
  "Customer Name": "Raynell Layne",
  "Model": "Colorado",
  "Color": "Pink",
  "Payment Mode": "Credit Card",
  "Delivery Date": "05/30/2016",
  "Amount": "2785.49"
```

```
},
{
  "Customer Name": "Raye Whines",
  "Model": "4Runner",
  "Color": "Red",
  "Payment Mode": "Debit Card",
  "Delivery Date": "12/10/2016",
  "Amount": "9967.74"
},
{
  "Customer Name": "Virgina Aharoni",
  "Model": "TSX",
  "Color": "Pink",
  "Payment Mode": "Cash On Delivery",
  "Delivery Date": "10/23/2014",
  "Amount": "5584.33"
},
{
  "Customer Name": "Peta Cheshir",
  "Model": "Pathfinder",
  "Color": "Red",
  "Payment Mode": "Net Banking",
  "Delivery Date": "12/24/2015",
  "Amount": "5286.53"
},
{
  "Customer Name": "Jule Urion",
  "Model": "Charger",
  "Color": "Violet",
  "Payment Mode": "Debit Card",
  "Delivery Date": "11/20/2013",
  "Amount": "13511.91"
},
{
  "Customer Name": "Lew Gilyatt",
  "Model": "Bonneville",
  "Color": "Crimson",
  "Payment Mode": "Credit Card",
  "Delivery Date": "11/19/2013",
  "Amount": "6498.19"
},
{
  "Customer Name": "Jobey Fortun",
  "Model": "B-Series",
  "Color": "Blue",
  "Payment Mode": "Net Banking",
  "Delivery Date": "10/30/2014",
  "Amount": "10359.67"
},
{
  "Customer Name": "Blondie Crump",
  "Model": "Voyager",
  "Color": "Turquoise",
  "Payment Mode": "Credit Card",
  "Delivery Date": "04/06/2018",
  "Amount": "8118.39"
},
}
```



```
{
  "Customer Name": "Florentia Binns",
  "Model": "Grand Prix",
  "Color": "Orange",
  "Payment Mode": "Cash On Delivery",
  "Delivery Date": "10/13/2016",
  "Amount": "10204.37"
},
{
  "Customer Name": "Jaquelin Galtone",
  "Model": "Sunbird",
  "Color": "Red",
  "Payment Mode": "Net Banking",
  "Delivery Date": "10/22/2013",
  "Amount": "6528.06"
},
{
  "Customer Name": "Hakeem Easseby",
  "Model": "Mirage",
  "Color": "Crimson",
  "Payment Mode": "Debit Card",
  "Delivery Date": "9/12/2014",
  "Amount": "5619.25"
},
{
  "Customer Name": "Nickolaus Gidman",
  "Model": "XK",
  "Color": "Orange",
  "Payment Mode": "Debit Card",
  "Delivery Date": "05/12/2016",
  "Amount": "5091.43"
},
{
  "Customer Name": "Jenine Iglesia",
  "Model": "Accord",
  "Color": "Orange",
  "Payment Mode": "Debit Card",
  "Delivery Date": "09/03/2018",
  "Amount": "14566.08"
},
{
  "Customer Name": "Fax Witherspoon",
  "Model": "Range Rover Sport",
  "Color": "Orange",
  "Payment Mode": "Credit Card",
  "Delivery Date": "2/22/2018",
  "Amount": "5284.87"
}
];
```

2. Then, import and define the Spreadsheet component in the **src/app/page.tsx** file, as shown below:

#### **PAGE.TSX**

```
'use client'
```

```

import { SpreadsheetComponent, SheetsDirective, SheetDirective,
ColumnsDirective, RangesDirective, RangeDirective, RowsDirective,
RowDirective, CellsDirective, CellDirective, CellStyleModel, ColumnDirective
} from '@syncfusion/ej2-react-spreadsheet';
import { defaultData } from './datasource';
export default function Home(this: any) {
let spreadsheet!: SpreadsheetComponent;
const boldRight: CellStyleModel = { fontWeight: 'bold', textAlign: 'right'
};
const bold: CellStyleModel = { fontWeight: 'bold' };
function onCreated(): void {
spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center',
verticalAlign: 'middle' }, 'A1:F1');
spreadsheet.numberFormat('$#,##0.00', 'F2:F31');
}
return (
<>
<h2>Syncfusion React Spreadsheet Component</h2>
<SpreadsheetComponent
openUrl='https://services.syncfusion.com/react/production/api/spreadsheet/op
en'
saveUrl='https://services.syncfusion.com/react/production/api/spreadsheet/sa
ve'
ref={(Obj) => { spreadsheet = Obj as SpreadsheetComponent }}
created={onCreated.bind(this)} >
<SheetsDirective>
<SheetDirective name="Car Sales Report">
<RangesDirective>
<RangeDirective dataSource={defaultData}></RangeDirective>
</RangesDirective>
<RowsDirective>
<RowDirective index={30}>
<CellsDirective>
<CellDirective index={4} value="Total Amount:"
style={boldRight}></CellDirective>
<CellDirective formula="=SUM(F2:F30)" style={bold}></CellDirective>
</CellsDirective>
</RowDirective>
</RowsDirective>
<ColumnsDirective>
<ColumnDirective width={180}></ColumnDirective>
<ColumnDirective width={130}></ColumnDirective>
<ColumnDirective width={130}></ColumnDirective>
<ColumnDirective width={180}></ColumnDirective>
<ColumnDirective width={130}></ColumnDirective>
<ColumnDirective width={120}></ColumnDirective>
</ColumnsDirective>
</SheetDirective>
</SheetsDirective>
</SpreadsheetComponent>
</>
)
}

```

### Run the application

To run the application, use the following command:

#### **NPM**

```
npm run dev
```

#### **YARN**

```
yarn run dev
```

To learn more about the functionality of the Spreadsheet component, refer to the [documentation](#).

[View the NEXT.js spreadsheet sample in the GitHub repository.](#)

### Data binding in React Spreadsheet component

The Spreadsheet uses [DataManager](#), which supports both RESTful JSON data services and local JavaScript object array binding to a range. The `dataSource` property can be assigned either with the instance of [Link to the Video](#) or JavaScript object array collection.

To get start quickly with Data Binding, you can check on this video:

To bind data to a cell, use `cell data binding` support.

#### Local data

To bind local data to the Spreadsheet, you can assign a JavaScript object array to the `dataSource` property.

Refer to the following code example for local data binding.

#### **APP.JSX**

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
  return (
    <SpreadsheetComponent>
      <SheetsDirective>
        <SheetDirective>
          <RangesDirective>
            <RangeDirective dataSource={data}></RangeDirective>
          </RangesDirective>
          <ColumnsDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={110}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={180}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
          </ColumnsDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  );
}
```

```

        </SheetsDirective>
      </SpreadsheetComponent>
    );
  };
  export default App;
  const root = createRoot(document.getElementById('root'));
  root.render(<App />);
  {% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
  return (
    <SpreadsheetComponent>
      <SheetsDirective>
        <SheetDirective>
          <RangesDirective>
            <RangeDirective dataSource={data}></RangeDirective>
          </RangesDirective>
          <ColumnsDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={110}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={180}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
          </ColumnsDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  );
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

### DATASOURCE.JSX

```

/**
 * Default data source
 */
export let data = [{
  OrderID: 10248,
  CustomerID: 'VINET',
  EmployeeID: 5,
  ShipName: 'Vins et alcools Chevalier',
  ShipCity: 'Reims',
  ShipAddress: '59 rue de l'Abbaye'
}];

```

```

    },
    {
      OrderID: 10249,
      CustomerID: 'TOMSP',
      EmployeeID: 6,
      ShipName: 'Toms Spezialitäten',
      ShipCity: 'Münster',
      ShipAddress: 'Luisenstr. 48'
    },
    {
      OrderID: 10250,
      CustomerID: 'HANAR',
      EmployeeID: 4,
      ShipName: 'Hanari Carnes',
      ShipCity: 'Rio de Janeiro',
      ShipAddress: 'Rua do Paço, 67'
    },
    {
      OrderID: 10251,
      CustomerID: 'VICTE',
      EmployeeID: 3,
      ShipName: 'Victuailles en stock',
      ShipCity: 'Lyon',
      ShipAddress: '2, rue du Commerce'
    },
    {
      OrderID: 10252,
      CustomerID: 'SUPRD',
      EmployeeID: 4,
      ShipName: 'Suprêmes délices',
      ShipCity: 'Charleroi',
      ShipAddress: 'Boulevard Tirou, 255'
    }
  ]];

```

**DATASOURCE.TSX**

```

/**
 * Default data source
 */
export let data: Object[] = [{
  OrderID: 10248,
  CustomerID: 'VINET',
  EmployeeID: 5,
  ShipName: 'Vins et alcools Chevalier',
  ShipCity: 'Reims',
  ShipAddress: '59 rue de l'Abbaye'
},
{
  OrderID: 10249,
  CustomerID: 'TOMSP',
  EmployeeID: 6,
  ShipName: 'Toms Spezialitäten',
  ShipCity: 'Münster',
  ShipAddress: 'Luisenstr. 48'
},
{

```

```

    OrderID: 10250,
    CustomerID: 'HANAR',
    EmployeeID: 4,
    ShipName: 'Hanari Carnes',
    ShipCity: 'Rio de Janeiro',
    ShipAddress: 'Rua do Paço, 67'
  },
  {
    OrderID: 10251,
    CustomerID: 'VICTE',
    EmployeeID: 3,
    ShipName: 'Victuailles en stock',
    ShipCity: 'Lyon',
    ShipAddress: '2, rue du Commerce'
  },
  {
    OrderID: 10252,
    CustomerID: 'SUPRD',
    EmployeeID: 4,
    ShipName: 'Suprêmes délices',
    ShipCity: 'Charleroi',
    ShipAddress: 'Boulevard Tirou, 255'
  }
];

```

The local data source can also be provided as an instance of the [DataManager](#). By default, [DataManager](#) uses [JsonAdaptor](#) for local data-binding.

### Remote data

To bind remote data to the Spreadsheet control, assign service data as an instance of [DataManager](#) to the `dataSource` property. To interact with remote data source, provide the service endpoint `url`.

Refer to the following code example for remote data binding.

### APP.JSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { DataManager, Query } from '@syncfusion/ej2-data';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from '@syncfusion/ej2-react-spreadsheet';
function App() {
  const query = new Query().select(['OrderID', 'CustomerID', 'ShipName',
'ShipCity', 'ShipCountry', 'Freight']).take(200);
  const data = new DataManager({
    url: 'https://services.syncfusion.com/react/production/api/Orders',
    crossDomain: true
  });
  return (
    <SpreadsheetComponent>
      <SheetsDirective>
        <SheetDirective name='Shipment Details'>
          <RangesDirective>

```

```

        <RangeDirective dataSource={data}
query={query}></RangeDirective>
    </RangesDirective>
    <ColumnsDirective>
        <ColumnDirective width={100}></ColumnDirective>
        <ColumnDirective width={110}></ColumnDirective>
        <ColumnDirective width={100}></ColumnDirective>
        <ColumnDirective width={180}></ColumnDirective>
        <ColumnDirective width={130}></ColumnDirective>
        <ColumnDirective width={130}></ColumnDirective>
    </ColumnsDirective>
</SheetDirective>
</SheetsDirective>
</SpreadsheetComponent>
    );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { DataManager, Query } from '@syncfusion/ej2-data';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
function App() {
    const query: Query = new Query().select(['OrderID', 'CustomerID',
'ShipName', 'ShipCity', 'ShipCountry', 'Freight']).take(200);
    const data: DataManager = new DataManager({
        url: 'https://services.syncfusion.com/react/production/api/Orders',
        crossDomain: true
    });
    return (
        <SpreadsheetComponent>
            <SheetsDirective>
                <SheetDirective name='Shipment Details'>
                    <RangesDirective>
                        <RangeDirective dataSource={data}
query={query}></RangeDirective>
                    </RangesDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={100}></ColumnDirective>
                        <ColumnDirective width={110}></ColumnDirective>
                        <ColumnDirective width={100}></ColumnDirective>
                        <ColumnDirective width={180}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
            </SheetsDirective>
        </SpreadsheetComponent>
    );
}

```

```

    </SpreadsheetComponent>
  );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

## DATASOURCE.JSX

```

/**
 * Default data source
 */
export let defaultData = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
    Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
  { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
    Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
  { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
  { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
'12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
70 },
  { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
    Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
  { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
    Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
  { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
    Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
  { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
    Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
  { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
    Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
/**
 * Grid datasource
 */
export function getTradeData(dataCount) {
  let employees = [
    'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
    'Fuller', 'Leverling', 'Peacock',
    'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
    'Dodsworth', 'Anne',
    'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
    'Jack', 'Rose'
  ];
  let designation = ['Manager', 'CFO', 'Designer', 'Developer', 'Program
Directory', 'System Analyst', 'Project Lead'];
  let mail = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
'journrapide.com'];
  let location = ['UK', 'USA', 'Sweden', 'France', 'Canada', 'Argentina',
'Austria', 'Germany', 'Mexico'];
  let status = ['Active', 'Inactive'];
  let trustworthiness = ['Perfect', 'Sufficient', 'Insufficient'];
}

```



```

    let tradeData = [];
    let address = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do Paço, 67',
'2, rue du Commerce', 'Boulevard Tirou, 255',
    'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
Mercado, 12',
    'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
    'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
    'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
43', '5a Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
    'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
    'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
    'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
    'Taucherstraße 10', 'Taucherstraße 10',
    'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
Circus', 'Av. dos Lusíadas, 23',
    'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
123', '2817 Milton Dr.'];
    let employeeimg = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
        dataCount = parseInt(dataCount, 10);
    }
    for (let i = 1; i <= dataCount; i++) {
        let code = 10000;
        tradeData.push({
            'EmployeeID': code + i,
            'Employees': employees[Math.floor(Math.random() *
employees.length)] + ' ' + employees[Math.floor(Math.random() *
employees.length)],
            'Designation': designation[Math.floor(Math.random() *
designation.length)],
            'Location': location[Math.floor(Math.random() *
location.length)],
            'Status': status[Math.floor(Math.random() * status.length)],
            'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
            'Rating': Math.floor(Math.random() * 5),
            'Software': Math.floor(Math.random() * 100),
            'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
            'CurrentSalary': Math.floor((Math.random() * 100000)),
            'Address': address[Math.floor(Math.random() * address.length)],
        });
        let employee = 'Employees';
        let emp = tradeData[i - 1][employee];
        let sName = emp.substr(0, emp.indexOf(' ')).toLowerCase();
        let empmail = 'Mail';
        tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }
    return tradeData;
}
export let tradeData = [
    {
        "EmployeeID": 10001,

```

```

    "Employees": "Laura Nancy",
    "Designation": "Designer",
    "Location": "France",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 69,
    "EmployeeImg": "usermale",
    "CurrentSalary": 84194,
    "Address": "Taucherstraße 10",
    "Mail": "laural5@jourrapide.com"
  },
  {
    "EmployeeID": 10002,
    "Employees": "Zachery Van",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 99,
    "EmployeeImg": "usermale",
    "CurrentSalary": 55349,
    "Address": "5a Ave. Los Palos Grandes",
    "Mail": "zachery109@sample.com"
  },
  {
    "EmployeeID": 10003,
    "Employees": "Rose Fuller",
    "Designation": "CFO",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 1,
    "EmployeeImg": "usermale",
    "CurrentSalary": 16477,
    "Address": "2817 Milton Dr.",
    "Mail": "rose55@rpy.com"
  },
  {
    "EmployeeID": 10004,
    "Employees": "Jack Bergs",
    "Designation": "Manager",
    "Location": "Mexico",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 36,
    "EmployeeImg": "usermale",
    "CurrentSalary": 49040,
    "Address": "2, rue du Commerce",
    "Mail": "jack30@sample.com"
  },
  {
    "EmployeeID": 10005,
    "Employees": "Vinet Bergs",

```

```

    "Designation": "Program Directory",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 39,
    "EmployeeImg": "usermale",
    "CurrentSalary": 5495,
    "Address": "Rua da Panificadora, 12",
    "Mail": "vinet32@jourrapide.com"
  },
  {
    "EmployeeID": 10006,
    "Employees": "Buchanan Van",
    "Designation": "Designer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 42182,
    "Address": "24, place Kléber",
    "Mail": "buchanan18@mail.com"
  },
  {
    "EmployeeID": 10007,
    "Employees": "Dodsworth Nancy",
    "Designation": "Project Lead",
    "Location": "USA",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 35776,
    "Address": "Rua do Paço, 67",
    "Mail": "dodsworth84@mail.com"
  },
  {
    "EmployeeID": 10008,
    "Employees": "Laura Jack",
    "Designation": "Developer",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 89,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25108,
    "Address": "Rua da Panificadora, 12",
    "Mail": "laura82@mail.com"
  },
  {
    "EmployeeID": 10009,
    "Employees": "Anne Fuller",
    "Designation": "Program Directory",

```

```
    "Location": "Mexico",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 32568,
    "Address": "Gran Vía, 1",
    "Mail": "anne97@jourrapide.com"
  },
  {
    "EmployeeID": 10010,
    "Employees": "Buchanan Andrew",
    "Designation": "Designer",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 62,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 12320,
    "Address": "P.O. Box 555",
    "Mail": "buchanan50@jourrapide.com"
  },
  {
    "EmployeeID": 10011,
    "Employees": "Andrew Janet",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 20890,
    "Address": "Starenweg 5",
    "Mail": "andrew63@mail.com"
  },
  {
    "EmployeeID": 10012,
    "Employees": "Margaret Tamer",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 7,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 22337,
    "Address": "Magazinweg 7",
    "Mail": "margaret26@mail.com"
  },
  {
    "EmployeeID": 10013,
    "Employees": "Tamer Fuller",
    "Designation": "CFO",
    "Location": "Canada",
```

```

    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 89181,
    "Address": "Taucherstraße 10",
    "Mail": "tamer40@arpy.com"
  },
  {
    "EmployeeID": 10014,
    "Employees": "Tamer Anne",
    "Designation": "CFO",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 18,
    "EmployeeImg": "usermale",
    "CurrentSalary": 20998,
    "Address": "Taucherstraße 10",
    "Mail": "tamer68@arpy.com"
  },
  {
    "EmployeeID": 10015,
    "Employees": "Anton Davolio",
    "Designation": "Project Lead",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 4,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 48232,
    "Address": "Luisenstr. 48",
    "Mail": "anton46@mail.com"
  },
  {
    "EmployeeID": 10016,
    "Employees": "Buchanan Buchanan",
    "Designation": "System Analyst",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "usermale",
    "CurrentSalary": 43041,
    "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
    "Mail": "buchanan68@mail.com"
  },
  {
    "EmployeeID": 10017,
    "Employees": "King Buchanan",
    "Designation": "Program Directory",
    "Location": "Sweden",
    "Status": "Active",

```

```
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 44,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 25259,
    "Address": "Magazinweg 7",
    "Mail": "king80@jourrapide.com"
  },
  {
    "EmployeeID": 10018,
    "Employees": "Rose Michael",
    "Designation": "Project Lead",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 31,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 91156,
    "Address": "Fauntleroy Circus",
    "Mail": "rose75@mail.com"
  },
  {
    "EmployeeID": 10019,
    "Employees": "King Bergs",
    "Designation": "Developer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 2,
    "Software": 29,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 28826,
    "Address": "2817 Milton Dr.",
    "Mail": "king57@jourrapide.com"
  },
  {
    "EmployeeID": 10020,
    "Employees": "Davolio Fuller",
    "Designation": "Designer",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 3,
    "Software": 35,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 71035,
    "Address": "Gran Vía, 1",
    "Mail": "davolio29@arpy.com"
  },
  {
    "EmployeeID": 10021,
    "Employees": "Rose Rose",
    "Designation": "CFO",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Perfect",
```

```

    "Rating": 3,
    "Software": 38,
    "EmployeeImg": "usermale",
    "CurrentSalary": 68123,
    "Address": "Rua do Mercado, 12",
    "Mail": "rose54@arpy.com"
  },
  {
    "EmployeeID": 10022,
    "Employees": "Andrew Michael",
    "Designation": "Program Directory",
    "Location": "UK",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 2,
    "Software": 61,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 75470,
    "Address": "2, rue du Commerce",
    "Mail": "andrew88@jourrapide.com"
  },
  {
    "EmployeeID": 10023,
    "Employees": "Davolio Kathryn",
    "Designation": "Manager",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 25,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25234,
    "Address": "Hauptstr. 31",
    "Mail": "davolio42@sample.com"
  },
  {
    "EmployeeID": 10024,
    "Employees": "Anne Fleet",
    "Designation": "System Analyst",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 8341,
    "Address": "59 rue de l'Abbaye",
    "Mail": "anne86@arpy.com"
  },
  {
    "EmployeeID": 10025,
    "Employees": "Margaret Andrew",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,

```

```
    "Software": 51,  
    "EmployeeImg": "userfemale",  
    "CurrentSalary": 84975,  
    "Address": "P.O. Box 555",  
    "Mail": "margaret41@arpy.com"  
  },  
  {  
    "EmployeeID": 10026,  
    "Employees": "Kathryn Laura",  
    "Designation": "Project Lead",  
    "Location": "Austria",  
    "Status": "Active",  
    "Trustworthiness": "Insufficient",  
    "Rating": 3,  
    "Software": 48,  
    "EmployeeImg": "usermale",  
    "CurrentSalary": 97282,  
    "Address": "Avda. Azteca 123",  
    "Mail": "kathryn82@rpy.com"  
  },  
  {  
    "EmployeeID": 10027,  
    "Employees": "Michael Michael",  
    "Designation": "Developer",  
    "Location": "UK",  
    "Status": "Inactive",  
    "Trustworthiness": "Perfect",  
    "Rating": 4,  
    "Software": 16,  
    "EmployeeImg": "usermale",  
    "CurrentSalary": 4184,  
    "Address": "Rua do Paço, 67",  
    "Mail": "michael58@jourrapide.com"  
  },  
  {  
    "EmployeeID": 10028,  
    "Employees": "Leverling Vinet",  
    "Designation": "Project Lead",  
    "Location": "Germany",  
    "Status": "Inactive",  
    "Trustworthiness": "Perfect",  
    "Rating": 0,  
    "Software": 57,  
    "EmployeeImg": "userfemale",  
    "CurrentSalary": 38370,  
    "Address": "59 rue de l'Abbaye",  
    "Mail": "leverling102@sample.com"  
  },  
  {  
    "EmployeeID": 10029,  
    "Employees": "Rose Jack",  
    "Designation": "Developer",  
    "Location": "UK",  
    "Status": "Active",  
    "Trustworthiness": "Perfect",  
    "Rating": 0,  
    "Software": 46,
```



```

    "EmployeeImg": "userfemale",
    "CurrentSalary": 84790,
    "Address": "Rua do Mercado, 12",
    "Mail": "rose108@jourrapide.com"
  },
  {
    "EmployeeID": 10030,
    "Employees": "Vinet Van",
    "Designation": "Developer",
    "Location": "USA",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 40,
    "EmployeeImg": "usermale",
    "CurrentSalary": 71005,
    "Address": "Gran Vía, 1",
    "Mail": "vinet90@jourrapide.com"
  }
];

```

### DATASOURCE.TSX

```

/**
 * Default data source
 */
export let defaultData: Object[] = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
    Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
  { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
    Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
  { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54 AM',
    Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
  { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time: '12:43:59 AM',
    Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit: 70 },
  { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
    Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
  { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
    Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
  { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
    Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
  { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
    Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
  { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
    Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
/**
 * Grid datasource
 */
export function getTradeData(dataCount?: number): object {
  let employees: string[] = [
    'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
    'Fuller', 'Leverling', 'Peacock',

```

```

    'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
    'Dodsworth', 'Anne',
    'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
    'Jack', 'Rose'];
    let designation: string[] = ['Manager', 'CFO', 'Designer', 'Developer',
    'Program Directory', 'System Analyst', 'Project Lead'];
    let mail: string[] = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
    'journrapide.com'];
    let location: string[] = ['UK', 'USA', 'Sweden', 'France', 'Canada',
    'Argentina', 'Austria', 'Germany', 'Mexico'];
    let status: string[] = ['Active', 'Inactive'];
    let trustworthiness: string[] = ['Perfect', 'Sufficient',
    'Insufficient'];
    let tradeData: Object[] = [];
    let address: string[] = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do
    Paço, 67', '2, rue du Commerce', 'Boulevard Tirou, 255',
    'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
    Mercado, 12',
    'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
    'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
    Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
    'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
    43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
    'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
    10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
    'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
    7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
    'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
    'Taucherstraße 10', 'Taucherstraße 10',
    'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
    Circus', 'Av. dos Lusíadas, 23',
    'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
    123', '2817 Milton Dr.'];
    let employeeimg: string[] = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
        dataCount = parseInt(dataCount, 10);
    }
    for (let i: number = 1; i <= dataCount; i++) {
        let code: any = 10000;
        tradeData.push({
            'EmployeeID': code + i,
            'Employees':
                employees[Math.floor(Math.random() * employees.length)] + '
' + employees[Math.floor(Math.random() * employees.length)],
            'Designation': designation[Math.floor(Math.random() *
designation.length)],
            'Location': location[Math.floor(Math.random() *
location.length)],
            'Status': status[Math.floor(Math.random() * status.length)],
            'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
            'Rating': Math.floor(Math.random() * 5),
            'Software': Math.floor(Math.random() * 100),
            'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
            'CurrentSalary': Math.floor((Math.random() * 100000)),
            'Address': address[Math.floor(Math.random() * address.length)],

```

```

    });
    let employee: string = 'Employees';
    let emp: string = tradeData[i - 1][employee];
    let sName: string = emp.substr(0, emp.indexOf(' ')).toLowerCase();
    let empmail: string = 'Mail';
    tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
  }
  return tradeData;
}
export let tradeData: Object[] = [
  {
    "EmployeeID": 10001,
    "Employees": "Laura Nancy",
    "Designation": "Designer",
    "Location": "France",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 69,
    "EmployeeImg": "usermale",
    "CurrentSalary": 84194,
    "Address": "Taucherstraße 10",
    "Mail": "laura15@jourrapide.com"
  },
  {
    "EmployeeID": 10002,
    "Employees": "Zachery Van",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 99,
    "EmployeeImg": "usermale",
    "CurrentSalary": 55349,
    "Address": "5ª Ave. Los Palos Grandes",
    "Mail": "zachery109@sample.com"
  },
  {
    "EmployeeID": 10003,
    "Employees": "Rose Fuller",
    "Designation": "CFO",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 1,
    "EmployeeImg": "usermale",
    "CurrentSalary": 16477,
    "Address": "2817 Milton Dr.",
    "Mail": "rose55@rpy.com"
  },
  {
    "EmployeeID": 10004,
    "Employees": "Jack Bergrs",
    "Designation": "Manager",

```

```
    "Location": "Mexico",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 36,
    "EmployeeImg": "usermale",
    "CurrentSalary": 49040,
    "Address": "2, rue du Commerce",
    "Mail": "jack30@sample.com"
  },
  {
    "EmployeeID": 10005,
    "Employees": "Vinet Bergs",
    "Designation": "Program Directory",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 39,
    "EmployeeImg": "usermale",
    "CurrentSalary": 5495,
    "Address": "Rua da Panificadora, 12",
    "Mail": "vinet32@jourrapide.com"
  },
  {
    "EmployeeID": 10006,
    "Employees": "Buchanan Van",
    "Designation": "Designer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 42182,
    "Address": "24, place Kléber",
    "Mail": "buchanan18@mail.com"
  },
  {
    "EmployeeID": 10007,
    "Employees": "Dodsworth Nancy",
    "Designation": "Project Lead",
    "Location": "USA",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 35776,
    "Address": "Rua do Paço, 67",
    "Mail": "dodsworth84@mail.com"
  },
  {
    "EmployeeID": 10008,
    "Employees": "Laura Jack",
    "Designation": "Developer",
    "Location": "Austria",
```

```

    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 89,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25108,
    "Address": "Rua da Panificadora, 12",
    "Mail": "laura82@mail.com"
  },
  {
    "EmployeeID": 10009,
    "Employees": "Anne Fuller",
    "Designation": "Program Directory",
    "Location": "Mexico",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 32568,
    "Address": "Gran Vía, 1",
    "Mail": "anne97@jourrapide.com"
  },
  {
    "EmployeeID": 10010,
    "Employees": "Buchanan Andrew",
    "Designation": "Designer",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 62,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 12320,
    "Address": "P.O. Box 555",
    "Mail": "buchanan50@jourrapide.com"
  },
  {
    "EmployeeID": 10011,
    "Employees": "Andrew Janet",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 20890,
    "Address": "Starenweg 5",
    "Mail": "andrew63@mail.com"
  },
  {
    "EmployeeID": 10012,
    "Employees": "Margaret Tamer",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",

```

```

    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 7,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 22337,
    "Address": "Magazinweg 7",
    "Mail": "margaret26@mail.com"
  },
  {
    "EmployeeID": 10013,
    "Employees": "Tamer Fuller",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 89181,
    "Address": "Taucherstraße 10",
    "Mail": "tamer40@arpy.com"
  },
  {
    "EmployeeID": 10014,
    "Employees": "Tamer Anne",
    "Designation": "CFO",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 18,
    "EmployeeImg": "usermale",
    "CurrentSalary": 20998,
    "Address": "Taucherstraße 10",
    "Mail": "tamer68@arpy.com"
  },
  {
    "EmployeeID": 10015,
    "Employees": "Anton Davolio",
    "Designation": "Project Lead",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 4,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 48232,
    "Address": "Luisenstr. 48",
    "Mail": "anton46@mail.com"
  },
  {
    "EmployeeID": 10016,
    "Employees": "Buchanan Buchanan",
    "Designation": "System Analyst",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",

```

```

    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "usermale",
    "CurrentSalary": 43041,
    "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
    "Mail": "buchanan68@mail.com"
  },
  {
    "EmployeeID": 10017,
    "Employees": "King Buchanan",
    "Designation": "Program Directory",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 44,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 25259,
    "Address": "Magazinweg 7",
    "Mail": "king80@jourrapide.com"
  },
  {
    "EmployeeID": 10018,
    "Employees": "Rose Michael",
    "Designation": "Project Lead",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 31,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 91156,
    "Address": "Fauntleroy Circus",
    "Mail": "rose75@mail.com"
  },
  {
    "EmployeeID": 10019,
    "Employees": "King Bergs",
    "Designation": "Developer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 2,
    "Software": 29,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 28826,
    "Address": "2817 Milton Dr.",
    "Mail": "king57@jourrapide.com"
  },
  {
    "EmployeeID": 10020,
    "Employees": "Davolio Fuller",
    "Designation": "Designer",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 3,

```

```
"Software": 35,  
"EmployeeImg": "userfemale",  
"CurrentSalary": 71035,  
"Address": "Gran Vía, 1",  
"Mail": "davolio29@arpy.com"  
},  
{  
  "EmployeeID": 10021,  
  "Employees": "Rose Rose",  
  "Designation": "CFO",  
  "Location": "Germany",  
  "Status": "Active",  
  "Trustworthiness": "Perfect",  
  "Rating": 3,  
  "Software": 38,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 68123,  
  "Address": "Rua do Mercado, 12",  
  "Mail": "rose54@arpy.com"  
},  
{  
  "EmployeeID": 10022,  
  "Employees": "Andrew Michael",  
  "Designation": "Program Directory",  
  "Location": "UK",  
  "Status": "Inactive",  
  "Trustworthiness": "Insufficient",  
  "Rating": 2,  
  "Software": 61,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 75470,  
  "Address": "2, rue du Commerce",  
  "Mail": "andrew88@jourrapide.com"  
},  
{  
  "EmployeeID": 10023,  
  "Employees": "Davolio Kathryn",  
  "Designation": "Manager",  
  "Location": "Germany",  
  "Status": "Active",  
  "Trustworthiness": "Perfect",  
  "Rating": 3,  
  "Software": 25,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 25234,  
  "Address": "Hauptstr. 31",  
  "Mail": "davolio42@sample.com"  
},  
{  
  "EmployeeID": 10024,  
  "Employees": "Anne Fleet",  
  "Designation": "System Analyst",  
  "Location": "UK",  
  "Status": "Active",  
  "Trustworthiness": "Perfect",  
  "Rating": 3,  
  "Software": 0,
```



```
"EmployeeImg": "userfemale",
"CurrentSalary": 8341,
"Address": "59 rue de l'Abbaye",
"Mail": "anne86@arpy.com"
},
{
  "EmployeeID": 10025,
  "Employees": "Margaret Andrew",
  "Designation": "System Analyst",
  "Location": "Germany",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 51,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 84975,
  "Address": "P.O. Box 555",
  "Mail": "margaret41@arpy.com"
},
{
  "EmployeeID": 10026,
  "Employees": "Kathryn Laura",
  "Designation": "Project Lead",
  "Location": "Austria",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 48,
  "EmployeeImg": "usermale",
  "CurrentSalary": 97282,
  "Address": "Avda. Azteca 123",
  "Mail": "kathryn82@rpy.com"
},
{
  "EmployeeID": 10027,
  "Employees": "Michael Michael",
  "Designation": "Developer",
  "Location": "UK",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 4,
  "Software": 16,
  "EmployeeImg": "usermale",
  "CurrentSalary": 4184,
  "Address": "Rua do Paço, 67",
  "Mail": "michael58@jourrapide.com"
},
{
  "EmployeeID": 10028,
  "Employees": "Leverling Vinet",
  "Designation": "Project Lead",
  "Location": "Germany",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 57,
  "EmployeeImg": "userfemale",
```

```

    "CurrentSalary": 38370,
    "Address": "59 rue de l'Abbaye",
    "Mail": "leverling102@sample.com"
  },
  {
    "EmployeeID": 10029,
    "Employees": "Rose Jack",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 46,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84790,
    "Address": "Rua do Mercado, 12",
    "Mail": "rose108@jourrapide.com"
  },
  {
    "EmployeeID": 10030,
    "Employees": "Vinet Van",
    "Designation": "Developer",
    "Location": "USA",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 40,
    "EmployeeImg": "usermale",
    "CurrentSalary": 71005,
    "Address": "Gran Vía, 1",
    "Mail": "vinet90@jourrapide.com"
  }
]

```

By default, **DataManager** uses **ODataAdaptor** for remote data-binding.

#### *Binding with OData services*

OData is a standardized protocol for creating and consuming data. You can retrieve data from OData service using the DataManager. Refer to the following code example for remote Data binding using OData service.

#### **APP.JSX**

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { DataManager, ODataAdaptor } from '@syncfusion/ej2-data';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
function App() {
  const spreadsheetRef = React.useRef(null);
  const data = (new DataManager({
    adaptor: new ODataAdaptor(),
    url: 'https://services.syncfusion.com/react/production/api/Orders'

```

```

    ));
    React.useEffect(() => {
        let spreadsheet = spreadsheetRef.current
        // Applies cell and number formatting to specified range of the
        active sheet
        if (spreadsheet) {
            spreadsheet.cellFormat({ fontWeight: 'bold', textAlign:
'center', verticalAlign: 'middle' }, 'A1:K1');
        }
    }, []);
    return (
        <SpreadsheetComponent ref={spreadsheetRef}>
            <SheetsDirective>
                <SheetDirective name='Order Details'>
                    <RangesDirective>
                        <RangeDirective dataSource={data}></RangeDirective>
                    </RangesDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={80}></ColumnDirective>
                        <ColumnDirective width={80}></ColumnDirective>
                        <ColumnDirective width={80}></ColumnDirective>
                        <ColumnDirective width={80}></ColumnDirective>
                        <ColumnDirective width={80}></ColumnDirective>
                        <ColumnDirective width={80}></ColumnDirective>
                        <ColumnDirective width={280}></ColumnDirective>
                        <ColumnDirective width={180}></ColumnDirective>
                        <ColumnDirective width={80}></ColumnDirective>
                        <ColumnDirective width={180}></ColumnDirective>
                        <ColumnDirective width={180}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
            </SheetsDirective>
        </SpreadsheetComponent>
    );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

## APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { DataManager, ODataAdaptor } from '@syncfusion/ej2-data';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
function App() {
    const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
    const data: DataManager = (new DataManager({
        adaptor: new ODataAdaptor(),
        url: 'https://services.syncfusion.com/react/production/api/Orders'
    }));

```

```

    React.useEffect(() => {
        let spreadsheet = spreadsheetRef.current
        // Applies cell and number formatting to specified range of the
        active sheet
        if (spreadsheet) {
            spreadsheet.cellFormat({ fontWeight: 'bold', textAlign:
'center', verticalAlign: 'middle' }, 'A1:K1');
        }
    }, []);
    return (
        <SpreadsheetComponent ref={spreadsheetRef}>
            <SheetsDirective>
                <SheetDirective name='Order Details'>
                    <RangesDirective>
                        <RangeDirective dataSource={data}></RangeDirective>
                    </RangesDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={80}></ColumnDirective>
                        <ColumnDirective width={80}></ColumnDirective>
                        <ColumnDirective width={80}></ColumnDirective>
                        <ColumnDirective width={80}></ColumnDirective>
                        <ColumnDirective width={80}></ColumnDirective>
                        <ColumnDirective width={80}></ColumnDirective>
                        <ColumnDirective width={280}></ColumnDirective>
                        <ColumnDirective width={180}></ColumnDirective>
                        <ColumnDirective width={80}></ColumnDirective>
                        <ColumnDirective width={180}></ColumnDirective>
                        <ColumnDirective width={180}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
            </SheetsDirective>
        </SpreadsheetComponent>
    );
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

## DATASOURCE.JSX

```

/**
 * Default data source
 */
export let defaultData = [
    { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
    { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
    { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
    { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
    { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
'12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
70 },

```

```

    { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
      Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
    { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
      Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
    { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
      Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
    { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
      Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
    { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
      Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
  ];
  /**
   * Grid datasource
   */
  export function getTradeData(dataCount) {
    let employees = [
      'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
      'Fuller', 'Leverling', 'Peacock',
      'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
      'Dodsworth', 'Anne',
      'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
      'Jack', 'Rose'
    ];
    let designation = ['Manager', 'CFO', 'Designer', 'Developer', 'Program
      Directory', 'System Analyst', 'Project Lead'];
    let mail = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
      'journrapide.com'];
    let location = ['UK', 'USA', 'Sweden', 'France', 'Canada', 'Argentina',
      'Austria', 'Germany', 'Mexico'];
    let status = ['Active', 'Inactive'];
    let trustworthiness = ['Perfect', 'Sufficient', 'Insufficient'];
    let tradeData = [];
    let address = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do Paço, 67',
      '2, rue du Commerce', 'Boulevard Tirou, 255',
      'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
      Mercado, 12',
      'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
      'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
      Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
      'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
      43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
      'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
      10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
      'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
      7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
      'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
      'Taucherstraße 10', 'Taucherstraße 10',
      'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
      Circus', 'Av. dos Lusíadas, 23',
      'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
      123', '2817 Milton Dr.'];
    let employeeimg = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
      dataCount = parseInt(dataCount, 10);
    }
    for (let i = 1; i <= dataCount; i++) {
      let code = 10000;

```

```

        tradeData.push({
            'EmployeeID': code + i,
            'Employees': employees[Math.floor(Math.random() *
employees.length)] + ' ' + employees[Math.floor(Math.random() *
employees.length)],
            'Designation': designation[Math.floor(Math.random() *
designation.length)],
            'Location': location[Math.floor(Math.random() *
location.length)],
            'Status': status[Math.floor(Math.random() * status.length)],
            'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
            'Rating': Math.floor(Math.random() * 5),
            'Software': Math.floor(Math.random() * 100),
            'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
            'CurrentSalary': Math.floor((Math.random() * 100000)),
            'Address': address[Math.floor(Math.random() * address.length)],
        });
        let employee = 'Employees';
        let emp = tradeData[i - 1][employee];
        let sName = emp.substr(0, emp.indexOf(' ')).toLowerCase();
        let empmail = 'Mail';
        tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }
    return tradeData;
}
export let tradeData = [
    {
        "EmployeeID": 10001,
        "Employees": "Laura Nancy",
        "Designation": "Designer",
        "Location": "France",
        "Status": "Inactive",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
        "Software": 69,
        "EmployeeImg": "usermale",
        "CurrentSalary": 84194,
        "Address": "Taucherstraße 10",
        "Mail": "laura15@jourrapide.com"
    },
    {
        "EmployeeID": 10002,
        "Employees": "Zachery Van",
        "Designation": "CFO",
        "Location": "Canada",
        "Status": "Inactive",
        "Trustworthiness": "Insufficient",
        "Rating": 3,
        "Software": 99,
        "EmployeeImg": "usermale",
        "CurrentSalary": 55349,
        "Address": "5ª Ave. Los Palos Grandes",
        "Mail": "zachery109@sample.com"
    },

```

```

{
  "EmployeeID": 10003,
  "Employees": "Rose Fuller",
  "Designation": "CFO",
  "Location": "France",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 1,
  "EmployeeImg": "usermale",
  "CurrentSalary": 16477,
  "Address": "2817 Milton Dr.",
  "Mail": "rose55@rpy.com"
},
{
  "EmployeeID": 10004,
  "Employees": "Jack Bergs",
  "Designation": "Manager",
  "Location": "Mexico",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 36,
  "EmployeeImg": "usermale",
  "CurrentSalary": 49040,
  "Address": "2, rue du Commerce",
  "Mail": "jack30@sample.com"
},
{
  "EmployeeID": 10005,
  "Employees": "Vinet Bergs",
  "Designation": "Program Directory",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 39,
  "EmployeeImg": "usermale",
  "CurrentSalary": 5495,
  "Address": "Rua da Panificadora, 12",
  "Mail": "vinet32@jourrapide.com"
},
{
  "EmployeeID": 10006,
  "Employees": "Buchanan Van",
  "Designation": "Designer",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 4,
  "Software": 78,
  "EmployeeImg": "usermale",
  "CurrentSalary": 42182,
  "Address": "24, place Kléber",
  "Mail": "buchanan18@mail.com"
},
{

```

```

      "EmployeeID": 10007,
      "Employees": "Dodsworth Nancy",
      "Designation": "Project Lead",
      "Location": "USA",
      "Status": "Inactive",
      "Trustworthiness": "Sufficient",
      "Rating": 0,
      "Software": 0,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 35776,
      "Address": "Rua do Paço, 67",
      "Mail": "dodsworth84@mail.com"
    },
    {
      "EmployeeID": 10008,
      "Employees": "Laura Jack",
      "Designation": "Developer",
      "Location": "Austria",
      "Status": "Inactive",
      "Trustworthiness": "Perfect",
      "Rating": 3,
      "Software": 89,
      "EmployeeImg": "usermale",
      "CurrentSalary": 25108,
      "Address": "Rua da Panificadora, 12",
      "Mail": "laura82@mail.com"
    },
    {
      "EmployeeID": 10009,
      "Employees": "Anne Fuller",
      "Designation": "Program Directory",
      "Location": "Mexico",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 0,
      "Software": 19,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 32568,
      "Address": "Gran Vía, 1",
      "Mail": "anne97@jourrapide.com"
    },
    {
      "EmployeeID": 10010,
      "Employees": "Buchanan Andrew",
      "Designation": "Designer",
      "Location": "Austria",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 1,
      "Software": 62,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 12320,
      "Address": "P.O. Box 555",
      "Mail": "buchanan50@jourrapide.com"
    },
    {
      "EmployeeID": 10011,

```



```

    "Employees": "Andrew Janet",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 20890,
    "Address": "Starenweg 5",
    "Mail": "andrew63@mail.com"
  },
  {
    "EmployeeID": 10012,
    "Employees": "Margaret Tamer",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 7,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 22337,
    "Address": "Magazinweg 7",
    "Mail": "margaret26@mail.com"
  },
  {
    "EmployeeID": 10013,
    "Employees": "Tamer Fuller",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 89181,
    "Address": "Taucherstraße 10",
    "Mail": "tamer40@arpy.com"
  },
  {
    "EmployeeID": 10014,
    "Employees": "Tamer Anne",
    "Designation": "CFO",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 18,
    "EmployeeImg": "usermale",
    "CurrentSalary": 20998,
    "Address": "Taucherstraße 10",
    "Mail": "tamer68@arpy.com"
  },
  {
    "EmployeeID": 10015,
    "Employees": "Anton Davolio",

```

```

    "Designation": "Project Lead",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 4,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 48232,
    "Address": "Luisenstr. 48",
    "Mail": "anton46@mail.com"
  },
  {
    "EmployeeID": 10016,
    "Employees": "Buchanan Buchanan",
    "Designation": "System Analyst",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "usermale",
    "CurrentSalary": 43041,
    "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
    "Mail": "buchanan68@mail.com"
  },
  {
    "EmployeeID": 10017,
    "Employees": "King Buchanan",
    "Designation": "Program Directory",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 44,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 25259,
    "Address": "Magazinweg 7",
    "Mail": "king80@jourrapide.com"
  },
  {
    "EmployeeID": 10018,
    "Employees": "Rose Michael",
    "Designation": "Project Lead",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 31,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 91156,
    "Address": "Fauntleroy Circus",
    "Mail": "rose75@mail.com"
  },
  {
    "EmployeeID": 10019,
    "Employees": "King Bergs",
    "Designation": "Developer",

```

```

        "Location": "Germany",
        "Status": "Active",
        "Trustworthiness": "Sufficient",
        "Rating": 2,
        "Software": 29,
        "EmployeeImg": "userfemale",
        "CurrentSalary": 28826,
        "Address": "2817 Milton Dr.",
        "Mail": "king57@jourrapide.com"
    },
    {
        "EmployeeID": 10020,
        "Employees": "Davolio Fuller",
        "Designation": "Designer",
        "Location": "Canada",
        "Status": "Inactive",
        "Trustworthiness": "Sufficient",
        "Rating": 3,
        "Software": 35,
        "EmployeeImg": "userfemale",
        "CurrentSalary": 71035,
        "Address": "Gran Vía, 1",
        "Mail": "davolio29@arpy.com"
    },
    {
        "EmployeeID": 10021,
        "Employees": "Rose Rose",
        "Designation": "CFO",
        "Location": "Germany",
        "Status": "Active",
        "Trustworthiness": "Perfect",
        "Rating": 3,
        "Software": 38,
        "EmployeeImg": "usermale",
        "CurrentSalary": 68123,
        "Address": "Rua do Mercado, 12",
        "Mail": "rose54@arpy.com"
    },
    {
        "EmployeeID": 10022,
        "Employees": "Andrew Michael",
        "Designation": "Program Directory",
        "Location": "UK",
        "Status": "Inactive",
        "Trustworthiness": "Insufficient",
        "Rating": 2,
        "Software": 61,
        "EmployeeImg": "userfemale",
        "CurrentSalary": 75470,
        "Address": "2, rue du Commerce",
        "Mail": "andrew88@jourrapide.com"
    },
    {
        "EmployeeID": 10023,
        "Employees": "Davolio Kathryn",
        "Designation": "Manager",
        "Location": "Germany",

```

```

    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 25,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25234,
    "Address": "Hauptstr. 31",
    "Mail": "davolio42@sample.com"
  },
  {
    "EmployeeID": 10024,
    "Employees": "Anne Fleet",
    "Designation": "System Analyst",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 8341,
    "Address": "59 rue de lAbbaye",
    "Mail": "anne86@arpy.com"
  },
  {
    "EmployeeID": 10025,
    "Employees": "Margaret Andrew",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 51,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84975,
    "Address": "P.O. Box 555",
    "Mail": "margaret41@arpy.com"
  },
  {
    "EmployeeID": 10026,
    "Employees": "Kathryn Laura",
    "Designation": "Project Lead",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 48,
    "EmployeeImg": "usermale",
    "CurrentSalary": 97282,
    "Address": "Avda. Azteca 123",
    "Mail": "kathryn82@rpy.com"
  },
  {
    "EmployeeID": 10027,
    "Employees": "Michael Michael",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Inactive",

```

```

        "Trustworthiness": "Perfect",
        "Rating": 4,
        "Software": 16,
        "EmployeeImg": "usermale",
        "CurrentSalary": 4184,
        "Address": "Rua do Paço, 67",
        "Mail": "michael58@jourrapide.com"
    },
    {
        "EmployeeID": 10028,
        "Employees": "Leverling Vinet",
        "Designation": "Project Lead",
        "Location": "Germany",
        "Status": "Inactive",
        "Trustworthiness": "Perfect",
        "Rating": 0,
        "Software": 57,
        "EmployeeImg": "userfemale",
        "CurrentSalary": 38370,
        "Address": "59 rue de l'Abbaye",
        "Mail": "leverling102@sample.com"
    },
    {
        "EmployeeID": 10029,
        "Employees": "Rose Jack",
        "Designation": "Developer",
        "Location": "UK",
        "Status": "Active",
        "Trustworthiness": "Perfect",
        "Rating": 0,
        "Software": 46,
        "EmployeeImg": "userfemale",
        "CurrentSalary": 84790,
        "Address": "Rua do Mercado, 12",
        "Mail": "rose108@jourrapide.com"
    },
    {
        "EmployeeID": 10030,
        "Employees": "Vinet Van",
        "Designation": "Developer",
        "Location": "USA",
        "Status": "Active",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
        "Software": 40,
        "EmployeeImg": "usermale",
        "CurrentSalary": 71005,
        "Address": "Gran Vía, 1",
        "Mail": "vinet90@jourrapide.com"
    }
];

```

**DATASOURCE.TSX**

```

/**
 * Default data source

```

```

*/
export let defaultData: Object[] = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
    Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
  { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
    Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
  { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
  { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
'12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
70 },
  { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
    Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
  { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
    Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
  { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
    Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
  { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
    Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
  { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
    Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
/**
 * Grid datasource
 */
export function getTradeData(dataCount?: number): object {
  let employees: string[] = [
    'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
    'Fuller', 'Leverling', 'Peacock',
    'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
    'Dodsworth', 'Anne',
    'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
    'Jack', 'Rose'];
  let designation: string[] = ['Manager', 'CFO', 'Designer', 'Developer',
    'Program Directory', 'System Analyst', 'Project Lead'];
  let mail: string[] = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
    'jourrapide.com'];
  let location: string[] = ['UK', 'USA', 'Sweden', 'France', 'Canada',
    'Argentina', 'Austria', 'Germany', 'Mexico'];
  let status: string[] = ['Active', 'Inactive'];
  let trustworthiness: string[] = ['Perfect', 'Sufficient',
    'Insufficient'];
  let tradeData: Object[] = [];
  let address: string[] = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do
Paço, 67', '2, rue du Commerce', 'Boulevard Tirou, 255',
    'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
Mercado, 12',
    'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
    'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
    'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
43', '5a Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
    'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',

```

```

    'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
    'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
    'Taucherstraße 10', 'Taucherstraße 10',
    'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
Circus', 'Av. dos Lusíadas, 23',
    'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
123', '2817 Milton Dr.'];
    let employeeimg: string[] = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
        dataCount = parseInt(dataCount, 10);
    }
    for (let i: number = 1; i <= dataCount; i++) {
        let code: any = 10000;
        tradeData.push({
            'EmployeeID': code + i,
            'Employees':
                employees[Math.floor(Math.random() * employees.length)] + '
' + employees[Math.floor(Math.random() * employees.length)],
            'Designation': designation[Math.floor(Math.random() *
designation.length)],
            'Location': location[Math.floor(Math.random() *
location.length)],
            'Status': status[Math.floor(Math.random() * status.length)],
            'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
            'Rating': Math.floor(Math.random() * 5),
            'Software': Math.floor(Math.random() * 100),
            'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
            'CurrentSalary': Math.floor((Math.random() * 100000)),
            'Address': address[Math.floor(Math.random() * address.length)],
        });
        let employee: string = 'Employees';
        let emp: string = tradeData[i - 1][employee];
        let sName: string = emp.substr(0, emp.indexOf(' ')).toLowerCase();
        let empmail: string = 'Mail';
        tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }
    return tradeData;
}
export let tradeData: Object[] = [
    {
        "EmployeeID": 10001,
        "Employees": "Laura Nancy",
        "Designation": "Designer",
        "Location": "France",
        "Status": "Inactive",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
        "Software": 69,
        "EmployeeImg": "usermale",
        "CurrentSalary": 84194,
        "Address": "Taucherstraße 10",
        "Mail": "laura15@jourrapide.com"
    },

```

```
{
  "EmployeeID": 10002,
  "Employees": "Zachery Van",
  "Designation": "CFO",
  "Location": "Canada",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 99,
  "EmployeeImg": "usermale",
  "CurrentSalary": 55349,
  "Address": "5a Ave. Los Palos Grandes",
  "Mail": "zachery109@sample.com"
},
{
  "EmployeeID": 10003,
  "Employees": "Rose Fuller",
  "Designation": "CFO",
  "Location": "France",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 1,
  "EmployeeImg": "usermale",
  "CurrentSalary": 16477,
  "Address": "2817 Milton Dr.",
  "Mail": "rose55@rpy.com"
},
{
  "EmployeeID": 10004,
  "Employees": "Jack Bergs",
  "Designation": "Manager",
  "Location": "Mexico",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 36,
  "EmployeeImg": "usermale",
  "CurrentSalary": 49040,
  "Address": "2, rue du Commerce",
  "Mail": "jack30@sample.com"
},
{
  "EmployeeID": 10005,
  "Employees": "Vinet Bergs",
  "Designation": "Program Directory",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 39,
  "EmployeeImg": "usermale",
  "CurrentSalary": 5495,
  "Address": "Rua da Panificadora, 12",
  "Mail": "vinet32@jourrapide.com"
},
{
```



```
"EmployeeID": 10006,  
"Employees": "Buchanan Van",  
"Designation": "Designer",  
"Location": "Germany",  
"Status": "Active",  
"Trustworthiness": "Insufficient",  
"Rating": 4,  
"Software": 78,  
"EmployeeImg": "usermale",  
"CurrentSalary": 42182,  
"Address": "24, place Kléber",  
"Mail": "buchanan18@mail.com"  
},  
{  
  "EmployeeID": 10007,  
  "Employees": "Dodsworth Nancy",  
  "Designation": "Project Lead",  
  "Location": "USA",  
  "Status": "Inactive",  
  "Trustworthiness": "Sufficient",  
  "Rating": 0,  
  "Software": 0,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 35776,  
  "Address": "Rua do Paço, 67",  
  "Mail": "dodsworth84@mail.com"  
},  
{  
  "EmployeeID": 10008,  
  "Employees": "Laura Jack",  
  "Designation": "Developer",  
  "Location": "Austria",  
  "Status": "Inactive",  
  "Trustworthiness": "Perfect",  
  "Rating": 3,  
  "Software": 89,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 25108,  
  "Address": "Rua da Panificadora, 12",  
  "Mail": "laura82@mail.com"  
},  
{  
  "EmployeeID": 10009,  
  "Employees": "Anne Fuller",  
  "Designation": "Program Directory",  
  "Location": "Mexico",  
  "Status": "Active",  
  "Trustworthiness": "Insufficient",  
  "Rating": 0,  
  "Software": 19,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 32568,  
  "Address": "Gran Vía, 1",  
  "Mail": "anne97@jourrapide.com"  
},  
{  
  "EmployeeID": 10010,
```

```
"Employees": "Buchanan Andrew",
"Designation": "Designer",
"Location": "Austria",
>Status": "Active",
"Trustworthiness": "Insufficient",
"Rating": 1,
"Software": 62,
"EmployeeImg": "userfemale",
"CurrentSalary": 12320,
"Address": "P.O. Box 555",
"Mail": "buchanan50@jourrapide.com"
},
{
  "EmployeeID": 10011,
  "Employees": "Andrew Janet",
  "Designation": "System Analyst",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 8,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 20890,
  "Address": "Starenweg 5",
  "Mail": "andrew63@mail.com"
},
{
  "EmployeeID": 10012,
  "Employees": "Margaret Tamer",
  "Designation": "System Analyst",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 4,
  "Software": 7,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 22337,
  "Address": "Magazinweg 7",
  "Mail": "margaret26@mail.com"
},
{
  "EmployeeID": 10013,
  "Employees": "Tamer Fuller",
  "Designation": "CFO",
  "Location": "Canada",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 78,
  "EmployeeImg": "usermale",
  "CurrentSalary": 89181,
  "Address": "Taucherstraße 10",
  "Mail": "tamer40@arpy.com"
},
{
  "EmployeeID": 10014,
  "Employees": "Tamer Anne",
```

```

    "Designation": "CFO",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 18,
    "EmployeeImg": "usermale",
    "CurrentSalary": 20998,
    "Address": "Taucherstraße 10",
    "Mail": "tamer68@arpy.com"
  },
  {
    "EmployeeID": 10015,
    "Employees": "Anton Davolio",
    "Designation": "Project Lead",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 4,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 48232,
    "Address": "Luisenstr. 48",
    "Mail": "anton46@mail.com"
  },
  {
    "EmployeeID": 10016,
    "Employees": "Buchanan Buchanan",
    "Designation": "System Analyst",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "usermale",
    "CurrentSalary": 43041,
    "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
    "Mail": "buchanan68@mail.com"
  },
  {
    "EmployeeID": 10017,
    "Employees": "King Buchanan",
    "Designation": "Program Directory",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 44,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 25259,
    "Address": "Magazinweg 7",
    "Mail": "king80@jourrapide.com"
  },
  {
    "EmployeeID": 10018,
    "Employees": "Rose Michael",
    "Designation": "Project Lead",

```

```

    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 31,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 91156,
    "Address": "Fauntleroy Circus",
    "Mail": "rose75@mail.com"
  },
  {
    "EmployeeID": 10019,
    "Employees": "King Bergs",
    "Designation": "Developer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 2,
    "Software": 29,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 28826,
    "Address": "2817 Milton Dr.",
    "Mail": "king57@jourrapide.com"
  },
  {
    "EmployeeID": 10020,
    "Employees": "Davolio Fuller",
    "Designation": "Designer",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 3,
    "Software": 35,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 71035,
    "Address": "Gran Vía, 1",
    "Mail": "davolio29@arpy.com"
  },
  {
    "EmployeeID": 10021,
    "Employees": "Rose Rose",
    "Designation": "CFO",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 38,
    "EmployeeImg": "usermale",
    "CurrentSalary": 68123,
    "Address": "Rua do Mercado, 12",
    "Mail": "rose54@arpy.com"
  },
  {
    "EmployeeID": 10022,
    "Employees": "Andrew Michael",
    "Designation": "Program Directory",
    "Location": "UK",

```

```

    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 2,
    "Software": 61,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 75470,
    "Address": "2, rue du Commerce",
    "Mail": "andrew88@jourrapide.com"
  },
  {
    "EmployeeID": 10023,
    "Employees": "Davolio Kathryn",
    "Designation": "Manager",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 25,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25234,
    "Address": "Hauptstr. 31",
    "Mail": "davolio42@sample.com"
  },
  {
    "EmployeeID": 10024,
    "Employees": "Anne Fleet",
    "Designation": "System Analyst",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 8341,
    "Address": "59 rue de l'Abbaye",
    "Mail": "anne86@arpy.com"
  },
  {
    "EmployeeID": 10025,
    "Employees": "Margaret Andrew",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 51,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84975,
    "Address": "P.O. Box 555",
    "Mail": "margaret41@arpy.com"
  },
  {
    "EmployeeID": 10026,
    "Employees": "Kathryn Laura",
    "Designation": "Project Lead",
    "Location": "Austria",
    "Status": "Active",

```

```
"Trustworthiness": "Insufficient",
"Rating": 3,
"Software": 48,
"EmployeeImg": "usermale",
"CurrentSalary": 97282,
"Address": "Avda. Azteca 123",
"Mail": "kathryn82@rpy.com"
},
{
  "EmployeeID": 10027,
  "Employees": "Michael Michael",
  "Designation": "Developer",
  "Location": "UK",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 4,
  "Software": 16,
  "EmployeeImg": "usermale",
  "CurrentSalary": 4184,
  "Address": "Rua do Paço, 67",
  "Mail": "michael58@jourrapide.com"
},
{
  "EmployeeID": 10028,
  "Employees": "Leverling Vinet",
  "Designation": "Project Lead",
  "Location": "Germany",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 57,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 38370,
  "Address": "59 rue de l'Abbaye",
  "Mail": "leverling102@sample.com"
},
{
  "EmployeeID": 10029,
  "Employees": "Rose Jack",
  "Designation": "Developer",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 46,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 84790,
  "Address": "Rua do Mercado, 12",
  "Mail": "rosl08@jourrapide.com"
},
{
  "EmployeeID": 10030,
  "Employees": "Vinet Van",
  "Designation": "Developer",
  "Location": "USA",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
```

```

    "Rating": 0,
    "Software": 40,
    "EmployeeImg": "usermale",
    "CurrentSalary": 71005,
    "Address": "Gran Via, 1",
    "Mail": "vinet90@jourrapide.com"
  }
]

```

### Web API

You can use WebApiAdaptor to bind spreadsheet with Web API created using OData endpoint.

### APP.JSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { DataManager, WebApiAdaptor } from '@syncfusion/ej2-data';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
function App() {
  const spreadsheetRef = React.useRef(null);
  const data = (new DataManager({ adaptor: new WebApiAdaptor(), url:
'https://services.syncfusion.com/react/production/api/Orders' }));
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    // Applies cell and number formatting to specified range of the
active sheet
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign:
'center', verticalAlign: 'middle' }, 'A1:K1');
    }
  }, []);
  return (
    <SpreadsheetComponent ref={spreadsheetRef}>
      <SheetsDirective>
        <SheetDirective name='Order Details'>
          <RangesDirective>
            <RangeDirective dataSource={data}></RangeDirective>
          </RangesDirective>
          <ColumnsDirective>
            <ColumnDirective width={80}></ColumnDirective>
            <ColumnDirective width={80}></ColumnDirective>
            <ColumnDirective width={80}></ColumnDirective>
            <ColumnDirective width={80}></ColumnDirective>
            <ColumnDirective width={80}></ColumnDirective>
            <ColumnDirective width={80}></ColumnDirective>
            <ColumnDirective width={280}></ColumnDirective>
            <ColumnDirective width={180}></ColumnDirective>
            <ColumnDirective width={80}></ColumnDirective>
            <ColumnDirective width={180}></ColumnDirective>
            <ColumnDirective width={180}></ColumnDirective>
          </ColumnsDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  );
}

```

```

        </SheetsDirective>
    </SpreadsheetComponent>
    );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { DataManager, WebApiAdaptor } from '@syncfusion/ej2-data';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
function App() {
    const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
    const data: DataManager = (new DataManager({ adaptor: new
WebApiAdaptor(), url:
'https://services.syncfusion.com/react/production/api/Orders' }));
    React.useEffect(() => {
        let spreadsheet = spreadsheetRef.current;
        // Applies cell and number formatting to specified range of the
active sheet
        if (spreadsheet) {
            spreadsheet.cellFormat({ fontWeight: 'bold', textAlign:
'center', verticalAlign: 'middle' }, 'A1:K1');
        }
    }, []);
    return (
        <SpreadsheetComponent ref={spreadsheetRef}>
            <SheetsDirective>
                <SheetDirective name='Order Details'>
                    <RangesDirective>
                        <RangeDirective dataSource={data}></RangeDirective>
                    </RangesDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={80}></ColumnDirective>
                        <ColumnDirective width={80}></ColumnDirective>
                        <ColumnDirective width={80}></ColumnDirective>
                        <ColumnDirective width={80}></ColumnDirective>
                        <ColumnDirective width={80}></ColumnDirective>
                        <ColumnDirective width={80}></ColumnDirective>
                        <ColumnDirective width={280}></ColumnDirective>
                        <ColumnDirective width={180}></ColumnDirective>
                        <ColumnDirective width={80}></ColumnDirective>
                        <ColumnDirective width={180}></ColumnDirective>
                        <ColumnDirective width={180}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
            </SheetsDirective>
        </SpreadsheetComponent>
    );
}

```



```

    );
  };
  export default App;
  const root = createRoot(document.getElementById('root')!);
  root.render(<App />);
  {% endraw %}

```

### DATASOURCE.JSX

```

/**
 * Default data source
 */
export let defaultData = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
    Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
  { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
    Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
  { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
  { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
'12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
70 },
  { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
    Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
  { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
    Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
  { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
    Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
  { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
    Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
  { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
    Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
/**
 * Grid datasource
 */
export function getTradeData(dataCount) {
  let employees = [
    'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
    'Fuller', 'Leverling', 'Peacock',
    'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
    'Dodsworth', 'Anne',
    'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
    'Jack', 'Rose'
  ];
  let designation = ['Manager', 'CFO', 'Designer', 'Developer', 'Program
Directory', 'System Analyst', 'Project Lead'];
  let mail = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
'jourrapide.com'];
  let location = ['UK', 'USA', 'Sweden', 'France', 'Canada', 'Argentina',
'Austria', 'Germany', 'Mexico'];
  let status = ['Active', 'Inactive'];
  let trustworthiness = ['Perfect', 'Sufficient', 'Insufficient'];
  let tradeData = [];

```

```

    let address = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do Paço, 67',
'2, rue du Commerce', 'Boulevard Tirou, 255',
'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
Mercado, 12',
'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
'Taucherstraße 10', 'Taucherstraße 10',
'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
Circus', 'Av. dos Lusíadas, 23',
'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
123', '2817 Milton Dr.'];
    let employeeimg = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
        dataCount = parseInt(dataCount, 10);
    }
    for (let i = 1; i <= dataCount; i++) {
        let code = 10000;
        tradeData.push({
            'EmployeeID': code + i,
            'Employees': employees[Math.floor(Math.random() *
employees.length)] + ' ' + employees[Math.floor(Math.random() *
employees.length)],
            'Designation': designation[Math.floor(Math.random() *
designation.length)],
            'Location': location[Math.floor(Math.random() *
location.length)],
            'Status': status[Math.floor(Math.random() * status.length)],
            'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
            'Rating': Math.floor(Math.random() * 5),
            'Software': Math.floor(Math.random() * 100),
            'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
            'CurrentSalary': Math.floor((Math.random() * 100000)),
            'Address': address[Math.floor(Math.random() * address.length)],
        });
        let employee = 'Employees';
        let emp = tradeData[i - 1][employee];
        let sName = emp.substr(0, emp.indexOf(' ')).toLowerCase();
        let empmail = 'Mail';
        tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }
    return tradeData;
}
export let tradeData = [
    {
        "EmployeeID": 10001,
        "Employees": "Laura Nancy",

```

```

    "Designation": "Designer",
    "Location": "France",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 69,
    "EmployeeImg": "usermale",
    "CurrentSalary": 84194,
    "Address": "Taucherstraße 10",
    "Mail": "laura15@jourrapide.com"
  },
  {
    "EmployeeID": 10002,
    "Employees": "Zachery Van",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 99,
    "EmployeeImg": "usermale",
    "CurrentSalary": 55349,
    "Address": "5a Ave. Los Palos Grandes",
    "Mail": "zachery109@sample.com"
  },
  {
    "EmployeeID": 10003,
    "Employees": "Rose Fuller",
    "Designation": "CFO",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 1,
    "EmployeeImg": "usermale",
    "CurrentSalary": 16477,
    "Address": "2817 Milton Dr.",
    "Mail": "rose55@rpy.com"
  },
  {
    "EmployeeID": 10004,
    "Employees": "Jack Bergs",
    "Designation": "Manager",
    "Location": "Mexico",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 36,
    "EmployeeImg": "usermale",
    "CurrentSalary": 49040,
    "Address": "2, rue du Commerce",
    "Mail": "jack30@sample.com"
  },
  {
    "EmployeeID": 10005,
    "Employees": "Vinet Bergs",
    "Designation": "Program Directory",

```

```

      "Location": "UK",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 1,
      "Software": 39,
      "EmployeeImg": "usermale",
      "CurrentSalary": 5495,
      "Address": "Rua da Panificadora, 12",
      "Mail": "vinet32@jourrapide.com"
    },
    {
      "EmployeeID": 10006,
      "Employees": "Buchanan Van",
      "Designation": "Designer",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 4,
      "Software": 78,
      "EmployeeImg": "usermale",
      "CurrentSalary": 42182,
      "Address": "24, place Kléber",
      "Mail": "buchanan18@mail.com"
    },
    {
      "EmployeeID": 10007,
      "Employees": "Dodsworth Nancy",
      "Designation": "Project Lead",
      "Location": "USA",
      "Status": "Inactive",
      "Trustworthiness": "Sufficient",
      "Rating": 0,
      "Software": 0,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 35776,
      "Address": "Rua do Paço, 67",
      "Mail": "dodsworth84@mail.com"
    },
    {
      "EmployeeID": 10008,
      "Employees": "Laura Jack",
      "Designation": "Developer",
      "Location": "Austria",
      "Status": "Inactive",
      "Trustworthiness": "Perfect",
      "Rating": 3,
      "Software": 89,
      "EmployeeImg": "usermale",
      "CurrentSalary": 25108,
      "Address": "Rua da Panificadora, 12",
      "Mail": "laura82@mail.com"
    },
    {
      "EmployeeID": 10009,
      "Employees": "Anne Fuller",
      "Designation": "Program Directory",
      "Location": "Mexico",

```

```

    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 32568,
    "Address": "Gran Vía, 1",
    "Mail": "anne97@jourrapide.com"
  },
  {
    "EmployeeID": 10010,
    "Employees": "Buchanan Andrew",
    "Designation": "Designer",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 62,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 12320,
    "Address": "P.O. Box 555",
    "Mail": "buchanan50@jourrapide.com"
  },
  {
    "EmployeeID": 10011,
    "Employees": "Andrew Janet",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 20890,
    "Address": "Starenweg 5",
    "Mail": "andrew63@mail.com"
  },
  {
    "EmployeeID": 10012,
    "Employees": "Margaret Tamer",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 7,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 22337,
    "Address": "Magazinweg 7",
    "Mail": "margaret26@mail.com"
  },
  {
    "EmployeeID": 10013,
    "Employees": "Tamer Fuller",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Active",

```

```

    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 89181,
    "Address": "Taucherstraße 10",
    "Mail": "tamer40@arpy.com"
  },
  {
    "EmployeeID": 10014,
    "Employees": "Tamer Anne",
    "Designation": "CFO",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 18,
    "EmployeeImg": "usermale",
    "CurrentSalary": 20998,
    "Address": "Taucherstraße 10",
    "Mail": "tamer68@arpy.com"
  },
  {
    "EmployeeID": 10015,
    "Employees": "Anton Davolio",
    "Designation": "Project Lead",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 4,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 48232,
    "Address": "Luisenstr. 48",
    "Mail": "anton46@mail.com"
  },
  {
    "EmployeeID": 10016,
    "Employees": "Buchanan Buchanan",
    "Designation": "System Analyst",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "usermale",
    "CurrentSalary": 43041,
    "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
    "Mail": "buchanan68@mail.com"
  },
  {
    "EmployeeID": 10017,
    "Employees": "King Buchanan",
    "Designation": "Program Directory",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",

```

```

    "Rating": 0,
    "Software": 44,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 25259,
    "Address": "Magazinweg 7",
    "Mail": "king80@jourrapide.com"
  },
  {
    "EmployeeID": 10018,
    "Employees": "Rose Michael",
    "Designation": "Project Lead",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 31,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 91156,
    "Address": "Fauntleroy Circus",
    "Mail": "rose75@mail.com"
  },
  {
    "EmployeeID": 10019,
    "Employees": "King Bergs",
    "Designation": "Developer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 2,
    "Software": 29,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 28826,
    "Address": "2817 Milton Dr.",
    "Mail": "king57@jourrapide.com"
  },
  {
    "EmployeeID": 10020,
    "Employees": "Davolio Fuller",
    "Designation": "Designer",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 3,
    "Software": 35,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 71035,
    "Address": "Gran Vía, 1",
    "Mail": "davolio29@arpy.com"
  },
  {
    "EmployeeID": 10021,
    "Employees": "Rose Rose",
    "Designation": "CFO",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,

```

```

    "Software": 38,
    "EmployeeImg": "usermale",
    "CurrentSalary": 68123,
    "Address": "Rua do Mercado, 12",
    "Mail": "rose54@arpy.com"
  },
  {
    "EmployeeID": 10022,
    "Employees": "Andrew Michael",
    "Designation": "Program Directory",
    "Location": "UK",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 2,
    "Software": 61,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 75470,
    "Address": "2, rue du Commerce",
    "Mail": "andrew88@jourrapide.com"
  },
  {
    "EmployeeID": 10023,
    "Employees": "Davolio Kathryn",
    "Designation": "Manager",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 25,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25234,
    "Address": "Hauptstr. 31",
    "Mail": "davolio42@sample.com"
  },
  {
    "EmployeeID": 10024,
    "Employees": "Anne Fleet",
    "Designation": "System Analyst",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 8341,
    "Address": "59 rue de l'Abbaye",
    "Mail": "anne86@arpy.com"
  },
  {
    "EmployeeID": 10025,
    "Employees": "Margaret Andrew",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 51,

```



```

      "EmployeeImg": "userfemale",
      "CurrentSalary": 84975,
      "Address": "P.O. Box 555",
      "Mail": "margaret41@arpy.com"
    },
    {
      "EmployeeID": 10026,
      "Employees": "Kathryn Laura",
      "Designation": "Project Lead",
      "Location": "Austria",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 3,
      "Software": 48,
      "EmployeeImg": "usermale",
      "CurrentSalary": 97282,
      "Address": "Avda. Azteca 123",
      "Mail": "kathryn82@rpy.com"
    },
    {
      "EmployeeID": 10027,
      "Employees": "Michael Michael",
      "Designation": "Developer",
      "Location": "UK",
      "Status": "Inactive",
      "Trustworthiness": "Perfect",
      "Rating": 4,
      "Software": 16,
      "EmployeeImg": "usermale",
      "CurrentSalary": 4184,
      "Address": "Rua do Paço, 67",
      "Mail": "michael58@jourrapide.com"
    },
    {
      "EmployeeID": 10028,
      "Employees": "Leverling Vinet",
      "Designation": "Project Lead",
      "Location": "Germany",
      "Status": "Inactive",
      "Trustworthiness": "Perfect",
      "Rating": 0,
      "Software": 57,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 38370,
      "Address": "59 rue de l'Abbaye",
      "Mail": "leverling102@sample.com"
    },
    {
      "EmployeeID": 10029,
      "Employees": "Rose Jack",
      "Designation": "Developer",
      "Location": "UK",
      "Status": "Active",
      "Trustworthiness": "Perfect",
      "Rating": 0,
      "Software": 46,
      "EmployeeImg": "userfemale",

```

```

      "CurrentSalary": 84790,
      "Address": "Rua do Mercado, 12",
      "Mail": "rose108@jourrapide.com"
    },
    {
      "EmployeeID": 10030,
      "Employees": "Vinet Van",
      "Designation": "Developer",
      "Location": "USA",
      "Status": "Active",
      "Trustworthiness": "Sufficient",
      "Rating": 0,
      "Software": 40,
      "EmployeeImg": "usermale",
      "CurrentSalary": 71005,
      "Address": "Gran Vía, 1",
      "Mail": "vinet90@jourrapide.com"
    }
  ]
};

```

### DATASOURCE.TSX

```

/**
 * Default data source
 */
export let defaultData: Object[] = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
    Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
  { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
    Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
  { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54 AM',
    Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
  { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time: '12:43:59 AM',
    Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit: 70 },
  { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
    Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
  { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
    Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
  { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
    Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
  { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
    Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
  { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
    Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
/**
 * Grid datasource
 */
export function getTradeData(dataCount?: number): object {
  let employees: string[] = [
    'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
    'Fuller', 'Leverling', 'Peacock',

```

```

    'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
    'Dodsworth', 'Anne',
    'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
    'Jack', 'Rose'];
    let designation: string[] = ['Manager', 'CFO', 'Designer', 'Developer',
    'Program Directory', 'System Analyst', 'Project Lead'];
    let mail: string[] = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
    'journrapide.com'];
    let location: string[] = ['UK', 'USA', 'Sweden', 'France', 'Canada',
    'Argentina', 'Austria', 'Germany', 'Mexico'];
    let status: string[] = ['Active', 'Inactive'];
    let trustworthiness: string[] = ['Perfect', 'Sufficient',
    'Insufficient'];
    let tradeData: Object[] = [];
    let address: string[] = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do
    Paço, 67', '2, rue du Commerce', 'Boulevard Tirou, 255',
    'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
    Mercado, 12',
    'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
    'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
    Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
    'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
    43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
    'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
    10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
    'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
    7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
    'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
    'Taucherstraße 10', 'Taucherstraße 10',
    'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
    Circus', 'Av. dos Lusíadas, 23',
    'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
    123', '2817 Milton Dr.'];
    let employeeimg: string[] = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
        dataCount = parseInt(dataCount, 10);
    }
    for (let i: number = 1; i <= dataCount; i++) {
        let code: any = 10000;
        tradeData.push({
            'EmployeeID': code + i,
            'Employees':
                employees[Math.floor(Math.random() * employees.length)] + '
' + employees[Math.floor(Math.random() * employees.length)],
            'Designation': designation[Math.floor(Math.random() *
designation.length)],
            'Location': location[Math.floor(Math.random() *
location.length)],
            'Status': status[Math.floor(Math.random() * status.length)],
            'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
            'Rating': Math.floor(Math.random() * 5),
            'Software': Math.floor(Math.random() * 100),
            'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
            'CurrentSalary': Math.floor((Math.random() * 100000)),
            'Address': address[Math.floor(Math.random() * address.length)],

```

```

    });
    let employee: string = 'Employees';
    let emp: string = tradeData[i - 1][employee];
    let sName: string = emp.substr(0, emp.indexOf(' ')).toLowerCase();
    let empmail: string = 'Mail';
    tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
  }
  return tradeData;
}
export let tradeData: Object[] = [
  {
    "EmployeeID": 10001,
    "Employees": "Laura Nancy",
    "Designation": "Designer",
    "Location": "France",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 69,
    "EmployeeImg": "usermale",
    "CurrentSalary": 84194,
    "Address": "Taucherstraße 10",
    "Mail": "laura15@jourrapide.com"
  },
  {
    "EmployeeID": 10002,
    "Employees": "Zachery Van",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 99,
    "EmployeeImg": "usermale",
    "CurrentSalary": 55349,
    "Address": "5ª Ave. Los Palos Grandes",
    "Mail": "zachery109@sample.com"
  },
  {
    "EmployeeID": 10003,
    "Employees": "Rose Fuller",
    "Designation": "CFO",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 1,
    "EmployeeImg": "usermale",
    "CurrentSalary": 16477,
    "Address": "2817 Milton Dr.",
    "Mail": "rose55@rpy.com"
  },
  {
    "EmployeeID": 10004,
    "Employees": "Jack Bergrs",
    "Designation": "Manager",

```

```
"Location": "Mexico",
>Status": "Inactive",
>Trustworthiness": "Insufficient",
>Rating": 3,
>Software": 36,
>EmployeeImg": "usermale",
>CurrentSalary": 49040,
>Address": "2, rue du Commerce",
>Mail": "jack30@sample.com"
},
{
  "EmployeeID": 10005,
  "Employees": "Vinet Bergs",
  "Designation": "Program Directory",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 39,
  "EmployeeImg": "usermale",
  "CurrentSalary": 5495,
  "Address": "Rua da Panificadora, 12",
  "Mail": "vinet32@jourrapide.com"
},
{
  "EmployeeID": 10006,
  "Employees": "Buchanan Van",
  "Designation": "Designer",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 4,
  "Software": 78,
  "EmployeeImg": "usermale",
  "CurrentSalary": 42182,
  "Address": "24, place Kléber",
  "Mail": "buchanan18@mail.com"
},
{
  "EmployeeID": 10007,
  "Employees": "Dodsworth Nancy",
  "Designation": "Project Lead",
  "Location": "USA",
  "Status": "Inactive",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 0,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 35776,
  "Address": "Rua do Paço, 67",
  "Mail": "dodsworth84@mail.com"
},
{
  "EmployeeID": 10008,
  "Employees": "Laura Jack",
  "Designation": "Developer",
  "Location": "Austria",
```

```
"Status": "Inactive",
"Trustworthiness": "Perfect",
"Rating": 3,
"Software": 89,
"EmployeeImg": "usermale",
"CurrentSalary": 25108,
"Address": "Rua da Panificadora, 12",
"Mail": "laura82@mail.com"
},
{
  "EmployeeID": 10009,
  "Employees": "Anne Fuller",
  "Designation": "Program Directory",
  "Location": "Mexico",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 0,
  "Software": 19,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 32568,
  "Address": "Gran Vía, 1",
  "Mail": "anne97@jourrapide.com"
},
{
  "EmployeeID": 10010,
  "Employees": "Buchanan Andrew",
  "Designation": "Designer",
  "Location": "Austria",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 62,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 12320,
  "Address": "P.O. Box 555",
  "Mail": "buchanan50@jourrapide.com"
},
{
  "EmployeeID": 10011,
  "Employees": "Andrew Janet",
  "Designation": "System Analyst",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 8,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 20890,
  "Address": "Starenweg 5",
  "Mail": "andrew63@mail.com"
},
{
  "EmployeeID": 10012,
  "Employees": "Margaret Tamer",
  "Designation": "System Analyst",
  "Location": "Germany",
  "Status": "Active",
```

```
"Trustworthiness": "Insufficient",
"Rating": 4,
"Software": 7,
"EmployeeImg": "userfemale",
"CurrentSalary": 22337,
"Address": "Magazinweg 7",
"Mail": "margaret26@mail.com"
},
{
  "EmployeeID": 10013,
  "Employees": "Tamer Fuller",
  "Designation": "CFO",
  "Location": "Canada",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 78,
  "EmployeeImg": "usermale",
  "CurrentSalary": 89181,
  "Address": "Taucherstraße 10",
  "Mail": "tamer40@arpy.com"
},
{
  "EmployeeID": 10014,
  "Employees": "Tamer Anne",
  "Designation": "CFO",
  "Location": "Sweden",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 18,
  "EmployeeImg": "usermale",
  "CurrentSalary": 20998,
  "Address": "Taucherstraße 10",
  "Mail": "tamer68@arpy.com"
},
{
  "EmployeeID": 10015,
  "Employees": "Anton Davolio",
  "Designation": "Project Lead",
  "Location": "France",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 4,
  "Software": 8,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 48232,
  "Address": "Luisenstr. 48",
  "Mail": "anton46@mail.com"
},
{
  "EmployeeID": 10016,
  "Employees": "Buchanan Buchanan",
  "Designation": "System Analyst",
  "Location": "Austria",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
```

```

    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "usermale",
    "CurrentSalary": 43041,
    "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
    "Mail": "buchanan68@mail.com"
  },
  {
    "EmployeeID": 10017,
    "Employees": "King Buchanan",
    "Designation": "Program Directory",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 44,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 25259,
    "Address": "Magazinweg 7",
    "Mail": "king80@jourrapide.com"
  },
  {
    "EmployeeID": 10018,
    "Employees": "Rose Michael",
    "Designation": "Project Lead",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 31,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 91156,
    "Address": "Fauntleroy Circus",
    "Mail": "rose75@mail.com"
  },
  {
    "EmployeeID": 10019,
    "Employees": "King Bergs",
    "Designation": "Developer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 2,
    "Software": 29,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 28826,
    "Address": "2817 Milton Dr.",
    "Mail": "king57@jourrapide.com"
  },
  {
    "EmployeeID": 10020,
    "Employees": "Davolio Fuller",
    "Designation": "Designer",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 3,

```



```
"Software": 35,  
"EmployeeImg": "userfemale",  
"CurrentSalary": 71035,  
"Address": "Gran Vía, 1",  
"Mail": "davolio29@arpy.com"  
},  
{  
  "EmployeeID": 10021,  
  "Employees": "Rose Rose",  
  "Designation": "CFO",  
  "Location": "Germany",  
  "Status": "Active",  
  "Trustworthiness": "Perfect",  
  "Rating": 3,  
  "Software": 38,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 68123,  
  "Address": "Rua do Mercado, 12",  
  "Mail": "rose54@arpy.com"  
},  
{  
  "EmployeeID": 10022,  
  "Employees": "Andrew Michael",  
  "Designation": "Program Directory",  
  "Location": "UK",  
  "Status": "Inactive",  
  "Trustworthiness": "Insufficient",  
  "Rating": 2,  
  "Software": 61,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 75470,  
  "Address": "2, rue du Commerce",  
  "Mail": "andrew88@jourrapide.com"  
},  
{  
  "EmployeeID": 10023,  
  "Employees": "Davolio Kathryn",  
  "Designation": "Manager",  
  "Location": "Germany",  
  "Status": "Active",  
  "Trustworthiness": "Perfect",  
  "Rating": 3,  
  "Software": 25,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 25234,  
  "Address": "Hauptstr. 31",  
  "Mail": "davolio42@sample.com"  
},  
{  
  "EmployeeID": 10024,  
  "Employees": "Anne Fleet",  
  "Designation": "System Analyst",  
  "Location": "UK",  
  "Status": "Active",  
  "Trustworthiness": "Perfect",  
  "Rating": 3,  
  "Software": 0,
```

```
"EmployeeImg": "userfemale",
"CurrentSalary": 8341,
"Address": "59 rue de l'Abbaye",
"Mail": "anne86@arpy.com"
},
{
  "EmployeeID": 10025,
  "Employees": "Margaret Andrew",
  "Designation": "System Analyst",
  "Location": "Germany",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 51,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 84975,
  "Address": "P.O. Box 555",
  "Mail": "margaret41@arpy.com"
},
{
  "EmployeeID": 10026,
  "Employees": "Kathryn Laura",
  "Designation": "Project Lead",
  "Location": "Austria",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 48,
  "EmployeeImg": "usermale",
  "CurrentSalary": 97282,
  "Address": "Avda. Azteca 123",
  "Mail": "kathryn82@rpy.com"
},
{
  "EmployeeID": 10027,
  "Employees": "Michael Michael",
  "Designation": "Developer",
  "Location": "UK",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 4,
  "Software": 16,
  "EmployeeImg": "usermale",
  "CurrentSalary": 4184,
  "Address": "Rua do Paço, 67",
  "Mail": "michael58@jourrapide.com"
},
{
  "EmployeeID": 10028,
  "Employees": "Leverling Vinet",
  "Designation": "Project Lead",
  "Location": "Germany",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 57,
  "EmployeeImg": "userfemale",
```

```

    "CurrentSalary": 38370,
    "Address": "59 rue de l'Abbaye",
    "Mail": "leverling102@sample.com"
  },
  {
    "EmployeeID": 10029,
    "Employees": "Rose Jack",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 46,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84790,
    "Address": "Rua do Mercado, 12",
    "Mail": "rosel108@jourrapide.com"
  },
  {
    "EmployeeID": 10030,
    "Employees": "Vinet Van",
    "Designation": "Developer",
    "Location": "USA",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 40,
    "EmployeeImg": "usermale",
    "CurrentSalary": 71005,
    "Address": "Gran Vía, 1",
    "Mail": "vinet90@jourrapide.com"
  }
]

```

### Cell data binding

The Spreadsheet control can bind the data to individual cell in a sheet . To achieve this you can use the `value` property.

Refer to the following code example for cell data binding.

### APP.JSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RowsDirective, CellsDirective, RowDirective, CellDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { ColumnsDirective, ColumnDirective } from '@syncfusion/ej2-react-
spreadsheet';
function App() {
  return (
    <SpreadsheetComponent>
      <SheetsDirective>
        <SheetDirective selectedRange='D13' name='PriceDetails'>
          <RowsDirective>

```

```

        <RowDirective>
          <CellsDirective>
            <CellDirective value='Item Name'
></CellDirective>
            <CellDirective value='Quantity'
></CellDirective>
            <CellDirective value='Price'
></CellDirective>
            <CellDirective value='Amount'
></CellDirective>
            <CellDirective value='Stock Detail'
></CellDirective>
            <CellDirective value='Website'
></CellDirective>
          </CellsDirective>
        </RowDirective>
        <RowDirective>
          <CellsDirective>
            <CellDirective value='Casual Shoes'
></CellDirective>
            <CellDirective value='10' ></CellDirective>
            <CellDirective value='$20' ></CellDirective>
            <CellDirective value='$200'
></CellDirective>
            <CellDirective value='OUT OF STOCK'
></CellDirective>
            <CellDirective
value='Amazon'></CellDirective>
          </CellsDirective>
        </RowDirective>
        <RowDirective>
          <CellsDirective>
            <CellDirective value='Sports Shoes'
></CellDirective>
            <CellDirective value='20' ></CellDirective>
            <CellDirective value='$30' ></CellDirective>
            <CellDirective value='$600'></CellDirective>
            <CellDirective value='IN
STOCK'></CellDirective>
            <CellDirective
value='Overstock'></CellDirective>
          </CellsDirective>
        </RowDirective>
        <RowDirective>
          <CellsDirective>
            <CellDirective value='Formal Shoes'
></CellDirective>
            <CellDirective value='20' ></CellDirective>
            <CellDirective value='$15' ></CellDirective>
            <CellDirective value='$300'></CellDirective>
            <CellDirective value='IN
STOCK'></CellDirective>
            <CellDirective
value='AliExpress'></CellDirective>
          </CellsDirective>
        </RowDirective>
      </RowDirective>

```

```

        <CellsDirective>
          <CellDirective value='Sandals & Floaters'
></CellDirective>
          <CellDirective value='15' ></CellDirective>
          <CellDirective value='$20' ></CellDirective>
          <CellDirective value='$300'
></CellDirective>
          <CellDirective value='OUT OF STOCK'
></CellDirective>
          <CellDirective
value='AliBaba'></CellDirective>
        </CellsDirective>
      </RowDirective>
    </RowDirective>
    <CellsDirective>
      <CellDirective value='Flip-Flops & Slippers'
></CellDirective>
      <CellDirective value='30' ></CellDirective>
      <CellDirective value='$10' ></CellDirective>
      <CellDirective value='$300'></CellDirective>
      <CellDirective value='IN STOCK'
></CellDirective>
      <CellDirective
value='Taobao'></CellDirective>
    </CellsDirective>
  </RowDirective>
</RowsDirective>
<ColumnsDirective>
  <ColumnDirective width={110}></ColumnDirective>
  <ColumnDirective width={115}></ColumnDirective>
  <ColumnDirective width={110}></ColumnDirective>
  <ColumnDirective width={100}></ColumnDirective>
  <ColumnDirective width={110}></ColumnDirective>
  <ColumnDirective width={100}></ColumnDirective>
</ColumnsDirective>
</SheetDirective>
</SheetsDirective>
</SpreadsheetComponent>
  );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RowsDirective, CellsDirective, RowDirective, CellDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { ColumnsDirective, ColumnDirective } from '@syncfusion/ej2-react-
spreadsheet';
function App() {

```

```

    return (
      <SpreadsheetComponent>
        <SheetsDirective>
          <SheetDirective selectedRange='D13' name='PriceDetails'>
            <RowsDirective>
              <RowDirective>
                <CellsDirective>
                  <CellDirective value='Item Name'
                ></CellDirective>
                  <CellDirective value='Quantity'
                ></CellDirective>
                  <CellDirective value='Price'
                ></CellDirective>
                  <CellDirective value='Amount'
                ></CellDirective>
                  <CellDirective value='Stock Detail'
                ></CellDirective>
                  <CellDirective value='Website'
                ></CellDirective>
                </CellsDirective>
              </RowDirective>
              <RowDirective>
                <CellsDirective>
                  <CellDirective value='Casual Shoes'
                ></CellDirective>
                  <CellDirective value='10' ></CellDirective>
                  <CellDirective value='$20' ></CellDirective>
                  <CellDirective value='$200'
                ></CellDirective>
                  <CellDirective value='OUT OF STOCK'
                ></CellDirective>
                  <CellDirective
value='Amazon'></CellDirective>
                </CellsDirective>
              </RowDirective>
              <RowDirective>
                <CellsDirective>
                  <CellDirective value='Sports Shoes'
                ></CellDirective>
                  <CellDirective value='20' ></CellDirective>
                  <CellDirective value='$30' ></CellDirective>
                  <CellDirective value='$600'></CellDirective>
                  <CellDirective value='IN
STOCK'></CellDirective>
                  <CellDirective
value='Overstock'></CellDirective>
                </CellsDirective>
              </RowDirective>
              <RowDirective>
                <CellsDirective>
                  <CellDirective value='Formal Shoes'
                ></CellDirective>
                  <CellDirective value='20' ></CellDirective>
                  <CellDirective value='$15' ></CellDirective>
                  <CellDirective value='$300'></CellDirective>
                  <CellDirective value='IN
STOCK'></CellDirective>
                </CellsDirective>
              </RowDirective>
            </RowsDirective>
          </SheetDirective>
        </SheetsDirective>
      </SpreadsheetComponent>
    )
  
```

```

        <CellDirective
value='AliExpress'></CellDirective>
      </CellsDirective>
    </RowDirective>
  </RowDirective>
  <CellsDirective>
    <CellDirective value='Sandals & Floaters'
></CellDirective>

    <CellDirective value='15' ></CellDirective>
    <CellDirective value='$20' ></CellDirective>
    <CellDirective value='$300'

></CellDirective>

    <CellDirective value='OUT OF STOCK'

></CellDirective>

    <CellDirective
value='AliBaba'></CellDirective>
      </CellsDirective>
    </RowDirective>
  </RowDirective>
  <CellsDirective>
    <CellDirective value='Flip-Flops & Slippers'
></CellDirective>

    <CellDirective value='30' ></CellDirective>
    <CellDirective value='$10' ></CellDirective>
    <CellDirective value='$300'></CellDirective>
    <CellDirective value='IN STOCK'

></CellDirective>

    <CellDirective
value='Taobao'></CellDirective>
      </CellsDirective>
    </RowDirective>
  </RowsDirective>
</ColumnsDirective>
  <ColumnDirective width={110}></ColumnDirective>
  <ColumnDirective width={115}></ColumnDirective>
  <ColumnDirective width={110}></ColumnDirective>
  <ColumnDirective width={100}></ColumnDirective>
  <ColumnDirective width={110}></ColumnDirective>
  <ColumnDirective width={100}></ColumnDirective>
</ColumnsDirective>
</SheetDirective>
</SheetsDirective>
</SpreadsheetComponent>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

## DATASOURCE.JSX

```

/**
 * Default data source
 */
export let defaultData = [

```

```

    { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
      Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
    { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
      Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
    { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
      Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
    { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
      AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
    { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
      '12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
      70 },
    { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
      Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
    { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
      Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
    { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
      Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
    { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
      Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
    { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
      Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
  ];
  /**
   * Grid datasource
   */
  export function getTradeData(dataCount) {
    let employees = [
      'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
      'Fuller', 'Leverling', 'Peacock',
      'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
      'Dodsworth', 'Anne',
      'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
      'Jack', 'Rose'
    ];
    let designation = ['Manager', 'CFO', 'Designer', 'Developer', 'Program
      Directory', 'System Analyst', 'Project Lead'];
    let mail = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
      'jourrapide.com'];
    let location = ['UK', 'USA', 'Sweden', 'France', 'Canada', 'Argentina',
      'Austria', 'Germany', 'Mexico'];
    let status = ['Active', 'Inactive'];
    let trustworthiness = ['Perfect', 'Sufficient', 'Insufficient'];
    let tradeData = [];
    let address = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do Paço, 67',
      '2, rue du Commerce', 'Boulevard Tirou, 255',
      'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
      Mercado, 12',
      'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
      'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
      Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
      'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
      43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
      'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
      10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
      'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
      7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',

```



```

    'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
    'Taucherstraße 10', 'Taucherstraße 10',
    'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
    Circus', 'Av. dos Lusíadas, 23',
    'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
    123', '2817 Milton Dr.'];
    let employeeimg = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
        dataCount = parseInt(dataCount, 10);
    }
    for (let i = 1; i <= dataCount; i++) {
        let code = 10000;
        tradeData.push({
            'EmployeeID': code + i,
            'Employees': employees[Math.floor(Math.random() *
employees.length)] + ' ' + employees[Math.floor(Math.random() *
employees.length)],
            'Designation': designation[Math.floor(Math.random() *
designation.length)],
            'Location': location[Math.floor(Math.random() *
location.length)],
            'Status': status[Math.floor(Math.random() * status.length)],
            'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
            'Rating': Math.floor(Math.random() * 5),
            'Software': Math.floor(Math.random() * 100),
            'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
            'CurrentSalary': Math.floor((Math.random() * 100000)),
            'Address': address[Math.floor(Math.random() * address.length)],
        });
        let employee = 'Employees';
        let emp = tradeData[i - 1][employee];
        let sName = emp.substr(0, emp.indexOf(' ')).toLowerCase();
        let empmail = 'Mail';
        tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }
    return tradeData;
}
export let tradeData = [
    {
        "EmployeeID": 10001,
        "Employees": "Laura Nancy",
        "Designation": "Designer",
        "Location": "France",
        "Status": "Inactive",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
        "Software": 69,
        "EmployeeImg": "usermale",
        "CurrentSalary": 84194,
        "Address": "Taucherstraße 10",
        "Mail": "laura15@jourrapide.com"
    },
    {
        "EmployeeID": 10002,

```

```

    "Employees": "Zachery Van",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 99,
    "EmployeeImg": "usermale",
    "CurrentSalary": 55349,
    "Address": "5a Ave. Los Palos Grandes",
    "Mail": "zachery109@sample.com"
  },
  {
    "EmployeeID": 10003,
    "Employees": "Rose Fuller",
    "Designation": "CFO",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 1,
    "EmployeeImg": "usermale",
    "CurrentSalary": 16477,
    "Address": "2817 Milton Dr.",
    "Mail": "rose55@rpy.com"
  },
  {
    "EmployeeID": 10004,
    "Employees": "Jack Bergs",
    "Designation": "Manager",
    "Location": "Mexico",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 36,
    "EmployeeImg": "usermale",
    "CurrentSalary": 49040,
    "Address": "2, rue du Commerce",
    "Mail": "jack30@sample.com"
  },
  {
    "EmployeeID": 10005,
    "Employees": "Vinet Bergs",
    "Designation": "Program Directory",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 39,
    "EmployeeImg": "usermale",
    "CurrentSalary": 5495,
    "Address": "Rua da Panificadora, 12",
    "Mail": "vinet32@jourrapide.com"
  },
  {
    "EmployeeID": 10006,
    "Employees": "Buchanan Van",

```

```

    "Designation": "Designer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 42182,
    "Address": "24, place Kléber",
    "Mail": "buchanan18@mail.com"
  },
  {
    "EmployeeID": 10007,
    "Employees": "Dodsworth Nancy",
    "Designation": "Project Lead",
    "Location": "USA",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 35776,
    "Address": "Rua do Paço, 67",
    "Mail": "dodsworth84@mail.com"
  },
  {
    "EmployeeID": 10008,
    "Employees": "Laura Jack",
    "Designation": "Developer",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 89,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25108,
    "Address": "Rua da Panificadora, 12",
    "Mail": "laura82@mail.com"
  },
  {
    "EmployeeID": 10009,
    "Employees": "Anne Fuller",
    "Designation": "Program Directory",
    "Location": "Mexico",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 32568,
    "Address": "Gran Vía, 1",
    "Mail": "anne97@jourrapide.com"
  },
  {
    "EmployeeID": 10010,
    "Employees": "Buchanan Andrew",
    "Designation": "Designer",

```

```

    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 62,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 12320,
    "Address": "P.O. Box 555",
    "Mail": "buchanan50@jourrapide.com"
  },
  {
    "EmployeeID": 10011,
    "Employees": "Andrew Janet",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 20890,
    "Address": "Starenweg 5",
    "Mail": "andrew63@mail.com"
  },
  {
    "EmployeeID": 10012,
    "Employees": "Margaret Tamer",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 7,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 22337,
    "Address": "Magazinweg 7",
    "Mail": "margaret26@mail.com"
  },
  {
    "EmployeeID": 10013,
    "Employees": "Tamer Fuller",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 89181,
    "Address": "Taucherstraße 10",
    "Mail": "tamer40@arpy.com"
  },
  {
    "EmployeeID": 10014,
    "Employees": "Tamer Anne",
    "Designation": "CFO",
    "Location": "Sweden",

```

```

    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 18,
    "EmployeeImg": "usermale",
    "CurrentSalary": 20998,
    "Address": "Taucherstraße 10",
    "Mail": "tamer68@arpy.com"
  },
  {
    "EmployeeID": 10015,
    "Employees": "Anton Davolio",
    "Designation": "Project Lead",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 4,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 48232,
    "Address": "Luisenstr. 48",
    "Mail": "anton46@mail.com"
  },
  {
    "EmployeeID": 10016,
    "Employees": "Buchanan Buchanan",
    "Designation": "System Analyst",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "usermale",
    "CurrentSalary": 43041,
    "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
    "Mail": "buchanan68@mail.com"
  },
  {
    "EmployeeID": 10017,
    "Employees": "King Buchanan",
    "Designation": "Program Directory",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 44,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 25259,
    "Address": "Magazinweg 7",
    "Mail": "king80@jourrapide.com"
  },
  {
    "EmployeeID": 10018,
    "Employees": "Rose Michael",
    "Designation": "Project Lead",
    "Location": "Canada",
    "Status": "Active",

```

```

    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 31,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 91156,
    "Address": "Fauntleroy Circus",
    "Mail": "rose75@mail.com"
  },
  {
    "EmployeeID": 10019,
    "Employees": "King Bergs",
    "Designation": "Developer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 2,
    "Software": 29,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 28826,
    "Address": "2817 Milton Dr.",
    "Mail": "king57@jourrapide.com"
  },
  {
    "EmployeeID": 10020,
    "Employees": "Davolio Fuller",
    "Designation": "Designer",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 3,
    "Software": 35,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 71035,
    "Address": "Gran Vía, 1",
    "Mail": "davolio29@arpy.com"
  },
  {
    "EmployeeID": 10021,
    "Employees": "Rose Rose",
    "Designation": "CFO",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 38,
    "EmployeeImg": "usermale",
    "CurrentSalary": 68123,
    "Address": "Rua do Mercado, 12",
    "Mail": "rose54@arpy.com"
  },
  {
    "EmployeeID": 10022,
    "Employees": "Andrew Michael",
    "Designation": "Program Directory",
    "Location": "UK",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",

```

```
    "Rating": 2,  
    "Software": 61,  
    "EmployeeImg": "userfemale",  
    "CurrentSalary": 75470,  
    "Address": "2, rue du Commerce",  
    "Mail": "andrew88@jourrapide.com"  
  },  
  {  
    "EmployeeID": 10023,  
    "Employees": "Davolio Kathryn",  
    "Designation": "Manager",  
    "Location": "Germany",  
    "Status": "Active",  
    "Trustworthiness": "Perfect",  
    "Rating": 3,  
    "Software": 25,  
    "EmployeeImg": "usermale",  
    "CurrentSalary": 25234,  
    "Address": "Hauptstr. 31",  
    "Mail": "davolio42@sample.com"  
  },  
  {  
    "EmployeeID": 10024,  
    "Employees": "Anne Fleet",  
    "Designation": "System Analyst",  
    "Location": "UK",  
    "Status": "Active",  
    "Trustworthiness": "Perfect",  
    "Rating": 3,  
    "Software": 0,  
    "EmployeeImg": "userfemale",  
    "CurrentSalary": 8341,  
    "Address": "59 rue de l'Abbaye",  
    "Mail": "anne86@arpy.com"  
  },  
  {  
    "EmployeeID": 10025,  
    "Employees": "Margaret Andrew",  
    "Designation": "System Analyst",  
    "Location": "Germany",  
    "Status": "Inactive",  
    "Trustworthiness": "Insufficient",  
    "Rating": 3,  
    "Software": 51,  
    "EmployeeImg": "userfemale",  
    "CurrentSalary": 84975,  
    "Address": "P.O. Box 555",  
    "Mail": "margaret41@arpy.com"  
  },  
  {  
    "EmployeeID": 10026,  
    "Employees": "Kathryn Laura",  
    "Designation": "Project Lead",  
    "Location": "Austria",  
    "Status": "Active",  
    "Trustworthiness": "Insufficient",  
    "Rating": 3,
```

```

    "Software": 48,
    "EmployeeImg": "usermale",
    "CurrentSalary": 97282,
    "Address": "Avda. Azteca 123",
    "Mail": "kathryn82@rpy.com"
  },
  {
    "EmployeeID": 10027,
    "Employees": "Michael Michael",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 16,
    "EmployeeImg": "usermale",
    "CurrentSalary": 4184,
    "Address": "Rua do Paço, 67",
    "Mail": "michael58@jourrapide.com"
  },
  {
    "EmployeeID": 10028,
    "Employees": "Leverling Vinet",
    "Designation": "Project Lead",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 57,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 38370,
    "Address": "59 rue de l'Abbaye",
    "Mail": "leverling102@sample.com"
  },
  {
    "EmployeeID": 10029,
    "Employees": "Rose Jack",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 46,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84790,
    "Address": "Rua do Mercado, 12",
    "Mail": "rosl08@jourrapide.com"
  },
  {
    "EmployeeID": 10030,
    "Employees": "Vinet Van",
    "Designation": "Developer",
    "Location": "USA",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 40,

```



```

        "EmployeeImg": "usermale",
        "CurrentSalary": 71005,
        "Address": "Gran Vía, 1",
        "Mail": "vinet90@jourrapide.com"
    }
];

```

### DATASOURCE.TSX

```

/**
 * Default data source
 */
export let defaultData: Object[] = [
    { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
      Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
    { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
      Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
    { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
      Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
    { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
      AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
    { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
      '12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
      70 },
    { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
      Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
    { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
      Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
    { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
      Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
    { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
      Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
    { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
      Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
/**
 * Grid datasource
 */
export function getTradeData(dataCount?: number): object {
    let employees: string[] = [
        'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
        'Fuller', 'Leverling', 'Peacock',
        'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
        'Dodsworth', 'Anne',
        'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
        'Jack', 'Rose'];
    let designation: string[] = ['Manager', 'CFO', 'Designer', 'Developer',
        'Program Directory', 'System Analyst', 'Project Lead'];
    let mail: string[] = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
        'jourrapide.com'];
    let location: string[] = ['UK', 'USA', 'Sweden', 'France', 'Canada',
        'Argentina', 'Austria', 'Germany', 'Mexico'];
    let status: string[] = ['Active', 'Inactive'];
    let trustworthiness: string[] = ['Perfect', 'Sufficient',
        'Insufficient'];
    let tradeData: Object[] = [];

```

```

let address: string[] = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do
Paço, 67', '2, rue du Commerce', 'Boulevard Tirou, 255',
'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
Mercado, 12',
'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
'Taucherstraße 10', 'Taucherstraße 10',
'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
Circus', 'Av. dos Lusíadas, 23',
'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
123', '2817 Milton Dr.'];
let employeeimg: string[] = ['usermale', 'userfemale'];
if (typeof dataCount === 'string') {
    dataCount = parseInt(dataCount, 10);
}
for (let i: number = 1; i <= dataCount; i++) {
    let code: any = 10000;
    tradeData.push({
        'EmployeeID': code + i,
        'Employees':
            employees[Math.floor(Math.random() * employees.length)] + '
' + employees[Math.floor(Math.random() * employees.length)],
        'Designation': designation[Math.floor(Math.random() *
designation.length)],
        'Location': location[Math.floor(Math.random() *
location.length)],
        'Status': status[Math.floor(Math.random() * status.length)],
        'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
        'Rating': Math.floor(Math.random() * 5),
        'Software': Math.floor(Math.random() * 100),
        'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
        'CurrentSalary': Math.floor((Math.random() * 100000)),
        'Address': address[Math.floor(Math.random() * address.length)],
    });
    let employee: string = 'Employees';
    let emp: string = tradeData[i - 1][employee];
    let sName: string = emp.substr(0, emp.indexOf(' ')).toLowerCase();
    let empmail: string = 'Mail';
    tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
}
return tradeData;
}
export let tradeData: Object[] = [
    {
        "EmployeeID": 10001,
        "Employees": "Laura Nancy",
    }
]

```

```
"Designation": "Designer",
"Location": "France",
"Status": "Inactive",
"Trustworthiness": "Sufficient",
"Rating": 0,
"Software": 69,
"EmployeeImg": "usermale",
"CurrentSalary": 84194,
"Address": "Taucherstraße 10",
"Mail": "laura15@jourrapide.com"
},
{
  "EmployeeID": 10002,
  "Employees": "Zachery Van",
  "Designation": "CFO",
  "Location": "Canada",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 99,
  "EmployeeImg": "usermale",
  "CurrentSalary": 55349,
  "Address": "5a Ave. Los Palos Grandes",
  "Mail": "zachery109@sample.com"
},
{
  "EmployeeID": 10003,
  "Employees": "Rose Fuller",
  "Designation": "CFO",
  "Location": "France",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 1,
  "EmployeeImg": "usermale",
  "CurrentSalary": 16477,
  "Address": "2817 Milton Dr.",
  "Mail": "rose55@rpy.com"
},
{
  "EmployeeID": 10004,
  "Employees": "Jack Bergs",
  "Designation": "Manager",
  "Location": "Mexico",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 36,
  "EmployeeImg": "usermale",
  "CurrentSalary": 49040,
  "Address": "2, rue du Commerce",
  "Mail": "jack30@sample.com"
},
{
  "EmployeeID": 10005,
  "Employees": "Vinet Bergs",
  "Designation": "Program Directory",
```

```
"Location": "UK",
"Status": "Active",
"Trustworthiness": "Insufficient",
"Rating": 1,
"Software": 39,
"EmployeeImg": "usermale",
"CurrentSalary": 5495,
"Address": "Rua da Panificadora, 12",
"Mail": "vinet32@jourrapide.com"
},
{
  "EmployeeID": 10006,
  "Employees": "Buchanan Van",
  "Designation": "Designer",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 4,
  "Software": 78,
  "EmployeeImg": "usermale",
  "CurrentSalary": 42182,
  "Address": "24, place Kléber",
  "Mail": "buchanan18@mail.com"
},
{
  "EmployeeID": 10007,
  "Employees": "Dodsworth Nancy",
  "Designation": "Project Lead",
  "Location": "USA",
  "Status": "Inactive",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 0,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 35776,
  "Address": "Rua do Paço, 67",
  "Mail": "dodsworth84@mail.com"
},
{
  "EmployeeID": 10008,
  "Employees": "Laura Jack",
  "Designation": "Developer",
  "Location": "Austria",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 89,
  "EmployeeImg": "usermale",
  "CurrentSalary": 25108,
  "Address": "Rua da Panificadora, 12",
  "Mail": "laura82@mail.com"
},
{
  "EmployeeID": 10009,
  "Employees": "Anne Fuller",
  "Designation": "Program Directory",
  "Location": "Mexico",
```

```

    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 32568,
    "Address": "Gran Vía, 1",
    "Mail": "anne97@jourrapide.com"
  },
  {
    "EmployeeID": 10010,
    "Employees": "Buchanan Andrew",
    "Designation": "Designer",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 62,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 12320,
    "Address": "P.O. Box 555",
    "Mail": "buchanan50@jourrapide.com"
  },
  {
    "EmployeeID": 10011,
    "Employees": "Andrew Janet",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 20890,
    "Address": "Starenweg 5",
    "Mail": "andrew63@mail.com"
  },
  {
    "EmployeeID": 10012,
    "Employees": "Margaret Tamer",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 7,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 22337,
    "Address": "Magazinweg 7",
    "Mail": "margaret26@mail.com"
  },
  {
    "EmployeeID": 10013,
    "Employees": "Tamer Fuller",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Active",

```

```

    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 89181,
    "Address": "Taucherstraße 10",
    "Mail": "tamer40@arpy.com"
  },
  {
    "EmployeeID": 10014,
    "Employees": "Tamer Anne",
    "Designation": "CFO",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 18,
    "EmployeeImg": "usermale",
    "CurrentSalary": 20998,
    "Address": "Taucherstraße 10",
    "Mail": "tamer68@arpy.com"
  },
  {
    "EmployeeID": 10015,
    "Employees": "Anton Davolio",
    "Designation": "Project Lead",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 4,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 48232,
    "Address": "Luisenstr. 48",
    "Mail": "anton46@mail.com"
  },
  {
    "EmployeeID": 10016,
    "Employees": "Buchanan Buchanan",
    "Designation": "System Analyst",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "usermale",
    "CurrentSalary": 43041,
    "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
    "Mail": "buchanan68@mail.com"
  },
  {
    "EmployeeID": 10017,
    "Employees": "King Buchanan",
    "Designation": "Program Directory",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",

```

```
"Rating": 0,  
"Software": 44,  
"EmployeeImg": "userfemale",  
"CurrentSalary": 25259,  
"Address": "Magazinweg 7",  
"Mail": "king80@jourrapide.com"  
},  
{  
  "EmployeeID": 10018,  
  "Employees": "Rose Michael",  
  "Designation": "Project Lead",  
  "Location": "Canada",  
  "Status": "Active",  
  "Trustworthiness": "Perfect",  
  "Rating": 4,  
  "Software": 31,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 91156,  
  "Address": "Fauntleroy Circus",  
  "Mail": "rose75@mail.com"  
},  
{  
  "EmployeeID": 10019,  
  "Employees": "King Bergs",  
  "Designation": "Developer",  
  "Location": "Germany",  
  "Status": "Active",  
  "Trustworthiness": "Sufficient",  
  "Rating": 2,  
  "Software": 29,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 28826,  
  "Address": "2817 Milton Dr.",  
  "Mail": "king57@jourrapide.com"  
},  
{  
  "EmployeeID": 10020,  
  "Employees": "Davolio Fuller",  
  "Designation": "Designer",  
  "Location": "Canada",  
  "Status": "Inactive",  
  "Trustworthiness": "Sufficient",  
  "Rating": 3,  
  "Software": 35,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 71035,  
  "Address": "Gran Vía, 1",  
  "Mail": "davolio29@arpy.com"  
},  
{  
  "EmployeeID": 10021,  
  "Employees": "Rose Rose",  
  "Designation": "CFO",  
  "Location": "Germany",  
  "Status": "Active",  
  "Trustworthiness": "Perfect",  
  "Rating": 3,
```

```
"Software": 38,  
"EmployeeImg": "usermale",  
"CurrentSalary": 68123,  
"Address": "Rua do Mercado, 12",  
"Mail": "rose54@arpy.com"  
},  
{  
  "EmployeeID": 10022,  
  "Employees": "Andrew Michael",  
  "Designation": "Program Directory",  
  "Location": "UK",  
  "Status": "Inactive",  
  "Trustworthiness": "Insufficient",  
  "Rating": 2,  
  "Software": 61,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 75470,  
  "Address": "2, rue du Commerce",  
  "Mail": "andrew88@jourrapide.com"  
},  
{  
  "EmployeeID": 10023,  
  "Employees": "Davolio Kathryn",  
  "Designation": "Manager",  
  "Location": "Germany",  
  "Status": "Active",  
  "Trustworthiness": "Perfect",  
  "Rating": 3,  
  "Software": 25,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 25234,  
  "Address": "Hauptstr. 31",  
  "Mail": "davolio42@sample.com"  
},  
{  
  "EmployeeID": 10024,  
  "Employees": "Anne Fleet",  
  "Designation": "System Analyst",  
  "Location": "UK",  
  "Status": "Active",  
  "Trustworthiness": "Perfect",  
  "Rating": 3,  
  "Software": 0,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 8341,  
  "Address": "59 rue de l'Abbaye",  
  "Mail": "anne86@arpy.com"  
},  
{  
  "EmployeeID": 10025,  
  "Employees": "Margaret Andrew",  
  "Designation": "System Analyst",  
  "Location": "Germany",  
  "Status": "Inactive",  
  "Trustworthiness": "Insufficient",  
  "Rating": 3,  
  "Software": 51,
```



```
"EmployeeImg": "userfemale",
"CurrentSalary": 84975,
"Address": "P.O. Box 555",
"Mail": "margaret41@arpy.com"
},
{
  "EmployeeID": 10026,
  "Employees": "Kathryn Laura",
  "Designation": "Project Lead",
  "Location": "Austria",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 48,
  "EmployeeImg": "usermale",
  "CurrentSalary": 97282,
  "Address": "Avda. Azteca 123",
  "Mail": "kathryn82@rpy.com"
},
{
  "EmployeeID": 10027,
  "Employees": "Michael Michael",
  "Designation": "Developer",
  "Location": "UK",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 4,
  "Software": 16,
  "EmployeeImg": "usermale",
  "CurrentSalary": 4184,
  "Address": "Rua do Paço, 67",
  "Mail": "michael58@jourrapide.com"
},
{
  "EmployeeID": 10028,
  "Employees": "Leverling Vinet",
  "Designation": "Project Lead",
  "Location": "Germany",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 57,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 38370,
  "Address": "59 rue de l'Abbaye",
  "Mail": "leverling102@sample.com"
},
{
  "EmployeeID": 10029,
  "Employees": "Rose Jack",
  "Designation": "Developer",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 46,
  "EmployeeImg": "userfemale",
```

```

    "CurrentSalary": 84790,
    "Address": "Rua do Mercado, 12",
    "Mail": "rose108@jourrapide.com"
  },
  {
    "EmployeeID": 10030,
    "Employees": "Vinet Van",
    "Designation": "Developer",
    "Location": "USA",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 40,
    "EmployeeImg": "usermale",
    "CurrentSalary": 71005,
    "Address": "Gran Vía, 1",
    "Mail": "vinet90@jourrapide.com"
  }
]

```

The cell data binding also supports formula, style, number format, and more.

#### Dynamic data binding and Datasource change event

You can dynamically change the datasource of the spreadsheet by changing the `dataSource` property of the `range` object of the `sheet`. The `dataSourceChanged` event handler will be triggered when editing, inserting, and deleting a row in the datasource range. This event will be triggered with a parameter named `action` which indicates the `edit`, `add` and `delete` actions for the respective ones.

The following table defines the arguments of the `dataSourceChanged` event.

Property	Type	Description
action	string	Indicates the type of action such as <code>edit</code> , <code>add</code> , and <code>delete</code> performed in the datasource range.
data	object[]	Modified data for <code>edit</code> action; New data for <code>add</code> action; Deleted data for <code>delete</code> action.
rangeIndex	number	Specifies the range index of the datasource.
sheetIndex	number	Specifies the sheet index of the datasource.

For `add` action, the value for all the fields will be `null` in the data. In the case that you do not want the primary key field to be null which needs to be updated in the backend service, you can use `edit` action after updating the primary key field to update in the backend service. <br><br>

For inserting a row at the end of the datasource range, you should insert a row below at the end of the range to trigger the `dataSourceChanged` event with action `add`.

#### APP.JSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';

```

```

import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { defaultData, itemData } from './datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  const dataSourceChanged = (args) => {
    appendElement(`Data source changed with <b>#160;${args.action}</b>
action<hr>`);
  };
  const btnClick = () => {
    const spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.sheets[0].ranges[0].dataSource = itemData;
    }
  };
  const clearBtnClick = () => {
    const eventLog = document.getElementById('EventLog');
    if (eventLog) {
      eventLog.innerHTML = "";
    }
  };
  const appendElement = (html) => {
    const span = document.createElement("span");
    span.innerHTML = html;
    const log = document.getElementById('EventLog');
    if (log) {
      log.insertBefore(span, log.firstChild);
    }
  };
  return (
    <div>
      <div><button id="changeDataBtn" className='e-btn'
onClick={btnClick}>Change Datasource</button>
      <SpreadsheetComponent ref={spreadsheetRef}
dataSourceChanged={dataSourceChanged}>
        <SheetsDirective>
          <SheetDirective>
            <RangesDirective>
              <RangeDirective dataSource={defaultData}></RangeDirective>
            </RangesDirective>
            <ColumnsDirective>
              <ColumnDirective width={180}></ColumnDirective>
              <ColumnDirective width={130}></ColumnDirective>
              <ColumnDirective width={130}></ColumnDirective>
              <ColumnDirective width={180}></ColumnDirective>
              <ColumnDirective width={130}></ColumnDirective>
              <ColumnDirective width={120}></ColumnDirective>
            </ColumnsDirective>
          </SheetDirective>
        </SheetsDirective>
      </SpreadsheetComponent>
    </div>
    <div>
      <h4><b>Event Trace</b></h4>
      <div id="evt">

```

```

        <div>
            <span id="EventLog"></span>
        </div>
        <button id="clearBtn" className='e-btn'
onClick={clearBtnClick}>Clear</button>
    </div>
</div>
</div>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, DataSourceChangedEventArgs, } from '@syncfusion/ej2-react-
spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { defaultData, itemData } from './datasource';
function App() {
    const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
    const dataSourceChanged = (args: DataSourceChangedEventArgs) => {
        appendElement(`Data source changed with <b>#${160};${args.action}</b>
action<hr>`);
    };
    const btnClick = (): void => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            spreadsheet.sheets[0].ranges[0].dataSource = itemData;
        }
    };
    const clearBtnClick = (): void => {
        const eventLog: HTMLElement = document.getElementById('EventLog')!;
        if (eventLog) {
            eventLog.innerHTML = "";
        }
    };
    const appendElement = (html: string): void => {
        const span: HTMLSpanElement = document.createElement("span");
        span.innerHTML = html;
        const log: HTMLElement = document.getElementById('EventLog')!;
        if (log) {
            log.insertBefore(span, log.firstChild);
        }
    };
    return (
        <div>
            <div>

```

```

    <button id="changeDataBtn" className='e-btn'
onClick={btnClick}>Change Datasource</button>
    <SpreadsheetComponent ref={spreadsheetRef}
dataSourceChanged={dataSourceChanged}>
      <SheetsDirective>
        <SheetDirective>
          <RangesDirective>
            <RangeDirective dataSource={defaultData}></RangeDirective>
          </RangesDirective>
          <ColumnsDirective>
            <ColumnDirective width={180}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
            <ColumnDirective width={180}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
            <ColumnDirective width={120}></ColumnDirective>
          </ColumnsDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  </div>
  <div>
    <h4><b>Event Trace</b></h4>
    <div id="evt">
      <div>
        <span id="EventLog"></span>
      </div>
      <button id="clearBtn" className='e-btn'
onClick={clearBtnClick}>Clear</button>
    </div>
  </div>
</div>
);
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

## DATASOURCE.JSX

```

/**
 * Default data source
 */
export let defaultData = [{
  OrderID: 10248,
  CustomerID: 'VINET',
  EmployeeID: 5,
  ShipName: 'Vins et alcools Chevalier',
  ShipCity: 'Reims',
  ShipAddress: '59 rue de l'Abbaye'
},
{
  OrderID: 10249,
  CustomerID: 'TOMSP',
  EmployeeID: 6,

```

```

      ShipName: 'Toms Spezialitäten',
      ShipCity: 'Münster',
      ShipAddress: 'Luisenstr. 48'
    },
    {
      OrderID: 10250,
      CustomerID: 'HANAR',
      EmployeeID: 4,
      ShipName: 'Hanari Carnes',
      ShipCity: 'Rio de Janeiro',
      ShipAddress: 'Rua do Paço, 67'
    },
    {
      OrderID: 10251,
      CustomerID: 'VICTE',
      EmployeeID: 3,
      ShipName: 'Victuailles en stock',
      ShipCity: 'Lyon',
      ShipAddress: '2, rue du Commerce'
    },
    {
      OrderID: 10252,
      CustomerID: 'SUPRD',
      EmployeeID: 4,
      ShipName: 'Suprêmes délices',
      ShipCity: 'Charleroi',
      ShipAddress: 'Boulevard Tirou, 255'
    }
  ]];
export let itemData = [{
  'Item Name': 'Casual Shoes',
  Date: '02/14/2019',
  Time: '11:34:32 AM',
  Quantity: 10,
  Price: 20,
  Amount: '=D2*E2',
  Discount: 1,
  Profit: 10
},
{
  'Item Name': 'Sports Shoes',
  Date: '06/11/2019',
  Time: '05:56:32 AM',
  Quantity: 20,
  Price: 30,
  Amount: '=D3*E3',
  Discount: 5,
  Profit: 50
},
{
  'Item Name': 'Formal Shoes',
  Date: '07/27/2019',
  Time: '03:32:44 AM',
  Quantity: 20,
  Price: 15,
  Amount: '=D4*E4',
  Discount: 7,
  Profit: 27
}

```

```

    },
    {
      'Item Name': 'Sandals & Floaters',
      Date: '11/21/2019',
      Time: '06:23:54 AM',
      Quantity: 15,
      Price: 20,
      Amount: '=D5*E5',
      Discount: 11,
      Profit: 67
    },
    {
      'Item Name': 'Flip- Flops & Slippers',
      Date: '06/23/2019',
      Time: '12:43:59 AM',
      Quantity: 30,
      Price: 10,
      Amount: '=D6*E6',
      Discount: 10,
      Profit: 70
    },
    {
      'Item Name': 'Sneakers',
      Date: '07/22/2019',
      Time: '10:55:53 AM',
      Quantity: 40,
      Price: 20,
      Amount: '=D7*E7',
      Discount: 13,
      Profit: 66
    },
    {
      'Item Name': 'Running Shoes',
      Date: '02/04/2019',
      Time: '03:44:34 AM',
      Quantity: 20,
      Price: 10,
      Amount: '=D8*E8',
      Discount: 3,
      Profit: 14
    },
    {
      'Item Name': 'Loafers',
      Date: '11/30/2019',
      Time: '03:12:52 AM',
      Quantity: 31,
      Price: 10,
      Amount: '=D9*E9',
      Discount: 6,
      Profit: 29
    },
    {
      'Item Name': 'Cricket Shoes',
      Date: '07/09/2019',
      Time: '11:32:14 AM',
      Quantity: 41,
      Price: 30,

```

```

        Amount: '=D10*E10',
        Discount: 12,
        Profit: 166
    },
    {
        'Item Name': 'T-Shirts',
        Date: '10/31/2019',
        Time: '12:01:44 AM',
        Quantity: 50,
        Price: 10,
        Amount: '=D11*E11',
        Discount: 9,
        Profit: 55
    }
];

```

### DATASOURCE.TSX

```

/**
 * Default data source
 */
export let defaultData: Object[] = [{
    OrderID: 10248,
    CustomerID: 'VINET',
    EmployeeID: 5,
    ShipName: 'Vins et alcools Chevalier',
    ShipCity: 'Reims',
    ShipAddress: '59 rue de l'Abbaye'
},
{
    OrderID: 10249,
    CustomerID: 'TOMSP',
    EmployeeID: 6,
    ShipName: 'Toms Spezialitäten',
    ShipCity: 'Münster',
    ShipAddress: 'Luisenstr. 48'
},
{
    OrderID: 10250,
    CustomerID: 'HANAR',
    EmployeeID: 4,
    ShipName: 'Hanari Carnes',
    ShipCity: 'Rio de Janeiro',
    ShipAddress: 'Rua do Paço, 67'
},
{
    OrderID: 10251,
    CustomerID: 'VICTE',
    EmployeeID: 3,
    ShipName: 'Victuailles en stock',
    ShipCity: 'Lyon',
    ShipAddress: '2, rue du Commerce'
},
{
    OrderID: 10252,
    CustomerID: 'SUPRD',

```



```

EmployeeID: 4,
ShipName: 'Suprêmes délices',
ShipCity: 'Charleroi',
ShipAddress: 'Boulevard Tirou, 255'
}];
export let itemData : Object[] = [{
  'Item Name': 'Casual Shoes',
  Date: '02/14/2019',
  Time: '11:34:32 AM',
  Quantity: 10,
  Price: 20,
  Amount: '=D2*E2',
  Discount: 1,
  Profit: 10
},
{
  'Item Name': 'Sports Shoes',
  Date: '06/11/2019',
  Time: '05:56:32 AM',
  Quantity: 20,
  Price: 30,
  Amount: '=D3*E3',
  Discount: 5,
  Profit: 50
},
{
  'Item Name': 'Formal Shoes',
  Date: '07/27/2019',
  Time: '03:32:44 AM',
  Quantity: 20,
  Price: 15,
  Amount: '=D4*E4',
  Discount: 7,
  Profit: 27
},
{
  'Item Name': 'Sandals & Floaters',
  Date: '11/21/2019',
  Time: '06:23:54 AM',
  Quantity: 15,
  Price: 20,
  Amount: '=D5*E5',
  Discount: 11,
  Profit: 67
},
{
  'Item Name': 'Flip- Flops & Slippers',
  Date: '06/23/2019',
  Time: '12:43:59 AM',
  Quantity: 30,
  Price: 10,
  Amount: '=D6*E6',
  Discount: 10,
  Profit: 70
},
{
  'Item Name': 'Sneakers',

```

```

    Date: '07/22/2019',
    Time: '10:55:53 AM',
    Quantity: 40,
    Price: 20,
    Amount: '=D7*E7',
    Discount: 13,
    Profit: 66
  },
  {
    'Item Name': 'Running Shoes',
    Date: '02/04/2019',
    Time: '03:44:34 AM',
    Quantity: 20,
    Price: 10,
    Amount: '=D8*E8',
    Discount: 3,
    Profit: 14
  },
  {
    'Item Name': 'Loafers',
    Date: '11/30/2019',
    Time: '03:12:52 AM',
    Quantity: 31,
    Price: 10,
    Amount: '=D9*E9',
    Discount: 6,
    Profit: 29
  },
  {
    'Item Name': 'Cricket Shoes',
    Date: '07/09/2019',
    Time: '11:32:14 AM',
    Quantity: 41,
    Price: 30,
    Amount: '=D10*E10',
    Discount: 12,
    Profit: 166
  },
  {
    'Item Name': 'T-Shirts',
    Date: '10/31/2019',
    Time: '12:01:44 AM',
    Quantity: 50,
    Price: 10,
    Amount: '=D11*E11',
    Discount: 9,
    Profit: 55
  }
];

```

### Note

You can refer to our [React Spreadsheet](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Spreadsheet example](#) to know how to present and manipulate data.

## See Also

- [Filtering](#)
- [Sorting](#)
- [Hyperlink](#)
- [Link to the Video](#)

## Open save in React Spreadsheet component

In import an excel file, it needs to be read and converted to client side Spreadsheet model. The converted client side Spreadsheet model is sent as JSON which is used to render Spreadsheet. Similarly, when you save the Spreadsheet, the client Spreadsheet model is sent to the server as JSON for processing and saved. Server configuration is used for this process.

To get start quickly with Open and Save, you can check on this video:

## Open

The Spreadsheet control opens an Excel document with its data, style, format, and more. To enable this feature, set [allowOpen](#) as `true` and assign service url to the [openUrl](#) property.

## User Interface:

In user interface you can open an Excel document by clicking **File > Open** menu item in ribbon.

The following sample shows the **Open** option by using the [openUrl](#) property in the Spreadsheet control. You can also use the [beforeOpen](#) event to trigger before opening an Excel file.

## APP.JSX

```
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
function App() {
    const beforeOpen = () => {};
    return (
        <SpreadsheetComponent allowOpen={true}
openUrl='https://services.syncfusion.com/react/production/api/spreadsheet/open' beforeOpen={beforeOpen} />
    );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
```

## APP.TSX

```
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
function App() {
    const beforeOpen = ():void => {};
    return (
        <SpreadsheetComponent allowOpen={true}
openUrl='https://services.syncfusion.com/react/production/api/spreadsheet/open' beforeOpen={beforeOpen} />
    );
};
```

```
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
```

Please find the below table for the beforeOpen event arguments.

Parameter	Type	Description
file	FileList or string or File	To get the file stream. <b>FileList</b> - contains length and item index. <b>File</b> - specifies the file lastModified and file name.
cancel	boolean	To prevent the open operation.
requestData	object	To provide the Form data.

\* Use **Ctrl + O** keyboard shortcut to open Excel documents.

\* The default value of the [allowOpen](#) property is **true**. For demonstration purpose, we have showcased the [allowOpen](#) property in previous code snippet.

#### Open an external URL excel file while initial load

You can achieve to access the remote excel file by using the [created](#) event. In this event you can fetch the excel file and convert it to a blob. Convert this blob to a file and [open](#) this file by using Spreadsheet component open method.

#### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
function App() {
  const spreadsheetRef = React.useRef(null);
  React.useEffect(() => {
    const fetchData = async () => {
      const response = await
fetch('https://cdn.syncfusion.com/scripts/spreadsheet/Sample.xlsx'); //
fetch the remote url
      const fileBlob = await response.blob(); // convert the excel
file to blob
      const file = new File([fileBlob], 'Sample.xlsx'); //convert the
blob into file
      let spreadsheet = spreadsheetRef.current;
      if (spreadsheet) {
        spreadsheet.open({ file }); // open the file into
Spreadsheet
      }
    };
    fetchData();
  }, []);
  return (
    <SpreadsheetComponent ref={spreadsheetRef}
    openUrl='https://services.syncfusion.com/react/production/api/spreadsheet/op
en' />
```

```

    );
  };
  export default App;
  const root = createRoot(document.getElementById('root'));
  root.render(<App />);
  {% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  React.useEffect(() => {
    const fetchData = async (): Promise<void> => {
      const response: Response = await
        fetch('https://cdn.syncfusion.com/scripts/spreadsheet/Sample.xlsx'); //
        fetch the remote url
      const fileBlob: Blob = await response.blob(); // convert the
      excel file to blob
      const file: File = new File([fileBlob], 'Sample.xlsx');
      //convert the blob into file
      let spreadsheet = spreadsheetRef.current;
      if (spreadsheet) {
        spreadsheet.open({ file }); // open the file into
        Spreadsheet
      }
    };
    fetchData();
  }, []);
  return (
    <SpreadsheetComponent ref={spreadsheetRef}
      openUrl='https://services.syncfusion.com/react/production/api/spreadsheet/open' />
    );
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

#### To add custom header during open

You can add your own custom header to the open action in the Spreadsheet. For processing the data, it has to be sent from server to client side and adding customer header can provide privacy to the data with the help of Authorization Token. Through the [beforeOpen](#) event, the custom header can be added to the request during open action.

### APP.JSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';

```

```
function App() {
  const beforeOpen = (args) => {
    args.requestData = { Authorization: 'YOUR TEXT' };
  };
  return (
    <SpreadsheetComponent
      openUrl="https://services.syncfusion.com/react/production/api/spreadsheet/open"
      beforeOpen={beforeOpen} />
  );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}
```

### APP.TSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { BeforeOpenEventArgs, SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
function App() {
  const beforeOpen = (args: BeforeOpenEventArgs): void => {
    args.requestData = { Authorization: 'YOUR TEXT' };
  };
  return (
    <SpreadsheetComponent
      openUrl="https://services.syncfusion.com/react/production/api/spreadsheet/open"
      beforeOpen={beforeOpen} />
  );
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}
```

### Open excel file into a read-only mode

You can open excel file into a read-only mode by using the [openComplete](#) event. In this event, you must protect all the sheets and lock its used range cells by using [protectSheet](#) and [lockCells](#) methods.

### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { getRangeAddress, SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
function App() {
  const spreadsheetRef = React.useRef(null);
  const openComplete = () => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      let sheets = spreadsheet.sheets;
      for (let index = 0; index < sheets.length; index++) {

```

```

        let name = spreadsheet.sheets[index].name;
        let protectSetting = { selectCells: true, formatCells: false
    };

        //To protect the sheet using sheet name
        spreadsheet.protectSheet(name, protectSetting);
        let address = getRangeAddress([0, 0,
sheets[index].usedRange.rowIndex, sheets[index].usedRange.colIndex,]);
        //To lock the used range cells
        spreadsheet.lockCells(name + '!' + address, true);
    }
    }
};
return (
    <SpreadsheetComponent ref={spreadsheetRef}
openComplete={openComplete}

openUrl="https://services.syncfusion.com/react/production/api/spreadsheet/op
en" />
    );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { getRangeAddress, ProtectSettingsModel, SheetModel,
SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
function App() {
    const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
    const openComplete = (): void => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            let sheets: SheetModel[] = spreadsheet.sheets;
            for (let index: number = 0; index < sheets.length; index++) {
                let name: string = spreadsheet.sheets[index].name as string;
                let protectSetting: ProtectSettingsModel = { selectCells: true,
formatCells: false };
                //To protect the sheet using sheet name
                spreadsheet.protectSheet(name, protectSetting);
                let address: string = getRangeAddress([0, 0,
sheets[index].usedRange.rowIndex as number, sheets[index].usedRange.colIndex
as number,]);
                //To lock the used range cells
                spreadsheet.lockCells(name + '!' + address, true);
            }
        }
    };
    return (
        <SpreadsheetComponent ref={spreadsheetRef} openComplete={openComplete}

```

```

openUrl="https://services.syncfusion.com/react/production/api/spreadsheet/open" />
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### External workbook confirmation dialog

When you open an excel file that contains external workbook references, you will see a confirmation dialog. This dialog allows you to either continue with the file opening or cancel the operation. This confirmation dialog will appear only if you set the `AllowExternalWorkbook` property value to **false** during the open request, as shown below. This prevents the spreadsheet from displaying inconsistent data.

```

`csharp
public IActionResult Open(IFormCollection openRequest)
{
    OpenRequest open = new OpenRequest();
    open.AllowExternalWorkbook = false;
    open.File = openRequest.Files[0];
    return Content(Workbook.Open(open));
}
`

```

This feature is only applicable when importing an Excel file and not when loading JSON data or binding cell data.

![External workbook confirmation dialog](./images/external-reference-dialog-alert%20.png)

### Supported file formats

The following list of Excel file formats are supported in Spreadsheet:

- MS Excel (.xlsx)
- MS Excel 97-2003 (.xls)
- Comma Separated Values (.csv)
- Excel Macro-Enabled Workbook (.xlsm)
- Excel Binary Workbook(.xlsb)

### Save

The Spreadsheet control saves its data, style, format, and more as Excel file document. To enable this feature, set `allowSave` as **true** and assign service url to the `saveUrl` property.

### User Interface:



In user interface, you can save Spreadsheet data as Excel document by clicking **File > Save As** menu item in ribbon.

The following sample shows the **Save** option by using the [saveUrl](#) property in the Spreadsheet control. You can also use the [beforeSave](#) event to trigger before saving the Spreadsheet as an Excel file.

#### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { defaultData } from './datasource';
function App() {
    const beforeSave = () => {};
    return (
        <SpreadsheetComponent allowSave={true}
saveUrl='https://services.syncfusion.com/react/production/api/spreadsheet/sa
ve' beforeSave={beforeSave}>
            <SheetsDirective>
                <SheetDirective>
                    <RangesDirective>
                        <RangeDirective
dataSource={defaultData}></RangeDirective>
                    </RangesDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={180}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
            </SheetsDirective>
        </SpreadsheetComponent>
    );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}
```

#### APP.TSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { defaultData } from './datasource';
function App() {
    const beforeSave = (): void => {};
    return (

```

```

    <SpreadsheetComponent allowSave={true}
    saveUrl='https://services.syncfusion.com/react/production/api/spreadsheet/save' beforeSave={beforeSave}>
      <SheetsDirective>
        <SheetDirective>
          <RangesDirective>
            <RangeDirective
dataSource={defaultData}></RangeDirective>
          </RangesDirective>
          <ColumnsDirective>
            <ColumnDirective width={180}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
          </ColumnsDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% enddraw %}

```

### DATASOURCE.JSX

```

/**
 * Default data source
 */
export let defaultData = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
  { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
  { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
  { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
'12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
70 },
  { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
  { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
  { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
  { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
  { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
/**
 * Grid datasource
 */
export function getTradeData(dataCount) {

```

```

let employees = [
  'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
  'Fuller', 'Leverling', 'Peacock',
  'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
  'Dodsworth', 'Anne',
  'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
  'Jack', 'Rose'
];
let designation = ['Manager', 'CFO', 'Designer', 'Developer', 'Program
Directory', 'System Analyst', 'Project Lead'];
let mail = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
'journrapide.com'];
let location = ['UK', 'USA', 'Sweden', 'France', 'Canada', 'Argentina',
'Austria', 'Germany', 'Mexico'];
let status = ['Active', 'Inactive'];
let trustworthiness = ['Perfect', 'Sufficient', 'Insufficient'];
let tradeData = [];
let address = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do Paço, 67',
'2, rue du Commerce', 'Boulevard Tirou, 255',
'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
Mercado, 12',
'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
'Taucherstraße 10', 'Taucherstraße 10',
'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
Circus', 'Av. dos Lusíadas, 23',
'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
123', '2817 Milton Dr.'];
let employeeimg = ['usermale', 'userfemale'];
if (typeof dataCount === 'string') {
  dataCount = parseInt(dataCount, 10);
}
for (let i = 1; i <= dataCount; i++) {
  let code = 10000;
  tradeData.push({
    'EmployeeID': code + i,
    'Employees': employees[Math.floor(Math.random() *
employees.length)] + ' ' + employees[Math.floor(Math.random() *
employees.length)],
    'Designation': designation[Math.floor(Math.random() *
designation.length)],
    'Location': location[Math.floor(Math.random() *
location.length)],
    'Status': status[Math.floor(Math.random() * status.length)],
    'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
    'Rating': Math.floor(Math.random() * 5),
    'Software': Math.floor(Math.random() * 100),

```

```

    'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
    'CurrentSalary': Math.floor((Math.random() * 100000)),
    'Address': address[Math.floor(Math.random() * address.length)],
  });
  let employee = 'Employees';
  let emp = tradeData[i - 1][employee];
  let sName = emp.substr(0, emp.indexOf(' ')).toLowerCase();
  let empmail = 'Mail';
  tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
}
return tradeData;
}
export let tradeData = [
{
  "EmployeeID": 10001,
  "Employees": "Laura Nancy",
  "Designation": "Designer",
  "Location": "France",
  "Status": "Inactive",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 69,
  "EmployeeImg": "usermale",
  "CurrentSalary": 84194,
  "Address": "Taucherstraße 10",
  "Mail": "laural5@jourrapide.com"
},
{
  "EmployeeID": 10002,
  "Employees": "Zachery Van",
  "Designation": "CFO",
  "Location": "Canada",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 99,
  "EmployeeImg": "usermale",
  "CurrentSalary": 55349,
  "Address": "5ª Ave. Los Palos Grandes",
  "Mail": "zachery109@sample.com"
},
{
  "EmployeeID": 10003,
  "Employees": "Rose Fuller",
  "Designation": "CFO",
  "Location": "France",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 1,
  "EmployeeImg": "usermale",
  "CurrentSalary": 16477,
  "Address": "2817 Milton Dr.",
  "Mail": "rose55@rpy.com"
},

```

```

{
  "EmployeeID": 10004,
  "Employees": "Jack Bergs",
  "Designation": "Manager",
  "Location": "Mexico",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 36,
  "EmployeeImg": "usermale",
  "CurrentSalary": 49040,
  "Address": "2, rue du Commerce",
  "Mail": "jack30@sample.com"
},
{
  "EmployeeID": 10005,
  "Employees": "Vinet Bergs",
  "Designation": "Program Directory",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 39,
  "EmployeeImg": "usermale",
  "CurrentSalary": 5495,
  "Address": "Rua da Panificadora, 12",
  "Mail": "vinet32@jourrapide.com"
},
{
  "EmployeeID": 10006,
  "Employees": "Buchanan Van",
  "Designation": "Designer",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 4,
  "Software": 78,
  "EmployeeImg": "usermale",
  "CurrentSalary": 42182,
  "Address": "24, place Kléber",
  "Mail": "buchanan18@mail.com"
},
{
  "EmployeeID": 10007,
  "Employees": "Dodsworth Nancy",
  "Designation": "Project Lead",
  "Location": "USA",
  "Status": "Inactive",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 0,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 35776,
  "Address": "Rua do Paço, 67",
  "Mail": "dodsworth84@mail.com"
},
{

```

```

      "EmployeeID": 10008,
      "Employees": "Laura Jack",
      "Designation": "Developer",
      "Location": "Austria",
      "Status": "Inactive",
      "Trustworthiness": "Perfect",
      "Rating": 3,
      "Software": 89,
      "EmployeeImg": "usermale",
      "CurrentSalary": 25108,
      "Address": "Rua da Panificadora, 12",
      "Mail": "laura82@mail.com"
    },
    {
      "EmployeeID": 10009,
      "Employees": "Anne Fuller",
      "Designation": "Program Directory",
      "Location": "Mexico",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 0,
      "Software": 19,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 32568,
      "Address": "Gran Vía, 1",
      "Mail": "anne97@jourrapide.com"
    },
    {
      "EmployeeID": 10010,
      "Employees": "Buchanan Andrew",
      "Designation": "Designer",
      "Location": "Austria",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 1,
      "Software": 62,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 12320,
      "Address": "P.O. Box 555",
      "Mail": "buchanan50@jourrapide.com"
    },
    {
      "EmployeeID": 10011,
      "Employees": "Andrew Janet",
      "Designation": "System Analyst",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 3,
      "Software": 8,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 20890,
      "Address": "Starenweg 5",
      "Mail": "andrew63@mail.com"
    },
    {
      "EmployeeID": 10012,

```

```

    "Employees": "Margaret Tamer",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 7,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 22337,
    "Address": "Magazinweg 7",
    "Mail": "margaret26@mail.com"
  },
  {
    "EmployeeID": 10013,
    "Employees": "Tamer Fuller",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 89181,
    "Address": "Taucherstraße 10",
    "Mail": "tamer40@arpy.com"
  },
  {
    "EmployeeID": 10014,
    "Employees": "Tamer Anne",
    "Designation": "CFO",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 18,
    "EmployeeImg": "usermale",
    "CurrentSalary": 20998,
    "Address": "Taucherstraße 10",
    "Mail": "tamer68@arpy.com"
  },
  {
    "EmployeeID": 10015,
    "Employees": "Anton Davolio",
    "Designation": "Project Lead",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 4,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 48232,
    "Address": "Luisenstr. 48",
    "Mail": "anton46@mail.com"
  },
  {
    "EmployeeID": 10016,
    "Employees": "Buchanan Buchanan",

```

```

    "Designation": "System Analyst",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "usermale",
    "CurrentSalary": 43041,
    "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
    "Mail": "buchanan68@mail.com"
  },
  {
    "EmployeeID": 10017,
    "Employees": "King Buchanan",
    "Designation": "Program Directory",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 44,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 25259,
    "Address": "Magazinweg 7",
    "Mail": "king80@jourrapide.com"
  },
  {
    "EmployeeID": 10018,
    "Employees": "Rose Michael",
    "Designation": "Project Lead",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 31,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 91156,
    "Address": "Fauntleroy Circus",
    "Mail": "rose75@mail.com"
  },
  {
    "EmployeeID": 10019,
    "Employees": "King Bergs",
    "Designation": "Developer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 2,
    "Software": 29,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 28826,
    "Address": "2817 Milton Dr.",
    "Mail": "king57@jourrapide.com"
  },
  {
    "EmployeeID": 10020,
    "Employees": "Davolio Fuller",
    "Designation": "Designer",

```



```

      "Location": "Canada",
      "Status": "Inactive",
      "Trustworthiness": "Sufficient",
      "Rating": 3,
      "Software": 35,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 71035,
      "Address": "Gran Vía, 1",
      "Mail": "davolio29@arpy.com"
    },
    {
      "EmployeeID": 10021,
      "Employees": "Rose Rose",
      "Designation": "CFO",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Perfect",
      "Rating": 3,
      "Software": 38,
      "EmployeeImg": "usermale",
      "CurrentSalary": 68123,
      "Address": "Rua do Mercado, 12",
      "Mail": "rose54@arpy.com"
    },
    {
      "EmployeeID": 10022,
      "Employees": "Andrew Michael",
      "Designation": "Program Directory",
      "Location": "UK",
      "Status": "Inactive",
      "Trustworthiness": "Insufficient",
      "Rating": 2,
      "Software": 61,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 75470,
      "Address": "2, rue du Commerce",
      "Mail": "andrew88@jourrapide.com"
    },
    {
      "EmployeeID": 10023,
      "Employees": "Davolio Kathryn",
      "Designation": "Manager",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Perfect",
      "Rating": 3,
      "Software": 25,
      "EmployeeImg": "usermale",
      "CurrentSalary": 25234,
      "Address": "Hauptstr. 31",
      "Mail": "davolio42@sample.com"
    },
    {
      "EmployeeID": 10024,
      "Employees": "Anne Fleet",
      "Designation": "System Analyst",
      "Location": "UK",

```

```

    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 8341,
    "Address": "59 rue de l'Abbaye",
    "Mail": "anne86@arpy.com"
  },
  {
    "EmployeeID": 10025,
    "Employees": "Margaret Andrew",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 51,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84975,
    "Address": "P.O. Box 555",
    "Mail": "margaret41@arpy.com"
  },
  {
    "EmployeeID": 10026,
    "Employees": "Kathryn Laura",
    "Designation": "Project Lead",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 48,
    "EmployeeImg": "usermale",
    "CurrentSalary": 97282,
    "Address": "Avda. Azteca 123",
    "Mail": "kathryn82@rpy.com"
  },
  {
    "EmployeeID": 10027,
    "Employees": "Michael Michael",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 16,
    "EmployeeImg": "usermale",
    "CurrentSalary": 4184,
    "Address": "Rua do Paço, 67",
    "Mail": "michael58@jourrapide.com"
  },
  {
    "EmployeeID": 10028,
    "Employees": "Leverling Vinet",
    "Designation": "Project Lead",
    "Location": "Germany",
    "Status": "Inactive",

```

```

        "Trustworthiness": "Perfect",
        "Rating": 0,
        "Software": 57,
        "EmployeeImg": "userfemale",
        "CurrentSalary": 38370,
        "Address": "59 rue de l'Abbaye",
        "Mail": "leverling102@sample.com"
    },
    {
        "EmployeeID": 10029,
        "Employees": "Rose Jack",
        "Designation": "Developer",
        "Location": "UK",
        "Status": "Active",
        "Trustworthiness": "Perfect",
        "Rating": 0,
        "Software": 46,
        "EmployeeImg": "userfemale",
        "CurrentSalary": 84790,
        "Address": "Rua do Mercado, 12",
        "Mail": "rose108@jourrapide.com"
    },
    {
        "EmployeeID": 10030,
        "Employees": "Vinet Van",
        "Designation": "Developer",
        "Location": "USA",
        "Status": "Active",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
        "Software": 40,
        "EmployeeImg": "usermale",
        "CurrentSalary": 71005,
        "Address": "Gran Vía, 1",
        "Mail": "vinet90@jourrapide.com"
    }
];

```

## DATASOURCE.TSX

```

/**
 * Default data source
 */
export let defaultData: Object[] = [
    { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
      Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
    { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
      Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
    { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
      Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
    { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54 AM',
      Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
    { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time: '12:43:59 AM',
      Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit: 70 },
];

```

```

    { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
      Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
    { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
      Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
    { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
      Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
    { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
      Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
    { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
      Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
  ];
  /**
   * Grid datasource
   */
  export function getTradeData(dataCount?: number): object {
    let employees: string[] = [
      'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
      'Fuller', 'Leverling', 'Peacock',
      'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
      'Dodsworth', 'Anne',
      'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
      'Jack', 'Rose'];
    let designation: string[] = ['Manager', 'CFO', 'Designer', 'Developer',
      'Program Directory', 'System Analyst', 'Project Lead'];
    let mail: string[] = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
      'journrapide.com'];
    let location: string[] = ['UK', 'USA', 'Sweden', 'France', 'Canada',
      'Argentina', 'Austria', 'Germany', 'Mexico'];
    let status: string[] = ['Active', 'Inactive'];
    let trustworthiness: string[] = ['Perfect', 'Sufficient',
      'Insufficient'];
    let tradeData: Object[] = [];
    let address: string[] = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do
      Paço, 67', '2, rue du Commerce', 'Boulevard Tirou, 255',
      'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
      Mercado, 12',
      'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
      'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
      Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
      'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
      43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
      'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
      10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
      'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
      7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
      'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
      'Taucherstraße 10', 'Taucherstraße 10',
      'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
      Circus', 'Av. dos Lusíadas, 23',
      'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
      123', '2817 Milton Dr.'];
    let employeeimg: string[] = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
      dataCount = parseInt(dataCount, 10);
    }
    for (let i: number = 1; i <= dataCount; i++) {
      let code: any = 10000;

```

```

        tradeData.push({
            'EmployeeID': code + i,
            'Employees':
                employees[Math.floor(Math.random() * employees.length)] + '
' + employees[Math.floor(Math.random() * employees.length)],
            'Designation': designation[Math.floor(Math.random() *
designation.length)],
            'Location': location[Math.floor(Math.random() *
location.length)],
            'Status': status[Math.floor(Math.random() * status.length)],
            'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
            'Rating': Math.floor(Math.random() * 5),
            'Software': Math.floor(Math.random() * 100),
            'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
            'CurrentSalary': Math.floor((Math.random() * 100000)),
            'Address': address[Math.floor(Math.random() * address.length)],
        });
        let employee: string = 'Employees';
        let emp: string = tradeData[i - 1][employee];
        let sName: string = emp.substr(0, emp.indexOf(' ')).toLowerCase();
        let empmail: string = 'Mail';
        tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }
    return tradeData;
}
export let tradeData: Object[] = [
    {
        "EmployeeID": 10001,
        "Employees": "Laura Nancy",
        "Designation": "Designer",
        "Location": "France",
        "Status": "Inactive",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
        "Software": 69,
        "EmployeeImg": "usermale",
        "CurrentSalary": 84194,
        "Address": "Taucherstraße 10",
        "Mail": "laura15@jourrapide.com"
    },
    {
        "EmployeeID": 10002,
        "Employees": "Zachery Van",
        "Designation": "CFO",
        "Location": "Canada",
        "Status": "Inactive",
        "Trustworthiness": "Insufficient",
        "Rating": 3,
        "Software": 99,
        "EmployeeImg": "usermale",
        "CurrentSalary": 55349,
        "Address": "5ª Ave. Los Palos Grandes",
        "Mail": "zachery109@sample.com"
    },

```

```

{
  "EmployeeID": 10003,
  "Employees": "Rose Fuller",
  "Designation": "CFO",
  "Location": "France",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 1,
  "EmployeeImg": "usermale",
  "CurrentSalary": 16477,
  "Address": "2817 Milton Dr.",
  "Mail": "rose55@rpy.com"
},
{
  "EmployeeID": 10004,
  "Employees": "Jack Bergs",
  "Designation": "Manager",
  "Location": "Mexico",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 36,
  "EmployeeImg": "usermale",
  "CurrentSalary": 49040,
  "Address": "2, rue du Commerce",
  "Mail": "jack30@sample.com"
},
{
  "EmployeeID": 10005,
  "Employees": "Vinet Bergs",
  "Designation": "Program Directory",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 39,
  "EmployeeImg": "usermale",
  "CurrentSalary": 5495,
  "Address": "Rua da Panificadora, 12",
  "Mail": "vinet32@jourrapide.com"
},
{
  "EmployeeID": 10006,
  "Employees": "Buchanan Van",
  "Designation": "Designer",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 4,
  "Software": 78,
  "EmployeeImg": "usermale",
  "CurrentSalary": 42182,
  "Address": "24, place Kléber",
  "Mail": "buchanan18@mail.com"
},
{

```

```
"EmployeeID": 10007,
"Employees": "Dodsworth Nancy",
"Designation": "Project Lead",
"Location": "USA",
"Status": "Inactive",
"Trustworthiness": "Sufficient",
"Rating": 0,
"Software": 0,
"EmployeeImg": "userfemale",
"CurrentSalary": 35776,
"Address": "Rua do Paço, 67",
"Mail": "dodsworth84@mail.com"
},
{
  "EmployeeID": 10008,
  "Employees": "Laura Jack",
  "Designation": "Developer",
  "Location": "Austria",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 89,
  "EmployeeImg": "usermale",
  "CurrentSalary": 25108,
  "Address": "Rua da Panificadora, 12",
  "Mail": "laura82@mail.com"
},
{
  "EmployeeID": 10009,
  "Employees": "Anne Fuller",
  "Designation": "Program Directory",
  "Location": "Mexico",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 0,
  "Software": 19,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 32568,
  "Address": "Gran Vía, 1",
  "Mail": "anne97@jourrapide.com"
},
{
  "EmployeeID": 10010,
  "Employees": "Buchanan Andrew",
  "Designation": "Designer",
  "Location": "Austria",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 62,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 12320,
  "Address": "P.O. Box 555",
  "Mail": "buchanan50@jourrapide.com"
},
{
  "EmployeeID": 10011,
```

```

    "Employees": "Andrew Janet",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 20890,
    "Address": "Starenweg 5",
    "Mail": "andrew63@mail.com"
  },
  {
    "EmployeeID": 10012,
    "Employees": "Margaret Tamer",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 7,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 22337,
    "Address": "Magazinweg 7",
    "Mail": "margaret26@mail.com"
  },
  {
    "EmployeeID": 10013,
    "Employees": "Tamer Fuller",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 89181,
    "Address": "Taucherstraße 10",
    "Mail": "tamer40@arpy.com"
  },
  {
    "EmployeeID": 10014,
    "Employees": "Tamer Anne",
    "Designation": "CFO",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 18,
    "EmployeeImg": "usermale",
    "CurrentSalary": 20998,
    "Address": "Taucherstraße 10",
    "Mail": "tamer68@arpy.com"
  },
  {
    "EmployeeID": 10015,
    "Employees": "Anton Davolio",

```



```

    "Designation": "Project Lead",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 4,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 48232,
    "Address": "Luisenstr. 48",
    "Mail": "anton46@mail.com"
  },
  {
    "EmployeeID": 10016,
    "Employees": "Buchanan Buchanan",
    "Designation": "System Analyst",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "usermale",
    "CurrentSalary": 43041,
    "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
    "Mail": "buchanan68@mail.com"
  },
  {
    "EmployeeID": 10017,
    "Employees": "King Buchanan",
    "Designation": "Program Directory",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 44,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 25259,
    "Address": "Magazinweg 7",
    "Mail": "king80@jourrapide.com"
  },
  {
    "EmployeeID": 10018,
    "Employees": "Rose Michael",
    "Designation": "Project Lead",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 31,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 91156,
    "Address": "Fauntleroy Circus",
    "Mail": "rose75@mail.com"
  },
  {
    "EmployeeID": 10019,
    "Employees": "King Bergrs",
    "Designation": "Developer",

```

```

    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 2,
    "Software": 29,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 28826,
    "Address": "2817 Milton Dr.",
    "Mail": "king57@jourrapide.com"
  },
  {
    "EmployeeID": 10020,
    "Employees": "Davolio Fuller",
    "Designation": "Designer",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 3,
    "Software": 35,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 71035,
    "Address": "Gran Vía, 1",
    "Mail": "davolio29@arpy.com"
  },
  {
    "EmployeeID": 10021,
    "Employees": "Rose Rose",
    "Designation": "CFO",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 38,
    "EmployeeImg": "usermale",
    "CurrentSalary": 68123,
    "Address": "Rua do Mercado, 12",
    "Mail": "rose54@arpy.com"
  },
  {
    "EmployeeID": 10022,
    "Employees": "Andrew Michael",
    "Designation": "Program Directory",
    "Location": "UK",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 2,
    "Software": 61,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 75470,
    "Address": "2, rue du Commerce",
    "Mail": "andrew88@jourrapide.com"
  },
  {
    "EmployeeID": 10023,
    "Employees": "Davolio Kathryn",
    "Designation": "Manager",
    "Location": "Germany",

```

```

    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 25,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25234,
    "Address": "Hauptstr. 31",
    "Mail": "davolio42@sample.com"
  },
  {
    "EmployeeID": 10024,
    "Employees": "Anne Fleet",
    "Designation": "System Analyst",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 8341,
    "Address": "59 rue de lAbbaye",
    "Mail": "anne86@arpy.com"
  },
  {
    "EmployeeID": 10025,
    "Employees": "Margaret Andrew",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 51,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84975,
    "Address": "P.O. Box 555",
    "Mail": "margaret41@arpy.com"
  },
  {
    "EmployeeID": 10026,
    "Employees": "Kathryn Laura",
    "Designation": "Project Lead",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 48,
    "EmployeeImg": "usermale",
    "CurrentSalary": 97282,
    "Address": "Avda. Azteca 123",
    "Mail": "kathryn82@rpy.com"
  },
  {
    "EmployeeID": 10027,
    "Employees": "Michael Michael",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Inactive",

```

```

    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 16,
    "EmployeeImg": "usermale",
    "CurrentSalary": 4184,
    "Address": "Rua do Paço, 67",
    "Mail": "michael58@jourrapide.com"
  },
  {
    "EmployeeID": 10028,
    "Employees": "Leverling Vinet",
    "Designation": "Project Lead",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 57,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 38370,
    "Address": "59 rue de l'Abbaye",
    "Mail": "leverling102@sample.com"
  },
  {
    "EmployeeID": 10029,
    "Employees": "Rose Jack",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 46,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84790,
    "Address": "Rua do Mercado, 12",
    "Mail": "rose108@jourrapide.com"
  },
  {
    "EmployeeID": 10030,
    "Employees": "Vinet Van",
    "Designation": "Developer",
    "Location": "USA",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 40,
    "EmployeeImg": "usermale",
    "CurrentSalary": 71005,
    "Address": "Gran Vía, 1",
    "Mail": "vinet90@jourrapide.com"
  }
]
```

Please find the below table for the beforeSave event arguments.

Parameter	Type	Description
-----------	------	-------------

| ---- | ---- | ---- |

| url | string | Specifies the save url. |

| fileName | string | Specifies the file name. |

| saveType | SaveType | Specifies the saveType like Xlsx, Xls, Csv and Pdf. |

| customParams | object | Passing the custom parameters from client to server while performing save operation. |

| isFullPost | boolean | It sends the form data from client to server, when set to true. It fetches the data from client to server and returns the data from server to client, when set to false. |

| needBlobData | boolean | You can get the blob data if set to true. |

| cancel | boolean | To prevent the save operations. |

\* Use **Ctrl + S** keyboard shortcut to save the Spreadsheet data as Excel file.

\* The default value of [allowSave](#) property is **true**. For demonstration purpose, we have showcased the [allowSave](#) property in previous code snippet.

\* Demo purpose only, we have used the online web service url link.

*To send and receive custom params from client to server*

Passing the custom parameters from client to server by using [beforeSave](#) event.

## APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { defaultData } from './datasource';
function App() {
  const [customParams, setCustomParams] = React.useState({});
  const beforeSave = (args) => {
    setCustomParams({ customParams: 'you can pass custom params in
server side' });
    args.customParams = customParams; // you can pass the custom params
  };
  return (
    <SpreadsheetComponent allowSave={true}
saveUrl='https://services.syncfusion.com/react/production/api/spreadsheet/sa
ve' beforeSave={beforeSave}>
      <SheetsDirective>
        <SheetDirective>
          <RangesDirective>
            <RangeDirective
dataSource={defaultData}></RangeDirective>
          </RangesDirective>
          <ColumnsDirective>
            <ColumnDirective width={180}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
          </ColumnsDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  );
}
```

```

        </ColumnsDirective>
      </SheetDirective>
    </SheetsDirective>
  </SpreadsheetComponent>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, BeforeSaveEventArgs } from '@syncfusion/ej2-react-
spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { defaultData } from './datasource';
function App() {
  const [customParams, setCustomParams] = React.useState({});
  const beforeSave = (args: BeforeSaveEventArgs): void => {
    setCustomParams({ customParams: 'you can pass custom params in
server side' });
    args.customParams = customParams; // you can pass the custom params
  };
  return (
    <SpreadsheetComponent allowSave={true}
saveUrl='https://services.syncfusion.com/react/production/api/spreadsheet/sa
ve' beforeSave={beforeSave}>
      <SheetsDirective>
        <SheetDirective>
          <RangesDirective>
            <RangeDirective
dataSource={defaultData}></RangeDirective>
          </RangesDirective>
          <ColumnsDirective>
            <ColumnDirective width={180}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
          </ColumnsDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  );
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

### DATASOURCE.JSX

```

/**
 * Default data source
 */
export let defaultData = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
    Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
  { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
    Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
  { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54 AM',
    Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
  { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time: '12:43:59 AM',
    Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit: 70 },
  { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
    Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
  { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
    Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
  { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
    Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
  { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
    Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
  { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
    Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
/**
 * Grid datasource
 */
export function getTradeData(dataCount) {
  let employees = [
    'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
    'Fuller', 'Leverling', 'Peacock',
    'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
    'Dodsworth', 'Anne',
    'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
    'Jack', 'Rose'
  ];
  let designation = ['Manager', 'CFO', 'Designer', 'Developer', 'Program
    Directory', 'System Analyst', 'Project Lead'];
  let mail = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
    'jourrapide.com'];
  let location = ['UK', 'USA', 'Sweden', 'France', 'Canada', 'Argentina',
    'Austria', 'Germany', 'Mexico'];
  let status = ['Active', 'Inactive'];
  let trustworthiness = ['Perfect', 'Sufficient', 'Insufficient'];
  let tradeData = [];
  let address = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do Paço, 67',
    '2, rue du Commerce', 'Boulevard Tirou, 255',
    'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
    Mercado, 12',
    'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
    'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
    Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
    'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
    43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',

```

```

    'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
    'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
    'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
    'Taucherstraße 10', 'Taucherstraße 10',
    'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
Circus', 'Av. dos Lusíadas, 23',
    'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
123', '2817 Milton Dr.'];
    let employeeimg = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
        dataCount = parseInt(dataCount, 10);
    }
    for (let i = 1; i <= dataCount; i++) {
        let code = 10000;
        tradeData.push({
            'EmployeeID': code + i,
            'Employees': employees[Math.floor(Math.random() *
employees.length)] + ' ' + employees[Math.floor(Math.random() *
employees.length)],
            'Designation': designation[Math.floor(Math.random() *
designation.length)],
            'Location': location[Math.floor(Math.random() *
location.length)],
            'Status': status[Math.floor(Math.random() * status.length)],
            'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
            'Rating': Math.floor(Math.random() * 5),
            'Software': Math.floor(Math.random() * 100),
            'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
            'CurrentSalary': Math.floor((Math.random() * 100000)),
            'Address': address[Math.floor(Math.random() * address.length)],
        });
        let employee = 'Employees';
        let emp = tradeData[i - 1][employee];
        let sName = emp.substr(0, emp.indexOf(' ')).toLowerCase();
        let empmail = 'Mail';
        tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }
    return tradeData;
}
export let tradeData = [
    {
        "EmployeeID": 10001,
        "Employees": "Laura Nancy",
        "Designation": "Designer",
        "Location": "France",
        "Status": "Inactive",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
        "Software": 69,
        "EmployeeImg": "usermale",
        "CurrentSalary": 84194,
        "Address": "Taucherstraße 10",
    }

```



```

    "Mail": "laura15@jourrapide.com"
  },
  {
    "EmployeeID": 10002,
    "Employees": "Zachery Van",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 99,
    "EmployeeImg": "usermale",
    "CurrentSalary": 55349,
    "Address": "5a Ave. Los Palos Grandes",
    "Mail": "zachery109@sample.com"
  },
  {
    "EmployeeID": 10003,
    "Employees": "Rose Fuller",
    "Designation": "CFO",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 1,
    "EmployeeImg": "usermale",
    "CurrentSalary": 16477,
    "Address": "2817 Milton Dr.",
    "Mail": "rose55@rpy.com"
  },
  {
    "EmployeeID": 10004,
    "Employees": "Jack Bergs",
    "Designation": "Manager",
    "Location": "Mexico",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 36,
    "EmployeeImg": "usermale",
    "CurrentSalary": 49040,
    "Address": "2, rue du Commerce",
    "Mail": "jack30@sample.com"
  },
  {
    "EmployeeID": 10005,
    "Employees": "Vinet Bergs",
    "Designation": "Program Directory",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 39,
    "EmployeeImg": "usermale",
    "CurrentSalary": 5495,
    "Address": "Rua da Panificadora, 12",
    "Mail": "vinet32@jourrapide.com"
  }

```

```
},
{
  "EmployeeID": 10006,
  "Employees": "Buchanan Van",
  "Designation": "Designer",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 4,
  "Software": 78,
  "EmployeeImg": "usermale",
  "CurrentSalary": 42182,
  "Address": "24, place Kléber",
  "Mail": "buchanan18@mail.com"
},
{
  "EmployeeID": 10007,
  "Employees": "Dodsworth Nancy",
  "Designation": "Project Lead",
  "Location": "USA",
  "Status": "Inactive",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 0,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 35776,
  "Address": "Rua do Paço, 67",
  "Mail": "dodsworth84@mail.com"
},
{
  "EmployeeID": 10008,
  "Employees": "Laura Jack",
  "Designation": "Developer",
  "Location": "Austria",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 89,
  "EmployeeImg": "usermale",
  "CurrentSalary": 25108,
  "Address": "Rua da Panificadora, 12",
  "Mail": "laura82@mail.com"
},
{
  "EmployeeID": 10009,
  "Employees": "Anne Fuller",
  "Designation": "Program Directory",
  "Location": "Mexico",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 0,
  "Software": 19,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 32568,
  "Address": "Gran Vía, 1",
  "Mail": "anne97@jourrapide.com"
},
},
```

```

{
  "EmployeeID": 10010,
  "Employees": "Buchanan Andrew",
  "Designation": "Designer",
  "Location": "Austria",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 62,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 12320,
  "Address": "P.O. Box 555",
  "Mail": "buchanan50@jourrapide.com"
},
{
  "EmployeeID": 10011,
  "Employees": "Andrew Janet",
  "Designation": "System Analyst",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 8,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 20890,
  "Address": "Starenweg 5",
  "Mail": "andrew63@mail.com"
},
{
  "EmployeeID": 10012,
  "Employees": "Margaret Tamer",
  "Designation": "System Analyst",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 4,
  "Software": 7,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 22337,
  "Address": "Magazinweg 7",
  "Mail": "margaret26@mail.com"
},
{
  "EmployeeID": 10013,
  "Employees": "Tamer Fuller",
  "Designation": "CFO",
  "Location": "Canada",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 78,
  "EmployeeImg": "usermale",
  "CurrentSalary": 89181,
  "Address": "Taucherstraße 10",
  "Mail": "tamer40@arpy.com"
},
{

```

```

    "EmployeeID": 10014,
    "Employees": "Tamer Anne",
    "Designation": "CFO",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 18,
    "EmployeeImg": "usermale",
    "CurrentSalary": 20998,
    "Address": "Taucherstraße 10",
    "Mail": "tamer68@arpy.com"
  },
  {
    "EmployeeID": 10015,
    "Employees": "Anton Davolio",
    "Designation": "Project Lead",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 4,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 48232,
    "Address": "Luisenstr. 48",
    "Mail": "anton46@mail.com"
  },
  {
    "EmployeeID": 10016,
    "Employees": "Buchanan Buchanan",
    "Designation": "System Analyst",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "usermale",
    "CurrentSalary": 43041,
    "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
    "Mail": "buchanan68@mail.com"
  },
  {
    "EmployeeID": 10017,
    "Employees": "King Buchanan",
    "Designation": "Program Directory",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 44,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 25259,
    "Address": "Magazinweg 7",
    "Mail": "king80@jourrapide.com"
  },
  {
    "EmployeeID": 10018,

```

```

    "Employees": "Rose Michael",
    "Designation": "Project Lead",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 31,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 91156,
    "Address": "Fauntleroy Circus",
    "Mail": "rose75@mail.com"
  },
  {
    "EmployeeID": 10019,
    "Employees": "King Bergs",
    "Designation": "Developer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 2,
    "Software": 29,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 28826,
    "Address": "2817 Milton Dr.",
    "Mail": "king57@jourrapide.com"
  },
  {
    "EmployeeID": 10020,
    "Employees": "Davolio Fuller",
    "Designation": "Designer",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 3,
    "Software": 35,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 71035,
    "Address": "Gran Vía, 1",
    "Mail": "davolio29@arpy.com"
  },
  {
    "EmployeeID": 10021,
    "Employees": "Rose Rose",
    "Designation": "CFO",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 38,
    "EmployeeImg": "usermale",
    "CurrentSalary": 68123,
    "Address": "Rua do Mercado, 12",
    "Mail": "rose54@arpy.com"
  },
  {
    "EmployeeID": 10022,
    "Employees": "Andrew Michael",

```

```

    "Designation": "Program Directory",
    "Location": "UK",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 2,
    "Software": 61,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 75470,
    "Address": "2, rue du Commerce",
    "Mail": "andrew88@jourrapide.com"
  },
  {
    "EmployeeID": 10023,
    "Employees": "Davolio Kathryn",
    "Designation": "Manager",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 25,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25234,
    "Address": "Hauptstr. 31",
    "Mail": "davolio42@sample.com"
  },
  {
    "EmployeeID": 10024,
    "Employees": "Anne Fleet",
    "Designation": "System Analyst",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 8341,
    "Address": "59 rue de l'Abbaye",
    "Mail": "anne86@arpy.com"
  },
  {
    "EmployeeID": 10025,
    "Employees": "Margaret Andrew",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 51,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84975,
    "Address": "P.O. Box 555",
    "Mail": "margaret41@arpy.com"
  },
  {
    "EmployeeID": 10026,
    "Employees": "Kathryn Laura",
    "Designation": "Project Lead",

```

```

    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 48,
    "EmployeeImg": "usermale",
    "CurrentSalary": 97282,
    "Address": "Avda. Azteca 123",
    "Mail": "kathryn82@rpy.com"
  },
  {
    "EmployeeID": 10027,
    "Employees": "Michael Michael",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 16,
    "EmployeeImg": "usermale",
    "CurrentSalary": 4184,
    "Address": "Rua do Paço, 67",
    "Mail": "michael58@jourrapide.com"
  },
  {
    "EmployeeID": 10028,
    "Employees": "Leverling Vinet",
    "Designation": "Project Lead",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 57,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 38370,
    "Address": "59 rue de l'Abbaye",
    "Mail": "leverling102@sample.com"
  },
  {
    "EmployeeID": 10029,
    "Employees": "Rose Jack",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 46,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84790,
    "Address": "Rua do Mercado, 12",
    "Mail": "rose108@jourrapide.com"
  },
  {
    "EmployeeID": 10030,
    "Employees": "Vinet Van",
    "Designation": "Developer",
    "Location": "USA",

```

```

        "Status": "Active",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
        "Software": 40,
        "EmployeeImg": "usermale",
        "CurrentSalary": 71005,
        "Address": "Gran Vía, 1",
        "Mail": "vinet90@jourrapide.com"
    }
];

```

## DATASOURCE.TSX

```

/**
 * Default data source
 */
export let defaultData: Object[] = [
    { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
      Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
    { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
      Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
    { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
      Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
    { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54 AM',
      Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
    { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time: '12:43:59 AM',
      Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit: 70 },
    { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
      Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
    { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
      Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
    { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
      Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
    { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
      Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
    { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
      Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
/**
 * Grid datasource
 */
export function getTradeData(dataCount?: number): object {
    let employees: string[] = [
        'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
        'Fuller', 'Leverling', 'Peacock',
        'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
        'Dodsworth', 'Anne',
        'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
        'Jack', 'Rose'];
    let designation: string[] = ['Manager', 'CFO', 'Designer', 'Developer',
        'Program Directory', 'System Analyst', 'Project Lead'];
    let mail: string[] = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
        'jourrapide.com'];
    let location: string[] = ['UK', 'USA', 'Sweden', 'France', 'Canada',
        'Argentina', 'Austria', 'Germany', 'Mexico'];
}

```



```

    let status: string[] = ['Active', 'Inactive'];
    let trustworthiness: string[] = ['Perfect', 'Sufficient',
'Insufficient'];
    let tradeData: Object[] = [];
    let address: string[] = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do
Paço, 67', '2, rue du Commerce', 'Boulevard Tirou, 255',
'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
Mercado, 12',
'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
'Taucherstraße 10', 'Taucherstraße 10',
'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
Circus', 'Av. dos Lusíadas, 23',
'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
123', '2817 Milton Dr.'];
    let employeeimg: string[] = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
        dataCount = parseInt(dataCount, 10);
    }
    for (let i: number = 1; i <= dataCount; i++) {
        let code: any = 10000;
        tradeData.push({
            'EmployeeID': code + i,
            'Employees':
                employees[Math.floor(Math.random() * employees.length)] + '
' + employees[Math.floor(Math.random() * employees.length)],
            'Designation': designation[Math.floor(Math.random() *
designation.length)],
            'Location': location[Math.floor(Math.random() *
location.length)],
            'Status': status[Math.floor(Math.random() * status.length)],
            'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
            'Rating': Math.floor(Math.random() * 5),
            'Software': Math.floor(Math.random() * 100),
            'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
            'CurrentSalary': Math.floor((Math.random() * 100000)),
            'Address': address[Math.floor(Math.random() * address.length)],
        });
        let employee: string = 'Employees';
        let emp: string = tradeData[i - 1][employee];
        let sName: string = emp.substr(0, emp.indexOf(' ')).toLowerCase();
        let empmail: string = 'Mail';
        tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }
    return tradeData;
}

```

```
export let tradeData: Object[] = [
  {
    "EmployeeID": 10001,
    "Employees": "Laura Nancy",
    "Designation": "Designer",
    "Location": "France",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 69,
    "EmployeeImg": "usermale",
    "CurrentSalary": 84194,
    "Address": "Taucherstraße 10",
    "Mail": "laura15@jourrapide.com"
  },
  {
    "EmployeeID": 10002,
    "Employees": "Zachery Van",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 99,
    "EmployeeImg": "usermale",
    "CurrentSalary": 55349,
    "Address": "5a Ave. Los Palos Grandes",
    "Mail": "zachery109@sample.com"
  },
  {
    "EmployeeID": 10003,
    "Employees": "Rose Fuller",
    "Designation": "CFO",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 1,
    "EmployeeImg": "usermale",
    "CurrentSalary": 16477,
    "Address": "2817 Milton Dr.",
    "Mail": "rose55@rpy.com"
  },
  {
    "EmployeeID": 10004,
    "Employees": "Jack Bergs",
    "Designation": "Manager",
    "Location": "Mexico",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 36,
    "EmployeeImg": "usermale",
    "CurrentSalary": 49040,
    "Address": "2, rue du Commerce",
    "Mail": "jack30@sample.com"
  },
]
```

```
{
  "EmployeeID": 10005,
  "Employees": "Vinet Bergs",
  "Designation": "Program Directory",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 39,
  "EmployeeImg": "usermale",
  "CurrentSalary": 5495,
  "Address": "Rua da Panificadora, 12",
  "Mail": "vinet32@jourrapide.com"
},
{
  "EmployeeID": 10006,
  "Employees": "Buchanan Van",
  "Designation": "Designer",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 4,
  "Software": 78,
  "EmployeeImg": "usermale",
  "CurrentSalary": 42182,
  "Address": "24, place Kléber",
  "Mail": "buchanan18@mail.com"
},
{
  "EmployeeID": 10007,
  "Employees": "Dodsworth Nancy",
  "Designation": "Project Lead",
  "Location": "USA",
  "Status": "Inactive",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 0,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 35776,
  "Address": "Rua do Paço, 67",
  "Mail": "dodsworth84@mail.com"
},
{
  "EmployeeID": 10008,
  "Employees": "Laura Jack",
  "Designation": "Developer",
  "Location": "Austria",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 89,
  "EmployeeImg": "usermale",
  "CurrentSalary": 25108,
  "Address": "Rua da Panificadora, 12",
  "Mail": "laura82@mail.com"
},
{
```

```
"EmployeeID": 10009,  
"Employees": "Anne Fuller",  
"Designation": "Program Directory",  
"Location": "Mexico",  
"Status": "Active",  
"Trustworthiness": "Insufficient",  
"Rating": 0,  
"Software": 19,  
"EmployeeImg": "userfemale",  
"CurrentSalary": 32568,  
"Address": "Gran Vía, 1",  
"Mail": "anne97@jourrapide.com"  
},  
{  
  "EmployeeID": 10010,  
  "Employees": "Buchanan Andrew",  
  "Designation": "Designer",  
  "Location": "Austria",  
  "Status": "Active",  
  "Trustworthiness": "Insufficient",  
  "Rating": 1,  
  "Software": 62,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 12320,  
  "Address": "P.O. Box 555",  
  "Mail": "buchanan50@jourrapide.com"  
},  
{  
  "EmployeeID": 10011,  
  "Employees": "Andrew Janet",  
  "Designation": "System Analyst",  
  "Location": "Germany",  
  "Status": "Active",  
  "Trustworthiness": "Insufficient",  
  "Rating": 3,  
  "Software": 8,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 20890,  
  "Address": "Starenweg 5",  
  "Mail": "andrew63@mail.com"  
},  
{  
  "EmployeeID": 10012,  
  "Employees": "Margaret Tamer",  
  "Designation": "System Analyst",  
  "Location": "Germany",  
  "Status": "Active",  
  "Trustworthiness": "Insufficient",  
  "Rating": 4,  
  "Software": 7,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 22337,  
  "Address": "Magazinweg 7",  
  "Mail": "margaret26@mail.com"  
},  
{  
  "EmployeeID": 10013,
```

```
"Employees": "Tamer Fuller",
"Designation": "CFO",
"Location": "Canada",
"Status": "Active",
"Trustworthiness": "Insufficient",
"Rating": 3,
"Software": 78,
"EmployeeImg": "usermale",
"CurrentSalary": 89181,
"Address": "Taucherstraße 10",
"Mail": "tamer40@arpy.com"
},
{
  "EmployeeID": 10014,
  "Employees": "Tamer Anne",
  "Designation": "CFO",
  "Location": "Sweden",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 18,
  "EmployeeImg": "usermale",
  "CurrentSalary": 20998,
  "Address": "Taucherstraße 10",
  "Mail": "tamer68@arpy.com"
},
{
  "EmployeeID": 10015,
  "Employees": "Anton Davolio",
  "Designation": "Project Lead",
  "Location": "France",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 4,
  "Software": 8,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 48232,
  "Address": "Luisenstr. 48",
  "Mail": "anton46@mail.com"
},
{
  "EmployeeID": 10016,
  "Employees": "Buchanan Buchanan",
  "Designation": "System Analyst",
  "Location": "Austria",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 19,
  "EmployeeImg": "usermale",
  "CurrentSalary": 43041,
  "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
  "Mail": "buchanan68@mail.com"
},
{
  "EmployeeID": 10017,
  "Employees": "King Buchanan",
```

```

    "Designation": "Program Directory",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 44,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 25259,
    "Address": "Magazinweg 7",
    "Mail": "king80@jourrapide.com"
  },
  {
    "EmployeeID": 10018,
    "Employees": "Rose Michael",
    "Designation": "Project Lead",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 31,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 91156,
    "Address": "Fauntleroy Circus",
    "Mail": "rose75@mail.com"
  },
  {
    "EmployeeID": 10019,
    "Employees": "King Bergs",
    "Designation": "Developer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 2,
    "Software": 29,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 28826,
    "Address": "2817 Milton Dr.",
    "Mail": "king57@jourrapide.com"
  },
  {
    "EmployeeID": 10020,
    "Employees": "Davolio Fuller",
    "Designation": "Designer",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 3,
    "Software": 35,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 71035,
    "Address": "Gran Vía, 1",
    "Mail": "davolio29@arpy.com"
  },
  {
    "EmployeeID": 10021,
    "Employees": "Rose Rose",
    "Designation": "CFO",

```

```

    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 38,
    "EmployeeImg": "usermale",
    "CurrentSalary": 68123,
    "Address": "Rua do Mercado, 12",
    "Mail": "rose54@arpy.com"
  },
  {
    "EmployeeID": 10022,
    "Employees": "Andrew Michael",
    "Designation": "Program Directory",
    "Location": "UK",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 2,
    "Software": 61,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 75470,
    "Address": "2, rue du Commerce",
    "Mail": "andrew88@jourrapide.com"
  },
  {
    "EmployeeID": 10023,
    "Employees": "Davolio Kathryn",
    "Designation": "Manager",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 25,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25234,
    "Address": "Hauptstr. 31",
    "Mail": "davolio42@sample.com"
  },
  {
    "EmployeeID": 10024,
    "Employees": "Anne Fleet",
    "Designation": "System Analyst",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 8341,
    "Address": "59 rue de l'Abbaye",
    "Mail": "anne86@arpy.com"
  },
  {
    "EmployeeID": 10025,
    "Employees": "Margaret Andrew",
    "Designation": "System Analyst",
    "Location": "Germany",

```

```

    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 51,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84975,
    "Address": "P.O. Box 555",
    "Mail": "margaret41@arpy.com"
  },
  {
    "EmployeeID": 10026,
    "Employees": "Kathryn Laura",
    "Designation": "Project Lead",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 48,
    "EmployeeImg": "usermale",
    "CurrentSalary": 97282,
    "Address": "Avda. Azteca 123",
    "Mail": "kathryn82@rpy.com"
  },
  {
    "EmployeeID": 10027,
    "Employees": "Michael Michael",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 16,
    "EmployeeImg": "usermale",
    "CurrentSalary": 4184,
    "Address": "Rua do Paço, 67",
    "Mail": "michael58@jourrapide.com"
  },
  {
    "EmployeeID": 10028,
    "Employees": "Leverling Vinet",
    "Designation": "Project Lead",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 57,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 38370,
    "Address": "59 rue de l'Abbaye",
    "Mail": "leverling102@sample.com"
  },
  {
    "EmployeeID": 10029,
    "Employees": "Rose Jack",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Active",

```



```

    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 46,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84790,
    "Address": "Rua do Mercado, 12",
    "Mail": "rose108@jourrapide.com"
  },
  {
    "EmployeeID": 10030,
    "Employees": "Vinet Van",
    "Designation": "Developer",
    "Location": "USA",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 40,
    "EmployeeImg": "usermale",
    "CurrentSalary": 71005,
    "Address": "Gran Vía, 1",
    "Mail": "vinet90@jourrapide.com"
  }
]

```

Server side code snippets:

`csharp

public IActionResult Save(SaveSettings saveSettings, string customParams)

{

Console.WriteLine(customParams); // you can get the custom params in controller side

return Workbook.Save(saveSettings);

}

`

*To add custom header during save*

You can add your own custom header to the save action in the Spreadsheet. For processing the data, it has to be sent from client to server side and adding customer header can provide privacy to the data with the help of Authorization Token. Through the [fileMenuItemSelect](#) event, the custom header can be added to the request during save action.

### APP.JSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { createElement } from '@syncfusion/ej2-base';
import { defaultData } from './datasource';
function App() {

```

```

const spreadsheetRef = React.useRef(null);
const fileMenuItemSelect = (args) => {
  let spreadsheet = spreadsheetRef.current;
  if (args.item.text === 'Microsoft Excel' && spreadsheet) {
    args.cancel = true;
    spreadsheet.saveAsJson().then((response) => {
      let formData = new FormData();
      formData.append('JSONData',
JSON.stringify(response.jsonObject.Workbook));
      formData.append('fileName', 'Sample');
      formData.append('saveType', 'Xlsx');
      fetch(
'https://services.syncfusion.com/react/production/api/spreadsheet/save',
        {
          method: 'POST',
          headers: { Authorization: 'YOUR TEXT' },
          body: formData,
          mode: 'no-cors'
        }
      ).then((response) => {
        response.blob().then((data) => {
          let anchor = createElement('a', {
            attrs: { download: 'Sample.xlsx' },
          });
          const url = URL.createObjectURL(data);
          anchor.href = url;
          document.body.appendChild(anchor);
          anchor.click();
          URL.revokeObjectURL(url);
          document.body.removeChild(anchor);
        });
      });
    });
  }
};
return (
  <SpreadsheetComponent ref={spreadsheetRef} allowSave={true}
fileMenuItemSelect={fileMenuItemSelect}

saveUrl="https://services.syncfusion.com/react/production/api/spreadsheet/sa
ve" >
    <SheetsDirective>
      <SheetDirective>
        <RangesDirective>
          <RangeDirective dataSource={defaultData}></RangeDirective>
        </RangesDirective>
        <ColumnsDirective>
          <ColumnDirective width={180}></ColumnDirective>
          <ColumnDirective width={130}></ColumnDirective>
          <ColumnDirective width={130}></ColumnDirective>
        </ColumnsDirective>
      </SheetDirective>
    </SheetsDirective>
  </SpreadsheetComponent>
);
};

```

```
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}
```

## APP.TSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, MenuSelectEventArgs } from '@syncfusion/ej2-react-
spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { createElement } from '@syncfusion/ej2-base';
import { defaultData } from './datasource';
function App() {
    const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
    const fileMenuItemSelect = (args: MenuSelectEventArgs): void => {
        let spreadsheet = spreadsheetRef.current;
        if (args.item.text === 'Microsoft Excel' && spreadsheet) {
            args.cancel = true;
            spreadsheet.saveAsJson().then((response) => {
                let formData: FormData = new FormData();
                formData.append('JSONData',
JSON.stringify(response.jsonObject.Workbook));
                formData.append('fileName', 'Sample');
                formData.append('saveType', 'Xlsx');
                fetch(

'https://services.syncfusion.com/react/production/api/spreadsheet/save',
                {
                    method: 'POST',
                    headers: { Authorization: 'YOUR TEXT' },
                    body: formData,
                    mode: 'no-cors'
                }
            ).then((response) => {
                response.blob().then((data) => {
                    let anchor: HTMLAnchorElement = createElement('a', {
                        attrs: { download: 'Sample.xlsx' },
                    });
                    const url: string = URL.createObjectURL(data);
                    anchor.href = url;
                    document.body.appendChild(anchor);
                    anchor.click();
                    URL.revokeObjectURL(url);
                    document.body.removeChild(anchor);
                });
            });
        }
    });
}
};
return (
```

```

    <SpreadsheetComponent ref={spreadsheetRef} allowSave={true}
    fileMenuItemSelect={fileMenuItemSelect}

    saveUrl="https://services.syncfusion.com/react/production/api/spreadsheet/save" >
      <SheetsDirective>
        <SheetDirective>
          <RangesDirective>
            <RangeDirective dataSource={defaultData}></RangeDirective>
          </RangesDirective>
          <ColumnsDirective>
            <ColumnDirective width={180}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
          </ColumnsDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  );
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% enddraw %}

```

### DATASOURCE.JSX

```

/**
 * Default data source
 */
export let defaultData = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
    Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
  { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
    Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
  { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
    AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
  { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
    '12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
    70 },
  { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
    Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
  { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
    Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
  { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
    Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
  { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
    Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
  { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
    Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
/**
 * Grid datasource
 */

```

```

export function getTradeData(dataCount) {
  let employees = [
    'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
    'Fuller', 'Leverling', 'Peacock',
    'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
    'Dodsworth', 'Anne',
    'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
    'Jack', 'Rose'
  ];
  let designation = ['Manager', 'CFO', 'Designer', 'Developer', 'Program
Directory', 'System Analyst', 'Project Lead'];
  let mail = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
'journrapide.com'];
  let location = ['UK', 'USA', 'Sweden', 'France', 'Canada', 'Argentina',
'Austria', 'Germany', 'Mexico'];
  let status = ['Active', 'Inactive'];
  let trustworthiness = ['Perfect', 'Sufficient', 'Insufficient'];
  let tradeData = [];
  let address = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do Paço, 67',
'2, rue du Commerce', 'Boulevard Tirou, 255',
'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
Mercado, 12',
'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
'Taucherstraße 10', 'Taucherstraße 10',
'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
Circus', 'Av. dos Lusíadas, 23',
'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
123', '2817 Milton Dr.'];
  let employeeimg = ['usermale', 'userfemale'];
  if (typeof dataCount === 'string') {
    dataCount = parseInt(dataCount, 10);
  }
  for (let i = 1; i <= dataCount; i++) {
    let code = 10000;
    tradeData.push({
      'EmployeeID': code + i,
      'Employees': employees[Math.floor(Math.random() *
employees.length)] + ' ' + employees[Math.floor(Math.random() *
employees.length)],
      'Designation': designation[Math.floor(Math.random() *
designation.length)],
      'Location': location[Math.floor(Math.random() *
location.length)],
      'Status': status[Math.floor(Math.random() * status.length)],
      'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
      'Rating': Math.floor(Math.random() * 5),
      'Software': Math.floor(Math.random() * 100),

```

```

    'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
    'CurrentSalary': Math.floor((Math.random() * 100000)),
    'Address': address[Math.floor(Math.random() * address.length)],
  });
  let employee = 'Employees';
  let emp = tradeData[i - 1][employee];
  let sName = emp.substr(0, emp.indexOf(' ')).toLowerCase();
  let empmail = 'Mail';
  tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
}
return tradeData;
}
export let tradeData = [
{
  "EmployeeID": 10001,
  "Employees": "Laura Nancy",
  "Designation": "Designer",
  "Location": "France",
  "Status": "Inactive",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 69,
  "EmployeeImg": "usermale",
  "CurrentSalary": 84194,
  "Address": "Taucherstraße 10",
  "Mail": "laural5@jourrapide.com"
},
{
  "EmployeeID": 10002,
  "Employees": "Zachery Van",
  "Designation": "CFO",
  "Location": "Canada",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 99,
  "EmployeeImg": "usermale",
  "CurrentSalary": 55349,
  "Address": "5ª Ave. Los Palos Grandes",
  "Mail": "zachery109@sample.com"
},
{
  "EmployeeID": 10003,
  "Employees": "Rose Fuller",
  "Designation": "CFO",
  "Location": "France",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 1,
  "EmployeeImg": "usermale",
  "CurrentSalary": 16477,
  "Address": "2817 Milton Dr.",
  "Mail": "rose55@rpy.com"
},

```

```

{
  "EmployeeID": 10004,
  "Employees": "Jack Bergs",
  "Designation": "Manager",
  "Location": "Mexico",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 36,
  "EmployeeImg": "usermale",
  "CurrentSalary": 49040,
  "Address": "2, rue du Commerce",
  "Mail": "jack30@sample.com"
},
{
  "EmployeeID": 10005,
  "Employees": "Vinet Bergs",
  "Designation": "Program Directory",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 39,
  "EmployeeImg": "usermale",
  "CurrentSalary": 5495,
  "Address": "Rua da Panificadora, 12",
  "Mail": "vinet32@jourrapide.com"
},
{
  "EmployeeID": 10006,
  "Employees": "Buchanan Van",
  "Designation": "Designer",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 4,
  "Software": 78,
  "EmployeeImg": "usermale",
  "CurrentSalary": 42182,
  "Address": "24, place Kléber",
  "Mail": "buchanan18@mail.com"
},
{
  "EmployeeID": 10007,
  "Employees": "Dodsworth Nancy",
  "Designation": "Project Lead",
  "Location": "USA",
  "Status": "Inactive",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 0,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 35776,
  "Address": "Rua do Paço, 67",
  "Mail": "dodsworth84@mail.com"
},
{

```

```

    "EmployeeID": 10008,
    "Employees": "Laura Jack",
    "Designation": "Developer",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 89,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25108,
    "Address": "Rua da Panificadora, 12",
    "Mail": "laura82@mail.com"
  },
  {
    "EmployeeID": 10009,
    "Employees": "Anne Fuller",
    "Designation": "Program Directory",
    "Location": "Mexico",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 32568,
    "Address": "Gran Vía, 1",
    "Mail": "anne97@jourrapide.com"
  },
  {
    "EmployeeID": 10010,
    "Employees": "Buchanan Andrew",
    "Designation": "Designer",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 62,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 12320,
    "Address": "P.O. Box 555",
    "Mail": "buchanan50@jourrapide.com"
  },
  {
    "EmployeeID": 10011,
    "Employees": "Andrew Janet",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 20890,
    "Address": "Starenweg 5",
    "Mail": "andrew63@mail.com"
  },
  {
    "EmployeeID": 10012,

```



```

    "Employees": "Margaret Tamer",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 7,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 22337,
    "Address": "Magazinweg 7",
    "Mail": "margaret26@mail.com"
  },
  {
    "EmployeeID": 10013,
    "Employees": "Tamer Fuller",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 89181,
    "Address": "Taucherstraße 10",
    "Mail": "tamer40@arpy.com"
  },
  {
    "EmployeeID": 10014,
    "Employees": "Tamer Anne",
    "Designation": "CFO",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 18,
    "EmployeeImg": "usermale",
    "CurrentSalary": 20998,
    "Address": "Taucherstraße 10",
    "Mail": "tamer68@arpy.com"
  },
  {
    "EmployeeID": 10015,
    "Employees": "Anton Davolio",
    "Designation": "Project Lead",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 4,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 48232,
    "Address": "Luisenstr. 48",
    "Mail": "anton46@mail.com"
  },
  {
    "EmployeeID": 10016,
    "Employees": "Buchanan Buchanan",

```

```

    "Designation": "System Analyst",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "usermale",
    "CurrentSalary": 43041,
    "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
    "Mail": "buchanan68@mail.com"
  },
  {
    "EmployeeID": 10017,
    "Employees": "King Buchanan",
    "Designation": "Program Directory",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 44,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 25259,
    "Address": "Magazinweg 7",
    "Mail": "king80@jourrapide.com"
  },
  {
    "EmployeeID": 10018,
    "Employees": "Rose Michael",
    "Designation": "Project Lead",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 31,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 91156,
    "Address": "Fauntleroy Circus",
    "Mail": "rose75@mail.com"
  },
  {
    "EmployeeID": 10019,
    "Employees": "King Bergs",
    "Designation": "Developer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 2,
    "Software": 29,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 28826,
    "Address": "2817 Milton Dr.",
    "Mail": "king57@jourrapide.com"
  },
  {
    "EmployeeID": 10020,
    "Employees": "Davolio Fuller",
    "Designation": "Designer",

```

```

      "Location": "Canada",
      "Status": "Inactive",
      "Trustworthiness": "Sufficient",
      "Rating": 3,
      "Software": 35,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 71035,
      "Address": "Gran Vía, 1",
      "Mail": "davolio29@arpy.com"
    },
    {
      "EmployeeID": 10021,
      "Employees": "Rose Rose",
      "Designation": "CFO",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Perfect",
      "Rating": 3,
      "Software": 38,
      "EmployeeImg": "usermale",
      "CurrentSalary": 68123,
      "Address": "Rua do Mercado, 12",
      "Mail": "rose54@arpy.com"
    },
    {
      "EmployeeID": 10022,
      "Employees": "Andrew Michael",
      "Designation": "Program Directory",
      "Location": "UK",
      "Status": "Inactive",
      "Trustworthiness": "Insufficient",
      "Rating": 2,
      "Software": 61,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 75470,
      "Address": "2, rue du Commerce",
      "Mail": "andrew88@jourrapide.com"
    },
    {
      "EmployeeID": 10023,
      "Employees": "Davolio Kathryn",
      "Designation": "Manager",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Perfect",
      "Rating": 3,
      "Software": 25,
      "EmployeeImg": "usermale",
      "CurrentSalary": 25234,
      "Address": "Hauptstr. 31",
      "Mail": "davolio42@sample.com"
    },
    {
      "EmployeeID": 10024,
      "Employees": "Anne Fleet",
      "Designation": "System Analyst",
      "Location": "UK",

```

```

    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 8341,
    "Address": "59 rue de l'Abbaye",
    "Mail": "anne86@arpy.com"
  },
  {
    "EmployeeID": 10025,
    "Employees": "Margaret Andrew",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 51,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84975,
    "Address": "P.O. Box 555",
    "Mail": "margaret41@arpy.com"
  },
  {
    "EmployeeID": 10026,
    "Employees": "Kathryn Laura",
    "Designation": "Project Lead",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 48,
    "EmployeeImg": "usermale",
    "CurrentSalary": 97282,
    "Address": "Avda. Azteca 123",
    "Mail": "kathryn82@rpy.com"
  },
  {
    "EmployeeID": 10027,
    "Employees": "Michael Michael",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 16,
    "EmployeeImg": "usermale",
    "CurrentSalary": 4184,
    "Address": "Rua do Paço, 67",
    "Mail": "michael58@jourrapide.com"
  },
  {
    "EmployeeID": 10028,
    "Employees": "Leverling Vinet",
    "Designation": "Project Lead",
    "Location": "Germany",
    "Status": "Inactive",

```

```

        "Trustworthiness": "Perfect",
        "Rating": 0,
        "Software": 57,
        "EmployeeImg": "userfemale",
        "CurrentSalary": 38370,
        "Address": "59 rue de l'Abbaye",
        "Mail": "leverling102@sample.com"
    },
    {
        "EmployeeID": 10029,
        "Employees": "Rose Jack",
        "Designation": "Developer",
        "Location": "UK",
        "Status": "Active",
        "Trustworthiness": "Perfect",
        "Rating": 0,
        "Software": 46,
        "EmployeeImg": "userfemale",
        "CurrentSalary": 84790,
        "Address": "Rua do Mercado, 12",
        "Mail": "rose108@jourrapide.com"
    },
    {
        "EmployeeID": 10030,
        "Employees": "Vinet Van",
        "Designation": "Developer",
        "Location": "USA",
        "Status": "Active",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
        "Software": 40,
        "EmployeeImg": "usermale",
        "CurrentSalary": 71005,
        "Address": "Gran Vía, 1",
        "Mail": "vinet90@jourrapide.com"
    }
];

```

### DATASOURCE.TSX

```

/**
 * Default data source
 */
export let defaultData: Object[] = [
    { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
      Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
    { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
      Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
    { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
      Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
    { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54 AM',
      Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
    { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time: '12:43:59 AM',
      Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit: 70 },
];

```

```

    { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
      Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
    { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
      Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
    { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
      Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
    { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
      Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
    { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
      Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
  ];
  /**
   * Grid datasource
   */
  export function getTradeData(dataCount?: number): object {
    let employees: string[] = [
      'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
      'Fuller', 'Leverling', 'Peacock',
      'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
      'Dodsworth', 'Anne',
      'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
      'Jack', 'Rose'];
    let designation: string[] = ['Manager', 'CFO', 'Designer', 'Developer',
      'Program Directory', 'System Analyst', 'Project Lead'];
    let mail: string[] = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
      'journrapide.com'];
    let location: string[] = ['UK', 'USA', 'Sweden', 'France', 'Canada',
      'Argentina', 'Austria', 'Germany', 'Mexico'];
    let status: string[] = ['Active', 'Inactive'];
    let trustworthiness: string[] = ['Perfect', 'Sufficient',
      'Insufficient'];
    let tradeData: Object[] = [];
    let address: string[] = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do
      Paço, 67', '2, rue du Commerce', 'Boulevard Tirou, 255',
      'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
      Mercado, 12',
      'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
      'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
      Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
      'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
      43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
      'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
      10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
      'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
      7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
      'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
      'Taucherstraße 10', 'Taucherstraße 10',
      'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
      Circus', 'Av. dos Lusíadas, 23',
      'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
      123', '2817 Milton Dr.'];
    let employeeimg: string[] = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
      dataCount = parseInt(dataCount, 10);
    }
    for (let i: number = 1; i <= dataCount; i++) {
      let code: any = 10000;

```

```

        tradeData.push({
            'EmployeeID': code + i,
            'Employees':
                employees[Math.floor(Math.random() * employees.length)] + '
' + employees[Math.floor(Math.random() * employees.length)],
            'Designation': designation[Math.floor(Math.random() *
designation.length)],
            'Location': location[Math.floor(Math.random() *
location.length)],
            'Status': status[Math.floor(Math.random() * status.length)],
            'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
            'Rating': Math.floor(Math.random() * 5),
            'Software': Math.floor(Math.random() * 100),
            'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
            'CurrentSalary': Math.floor((Math.random() * 100000)),
            'Address': address[Math.floor(Math.random() * address.length)],
        });
        let employee: string = 'Employees';
        let emp: string = tradeData[i - 1][employee];
        let sName: string = emp.substr(0, emp.indexOf(' ')).toLowerCase();
        let empmail: string = 'Mail';
        tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }
    return tradeData;
}
export let tradeData: Object[] = [
    {
        "EmployeeID": 10001,
        "Employees": "Laura Nancy",
        "Designation": "Designer",
        "Location": "France",
        "Status": "Inactive",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
        "Software": 69,
        "EmployeeImg": "usermale",
        "CurrentSalary": 84194,
        "Address": "Taucherstraße 10",
        "Mail": "laura15@jourrapide.com"
    },
    {
        "EmployeeID": 10002,
        "Employees": "Zachery Van",
        "Designation": "CFO",
        "Location": "Canada",
        "Status": "Inactive",
        "Trustworthiness": "Insufficient",
        "Rating": 3,
        "Software": 99,
        "EmployeeImg": "usermale",
        "CurrentSalary": 55349,
        "Address": "5ª Ave. Los Palos Grandes",
        "Mail": "zachery109@sample.com"
    },

```

```
{
  "EmployeeID": 10003,
  "Employees": "Rose Fuller",
  "Designation": "CFO",
  "Location": "France",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 1,
  "EmployeeImg": "usermale",
  "CurrentSalary": 16477,
  "Address": "2817 Milton Dr.",
  "Mail": "rose55@rpy.com"
},
{
  "EmployeeID": 10004,
  "Employees": "Jack Bergs",
  "Designation": "Manager",
  "Location": "Mexico",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 36,
  "EmployeeImg": "usermale",
  "CurrentSalary": 49040,
  "Address": "2, rue du Commerce",
  "Mail": "jack30@sample.com"
},
{
  "EmployeeID": 10005,
  "Employees": "Vinet Bergs",
  "Designation": "Program Directory",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 39,
  "EmployeeImg": "usermale",
  "CurrentSalary": 5495,
  "Address": "Rua da Panificadora, 12",
  "Mail": "vinet32@jourrapide.com"
},
{
  "EmployeeID": 10006,
  "Employees": "Buchanan Van",
  "Designation": "Designer",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 4,
  "Software": 78,
  "EmployeeImg": "usermale",
  "CurrentSalary": 42182,
  "Address": "24, place Kléber",
  "Mail": "buchanan18@mail.com"
},
{
```



```
"EmployeeID": 10007,  
"Employees": "Dodsworth Nancy",  
"Designation": "Project Lead",  
"Location": "USA",  
"Status": "Inactive",  
"Trustworthiness": "Sufficient",  
"Rating": 0,  
"Software": 0,  
"EmployeeImg": "userfemale",  
"CurrentSalary": 35776,  
"Address": "Rua do Paço, 67",  
"Mail": "dodsworth84@mail.com"  
},  
{  
  "EmployeeID": 10008,  
  "Employees": "Laura Jack",  
  "Designation": "Developer",  
  "Location": "Austria",  
  "Status": "Inactive",  
  "Trustworthiness": "Perfect",  
  "Rating": 3,  
  "Software": 89,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 25108,  
  "Address": "Rua da Panificadora, 12",  
  "Mail": "laura82@mail.com"  
},  
{  
  "EmployeeID": 10009,  
  "Employees": "Anne Fuller",  
  "Designation": "Program Directory",  
  "Location": "Mexico",  
  "Status": "Active",  
  "Trustworthiness": "Insufficient",  
  "Rating": 0,  
  "Software": 19,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 32568,  
  "Address": "Gran Vía, 1",  
  "Mail": "anne97@jourrapide.com"  
},  
{  
  "EmployeeID": 10010,  
  "Employees": "Buchanan Andrew",  
  "Designation": "Designer",  
  "Location": "Austria",  
  "Status": "Active",  
  "Trustworthiness": "Insufficient",  
  "Rating": 1,  
  "Software": 62,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 12320,  
  "Address": "P.O. Box 555",  
  "Mail": "buchanan50@jourrapide.com"  
},  
{  
  "EmployeeID": 10011,
```

```

    "Employees": "Andrew Janet",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 20890,
    "Address": "Starenweg 5",
    "Mail": "andrew63@mail.com"
  },
  {
    "EmployeeID": 10012,
    "Employees": "Margaret Tamer",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 7,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 22337,
    "Address": "Magazinweg 7",
    "Mail": "margaret26@mail.com"
  },
  {
    "EmployeeID": 10013,
    "Employees": "Tamer Fuller",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 89181,
    "Address": "Taucherstraße 10",
    "Mail": "tamer40@arpy.com"
  },
  {
    "EmployeeID": 10014,
    "Employees": "Tamer Anne",
    "Designation": "CFO",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 18,
    "EmployeeImg": "usermale",
    "CurrentSalary": 20998,
    "Address": "Taucherstraße 10",
    "Mail": "tamer68@arpy.com"
  },
  {
    "EmployeeID": 10015,
    "Employees": "Anton Davolio",

```

```

    "Designation": "Project Lead",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 4,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 48232,
    "Address": "Luisenstr. 48",
    "Mail": "anton46@mail.com"
  },
  {
    "EmployeeID": 10016,
    "Employees": "Buchanan Buchanan",
    "Designation": "System Analyst",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "usermale",
    "CurrentSalary": 43041,
    "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
    "Mail": "buchanan68@mail.com"
  },
  {
    "EmployeeID": 10017,
    "Employees": "King Buchanan",
    "Designation": "Program Directory",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 44,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 25259,
    "Address": "Magazinweg 7",
    "Mail": "king80@jourrapide.com"
  },
  {
    "EmployeeID": 10018,
    "Employees": "Rose Michael",
    "Designation": "Project Lead",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 31,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 91156,
    "Address": "Fauntleroy Circus",
    "Mail": "rose75@mail.com"
  },
  {
    "EmployeeID": 10019,
    "Employees": "King Bergrs",
    "Designation": "Developer",

```

```
"Location": "Germany",
>Status": "Active",
>Trustworthiness": "Sufficient",
>Rating": 2,
>Software": 29,
>EmployeeImg": "userfemale",
>CurrentSalary": 28826,
>Address": "2817 Milton Dr.",
>Mail": "king57@jourrapide.com"
},
{
  "EmployeeID": 10020,
  "Employees": "Davolio Fuller",
  "Designation": "Designer",
  "Location": "Canada",
  "Status": "Inactive",
  "Trustworthiness": "Sufficient",
  "Rating": 3,
  "Software": 35,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 71035,
  "Address": "Gran Vía, 1",
  "Mail": "davolio29@arpy.com"
},
{
  "EmployeeID": 10021,
  "Employees": "Rose Rose",
  "Designation": "CFO",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 38,
  "EmployeeImg": "usermale",
  "CurrentSalary": 68123,
  "Address": "Rua do Mercado, 12",
  "Mail": "rose54@arpy.com"
},
{
  "EmployeeID": 10022,
  "Employees": "Andrew Michael",
  "Designation": "Program Directory",
  "Location": "UK",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 2,
  "Software": 61,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 75470,
  "Address": "2, rue du Commerce",
  "Mail": "andrew88@jourrapide.com"
},
{
  "EmployeeID": 10023,
  "Employees": "Davolio Kathryn",
  "Designation": "Manager",
  "Location": "Germany",
```

```

    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 25,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25234,
    "Address": "Hauptstr. 31",
    "Mail": "davolio42@sample.com"
  },
  {
    "EmployeeID": 10024,
    "Employees": "Anne Fleet",
    "Designation": "System Analyst",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 8341,
    "Address": "59 rue de lAbbaye",
    "Mail": "anne86@arpy.com"
  },
  {
    "EmployeeID": 10025,
    "Employees": "Margaret Andrew",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 51,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84975,
    "Address": "P.O. Box 555",
    "Mail": "margaret41@arpy.com"
  },
  {
    "EmployeeID": 10026,
    "Employees": "Kathryn Laura",
    "Designation": "Project Lead",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 48,
    "EmployeeImg": "usermale",
    "CurrentSalary": 97282,
    "Address": "Avda. Azteca 123",
    "Mail": "kathryn82@rpy.com"
  },
  {
    "EmployeeID": 10027,
    "Employees": "Michael Michael",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Inactive",

```

```
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 16,
    "EmployeeImg": "usermale",
    "CurrentSalary": 4184,
    "Address": "Rua do Paço, 67",
    "Mail": "michael58@jourrapide.com"
  },
  {
    "EmployeeID": 10028,
    "Employees": "Leverling Vinet",
    "Designation": "Project Lead",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 57,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 38370,
    "Address": "59 rue de l'Abbaye",
    "Mail": "leverling102@sample.com"
  },
  {
    "EmployeeID": 10029,
    "Employees": "Rose Jack",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 46,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84790,
    "Address": "Rua do Mercado, 12",
    "Mail": "rose108@jourrapide.com"
  },
  {
    "EmployeeID": 10030,
    "Employees": "Vinet Van",
    "Designation": "Developer",
    "Location": "USA",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 40,
    "EmployeeImg": "usermale",
    "CurrentSalary": 71005,
    "Address": "Gran Vía, 1",
    "Mail": "vinet90@jourrapide.com"
  }
]
```

*To change the PDF orientation*

By default, the PDF document is created in **Portrait** orientation. You can change the orientation of the PDF document by using the `args.pdfLayoutSettings.orientation` argument settings in the [beforeSave](#) event.

The possible values are:

- **Portrait** - Used to display content in a vertical layout.
- **Landscape** - Used to display content in a horizontal layout.

**APP.JSX**

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { defaultData } from './datasource';
function App() {
    const beforeSave = (args) => {
        args.pdfLayoutSettings.orientation = 'Landscape'; // You can change
the orientation of the PDF document
    }
    return (<SpreadsheetComponent allowSave={true}
saveUrl='https://services.syncfusion.com/react/production/api/spreadsheet/sa
ve' beforeSave={beforeSave}>
        <SheetsDirective>
            <SheetDirective>
                <RangesDirective>
                    <RangeDirective
dataSource={defaultData}></RangeDirective>
                </RangesDirective>
                <ColumnsDirective>
                    <ColumnDirective width={180}></ColumnDirective>
                    <ColumnDirective width={130}></ColumnDirective>
                    <ColumnDirective width={130}></ColumnDirective>
                </ColumnsDirective>
            </SheetDirective>
        </SheetsDirective>
    </SpreadsheetComponent>);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}
```

**APP.TSX**

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
```

```

import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { defaultData } from './datasource';
function App() {
    const beforeSave = (args): void => {
        args.pdfLayoutSettings.orientation = 'Landscape'; // You can change
the orientation of the PDF document
    }
    return (<SpreadsheetComponent allowSave={true}
saveUrl='https://services.syncfusion.com/react/production/api/spreadsheet/sa
ve' beforeSave={beforeSave}>
        <SheetsDirective>
            <SheetDirective>
                <RangesDirective>
                    <RangeDirective
dataSource={defaultData}></RangeDirective>
                </RangesDirective>
                <ColumnsDirective>
                    <ColumnDirective width={180}></ColumnDirective>
                    <ColumnDirective width={130}></ColumnDirective>
                    <ColumnDirective width={130}></ColumnDirective>
                </ColumnsDirective>
            </SheetDirective>
        </SheetsDirective>
    </SpreadsheetComponent>);
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

### DATASOURCE.JSX

```

/**
 * Default data source
 */
export let defaultData = [
    { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
    { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
    { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
    { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
    { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
'12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
70 },
    { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
    { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
    { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
    { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },

```



```

    { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
      Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
  ];
  /**
   * Grid datasource
   */
  export function getTradeData(dataCount) {
    let employees = [
      'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
      'Fuller', 'Leverling', 'Peacock',
      'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
      'Dodsworth', 'Anne',
      'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
      'Jack', 'Rose'
    ];
    let designation = ['Manager', 'CFO', 'Designer', 'Developer', 'Program
      Directory', 'System Analyst', 'Project Lead'];
    let mail = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
      'jourrapide.com'];
    let location = ['UK', 'USA', 'Sweden', 'France', 'Canada', 'Argentina',
      'Austria', 'Germany', 'Mexico'];
    let status = ['Active', 'Inactive'];
    let trustworthiness = ['Perfect', 'Sufficient', 'Insufficient'];
    let tradeData = [];
    let address = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do Paço, 67',
      '2, rue du Commerce', 'Boulevard Tirou, 255',
      'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
      Mercado, 12',
      'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
      'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
      Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
      'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
      43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
      'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
      10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
      'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
      7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
      'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
      'Taucherstraße 10', 'Taucherstraße 10',
      'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
      Circus', 'Av. dos Lusíadas, 23',
      'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
      123', '2817 Milton Dr.'];
    let employeeimg = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
      dataCount = parseInt(dataCount, 10);
    }
    for (let i = 1; i <= dataCount; i++) {
      let code = 10000;
      tradeData.push({
        'EmployeeID': code + i,
        'Employees': employees[Math.floor(Math.random() *
          employees.length)] + ' ' + employees[Math.floor(Math.random() *
          employees.length)],
        'Designation': designation[Math.floor(Math.random() *
          designation.length)],

```

```

        'Location': location[Math.floor(Math.random() *
location.length)],
        'Status': status[Math.floor(Math.random() * status.length)],
        'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
        'Rating': Math.floor(Math.random() * 5),
        'Software': Math.floor(Math.random() * 100),
        'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
        'CurrentSalary': Math.floor((Math.random() * 100000)),
        'Address': address[Math.floor(Math.random() * address.length)],
    });
    let employee = 'Employees';
    let emp = tradeData[i - 1][employee];
    let sName = emp.substr(0, emp.indexOf(' ')).toLowerCase();
    let empmail = 'Mail';
    tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }
    return tradeData;
}
export let tradeData = [
    {
        "EmployeeID": 10001,
        "Employees": "Laura Nancy",
        "Designation": "Designer",
        "Location": "France",
        "Status": "Inactive",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
        "Software": 69,
        "EmployeeImg": "usermale",
        "CurrentSalary": 84194,
        "Address": "Taucherstraße 10",
        "Mail": "laura15@jourrapide.com"
    },
    {
        "EmployeeID": 10002,
        "Employees": "Zachery Van",
        "Designation": "CFO",
        "Location": "Canada",
        "Status": "Inactive",
        "Trustworthiness": "Insufficient",
        "Rating": 3,
        "Software": 99,
        "EmployeeImg": "usermale",
        "CurrentSalary": 55349,
        "Address": "5ª Ave. Los Palos Grandes",
        "Mail": "zachery109@sample.com"
    },
    {
        "EmployeeID": 10003,
        "Employees": "Rose Fuller",
        "Designation": "CFO",
        "Location": "France",
        "Status": "Active",
        "Trustworthiness": "Insufficient",

```

```

    "Rating": 1,
    "Software": 1,
    "EmployeeImg": "usermale",
    "CurrentSalary": 16477,
    "Address": "2817 Milton Dr.",
    "Mail": "rose55@rpy.com"
  },
  {
    "EmployeeID": 10004,
    "Employees": "Jack Bergs",
    "Designation": "Manager",
    "Location": "Mexico",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 36,
    "EmployeeImg": "usermale",
    "CurrentSalary": 49040,
    "Address": "2, rue du Commerce",
    "Mail": "jack30@sample.com"
  },
  {
    "EmployeeID": 10005,
    "Employees": "Vinet Bergs",
    "Designation": "Program Directory",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 39,
    "EmployeeImg": "usermale",
    "CurrentSalary": 5495,
    "Address": "Rua da Panificadora, 12",
    "Mail": "vinet32@jourrapide.com"
  },
  {
    "EmployeeID": 10006,
    "Employees": "Buchanan Van",
    "Designation": "Designer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 42182,
    "Address": "24, place Kléber",
    "Mail": "buchanan18@mail.com"
  },
  {
    "EmployeeID": 10007,
    "Employees": "Dodsworth Nancy",
    "Designation": "Project Lead",
    "Location": "USA",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 0,

```

```

    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 35776,
    "Address": "Rua do Paço, 67",
    "Mail": "dodsworth84@mail.com"
  },
  {
    "EmployeeID": 10008,
    "Employees": "Laura Jack",
    "Designation": "Developer",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 89,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25108,
    "Address": "Rua da Panificadora, 12",
    "Mail": "laura82@mail.com"
  },
  {
    "EmployeeID": 10009,
    "Employees": "Anne Fuller",
    "Designation": "Program Directory",
    "Location": "Mexico",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 32568,
    "Address": "Gran Vía, 1",
    "Mail": "anne97@jourrapide.com"
  },
  {
    "EmployeeID": 10010,
    "Employees": "Buchanan Andrew",
    "Designation": "Designer",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 62,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 12320,
    "Address": "P.O. Box 555",
    "Mail": "buchanan50@jourrapide.com"
  },
  {
    "EmployeeID": 10011,
    "Employees": "Andrew Janet",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 8,

```

```

      "EmployeeImg": "userfemale",
      "CurrentSalary": 20890,
      "Address": "Starenweg 5",
      "Mail": "andrew63@mail.com"
    },
    {
      "EmployeeID": 10012,
      "Employees": "Margaret Tamer",
      "Designation": "System Analyst",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 4,
      "Software": 7,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 22337,
      "Address": "Magazinweg 7",
      "Mail": "margaret26@mail.com"
    },
    {
      "EmployeeID": 10013,
      "Employees": "Tamer Fuller",
      "Designation": "CFO",
      "Location": "Canada",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 3,
      "Software": 78,
      "EmployeeImg": "usermale",
      "CurrentSalary": 89181,
      "Address": "Taucherstraße 10",
      "Mail": "tamer40@arpy.com"
    },
    {
      "EmployeeID": 10014,
      "Employees": "Tamer Anne",
      "Designation": "CFO",
      "Location": "Sweden",
      "Status": "Active",
      "Trustworthiness": "Sufficient",
      "Rating": 0,
      "Software": 18,
      "EmployeeImg": "usermale",
      "CurrentSalary": 20998,
      "Address": "Taucherstraße 10",
      "Mail": "tamer68@arpy.com"
    },
    {
      "EmployeeID": 10015,
      "Employees": "Anton Davolio",
      "Designation": "Project Lead",
      "Location": "France",
      "Status": "Active",
      "Trustworthiness": "Sufficient",
      "Rating": 4,
      "Software": 8,
      "EmployeeImg": "userfemale",

```

```

    "CurrentSalary": 48232,
    "Address": "Luisenstr. 48",
    "Mail": "anton46@mail.com"
  },
  {
    "EmployeeID": 10016,
    "Employees": "Buchanan Buchanan",
    "Designation": "System Analyst",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "usermale",
    "CurrentSalary": 43041,
    "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
    "Mail": "buchanan68@mail.com"
  },
  {
    "EmployeeID": 10017,
    "Employees": "King Buchanan",
    "Designation": "Program Directory",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 44,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 25259,
    "Address": "Magazinweg 7",
    "Mail": "king80@jourrapide.com"
  },
  {
    "EmployeeID": 10018,
    "Employees": "Rose Michael",
    "Designation": "Project Lead",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 31,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 91156,
    "Address": "Fauntleroy Circus",
    "Mail": "rose75@mail.com"
  },
  {
    "EmployeeID": 10019,
    "Employees": "King Bergs",
    "Designation": "Developer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 2,
    "Software": 29,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 28826,

```

```
      "Address": "2817 Milton Dr.",
      "Mail": "king57@jourrapide.com"
    },
    {
      "EmployeeID": 10020,
      "Employees": "Davolio Fuller",
      "Designation": "Designer",
      "Location": "Canada",
      "Status": "Inactive",
      "Trustworthiness": "Sufficient",
      "Rating": 3,
      "Software": 35,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 71035,
      "Address": "Gran Vía, 1",
      "Mail": "davolio29@arpy.com"
    },
    {
      "EmployeeID": 10021,
      "Employees": "Rose Rose",
      "Designation": "CFO",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Perfect",
      "Rating": 3,
      "Software": 38,
      "EmployeeImg": "usermale",
      "CurrentSalary": 68123,
      "Address": "Rua do Mercado, 12",
      "Mail": "rose54@arpy.com"
    },
    {
      "EmployeeID": 10022,
      "Employees": "Andrew Michael",
      "Designation": "Program Directory",
      "Location": "UK",
      "Status": "Inactive",
      "Trustworthiness": "Insufficient",
      "Rating": 2,
      "Software": 61,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 75470,
      "Address": "2, rue du Commerce",
      "Mail": "andrew88@jourrapide.com"
    },
    {
      "EmployeeID": 10023,
      "Employees": "Davolio Kathryn",
      "Designation": "Manager",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Perfect",
      "Rating": 3,
      "Software": 25,
      "EmployeeImg": "usermale",
      "CurrentSalary": 25234,
      "Address": "Hauptstr. 31",
```

```
    "Mail": "davolio42@sample.com"
  },
  {
    "EmployeeID": 10024,
    "Employees": "Anne Fleet",
    "Designation": "System Analyst",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 8341,
    "Address": "59 rue de l'Abbaye",
    "Mail": "anne86@arpy.com"
  },
  {
    "EmployeeID": 10025,
    "Employees": "Margaret Andrew",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 51,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84975,
    "Address": "P.O. Box 555",
    "Mail": "margaret41@arpy.com"
  },
  {
    "EmployeeID": 10026,
    "Employees": "Kathryn Laura",
    "Designation": "Project Lead",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 48,
    "EmployeeImg": "usermale",
    "CurrentSalary": 97282,
    "Address": "Avda. Azteca 123",
    "Mail": "kathryn82@rpy.com"
  },
  {
    "EmployeeID": 10027,
    "Employees": "Michael Michael",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 16,
    "EmployeeImg": "usermale",
    "CurrentSalary": 4184,
    "Address": "Rua do Paço, 67",
    "Mail": "michael58@jourrapide.com"
```



```

    },
    {
      "EmployeeID": 10028,
      "Employees": "Leverling Vinet",
      "Designation": "Project Lead",
      "Location": "Germany",
      "Status": "Inactive",
      "Trustworthiness": "Perfect",
      "Rating": 0,
      "Software": 57,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 38370,
      "Address": "59 rue de l'Abbaye",
      "Mail": "leverling102@sample.com"
    },
    {
      "EmployeeID": 10029,
      "Employees": "Rose Jack",
      "Designation": "Developer",
      "Location": "UK",
      "Status": "Active",
      "Trustworthiness": "Perfect",
      "Rating": 0,
      "Software": 46,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 84790,
      "Address": "Rua do Mercado, 12",
      "Mail": "rose108@jourrapide.com"
    },
    {
      "EmployeeID": 10030,
      "Employees": "Vinet Van",
      "Designation": "Developer",
      "Location": "USA",
      "Status": "Active",
      "Trustworthiness": "Sufficient",
      "Rating": 0,
      "Software": 40,
      "EmployeeImg": "usermale",
      "CurrentSalary": 71005,
      "Address": "Gran Vía, 1",
      "Mail": "vinet90@jourrapide.com"
    }
  ];

```

**DATASOURCE.TSX**

```

/**
 * Default data source
 */
export let defaultData: Object[] = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
    Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },

```

```

    { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
      Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
    { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
    { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
'12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
70 },
    { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
      Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
    { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
      Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
    { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
      Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
    { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
      Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
    { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
      Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
  ];
  /**
   * Grid datasource
   */
  export function getTradeData(dataCount?: number): object {
    let employees: string[] = [
      'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
      'Fuller', 'Leverling', 'Peacock',
      'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
      'Dodsworth', 'Anne',
      'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
      'Jack', 'Rose'];
    let designation: string[] = ['Manager', 'CFO', 'Designer', 'Developer',
      'Program Director', 'System Analyst', 'Project Lead'];
    let mail: string[] = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
      'jourrapide.com'];
    let location: string[] = ['UK', 'USA', 'Sweden', 'France', 'Canada',
      'Argentina', 'Austria', 'Germany', 'Mexico'];
    let status: string[] = ['Active', 'Inactive'];
    let trustworthiness: string[] = ['Perfect', 'Sufficient',
      'Insufficient'];
    let tradeData: Object[] = [];
    let address: string[] = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do
Paço, 67', '2, rue du Commerce', 'Boulevard Tirou, 255',
      'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
Mercado, 12',
      'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
      'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
      'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
      'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
      'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
      'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
      'Taucherstraße 10', 'Taucherstraße 10',
      'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
Circus', 'Av. dos Lusíadas, 23',

```

```

    'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
123', '2817 Milton Dr.'];
    let employeeimg: string[] = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
        dataCount = parseInt(dataCount, 10);
    }
    for (let i: number = 1; i <= dataCount; i++) {
        let code: any = 10000;
        tradeData.push({
            'EmployeeID': code + i,
            'Employees':
                employees[Math.floor(Math.random() * employees.length)] + '
' + employees[Math.floor(Math.random() * employees.length)],
            'Designation': designation[Math.floor(Math.random() *
designation.length)],
            'Location': location[Math.floor(Math.random() *
location.length)],
            'Status': status[Math.floor(Math.random() * status.length)],
            'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
            'Rating': Math.floor(Math.random() * 5),
            'Software': Math.floor(Math.random() * 100),
            'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
            'CurrentSalary': Math.floor((Math.random() * 100000)),
            'Address': address[Math.floor(Math.random() * address.length)],
        });
        let employee: string = 'Employees';
        let emp: string = tradeData[i - 1][employee];
        let sName: string = emp.substr(0, emp.indexOf(' ')).toLowerCase();
        let empmail: string = 'Mail';
        tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }
    return tradeData;
}
export let tradeData: Object[] = [
    {
        "EmployeeID": 10001,
        "Employees": "Laura Nancy",
        "Designation": "Designer",
        "Location": "France",
        "Status": "Inactive",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
        "Software": 69,
        "EmployeeImg": "usermale",
        "CurrentSalary": 84194,
        "Address": "Taucherstraße 10",
        "Mail": "laura15@jourrapide.com"
    },
    {
        "EmployeeID": 10002,
        "Employees": "Zachery Van",
        "Designation": "CFO",
        "Location": "Canada",
        "Status": "Inactive",
    }
]

```

```

    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 99,
    "EmployeeImg": "usermale",
    "CurrentSalary": 55349,
    "Address": "5ª Ave. Los Palos Grandes",
    "Mail": "zachery109@sample.com"
  },
  {
    "EmployeeID": 10003,
    "Employees": "Rose Fuller",
    "Designation": "CFO",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 1,
    "EmployeeImg": "usermale",
    "CurrentSalary": 16477,
    "Address": "2817 Milton Dr.",
    "Mail": "rose55@rpy.com"
  },
  {
    "EmployeeID": 10004,
    "Employees": "Jack Bergs",
    "Designation": "Manager",
    "Location": "Mexico",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 36,
    "EmployeeImg": "usermale",
    "CurrentSalary": 49040,
    "Address": "2, rue du Commerce",
    "Mail": "jack30@sample.com"
  },
  {
    "EmployeeID": 10005,
    "Employees": "Vinet Bergs",
    "Designation": "Program Directory",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 39,
    "EmployeeImg": "usermale",
    "CurrentSalary": 5495,
    "Address": "Rua da Panificadora, 12",
    "Mail": "vinet32@jourrapide.com"
  },
  {
    "EmployeeID": 10006,
    "Employees": "Buchanan Van",
    "Designation": "Designer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",

```

```
"Rating": 4,
"Software": 78,
"EmployeeImg": "usermale",
"CurrentSalary": 42182,
"Address": "24, place Kléber",
"Mail": "buchanan18@mail.com"
},
{
  "EmployeeID": 10007,
  "Employees": "Dodsworth Nancy",
  "Designation": "Project Lead",
  "Location": "USA",
  "Status": "Inactive",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 0,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 35776,
  "Address": "Rua do Paço, 67",
  "Mail": "dodsworth84@mail.com"
},
{
  "EmployeeID": 10008,
  "Employees": "Laura Jack",
  "Designation": "Developer",
  "Location": "Austria",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 89,
  "EmployeeImg": "usermale",
  "CurrentSalary": 25108,
  "Address": "Rua da Panificadora, 12",
  "Mail": "laura82@mail.com"
},
{
  "EmployeeID": 10009,
  "Employees": "Anne Fuller",
  "Designation": "Program Directory",
  "Location": "Mexico",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 0,
  "Software": 19,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 32568,
  "Address": "Gran Vía, 1",
  "Mail": "anne97@jourrapide.com"
},
{
  "EmployeeID": 10010,
  "Employees": "Buchanan Andrew",
  "Designation": "Designer",
  "Location": "Austria",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
```

```

    "Software": 62,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 12320,
    "Address": "P.O. Box 555",
    "Mail": "buchanan50@jourrapide.com"
  },
  {
    "EmployeeID": 10011,
    "Employees": "Andrew Janet",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 20890,
    "Address": "Starenweg 5",
    "Mail": "andrew63@mail.com"
  },
  {
    "EmployeeID": 10012,
    "Employees": "Margaret Tamer",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 7,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 22337,
    "Address": "Magazinweg 7",
    "Mail": "margaret26@mail.com"
  },
  {
    "EmployeeID": 10013,
    "Employees": "Tamer Fuller",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 89181,
    "Address": "Taucherstraße 10",
    "Mail": "tamer40@arpy.com"
  },
  {
    "EmployeeID": 10014,
    "Employees": "Tamer Anne",
    "Designation": "CFO",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 18,

```

```
"EmployeeImg": "usermale",
"CurrentSalary": 20998,
"Address": "Taucherstraße 10",
"Mail": "tamer68@arpy.com"
},
{
  "EmployeeID": 10015,
  "Employees": "Anton Davolio",
  "Designation": "Project Lead",
  "Location": "France",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 4,
  "Software": 8,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 48232,
  "Address": "Luisenstr. 48",
  "Mail": "anton46@mail.com"
},
{
  "EmployeeID": 10016,
  "Employees": "Buchanan Buchanan",
  "Designation": "System Analyst",
  "Location": "Austria",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 19,
  "EmployeeImg": "usermale",
  "CurrentSalary": 43041,
  "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
  "Mail": "buchanan68@mail.com"
},
{
  "EmployeeID": 10017,
  "Employees": "King Buchanan",
  "Designation": "Program Directory",
  "Location": "Sweden",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 44,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 25259,
  "Address": "Magazinweg 7",
  "Mail": "king80@jourrapide.com"
},
{
  "EmployeeID": 10018,
  "Employees": "Rose Michael",
  "Designation": "Project Lead",
  "Location": "Canada",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 4,
  "Software": 31,
  "EmployeeImg": "userfemale",
```

```
"CurrentSalary": 91156,  
"Address": "Fauntleroy Circus",  
"Mail": "rose75@mail.com"  
},  
{  
  "EmployeeID": 10019,  
  "Employees": "King Bergs",  
  "Designation": "Developer",  
  "Location": "Germany",  
  "Status": "Active",  
  "Trustworthiness": "Sufficient",  
  "Rating": 2,  
  "Software": 29,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 28826,  
  "Address": "2817 Milton Dr.",  
  "Mail": "king57@jourrapide.com"  
},  
{  
  "EmployeeID": 10020,  
  "Employees": "Davolio Fuller",  
  "Designation": "Designer",  
  "Location": "Canada",  
  "Status": "Inactive",  
  "Trustworthiness": "Sufficient",  
  "Rating": 3,  
  "Software": 35,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 71035,  
  "Address": "Gran Vía, 1",  
  "Mail": "davolio29@arpy.com"  
},  
{  
  "EmployeeID": 10021,  
  "Employees": "Rose Rose",  
  "Designation": "CFO",  
  "Location": "Germany",  
  "Status": "Active",  
  "Trustworthiness": "Perfect",  
  "Rating": 3,  
  "Software": 38,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 68123,  
  "Address": "Rua do Mercado, 12",  
  "Mail": "rose54@arpy.com"  
},  
{  
  "EmployeeID": 10022,  
  "Employees": "Andrew Michael",  
  "Designation": "Program Directory",  
  "Location": "UK",  
  "Status": "Inactive",  
  "Trustworthiness": "Insufficient",  
  "Rating": 2,  
  "Software": 61,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 75470,
```



```
"Address": "2, rue du Commerce",
"Mail": "andrew88@jourrapide.com"
},
{
  "EmployeeID": 10023,
  "Employees": "Davolio Kathryn",
  "Designation": "Manager",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 25,
  "EmployeeImg": "usermale",
  "CurrentSalary": 25234,
  "Address": "Hauptstr. 31",
  "Mail": "davolio42@sample.com"
},
{
  "EmployeeID": 10024,
  "Employees": "Anne Fleet",
  "Designation": "System Analyst",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 0,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 8341,
  "Address": "59 rue de l'Abbaye",
  "Mail": "anne86@arpy.com"
},
{
  "EmployeeID": 10025,
  "Employees": "Margaret Andrew",
  "Designation": "System Analyst",
  "Location": "Germany",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 51,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 84975,
  "Address": "P.O. Box 555",
  "Mail": "margaret41@arpy.com"
},
{
  "EmployeeID": 10026,
  "Employees": "Kathryn Laura",
  "Designation": "Project Lead",
  "Location": "Austria",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 48,
  "EmployeeImg": "usermale",
  "CurrentSalary": 97282,
  "Address": "Avda. Azteca 123",
```

```
"Mail": "kathryn82@rpy.com"
},
{
  "EmployeeID": 10027,
  "Employees": "Michael Michael",
  "Designation": "Developer",
  "Location": "UK",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 4,
  "Software": 16,
  "EmployeeImg": "usermale",
  "CurrentSalary": 4184,
  "Address": "Rua do Paço, 67",
  "Mail": "michael58@jourrapide.com"
},
{
  "EmployeeID": 10028,
  "Employees": "Leverling Vinet",
  "Designation": "Project Lead",
  "Location": "Germany",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 57,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 38370,
  "Address": "59 rue de l'Abbaye",
  "Mail": "leverling102@sample.com"
},
{
  "EmployeeID": 10029,
  "Employees": "Rose Jack",
  "Designation": "Developer",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 46,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 84790,
  "Address": "Rua do Mercado, 12",
  "Mail": "rose108@jourrapide.com"
},
{
  "EmployeeID": 10030,
  "Employees": "Vinet Van",
  "Designation": "Developer",
  "Location": "USA",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 40,
  "EmployeeImg": "usermale",
  "CurrentSalary": 71005,
  "Address": "Gran Vía, 1",
  "Mail": "vinet90@jourrapide.com"
```

```
}  
]
```

### Supported file formats

The following list of Excel file formats are supported in Spreadsheet:

- MS Excel (.xlsx)
- MS Excel 97-2003 (.xls)
- Comma Separated Values (.csv)
- Portable Document Format (.pdf)

### Methods

To save the Spreadsheet document as an `xlsx`, `xls`, `csv`, or `pdf` file, by using `save` method should be called with the `url`, `fileName` and `saveType` as parameters. The following code example shows to save the spreadsheet file as an `xlsx`, `xls`, `csv`, or `pdf` in the button click event.

### APP.JSX

```
{% raw %}  
import * as React from 'react';  
import { createRoot } from 'react-dom/client';  
import { SpreadsheetComponent, SheetsDirective, SheetDirective,  
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';  
import { RangeDirective, ColumnsDirective, ColumnDirective } from  
'@syncfusion/ej2-react-spreadsheet';  
import { DropDownButtonComponent } from '@syncfusion/ej2-react-  
splitbuttons';  
import { defaultData } from './datasource';  
function App() {  
    const spreadsheetRef = React.useRef(null);  
    const items = [  
        { text: "Save As xlsx" },  
        { text: "Save As xls" },  
        { text: "Save As csv" },  
        { text: "Save As pdf" }  
    ];  
    const itemSelect = (args) => {  
        let spreadsheet = spreadsheetRef.current;  
        if (spreadsheet) {  
            if (args.item.text === 'Save As xlsx')  
                spreadsheet.save({ url:  
'https://services.syncfusion.com/react/production/api/spreadsheet/save',  
fileName: "Sample", saveType: "Xlsx" });  
            if (args.item.text === 'Save As xls')  
                spreadsheet.save({ url:  
'https://services.syncfusion.com/react/production/api/spreadsheet/save',  
fileName: "Sample", saveType: "Xls" });  
            if (args.item.text === 'Save As csv')  
                spreadsheet.save({ url:  
'https://services.syncfusion.com/react/production/api/spreadsheet/save',  
fileName: "Sample", saveType: "Csv" });  
            if (args.item.text === 'Save As pdf')
```

```

        spreadsheet.save({ url:
'https://services.syncfusion.com/react/production/api/spreadsheet/save',
fileName: "Sample", saveType: "Pdf" });
    }
};
return (
    <div>
        <DropDownButtonComponent items={items} select={itemSelect}> Save
    </DropDownButtonComponent>
        <SpreadsheetComponent ref={spreadsheetRef} >
            <SheetsDirective>
                <SheetDirective>
                    <RangesDirective>
                        <RangeDirective
dataSource={defaultData}></RangeDirective>
                    </RangesDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={180}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
            </SheetsDirective>
        </SpreadsheetComponent>
    </div>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

**APP.TSX**

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { DropDownButtonComponent, ItemModel, MenuEventArgs } from
'@syncfusion/ej2-react-splitbuttons';
import { defaultData } from './datasource';
function App() {
    const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
    const items: ItemModel[] = [
        { text: "Save As xlsx" },
        { text: "Save As xls" },
        { text: "Save As csv" },
        { text: "Save As pdf" }
    ];
    const itemSelect = (args: MenuEventArgs): void => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            if (args.item.text === 'Save As xlsx')

```

```

        spreadsheet.save({ url:
'https://services.syncfusion.com/react/production/api/spreadsheet/save',
fileName: "Sample", saveType: "Xlsx" });
        if (args.item.text === 'Save As xls')
            spreadsheet.save({ url:
'https://services.syncfusion.com/react/production/api/spreadsheet/save',
fileName: "Sample", saveType: "Xls" });
        if (args.item.text === 'Save As csv')
            spreadsheet.save({ url:
'https://services.syncfusion.com/react/production/api/spreadsheet/save',
fileName: "Sample", saveType: "Csv" });
        if (args.item.text === 'Save As pdf')
            spreadsheet.save({ url:
'https://services.syncfusion.com/react/production/api/spreadsheet/save',
fileName: "Sample", saveType: "Pdf" });
    }
};
return (
    <div>
        <DropDownButtonComponent items={items} select={itemSelect}> Save
    </DropDownButtonComponent>
        <SpreadsheetComponent ref={spreadsheetRef} >
            <SheetsDirective>
                <SheetDirective>
                    <RangesDirective>
                        <RangeDirective
dataSource={defaultData}></RangeDirective>
                    </RangesDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={180}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
            </SheetsDirective>
        </SpreadsheetComponent>
    </div>
);
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

### DATASOURCE.JSX

```

/**
 * Default data source
 */
export let defaultData = [
    { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
    { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
    { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },

```

```

    { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54 AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
    { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time: '12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit: 70 },
    { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM', Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
    { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM', Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
    { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM', Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
    { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM', Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
    { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM', Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
  ];
  /**
   * Grid datasource
   */
  export function getTradeData(dataCount) {
    let employees = [
      'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
      'Fuller', 'Leverling', 'Peacock',
      'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
      'Dodsworth', 'Anne',
      'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
      'Jack', 'Rose'
    ];
    let designation = ['Manager', 'CFO', 'Designer', 'Developer', 'Program Directory', 'System Analyst', 'Project Lead'];
    let mail = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com', 'jourrapide.com'];
    let location = ['UK', 'USA', 'Sweden', 'France', 'Canada', 'Argentina', 'Austria', 'Germany', 'Mexico'];
    let status = ['Active', 'Inactive'];
    let trustworthiness = ['Perfect', 'Sufficient', 'Insufficient'];
    let tradeData = [];
    let address = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do Paço, 67', '2, rue du Commerce', 'Boulevard Tirou, 255', 'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do Mercado, 12', 'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6', 'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6', 'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz 43', '5a Ave. Los Palos Grandes', '1029 - 12th Ave. S.', 'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße 10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22', 'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg 7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1', 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7', 'Taucherstraße 10', 'Taucherstraße 10', 'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy Circus', 'Av. dos Lusíadas, 23', 'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca 123', '2817 Milton Dr.'];
    let employeeimg = ['usermale', 'userfemale'];
  }

```

```

    if (typeof dataCount === 'string') {
      dataCount = parseInt(dataCount, 10);
    }
    for (let i = 1; i <= dataCount; i++) {
      let code = 10000;
      tradeData.push({
        'EmployeeID': code + i,
        'Employees': employees[Math.floor(Math.random() *
employees.length)] + ' ' + employees[Math.floor(Math.random() *
employees.length)],
        'Designation': designation[Math.floor(Math.random() *
designation.length)],
        'Location': location[Math.floor(Math.random() *
location.length)],
        'Status': status[Math.floor(Math.random() * status.length)],
        'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
        'Rating': Math.floor(Math.random() * 5),
        'Software': Math.floor(Math.random() * 100),
        'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
        'CurrentSalary': Math.floor((Math.random() * 100000)),
        'Address': address[Math.floor(Math.random() * address.length)],
      });
      let employee = 'Employees';
      let emp = tradeData[i - 1][employee];
      let sName = emp.substr(0, emp.indexOf(' ')).toLowerCase();
      let empmail = 'Mail';
      tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }
    return tradeData;
  }
export let tradeData = [
  {
    "EmployeeID": 10001,
    "Employees": "Laura Nancy",
    "Designation": "Designer",
    "Location": "France",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 69,
    "EmployeeImg": "usermale",
    "CurrentSalary": 84194,
    "Address": "Taucherstraße 10",
    "Mail": "laural5@jourrapide.com"
  },
  {
    "EmployeeID": 10002,
    "Employees": "Zachery Van",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 99,

```

```

      "EmployeeImg": "usermale",
      "CurrentSalary": 55349,
      "Address": "5ª Ave. Los Palos Grandes",
      "Mail": "zachery109@sample.com"
    },
    {
      "EmployeeID": 10003,
      "Employees": "Rose Fuller",
      "Designation": "CFO",
      "Location": "France",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 1,
      "Software": 1,
      "EmployeeImg": "usermale",
      "CurrentSalary": 16477,
      "Address": "2817 Milton Dr.",
      "Mail": "rose55@rpy.com"
    },
    {
      "EmployeeID": 10004,
      "Employees": "Jack Bergs",
      "Designation": "Manager",
      "Location": "Mexico",
      "Status": "Inactive",
      "Trustworthiness": "Insufficient",
      "Rating": 3,
      "Software": 36,
      "EmployeeImg": "usermale",
      "CurrentSalary": 49040,
      "Address": "2, rue du Commerce",
      "Mail": "jack30@sample.com"
    },
    {
      "EmployeeID": 10005,
      "Employees": "Vinet Bergs",
      "Designation": "Program Directory",
      "Location": "UK",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 1,
      "Software": 39,
      "EmployeeImg": "usermale",
      "CurrentSalary": 5495,
      "Address": "Rua da Panificadora, 12",
      "Mail": "vinet32@jourrapide.com"
    },
    {
      "EmployeeID": 10006,
      "Employees": "Buchanan Van",
      "Designation": "Designer",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 4,
      "Software": 78,
      "EmployeeImg": "usermale",

```



```
"CurrentSalary": 42182,  
"Address": "24, place Kléber",  
"Mail": "buchanan18@mail.com"  
},  
{  
  "EmployeeID": 10007,  
  "Employees": "Dodsworth Nancy",  
  "Designation": "Project Lead",  
  "Location": "USA",  
  "Status": "Inactive",  
  "Trustworthiness": "Sufficient",  
  "Rating": 0,  
  "Software": 0,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 35776,  
  "Address": "Rua do Paço, 67",  
  "Mail": "dodsworth84@mail.com"  
},  
{  
  "EmployeeID": 10008,  
  "Employees": "Laura Jack",  
  "Designation": "Developer",  
  "Location": "Austria",  
  "Status": "Inactive",  
  "Trustworthiness": "Perfect",  
  "Rating": 3,  
  "Software": 89,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 25108,  
  "Address": "Rua da Panificadora, 12",  
  "Mail": "laura82@mail.com"  
},  
{  
  "EmployeeID": 10009,  
  "Employees": "Anne Fuller",  
  "Designation": "Program Directory",  
  "Location": "Mexico",  
  "Status": "Active",  
  "Trustworthiness": "Insufficient",  
  "Rating": 0,  
  "Software": 19,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 32568,  
  "Address": "Gran Vía, 1",  
  "Mail": "anne97@jourrapide.com"  
},  
{  
  "EmployeeID": 10010,  
  "Employees": "Buchanan Andrew",  
  "Designation": "Designer",  
  "Location": "Austria",  
  "Status": "Active",  
  "Trustworthiness": "Insufficient",  
  "Rating": 1,  
  "Software": 62,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 12320,
```

```

      "Address": "P.O. Box 555",
      "Mail": "buchanan50@jourrapide.com"
    },
    {
      "EmployeeID": 10011,
      "Employees": "Andrew Janet",
      "Designation": "System Analyst",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 3,
      "Software": 8,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 20890,
      "Address": "Starenweg 5",
      "Mail": "andrew63@mail.com"
    },
    {
      "EmployeeID": 10012,
      "Employees": "Margaret Tamer",
      "Designation": "System Analyst",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 4,
      "Software": 7,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 22337,
      "Address": "Magazinweg 7",
      "Mail": "margaret26@mail.com"
    },
    {
      "EmployeeID": 10013,
      "Employees": "Tamer Fuller",
      "Designation": "CFO",
      "Location": "Canada",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 3,
      "Software": 78,
      "EmployeeImg": "usermale",
      "CurrentSalary": 89181,
      "Address": "Taucherstraße 10",
      "Mail": "tamer40@arpy.com"
    },
    {
      "EmployeeID": 10014,
      "Employees": "Tamer Anne",
      "Designation": "CFO",
      "Location": "Sweden",
      "Status": "Active",
      "Trustworthiness": "Sufficient",
      "Rating": 0,
      "Software": 18,
      "EmployeeImg": "usermale",
      "CurrentSalary": 20998,
      "Address": "Taucherstraße 10",

```

```

    "Mail": "tamer68@arpy.com"
  },
  {
    "EmployeeID": 10015,
    "Employees": "Anton Davolio",
    "Designation": "Project Lead",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 4,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 48232,
    "Address": "Luisenstr. 48",
    "Mail": "anton46@mail.com"
  },
  {
    "EmployeeID": 10016,
    "Employees": "Buchanan Buchanan",
    "Designation": "System Analyst",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "usermale",
    "CurrentSalary": 43041,
    "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
    "Mail": "buchanan68@mail.com"
  },
  {
    "EmployeeID": 10017,
    "Employees": "King Buchanan",
    "Designation": "Program Directory",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 44,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 25259,
    "Address": "Magazinweg 7",
    "Mail": "king80@jourrapide.com"
  },
  {
    "EmployeeID": 10018,
    "Employees": "Rose Michael",
    "Designation": "Project Lead",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 31,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 91156,
    "Address": "Fauntleroy Circus",
    "Mail": "rose75@mail.com"
  }

```

```

    },
    {
      "EmployeeID": 10019,
      "Employees": "King Bergs",
      "Designation": "Developer",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Sufficient",
      "Rating": 2,
      "Software": 29,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 28826,
      "Address": "2817 Milton Dr.",
      "Mail": "king57@jourrapide.com"
    },
    {
      "EmployeeID": 10020,
      "Employees": "Davolio Fuller",
      "Designation": "Designer",
      "Location": "Canada",
      "Status": "Inactive",
      "Trustworthiness": "Sufficient",
      "Rating": 3,
      "Software": 35,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 71035,
      "Address": "Gran Vía, 1",
      "Mail": "davolio29@arpy.com"
    },
    {
      "EmployeeID": 10021,
      "Employees": "Rose Rose",
      "Designation": "CFO",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Perfect",
      "Rating": 3,
      "Software": 38,
      "EmployeeImg": "usermale",
      "CurrentSalary": 68123,
      "Address": "Rua do Mercado, 12",
      "Mail": "rose54@arpy.com"
    },
    {
      "EmployeeID": 10022,
      "Employees": "Andrew Michael",
      "Designation": "Program Directory",
      "Location": "UK",
      "Status": "Inactive",
      "Trustworthiness": "Insufficient",
      "Rating": 2,
      "Software": 61,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 75470,
      "Address": "2, rue du Commerce",
      "Mail": "andrew88@jourrapide.com"
    },
  ],

```

```

{
  "EmployeeID": 10023,
  "Employees": "Davolio Kathryn",
  "Designation": "Manager",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 25,
  "EmployeeImg": "usermale",
  "CurrentSalary": 25234,
  "Address": "Hauptstr. 31",
  "Mail": "davolio42@sample.com"
},
{
  "EmployeeID": 10024,
  "Employees": "Anne Fleet",
  "Designation": "System Analyst",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 0,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 8341,
  "Address": "59 rue de l'Abbaye",
  "Mail": "anne86@arpy.com"
},
{
  "EmployeeID": 10025,
  "Employees": "Margaret Andrew",
  "Designation": "System Analyst",
  "Location": "Germany",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 51,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 84975,
  "Address": "P.O. Box 555",
  "Mail": "margaret41@arpy.com"
},
{
  "EmployeeID": 10026,
  "Employees": "Kathryn Laura",
  "Designation": "Project Lead",
  "Location": "Austria",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 48,
  "EmployeeImg": "usermale",
  "CurrentSalary": 97282,
  "Address": "Avda. Azteca 123",
  "Mail": "kathryn82@rpy.com"
},
{

```

```
"EmployeeID": 10027,  
"Employees": "Michael Michael",  
"Designation": "Developer",  
"Location": "UK",  
"Status": "Inactive",  
"Trustworthiness": "Perfect",  
"Rating": 4,  
"Software": 16,  
"EmployeeImg": "usermale",  
"CurrentSalary": 4184,  
"Address": "Rua do Paço, 67",  
"Mail": "michael58@jourrapide.com"  
},  
{  
  "EmployeeID": 10028,  
  "Employees": "Leverling Vinet",  
  "Designation": "Project Lead",  
  "Location": "Germany",  
  "Status": "Inactive",  
  "Trustworthiness": "Perfect",  
  "Rating": 0,  
  "Software": 57,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 38370,  
  "Address": "59 rue de l'Abbaye",  
  "Mail": "leverling102@sample.com"  
},  
{  
  "EmployeeID": 10029,  
  "Employees": "Rose Jack",  
  "Designation": "Developer",  
  "Location": "UK",  
  "Status": "Active",  
  "Trustworthiness": "Perfect",  
  "Rating": 0,  
  "Software": 46,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 84790,  
  "Address": "Rua do Mercado, 12",  
  "Mail": "rose108@jourrapide.com"  
},  
{  
  "EmployeeID": 10030,  
  "Employees": "Vinet Van",  
  "Designation": "Developer",  
  "Location": "USA",  
  "Status": "Active",  
  "Trustworthiness": "Sufficient",  
  "Rating": 0,  
  "Software": 40,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 71005,  
  "Address": "Gran Vía, 1",  
  "Mail": "vinet90@jourrapide.com"  
}  
];
```

**DATASOURCE.TSX**

```

/**
 * Default data source
 */
export let defaultData: Object[] = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
    Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
  { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
    Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
  { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54 AM',
    Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
  { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time: '12:43:59 AM',
    Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit: 70 },
  { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
    Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
  { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
    Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
  { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
    Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
  { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
    Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
  { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
    Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
/**
 * Grid datasource
 */
export function getTradeData(dataCount?: number): object {
  let employees: string[] = [
    'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
    'Fuller', 'Leverling', 'Peacock',
    'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
    'Dodsworth', 'Anne',
    'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
    'Jack', 'Rose'];
  let designation: string[] = ['Manager', 'CFO', 'Designer', 'Developer',
    'Program Directory', 'System Analyst', 'Project Lead'];
  let mail: string[] = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
    'jourrapide.com'];
  let location: string[] = ['UK', 'USA', 'Sweden', 'France', 'Canada',
    'Argentina', 'Austria', 'Germany', 'Mexico'];
  let status: string[] = ['Active', 'Inactive'];
  let trustworthiness: string[] = ['Perfect', 'Sufficient',
    'Insufficient'];
  let tradeData: Object[] = [];
  let address: string[] = ['59 rue de lAbbaye', 'Luisenstr. 48', 'Rua do
    Paço, 67', '2, rue du Commerce', 'Boulevard Tirou, 255',
    'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
    Mercado, 12',
    'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',

```

```

'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
'Åkergratan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
'Taucherstraße 10', 'Taucherstraße 10',
'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
Circus', 'Av. dos Lusíadas, 23',
'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
123', '2817 Milton Dr.'];
let employeeimg: string[] = ['usermale', 'userfemale'];
if (typeof dataCount === 'string') {
    dataCount = parseInt(dataCount, 10);
}
for (let i: number = 1; i <= dataCount; i++) {
    let code: any = 10000;
    tradeData.push({
        'EmployeeID': code + i,
        'Employees':
            employees[Math.floor(Math.random() * employees.length)] + '
' + employees[Math.floor(Math.random() * employees.length)],
        'Designation': designation[Math.floor(Math.random() *
designation.length)],
        'Location': location[Math.floor(Math.random() *
location.length)],
        'Status': status[Math.floor(Math.random() * status.length)],
        'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
        'Rating': Math.floor(Math.random() * 5),
        'Software': Math.floor(Math.random() * 100),
        'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
        'CurrentSalary': Math.floor((Math.random() * 100000)),
        'Address': address[Math.floor(Math.random() * address.length)],
    });
    let employee: string = 'Employees';
    let emp: string = tradeData[i - 1][employee];
    let sName: string = emp.substr(0, emp.indexOf(' ')).toLowerCase();
    let empmail: string = 'Mail';
    tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
}
return tradeData;
}
export let tradeData: Object[] = [
    {
        "EmployeeID": 10001,
        "Employees": "Laura Nancy",
        "Designation": "Designer",
        "Location": "France",
        "Status": "Inactive",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
    }
]

```



```

    "Software": 69,
    "EmployeeImg": "usermale",
    "CurrentSalary": 84194,
    "Address": "Taucherstraße 10",
    "Mail": "laura15@jourrapide.com"
  },
  {
    "EmployeeID": 10002,
    "Employees": "Zachery Van",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 99,
    "EmployeeImg": "usermale",
    "CurrentSalary": 55349,
    "Address": "5a Ave. Los Palos Grandes",
    "Mail": "zachery109@sample.com"
  },
  {
    "EmployeeID": 10003,
    "Employees": "Rose Fuller",
    "Designation": "CFO",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 1,
    "EmployeeImg": "usermale",
    "CurrentSalary": 16477,
    "Address": "2817 Milton Dr.",
    "Mail": "rose55@rpy.com"
  },
  {
    "EmployeeID": 10004,
    "Employees": "Jack Bergs",
    "Designation": "Manager",
    "Location": "Mexico",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 36,
    "EmployeeImg": "usermale",
    "CurrentSalary": 49040,
    "Address": "2, rue du Commerce",
    "Mail": "jack30@sample.com"
  },
  {
    "EmployeeID": 10005,
    "Employees": "Vinet Bergs",
    "Designation": "Program Directory",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 39,

```

```

    "EmployeeImg": "usermale",
    "CurrentSalary": 5495,
    "Address": "Rua da Panificadora, 12",
    "Mail": "vinet32@jourrapide.com"
  },
  {
    "EmployeeID": 10006,
    "Employees": "Buchanan Van",
    "Designation": "Designer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 42182,
    "Address": "24, place Kléber",
    "Mail": "buchanan18@mail.com"
  },
  {
    "EmployeeID": 10007,
    "Employees": "Dodsworth Nancy",
    "Designation": "Project Lead",
    "Location": "USA",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 35776,
    "Address": "Rua do Paço, 67",
    "Mail": "dodsworth84@mail.com"
  },
  {
    "EmployeeID": 10008,
    "Employees": "Laura Jack",
    "Designation": "Developer",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 89,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25108,
    "Address": "Rua da Panificadora, 12",
    "Mail": "laura82@mail.com"
  },
  {
    "EmployeeID": 10009,
    "Employees": "Anne Fuller",
    "Designation": "Program Directory",
    "Location": "Mexico",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "userfemale",

```

```
"CurrentSalary": 32568,  
"Address": "Gran Vía, 1",  
"Mail": "anne97@jourrapide.com"  
},  
{  
  "EmployeeID": 10010,  
  "Employees": "Buchanan Andrew",  
  "Designation": "Designer",  
  "Location": "Austria",  
  "Status": "Active",  
  "Trustworthiness": "Insufficient",  
  "Rating": 1,  
  "Software": 62,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 12320,  
  "Address": "P.O. Box 555",  
  "Mail": "buchanan50@jourrapide.com"  
},  
{  
  "EmployeeID": 10011,  
  "Employees": "Andrew Janet",  
  "Designation": "System Analyst",  
  "Location": "Germany",  
  "Status": "Active",  
  "Trustworthiness": "Insufficient",  
  "Rating": 3,  
  "Software": 8,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 20890,  
  "Address": "Starenweg 5",  
  "Mail": "andrew63@mail.com"  
},  
{  
  "EmployeeID": 10012,  
  "Employees": "Margaret Tamer",  
  "Designation": "System Analyst",  
  "Location": "Germany",  
  "Status": "Active",  
  "Trustworthiness": "Insufficient",  
  "Rating": 4,  
  "Software": 7,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 22337,  
  "Address": "Magazinweg 7",  
  "Mail": "margaret26@mail.com"  
},  
{  
  "EmployeeID": 10013,  
  "Employees": "Tamer Fuller",  
  "Designation": "CFO",  
  "Location": "Canada",  
  "Status": "Active",  
  "Trustworthiness": "Insufficient",  
  "Rating": 3,  
  "Software": 78,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 89181,
```

```

    "Address": "Taucherstraße 10",
    "Mail": "tamer40@arpy.com"
  },
  {
    "EmployeeID": 10014,
    "Employees": "Tamer Anne",
    "Designation": "CFO",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 18,
    "EmployeeImg": "usermale",
    "CurrentSalary": 20998,
    "Address": "Taucherstraße 10",
    "Mail": "tamer68@arpy.com"
  },
  {
    "EmployeeID": 10015,
    "Employees": "Anton Davolio",
    "Designation": "Project Lead",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 4,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 48232,
    "Address": "Luisenstr. 48",
    "Mail": "anton46@mail.com"
  },
  {
    "EmployeeID": 10016,
    "Employees": "Buchanan Buchanan",
    "Designation": "System Analyst",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "usermale",
    "CurrentSalary": 43041,
    "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
    "Mail": "buchanan68@mail.com"
  },
  {
    "EmployeeID": 10017,
    "Employees": "King Buchanan",
    "Designation": "Program Directory",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 44,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 25259,
    "Address": "Magazinweg 7",

```

```

    "Mail": "king80@jourrapide.com"
  },
  {
    "EmployeeID": 10018,
    "Employees": "Rose Michael",
    "Designation": "Project Lead",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 31,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 91156,
    "Address": "Fauntleroy Circus",
    "Mail": "rose75@mail.com"
  },
  {
    "EmployeeID": 10019,
    "Employees": "King Bergrs",
    "Designation": "Developer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 2,
    "Software": 29,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 28826,
    "Address": "2817 Milton Dr.",
    "Mail": "king57@jourrapide.com"
  },
  {
    "EmployeeID": 10020,
    "Employees": "Davolio Fuller",
    "Designation": "Designer",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 3,
    "Software": 35,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 71035,
    "Address": "Gran Vía, 1",
    "Mail": "davolio29@arpy.com"
  },
  {
    "EmployeeID": 10021,
    "Employees": "Rose Rose",
    "Designation": "CFO",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 38,
    "EmployeeImg": "usermale",
    "CurrentSalary": 68123,
    "Address": "Rua do Mercado, 12",
    "Mail": "rose54@arpy.com"
  }

```

```
},
{
  "EmployeeID": 10022,
  "Employees": "Andrew Michael",
  "Designation": "Program Directory",
  "Location": "UK",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 2,
  "Software": 61,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 75470,
  "Address": "2, rue du Commerce",
  "Mail": "andrew88@jourrapide.com"
},
{
  "EmployeeID": 10023,
  "Employees": "Davolio Kathryn",
  "Designation": "Manager",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 25,
  "EmployeeImg": "usermale",
  "CurrentSalary": 25234,
  "Address": "Hauptstr. 31",
  "Mail": "davolio42@sample.com"
},
{
  "EmployeeID": 10024,
  "Employees": "Anne Fleet",
  "Designation": "System Analyst",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 0,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 8341,
  "Address": "59 rue de lAbbaye",
  "Mail": "anne86@arpy.com"
},
{
  "EmployeeID": 10025,
  "Employees": "Margaret Andrew",
  "Designation": "System Analyst",
  "Location": "Germany",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 51,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 84975,
  "Address": "P.O. Box 555",
  "Mail": "margaret41@arpy.com"
},
}
```

```
{
  "EmployeeID": 10026,
  "Employees": "Kathryn Laura",
  "Designation": "Project Lead",
  "Location": "Austria",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 48,
  "EmployeeImg": "usermale",
  "CurrentSalary": 97282,
  "Address": "Avda. Azteca 123",
  "Mail": "kathryn82@rpy.com"
},
{
  "EmployeeID": 10027,
  "Employees": "Michael Michael",
  "Designation": "Developer",
  "Location": "UK",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 4,
  "Software": 16,
  "EmployeeImg": "usermale",
  "CurrentSalary": 4184,
  "Address": "Rua do Paço, 67",
  "Mail": "michael58@jourrapide.com"
},
{
  "EmployeeID": 10028,
  "Employees": "Leverling Vinet",
  "Designation": "Project Lead",
  "Location": "Germany",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 57,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 38370,
  "Address": "59 rue de l'Abbaye",
  "Mail": "leverling102@sample.com"
},
{
  "EmployeeID": 10029,
  "Employees": "Rose Jack",
  "Designation": "Developer",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 46,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 84790,
  "Address": "Rua do Mercado, 12",
  "Mail": "rose108@jourrapide.com"
},
{
```

```
"EmployeeID": 10030,  
"Employees": "Vinet Van",  
"Designation": "Developer",  
"Location": "USA",  
"Status": "Active",  
"Trustworthiness": "Sufficient",  
"Rating": 0,  
"Software": 40,  
"EmployeeImg": "usermale",  
"CurrentSalary": 71005,  
"Address": "Gran Vía, 1",  
"Mail": "vinet90@jourrapide.com"  
}  
]
```

### Server Configuration

In Spreadsheet control, Excel import and export support processed in **server-side**, to use importing and exporting in your projects, it is required to create a server with any of the following web services.

- WebAPI
- WCF Service
- ASP.NET MVC Controller Action

The following code snippets shows server configuration using **WebAPI** service.

```
`csharp  
[Route("api/[controller]")]  
public class SpreadsheetController : Controller  
{  
    //To open excel file  
    [AcceptVerbs("Post")]  
    [HttpPost]  
    [EnableCors("AllowAllOrigins")]  
    [Route("Open")]  
    public IActionResult Open(IFormCollection openRequest)  
    {  
        OpenRequest open = new OpenRequest();  
        open.File = openRequest.Files[0];  
        return Content(Workbook.Open(open));  
    }  
    //To save as excel file  
    [AcceptVerbs("Post")]
```



```
[HttpPost]
[EnableCors("AllowAllOrigins")]
[Route("Save")]
public IActionResult Save([FromForm]SaveSettings saveSettings)
{
    return Workbook.Save(saveSettings);
}
}
```

### Server Dependencies

Open and save helper functions are shipped in the Syncfusion.EJ2.Spreadsheet package, which is available in Essential Studio and [nuget.org](https://nuget.org). Following list of dependencies required for Spreadsheet open and save operations.

- Syncfusion.EJ2
- Syncfusion.EJ2.Spreadsheet
- Syncfusion.Compression.Base
- Syncfusion.XlsIO.Base

And also refer [this](#) for more information.

### Note

You can refer to our [React Spreadsheet](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Spreadsheet example](#) to know how to present and manipulate data.

### See Also

- [Filtering](#)
- [Sorting](#)
- [Hyperlink](#)

## Worksheet in React Spreadsheet component

Worksheet is a collection of cells organized in the form of rows and columns that allows you to store, format, and manipulate the data.

### Add sheet

You can dynamically add or insert a sheet by one of the following ways,

- Click the **Add Sheet** button in the sheet tab. This will add a new empty sheet next to current active sheet.
- Right-click on the sheet tab, and then select **Insert** option from the context menu to insert a new empty sheet before the current active sheet.
- Using [insertSheet](#) method, you can insert one or more sheets at your desired index.

The following code example shows the insert sheet operation in spreadsheet.

### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { data } from '../datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  React.useEffect(() => {
    const insSheetModel = [{
      index: 1,
      name: 'Inserted Sheet',
      ranges: [{ dataSource: data }],
      columns: [
        { width: 150 },
        { width: 110 },
        { width: 110 },
        { width: 85 },
        { width: 85 },
        { width: 85 },
        { width: 85 },
        { width: 85 },
        { width: 85 },
      ]
    }];
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      // Applies style formatting to the active sheet before inserting
      // a new sheet
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center' }, 'A1:H1');
      spreadsheet.cellFormat({ textAlign: 'center' }, 'D2:H11');
      // inserting a new sheet with data at 1st index
      // You can also insert empty sheets by specifying the start and
      // end sheet index instead of sheet model
      spreadsheet.insertSheet(insSheetModel);
      // Applies style formatting for the inserted sheet
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center' }, 'Inserted Sheet!A1:H1');
      spreadsheet.cellFormat({ textAlign: 'center' }, 'Inserted Sheet!D2:H11');
    }
  }, []);
  return (
    <SpreadsheetComponent showRibbon={false} ref={spreadsheetRef}
    showFormulaBar={true}>
      <SheetsDirective>
        <SheetDirective name="Price Details">
          <RangesDirective>
            <RangeDirective dataSource={data}></RangeDirective>
          </RangesDirective>
          <ColumnsDirective>

```

```

        <ColumnDirective width={150}></ColumnDirective>
        <ColumnDirective width={110}></ColumnDirective>
        <ColumnDirective width={110}></ColumnDirective>
        <ColumnDirective width={85}></ColumnDirective>
        <ColumnDirective width={85}></ColumnDirective>
        <ColumnDirective width={85}></ColumnDirective>
        <ColumnDirective width={85}></ColumnDirective>
        <ColumnDirective width={85}></ColumnDirective>
      </ColumnsDirective>
    </SheetDirective>
  </SheetsDirective>
</SpreadsheetComponent>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, SheetModel } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  React.useEffect(() => {
    const insSheetModel: SheetModel[] = [{
      index: 1,
      name: 'Inserted Sheet',
      ranges: [{ dataSource: data }],
      columns: [{ width: 150 }, { width: 110 }, { width: 110 }, {
width: 85 }, { width: 85 }, { width: 85 }, { width: 85 }, { width: 85 }]
    }];
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      // Applies style formatting to the active sheet before inserting
a new sheet
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A1:H1');
      spreadsheet.cellFormat({ textAlign: 'center' }, 'D2:H11');
      // inserting a new sheet with data at 1st index
      // You can also insert empty sheets by specifying the start and
end sheet index instead of sheet model
      spreadsheet.insertSheet(insSheetModel);
      // Applies style formatting for the inserted sheet
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'Inserted Sheet!A1:H1');
      spreadsheet.cellFormat({ textAlign: 'center' }, 'Inserted
Sheet!D2:H11');
    }
  });
}

```

```

    }, []);
    return (
      <SpreadsheetComponent showRibbon={false} ref={spreadsheetRef}
        showformulaBar={true}>
        <SheetsDirective>
          <SheetDirective name="Price Details">
            <RangesDirective>
              <RangeDirective dataSource={data}></RangeDirective>
            </RangesDirective>
            <ColumnsDirective>
              <ColumnDirective width={150}></ColumnDirective>
              <ColumnDirective width={110}></ColumnDirective>
              <ColumnDirective width={110}></ColumnDirective>
              <ColumnDirective width={85}></ColumnDirective>
              <ColumnDirective width={85}></ColumnDirective>
              <ColumnDirective width={85}></ColumnDirective>
              <ColumnDirective width={85}></ColumnDirective>
              <ColumnDirective width={85}></ColumnDirective>
            </ColumnsDirective>
          </SheetDirective>
        </SheetsDirective>
      </SpreadsheetComponent>
    );
  };
  export default App;
  const root = createRoot(document.getElementById('root'));
  root.render(<App />);
  {% endraw %}

```

### DATASOURCE.JSX

```

/**
 * Insert sheet data source
 */
export let data = [
  { 'Item Name': 'Casual Shoes', 'Date': '02/14/2019', 'Time': '11:34:32 AM', 'Quantity': 10, 'Price': 20, 'Amount': '=D2*E2', 'Discount': 1, 'Profit': 10 },
  { 'Item Name': 'Sports Shoes', 'Date': '06/11/2019', 'Time': '05:56:32 AM', 'Quantity': 20, 'Price': 30, 'Amount': '=D3*E3', 'Discount': 5, 'Profit': 50 },
  { 'Item Name': 'Formal Shoes', 'Date': '07/27/2019', 'Time': '03:32:44 AM', 'Quantity': 20, 'Price': 15, 'Amount': '=D4*E4', 'Discount': 7, 'Profit': 27 },
  { 'Item Name': 'Sandals & Floaters', 'Date': '11/21/2019', 'Time': '06:23:54 AM', 'Quantity': 15, 'Price': 20, 'Amount': '=D5*E5', 'Discount': 11, 'Profit': 67 },
  { 'Item Name': 'Flip- Flops & Slippers', 'Date': '06/23/2019', 'Time': '12:43:59 AM', 'Quantity': 30, 'Price': 10, 'Amount': '=D6*E6', 'Discount': 10, 'Profit': 70 },
  { 'Item Name': 'Sneakers', 'Date': '07/22/2019', 'Time': '10:55:53 AM', 'Quantity': 40, 'Price': 20, 'Amount': '=D7*E7', 'Discount': 13, 'Profit': 66 },
  { 'Item Name': 'Running Shoes', 'Date': '02/04/2019', 'Time': '03:44:34 AM', 'Quantity': 20, 'Price': 10, 'Amount': '=D8*E8', 'Discount': 3, 'Profit': 14 },

```

```

    { 'Item Name': 'Loafers', 'Date': '11/30/2019', 'Time': '03:12:52 AM',
      'Quantity': 31, 'Price': 10, 'Amount': '=D9*E9', 'Discount': 6, 'Profit': 29
    },
    { 'Item Name': 'Cricket Shoes', 'Date': '07/09/2019', 'Time': '11:32:14
AM', 'Quantity': 41, 'Price': 30, 'Amount': '=D10*E10', 'Discount': 12,
      'Profit': 166 },
    { 'Item Name': 'T-Shirts', 'Date': '10/31/2019', 'Time': '12:01:44 AM',
      'Quantity': 50, 'Price': 10, 'Amount': '=D11*E11', 'Discount': 9, 'Profit':
55 }
  ];

```

### DATASOURCE.TSX

```

/**
 * Insert sheet data source
 */
export let data: Object[] = [
  { 'Item Name': 'Casual Shoes', 'Date': '02/14/2019', 'Time': '11:34:32
AM', 'Quantity': 10, 'Price': 20, 'Amount': '=D2*E2', 'Discount': 1,
    'Profit': 10 },
  { 'Item Name': 'Sports Shoes', 'Date': '06/11/2019', 'Time': '05:56:32
AM', 'Quantity': 20, 'Price': 30, 'Amount': '=D3*E3', 'Discount': 5,
    'Profit': 50 },
  { 'Item Name': 'Formal Shoes', 'Date': '07/27/2019', 'Time': '03:32:44
AM', 'Quantity': 20, 'Price': 15, 'Amount': '=D4*E4', 'Discount': 7,
    'Profit': 27 },
  { 'Item Name': 'Sandals & Floaters', 'Date': '11/21/2019', 'Time':
'06:23:54 AM', 'Quantity': 15, 'Price': 20, 'Amount': '=D5*E5', 'Discount':
11, 'Profit': 67 },
  { 'Item Name': 'Flip- Flops & Slippers', 'Date': '06/23/2019', 'Time':
'12:43:59 AM', 'Quantity': 30, 'Price': 10, 'Amount': '=D6*E6', 'Discount':
10, 'Profit': 70 },
  { 'Item Name': 'Sneakers', 'Date': '07/22/2019', 'Time': '10:55:53 AM',
    'Quantity': 40, 'Price': 20, 'Amount': '=D7*E7', 'Discount': 13, 'Profit':
66 },
  { 'Item Name': 'Running Shoes', 'Date': '02/04/2019', 'Time': '03:44:34
AM', 'Quantity': 20, 'Price': 10, 'Amount': '=D8*E8', 'Discount': 3,
    'Profit': 14 },
  { 'Item Name': 'Loafers', 'Date': '11/30/2019', 'Time': '03:12:52 AM',
    'Quantity': 31, 'Price': 10, 'Amount': '=D9*E9', 'Discount': 6, 'Profit': 29
  },
  { 'Item Name': 'Cricket Shoes', 'Date': '07/09/2019', 'Time': '11:32:14
AM', 'Quantity': 41, 'Price': 30, 'Amount': '=D10*E10', 'Discount': 12,
    'Profit': 166 },
  { 'Item Name': 'T-Shirts', 'Date': '10/31/2019', 'Time': '12:01:44 AM',
    'Quantity': 50, 'Price': 10, 'Amount': '=D11*E11', 'Discount': 9, 'Profit':
55 }
];

```

### Delete sheet

The Spreadsheet has support for removing an existing worksheet. You can dynamically delete the existing sheet by the following way,

- Right-click on the sheet tab, and then select **Delete** option from context menu.

- Using [delete](#) method to delete the sheets.

### Rename sheet

You can dynamically rename an existing worksheet in the following way,

- Right-click on the sheet tab, and then select **Rename** option from the context menu.

### Headers

By default, the row and column headers are visible in worksheets. You can dynamically show or hide worksheet headers by using one of the following ways,

- Switch to **View** tab, and then select **Hide Headers** option to hide both the row and column headers.
- Set **showHeaders** property in **sheets** as **true** or **false** to show or hide the headers at initial load. By default, the **showHeaders** property is enabled in each worksheet.

### Gridlines

Gridlines act as a border like appearance of cells. They are used to distinguish cells on the worksheet. You can dynamically show or hide gridlines by using one of the following ways,

- Switch to **View** tab, and then select **Hide Gridlines** option to hide the gridlines in worksheet.
- Set **showGridLines** property in **sheets** as **true** or **false** to show or hide the gridlines at initial load. By default, the **showGridLines** property is enabled in each worksheet.

The following code example shows the headers and gridlines operation in spreadsheet.

#### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A1:H1');
      spreadsheet.cellFormat({ textAlign: 'center' }, 'D2:H11');
      // The gridlines have been removed to set border for the range
of cells
      spreadsheet.setBorder({ border: '1px solid #e0e0e0' },
'A1:H11');
    }
  }, []);
  return (
```

```

        <SpreadsheetComponent ref={spreadsheetRef} showFormulaBar={false}>
          <SheetsDirective>
            <SheetDirective name='Price Details' showGridLines={false}
showHeaders={false}>
              <RangesDirective>
                <RangeDirective dataSource={data}></RangeDirective>
              </RangesDirective>
              <ColumnsDirective>
                <ColumnDirective width={150}></ColumnDirective>
                <ColumnDirective width={110}></ColumnDirective>
                <ColumnDirective width={110}></ColumnDirective>
                <ColumnDirective width={85}></ColumnDirective>
                <ColumnDirective width={85}></ColumnDirective>
                <ColumnDirective width={85}></ColumnDirective>
                <ColumnDirective width={85}></ColumnDirective>
              </ColumnsDirective>
            </SheetDirective>
          </SheetsDirective>
        </SpreadsheetComponent>
      );
    };
    export default App;
    const root = createRoot(document.getElementById('root'));
    root.render(<App />);
    {% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A1:H1');
      spreadsheet.cellFormat({ textAlign: 'center' }, 'D2:H11');
      // The gridlines have been removed to set border for the range
of cells
      spreadsheet.setBorder({ border: '1px solid #e0e0e0' },
'A1:H11');
    }
  }, []);
  return (
    <SpreadsheetComponent ref={spreadsheetRef} showFormulaBar={false}>
      <SheetsDirective>
        <SheetDirective name='Price Details' showGridLines={false}
showHeaders={false}>

```

```

        <RangesDirective>
          <RangeDirective dataSource={data}></RangeDirective>
        </RangesDirective>
        <ColumnsDirective>
          <ColumnDirective width={150}></ColumnDirective>
          <ColumnDirective width={110}></ColumnDirective>
          <ColumnDirective width={110}></ColumnDirective>
          <ColumnDirective width={85}></ColumnDirective>
          <ColumnDirective width={85}></ColumnDirective>
          <ColumnDirective width={85}></ColumnDirective>
          <ColumnDirective width={85}></ColumnDirective>
          <ColumnDirective width={85}></ColumnDirective>
        </ColumnsDirective>
      </SheetDirective>
    </SheetsDirective>
  </SpreadsheetComponent>
);
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

### DATASOURCE.JSX

```

/**
 * Ribbon customization data source
 */
export let data = [
  { 'Item Name': 'Casual Shoes', 'Date': '02/14/2019', 'Time': '11:34:32 AM', 'Quantity': 10, 'Price': 20, 'Amount': '=D2*E2', 'Discount': 1, 'Profit': 10 },
  { 'Item Name': 'Sports Shoes', 'Date': '06/11/2019', 'Time': '05:56:32 AM', 'Quantity': 20, 'Price': 30, 'Amount': '=D3*E3', 'Discount': 5, 'Profit': 50 },
  { 'Item Name': 'Formal Shoes', 'Date': '07/27/2019', 'Time': '03:32:44 AM', 'Quantity': 20, 'Price': 15, 'Amount': '=D4*E4', 'Discount': 7, 'Profit': 27 },
  { 'Item Name': 'Sandals & Floaters', 'Date': '11/21/2019', 'Time': '06:23:54 AM', 'Quantity': 15, 'Price': 20, 'Amount': '=D5*E5', 'Discount': 11, 'Profit': 67 },
  { 'Item Name': 'Flip- Flops & Slippers', 'Date': '06/23/2019', 'Time': '12:43:59 AM', 'Quantity': 30, 'Price': 10, 'Amount': '=D6*E6', 'Discount': 10, 'Profit': 70 },
  { 'Item Name': 'Sneakers', 'Date': '07/22/2019', 'Time': '10:55:53 AM', 'Quantity': 40, 'Price': 20, 'Amount': '=D7*E7', 'Discount': 13, 'Profit': 66 },
  { 'Item Name': 'Running Shoes', 'Date': '02/04/2019', 'Time': '03:44:34 AM', 'Quantity': 20, 'Price': 10, 'Amount': '=D8*E8', 'Discount': 3, 'Profit': 14 },
  { 'Item Name': 'Loafers', 'Date': '11/30/2019', 'Time': '03:12:52 AM', 'Quantity': 31, 'Price': 10, 'Amount': '=D9*E9', 'Discount': 6, 'Profit': 29 },
  { 'Item Name': 'Cricket Shoes', 'Date': '07/09/2019', 'Time': '11:32:14 AM', 'Quantity': 41, 'Price': 30, 'Amount': '=D10*E10', 'Discount': 12, 'Profit': 166 },
];

```



```
{ 'Item Name': 'T-Shirts', 'Date': '10/31/2019', 'Time': '12:01:44 AM',
'Quantity': 50, 'Price': 10, 'Amount': '=D11*E11', 'Discount': 9, 'Profit':
55 }
];
```

**DATASOURCE.TSX**

```
/**
 * Ribbon customization data source
 */
export let data: Object[] = [
  { 'Item Name': 'Casual Shoes', 'Date': '02/14/2019', 'Time': '11:34:32
AM', 'Quantity': 10, 'Price': 20, 'Amount': '=D2*E2', 'Discount': 1,
'Profit': 10 },
  { 'Item Name': 'Sports Shoes', 'Date': '06/11/2019', 'Time': '05:56:32
AM', 'Quantity': 20, 'Price': 30, 'Amount': '=D3*E3', 'Discount': 5,
'Profit': 50 },
  { 'Item Name': 'Formal Shoes', 'Date': '07/27/2019', 'Time': '03:32:44
AM', 'Quantity': 20, 'Price': 15, 'Amount': '=D4*E4', 'Discount': 7,
'Profit': 27 },
  { 'Item Name': 'Sandals & Floaters', 'Date': '11/21/2019', 'Time':
'06:23:54 AM', 'Quantity': 15, 'Price': 20, 'Amount': '=D5*E5', 'Discount':
11, 'Profit': 67 },
  { 'Item Name': 'Flip- Flops & Slippers', 'Date': '06/23/2019', 'Time':
'12:43:59 AM', 'Quantity': 30, 'Price': 10, 'Amount': '=D6*E6', 'Discount':
10, 'Profit': 70 },
  { 'Item Name': 'Sneakers', 'Date': '07/22/2019', 'Time': '10:55:53 AM',
'Quantity': 40, 'Price': 20, 'Amount': '=D7*E7', 'Discount': 13, 'Profit':
66 },
  { 'Item Name': 'Running Shoes', 'Date': '02/04/2019', 'Time': '03:44:34
AM', 'Quantity': 20, 'Price': 10, 'Amount': '=D8*E8', 'Discount': 3,
'Profit': 14 },
  { 'Item Name': 'Loafers', 'Date': '11/30/2019', 'Time': '03:12:52 AM',
'Quantity': 31, 'Price': 10, 'Amount': '=D9*E9', 'Discount': 6, 'Profit': 29
},
  { 'Item Name': 'Cricket Shoes', 'Date': '07/09/2019', 'Time': '11:32:14
AM', 'Quantity': 41, 'Price': 30, 'Amount': '=D10*E10', 'Discount': 12,
'Profit': 166 },
  { 'Item Name': 'T-Shirts', 'Date': '10/31/2019', 'Time': '12:01:44 AM',
'Quantity': 50, 'Price': 10, 'Amount': '=D11*E11', 'Discount': 9, 'Profit':
55 }
];
```

**Sheet visibility**

Hiding a worksheet can help prevent unauthorized or accidental changes to your file.

There are three visibility state as like Microsoft Excel,

| State | Description |

|-----|-----|

| **Visible** | You can see the worksheet once the component is loaded. |

| **Hidden** | This worksheet is not visible, but you can unhide by selecting the sheet from **List All Sheets** dropdown menu. |

| **VeryHidden** | This worksheet is not visible and cannot be unhidden. Changing the state property to **Visible** is the only way to view this sheet. |

The following code example shows the three types of sheet visibility state.

### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
    const spreadsheetRef = React.useRef(null);
    React.useEffect(() => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            // Applies style formatting to active visible sheet
            spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A1:H1');
            spreadsheet.cellFormat({ textAlign: 'center' }, 'D2:H11');
            // Applies style formatting to active hidden sheet
            spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'Hidden Sheet!A1:H1');
            spreadsheet.cellFormat({ textAlign: 'center' }, 'Hidden
Sheet!D2:H11');
        }
    }, []);
    return (
        <SpreadsheetComponent showFormulaBar={false} ref={spreadsheetRef}
openUrl={"https://services.syncfusion.com/react/production/api/spreadsheet/o
pen"}
saveUrl={"https://services.syncfusion.com/react/production/api/spreadsheet/s
ave"}>
            <SheetsDirective>
                <SheetDirective name='Visible Sheet' state={'Visible'}>
                    <RangesDirective>
                        <RangeDirective dataSource={data}></RangeDirective>
                    </RangesDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={150}></ColumnDirective>
                        <ColumnDirective width={110}></ColumnDirective>
                        <ColumnDirective width={110}></ColumnDirective>
                        <ColumnDirective width={85}></ColumnDirective>
                        <ColumnDirective width={85}></ColumnDirective>
                        <ColumnDirective width={85}></ColumnDirective>
                        <ColumnDirective width={85}></ColumnDirective>
                        <ColumnDirective width={85}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
                <SheetDirective name='Very Hidden Sheet'
state={'VeryHidden'}>
                    <RangesDirective>
```

```

        <RangeDirective dataSource={data}></RangeDirective>
      </RangesDirective>
      <ColumnsDirective>
        <ColumnDirective width={150}></ColumnDirective>
        <ColumnDirective width={110}></ColumnDirective>
        <ColumnDirective width={110}></ColumnDirective>
        <ColumnDirective width={85}></ColumnDirective>
        <ColumnDirective width={85}></ColumnDirective>
        <ColumnDirective width={85}></ColumnDirective>
        <ColumnDirective width={85}></ColumnDirective>
        <ColumnDirective width={85}></ColumnDirective>
      </ColumnsDirective>
    </SheetDirective>
    <SheetDirective name='Hidden Sheet' state={'Hidden'}>
      <RangesDirective>
        <RangeDirective dataSource={data}></RangeDirective>
      </RangesDirective>
      <ColumnsDirective>
        <ColumnDirective width={150}></ColumnDirective>
        <ColumnDirective width={110}></ColumnDirective>
        <ColumnDirective width={110}></ColumnDirective>
        <ColumnDirective width={85}></ColumnDirective>
        <ColumnDirective width={85}></ColumnDirective>
        <ColumnDirective width={85}></ColumnDirective>
        <ColumnDirective width={85}></ColumnDirective>
        <ColumnDirective width={85}></ColumnDirective>
      </ColumnsDirective>
    </SheetDirective>
  </SheetsDirective>
</SpreadsheetComponent>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      // Applies style formatting to active visible sheet
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A1:H1');
      spreadsheet.cellFormat({ textAlign: 'center' }, 'D2:H11');
    }
  });
}

```

```

        // Applies style formatting to active hidden sheet
        spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
    }, 'Hidden Sheet!A1:H1');
        spreadsheet.cellFormat({ textAlign: 'center' }, 'Hidden
    Sheet!D2:H11');
    }
    }, []);
    return (
        <SpreadsheetComponent showFormulaBar={false} ref={spreadsheetRef}
    openUrl={"https://services.syncfusion.com/react/production/api/spreadsheet/o
    pen"}

    saveUrl={"https://services.syncfusion.com/react/production/api/spreadsheet/s
    ave"}>
        <SheetsDirective>
            <SheetDirective name='Visible Sheet' state={'Visible'}>
                <RangesDirective>
                    <RangeDirective dataSource={data}></RangeDirective>
                </RangesDirective>
                <ColumnsDirective>
                    <ColumnDirective width={150}></ColumnDirective>
                    <ColumnDirective width={110}></ColumnDirective>
                    <ColumnDirective width={110}></ColumnDirective>
                    <ColumnDirective width={85}></ColumnDirective>
                    <ColumnDirective width={85}></ColumnDirective>
                    <ColumnDirective width={85}></ColumnDirective>
                    <ColumnDirective width={85}></ColumnDirective>
                    <ColumnDirective width={85}></ColumnDirective>
                </ColumnsDirective>
            </SheetDirective>
            <SheetDirective name='Very Hidden Sheet'
    state={'VeryHidden'}>
                <RangesDirective>
                    <RangeDirective dataSource={data}></RangeDirective>
                </RangesDirective>
                <ColumnsDirective>
                    <ColumnDirective width={150}></ColumnDirective>
                    <ColumnDirective width={110}></ColumnDirective>
                    <ColumnDirective width={110}></ColumnDirective>
                    <ColumnDirective width={85}></ColumnDirective>
                    <ColumnDirective width={85}></ColumnDirective>
                    <ColumnDirective width={85}></ColumnDirective>
                    <ColumnDirective width={85}></ColumnDirective>
                    <ColumnDirective width={85}></ColumnDirective>
                </ColumnsDirective>
            </SheetDirective>
            <SheetDirective name='Hidden Sheet' state={'Hidden'}>
                <RangesDirective>
                    <RangeDirective dataSource={data}></RangeDirective>
                </RangesDirective>
                <ColumnsDirective>
                    <ColumnDirective width={150}></ColumnDirective>
                    <ColumnDirective width={110}></ColumnDirective>
                    <ColumnDirective width={110}></ColumnDirective>
                    <ColumnDirective width={85}></ColumnDirective>
                    <ColumnDirective width={85}></ColumnDirective>
                    <ColumnDirective width={85}></ColumnDirective>
                </ColumnsDirective>
            </SheetDirective>
        </SheetsDirective>
    </SpreadsheetComponent>
    </pre>

```

```

        <ColumnDirective width={85}></ColumnDirective>
        <ColumnDirective width={85}></ColumnDirective>
    </ColumnsDirective>
    </SheetDirective>
  </SheetsDirective>
</SpreadsheetComponent>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### DATASOURCE.JSX

```

/**
 * Hide/show sheet data source
 */
export let data = [
  { 'Item Name': 'Casual Shoes', 'Date': '02/14/2019', 'Time': '11:34:32 AM', 'Quantity': 10, 'Price': 20, 'Amount': '=D2*E2', 'Discount': 1, 'Profit': 10 },
  { 'Item Name': 'Sports Shoes', 'Date': '06/11/2019', 'Time': '05:56:32 AM', 'Quantity': 20, 'Price': 30, 'Amount': '=D3*E3', 'Discount': 5, 'Profit': 50 },
  { 'Item Name': 'Formal Shoes', 'Date': '07/27/2019', 'Time': '03:32:44 AM', 'Quantity': 20, 'Price': 15, 'Amount': '=D4*E4', 'Discount': 7, 'Profit': 27 },
  { 'Item Name': 'Sandals & Floaters', 'Date': '11/21/2019', 'Time': '06:23:54 AM', 'Quantity': 15, 'Price': 20, 'Amount': '=D5*E5', 'Discount': 11, 'Profit': 67 },
  { 'Item Name': 'Flip- Flops & Slippers', 'Date': '06/23/2019', 'Time': '12:43:59 AM', 'Quantity': 30, 'Price': 10, 'Amount': '=D6*E6', 'Discount': 10, 'Profit': 70 },
  { 'Item Name': 'Sneakers', 'Date': '07/22/2019', 'Time': '10:55:53 AM', 'Quantity': 40, 'Price': 20, 'Amount': '=D7*E7', 'Discount': 13, 'Profit': 66 },
  { 'Item Name': 'Running Shoes', 'Date': '02/04/2019', 'Time': '03:44:34 AM', 'Quantity': 20, 'Price': 10, 'Amount': '=D8*E8', 'Discount': 3, 'Profit': 14 },
  { 'Item Name': 'Loafers', 'Date': '11/30/2019', 'Time': '03:12:52 AM', 'Quantity': 31, 'Price': 10, 'Amount': '=D9*E9', 'Discount': 6, 'Profit': 29 },
  { 'Item Name': 'Cricket Shoes', 'Date': '07/09/2019', 'Time': '11:32:14 AM', 'Quantity': 41, 'Price': 30, 'Amount': '=D10*E10', 'Discount': 12, 'Profit': 166 },
  { 'Item Name': 'T-Shirts', 'Date': '10/31/2019', 'Time': '12:01:44 AM', 'Quantity': 50, 'Price': 10, 'Amount': '=D11*E11', 'Discount': 9, 'Profit': 55 }
];

```

### DATASOURCE.TSX

```

/**
 * Hide/show sheet data source
 */

```

```
export let data: Object[] = [
  { 'Item Name': 'Casual Shoes', 'Date': '02/14/2019', 'Time': '11:34:32 AM', 'Quantity': 10, 'Price': 20, 'Amount': '=D2*E2', 'Discount': 1, 'Profit': 10 },
  { 'Item Name': 'Sports Shoes', 'Date': '06/11/2019', 'Time': '05:56:32 AM', 'Quantity': 20, 'Price': 30, 'Amount': '=D3*E3', 'Discount': 5, 'Profit': 50 },
  { 'Item Name': 'Formal Shoes', 'Date': '07/27/2019', 'Time': '03:32:44 AM', 'Quantity': 20, 'Price': 15, 'Amount': '=D4*E4', 'Discount': 7, 'Profit': 27 },
  { 'Item Name': 'Sandals & Floaters', 'Date': '11/21/2019', 'Time': '06:23:54 AM', 'Quantity': 15, 'Price': 20, 'Amount': '=D5*E5', 'Discount': 11, 'Profit': 67 },
  { 'Item Name': 'Flip- Flops & Slippers', 'Date': '06/23/2019', 'Time': '12:43:59 AM', 'Quantity': 30, 'Price': 10, 'Amount': '=D6*E6', 'Discount': 10, 'Profit': 70 },
  { 'Item Name': 'Sneakers', 'Date': '07/22/2019', 'Time': '10:55:53 AM', 'Quantity': 40, 'Price': 20, 'Amount': '=D7*E7', 'Discount': 13, 'Profit': 66 },
  { 'Item Name': 'Running Shoes', 'Date': '02/04/2019', 'Time': '03:44:34 AM', 'Quantity': 20, 'Price': 10, 'Amount': '=D8*E8', 'Discount': 3, 'Profit': 14 },
  { 'Item Name': 'Loafers', 'Date': '11/30/2019', 'Time': '03:12:52 AM', 'Quantity': 31, 'Price': 10, 'Amount': '=D9*E9', 'Discount': 6, 'Profit': 29 },
  { 'Item Name': 'Cricket Shoes', 'Date': '07/09/2019', 'Time': '11:32:14 AM', 'Quantity': 41, 'Price': 30, 'Amount': '=D10*E10', 'Discount': 12, 'Profit': 166 },
  { 'Item Name': 'T-Shirts', 'Date': '10/31/2019', 'Time': '12:01:44 AM', 'Quantity': 50, 'Price': 10, 'Amount': '=D11*E11', 'Discount': 9, 'Profit': 55 }
];
```

### Note

You can refer to our [React Spreadsheet](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Spreadsheet example](#) to know how to present and manipulate data.

### See Also

- [Sheet protection](#)
- [Rows and columns](#)
- [Cell range](#)
- [Formatting](#)[Link to the Video](#)

### Cell range in React Spreadsheet component

A group of cells in a sheet is known as cell range.

To get start quickly with Cell Range, you can check on this video:

### Wrap text

Wrap text allows you to display large content as multiple lines in a single cell. By default, the wrap text support is enabled. Use the [allowWrap](#) property to enable or disable the wrap text support in spreadsheet.

Wrap text can be applied or removed to a cell or range of cells in the following ways,

- Using the `wrap` property in `cell`, you can enable or disable wrap text to a cell at initial load.
- Select or deselect wrap button from ribbon toolbar to apply or remove the wrap text to the selected range.
- Using the `wrap` method, you can apply or remove the wrap text once the component is loaded.

The following code example shows the wrap text functionality in spreadsheet.

### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective, CellDirective, CellsDirective
} from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A1:H1');
      spreadsheet.cellFormat({ verticalAlign: 'middle' }, 'A1:H5');
      spreadsheet.cellFormat({ textAlign: 'center' }, 'A2:B5');
      spreadsheet.cellFormat({ textAlign: 'center' }, 'D2:D5');
      // To wrap the cells from E2 to E5 range
      spreadsheet.wrap('E2:E5');
      // To unwrap the H3 cell
      spreadsheet.wrap('H3', false);
    }
  }, []);
  return (
    <div>
      <SpreadsheetComponent ref={spreadsheetRef}
showFormulaBar={false} >
        <SheetsDirective>
          <SheetDirective name={"Movie List"}>
            <RowsDirective>
              <RowDirective height={30}>
                </RowDirective>
                <RowDirective>
                  <CellsDirective>
                    <CellDirective index={7}
wrap={true}></CellDirective>
                  </CellsDirective>
```

```

        </RowDirective>
        <RowDirective>
            <CellsDirective>
                <CellDirective index={7}>
wrap={true}></CellDirective>
            </CellsDirective>
        </RowDirective>
        <RowDirective>
            <CellsDirective>
                <CellDirective index={7}>
wrap={true}></CellDirective>
            </CellsDirective>
        </RowDirective>
        <RowDirective>
            <CellsDirective>
                <CellDirective index={7}>
wrap={true}></CellDirective>
            </CellsDirective>
        </RowDirective>
    </RowsDirective>
    <RangesDirective>
        <RangeDirective
dataSource={data}></RangeDirective>
    </RangesDirective>
    <ColumnsDirective>
        <ColumnDirective width={100}>
index={1}></ColumnDirective>
        <ColumnDirective width={140}></ColumnDirective>
        <ColumnDirective width={90}></ColumnDirective>
        <ColumnDirective width={150}></ColumnDirective>
        <ColumnDirective width={120}></ColumnDirective>
        <ColumnDirective width={90}></ColumnDirective>
        <ColumnDirective width={180}></ColumnDirective>
    </ColumnsDirective>
</SheetDirective>
</SheetsDirective>
</SpreadsheetComponent>
</div>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective, CellDirective, CellsDirective
} from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';

```



```

function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A1:H1');
      spreadsheet.cellFormat({ verticalAlign: 'middle' }, 'A1:H5');
      spreadsheet.cellFormat({ textAlign: 'center' }, 'A2:B5');
      spreadsheet.cellFormat({ textAlign: 'center' }, 'D2:D5');
      // To wrap the cells from E2 to E5 range
      spreadsheet.wrap('E2:E5');
      // To unwrap the H3 cell
      spreadsheet.wrap('H3', false);
    }
  }, []);
  return (
    <div>
      <SpreadsheetComponent ref={spreadsheetRef}
showFormulaBar={false} >
        <SheetsDirective>
          <SheetDirective name={"Movie List"}>
            <RowsDirective>
              <RowDirective height={30}>
                </RowDirective>
                <RowDirective>
                  <CellsDirective>
                    <CellDirective index={7}
wrap={true}></CellDirective>
                  </CellsDirective>
                </RowDirective>
                <RowDirective>
                  <CellsDirective>
                    <CellDirective index={7}
wrap={true}></CellDirective>
                  </CellsDirective>
                </RowDirective>
                <RowDirective>
                  <CellsDirective>
                    <CellDirective index={7}
wrap={true}></CellDirective>
                  </CellsDirective>
                </RowDirective>
                <RowDirective>
                  <CellsDirective>
                    <CellDirective index={7}
wrap={true}></CellDirective>
                  </CellsDirective>
                </RowDirective>
                <RangesDirective>
                  <RangeDirective
dataSource={data}></RangeDirective>
                </RangesDirective>
                <ColumnsDirective>
                  <ColumnDirective width={100}
index={1}></ColumnDirective>
            </SheetDirective>
          </RowsDirective>
        </RangesDirective>
      </SpreadsheetComponent>
    </div>
  );
}

```

```

        <ColumnDirective width={140}></ColumnDirective>
        <ColumnDirective width={90}></ColumnDirective>
        <ColumnDirective width={150}></ColumnDirective>
        <ColumnDirective width={120}></ColumnDirective>
        <ColumnDirective width={90}></ColumnDirective>
        <ColumnDirective width={180}></ColumnDirective>
      </ColumnsDirective>
    </SheetDirective>
  </SheetsDirective>
</SpreadsheetComponent>
</div>
);
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

### DATASOURCE.JSX

```

/**
 * Wrap text data source
 */
export let data = [
  {
    'No': '1',
    'Released on': 1994,
    'Title': 'Forrest Gump',
    'Rating': '5 Stars',
    'Casts': 'Tom Hanks, Robin Wright, Gary Sinise',
    'Directed By': 'Robert Zemeckis',
    'Genre': 'Drama',
    'Comments': 'Based on the 1986 novel of the same name by Winston
Groom'
  },
  {
    'No': '2',
    'Released on': 1946,
    'Title': 'It's a Wonderful Life',
    'Rating': '2 Stars',
    'Casts': 'James Stewart, Donna Reed, Lionel Barrymore',
    'Directed By': 'Frank Capra',
    'Genre': 'Drama',
    'Comments': 'Colorized version'
  },
  {
    'No': '3',
    'Released on': 1988,
    'Title': 'Big',
    'Rating': '4 Stars',
    'Casts': 'Tom Hanks, Elizabeth Perkins, Robert Loggia',
    'Directed By': 'Penny Marshall',
    'Genre': 'Comedy',
    'Comments': 'A thirteen-year-old boy wishes to be big, and his wish
comes true.'
  },

```

```

    {
      'No': '4',
      'Released on': 1954,
      'Title': 'Rear Window',
      'Rating': '4 Stars',
      'Casts': 'James Stewart, Grace Kelly, Wendell Corey',
      'Directed By': 'Alfred Hitchcock',
      'Genre': 'Suspense',
      'Comments': 'Truly suspenseful and masterfully crafted'
    }
  ];

```

### DATASOURCE.TSX

```

/**
 * Wrap text data source
 */
export let data: Object[] = [
  {
    'No': '1',
    'Released on': 1994,
    'Title': 'Forrest Gump',
    'Rating': '5 Stars',
    'Casts': 'Tom Hanks, Robin Wright, Gary Sinise',
    'Directed By': 'Robert Zemeckis',
    'Genre': 'Drama',
    'Comments': 'Based on the 1986 novel of the same name by Winston
Groom'
  },
  {
    'No': '2',
    'Released on': 1946,
    'Title': 'It's a Wonderful Life',
    'Rating': '2 Stars',
    'Casts': 'James Stewart, Donna Reed, Lionel Barrymore',
    'Directed By': 'Frank Capra',
    'Genre': 'Drama',
    'Comments': 'Colorized version'
  },
  {
    'No': '3',
    'Released on': 1988,
    'Title': 'Big',
    'Rating': '4 Stars',
    'Casts': 'Tom Hanks, Elizabeth Perkins, Robert Loggia',
    'Directed By': 'Penny Marshall',
    'Genre': 'Comedy',
    'Comments': 'A thirteen-year-old boy wishes to be big, and his wish
comes true.'
  },
  {
    'No': '4',
    'Released on': 1954,
    'Title': 'Rear Window',
    'Rating': '4 Stars',
    'Casts': 'James Stewart, Grace Kelly, Wendell Corey',

```

```

    'Directed By': 'Alfred Hitchcock',
    'Genre': 'Suspense',
    'Comments': 'Truly suspenseful and masterfully crafted'
  }
];

```

### Limitations of Wrap text

The following features have some limitations in wrap text:

- Sorting with wrap text applied data.
- Merge with wrap text

### Merge cells

Merge cells allows users to span two or more cells in the same row or column into a single cell. When cells with multiple values are merged, top-left most cell data will be the data for the merged cell. By default, the merge cells option is enabled. Use [allowMerge](#) property to enable or disable the merge cells option in spreadsheet.

You can merge the range of cells in the following ways,

- Set the `rowSpan` and `colSpan` property in `cell` to merge the number of cells at initial load.
- Select the range of cells and apply merge by selecting the desired option from ribbon toolbar.
- Use [merge](#) method to merge the range of cells, once the component is loaded.

The available merge options in spreadsheet are,

Type	Action
----- -----	
Merge All	Combines all the cells in a range in to a single cell (default).
Merge Horizontally	Combines cells in a range as row-wise.
Merge Vertically	Combines cells in a range as column-wise.
UnMerge	Splits the merged cells into multiple cells.

The following code example shows the merge cells operation in spreadsheet.

### APP.JSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective, CellsDirective, CellDirective
} from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;

```

```

    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
    }, 'A1:S1');
      spreadsheet.numberFormat('h:mm AM/PM', 'C1:S1');
      spreadsheet.cellFormat({ verticalAlign: 'middle' }, 'A1:S11');
      // Merging the `K4:M4` cells using method
      spreadsheet.merge('K4:M4');
      // Merging the 5th and 6th row cells across 11th, 12th and 13th
column
      spreadsheet.merge('K5:M6', 'Vertically');
      // Merging the 18th and 19th column cells across 2nd, 3rd and
4th row
      spreadsheet.merge('N4:O6', 'Horizontally');
    }
  }, []);
  return (
    <div>
      <SpreadsheetComponent ref={spreadsheetRef}
showFormulaBar={false}>
        <SheetsDirective>
          <SheetDirective name={"Merge Cells"}>
            <RowsDirective>
              <RowDirective height={35}></RowDirective>
              <RowDirective height={35}>
                <CellsDirective>
                  <CellDirective index={1}
rowSpan={2}></CellDirective>
                  <CellDirective
colSpan={2}></CellDirective>
                  <CellDirective index={6}
colSpan={3}></CellDirective>
                  <CellDirective index={10} rowSpan={2}
colSpan={3}></CellDirective>
                  <CellDirective index={13}
colSpan={2}></CellDirective>
                  <CellDirective index={17}
colSpan={2}></CellDirective>
                </CellsDirective>
              </RowDirective>
              <RowDirective height={35}>
                <CellsDirective>
                  <CellDirective index={3}
colSpan={3}></CellDirective>
                  <CellDirective index={6}
colSpan={4}></CellDirective>
                  <CellDirective index={13}
colSpan={3}></CellDirective>
                  <CellDirective index={17}
colSpan={2}></CellDirective>
                </CellsDirective>
              </RowDirective>
              <RowDirective height={35}>
                <CellsDirective>
                  <CellDirective index={2}
colSpan={3}></CellDirective>
                  <CellDirective index={5}
colSpan={2}></CellDirective>

```

```

                                <CellDirective index={7}>
colSpan={3}></CellDirective>
                                <CellDirective index={15}>
colSpan={2}></CellDirective>
                                <CellDirective index={17}>
colSpan={2}></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective height={35}>
                                  <CellsDirective>
                                    <CellDirective index={2}>
colSpan={3}></CellDirective>
                                    <CellDirective index={6}>
colSpan={4}></CellDirective>
                                    <CellDirective index={16}>
colSpan={2}></CellDirective>
                                    </CellsDirective>
                                    </RowDirective>
                                    <RowDirective height={35}>
                                      <CellsDirective>
                                        <CellDirective index={2}>
colSpan={4}></CellDirective>
                                        <CellDirective index={7}>
colSpan={3}></CellDirective>
                                        <CellDirective index={15}>
colSpan={2}></CellDirective>
                                        <CellDirective index={17}>
colSpan={2}></CellDirective>
                                      </CellsDirective>
                                      </RowDirective>
                                    </RowsDirective>
                                    <RangesDirective>
                                      <RangeDirective
dataSource={data}></RangeDirective>
                                      </RangesDirective>
                                    <ColumnsDirective>
                                      <ColumnDirective width={90}></ColumnDirective>
                                      <ColumnDirective width={150}></ColumnDirective>
                                      <ColumnDirective width={100}></ColumnDirective>
                                      <ColumnDirective width={100}></ColumnDirective>
                                      <ColumnDirective width={100}></ColumnDirective>
                                      <ColumnDirective width={100}></ColumnDirective>
                                      <ColumnDirective width={100}></ColumnDirective>
                                      <ColumnDirective width={100}></ColumnDirective>
                                      <ColumnDirective width={100}></ColumnDirective>
                                      <ColumnDirective width={100}></ColumnDirective>
                                      <ColumnDirective width={120}></ColumnDirective>
                                      <ColumnDirective width={120}></ColumnDirective>
                                      <ColumnDirective width={120}></ColumnDirective>
                                      <ColumnDirective width={120}></ColumnDirective>
                                      <ColumnDirective width={120}></ColumnDirective>
                                      <ColumnDirective width={100}></ColumnDirective>
                                      <ColumnDirective width={100}></ColumnDirective>
                                      <ColumnDirective width={100}></ColumnDirective>
                                      <ColumnDirective width={100}></ColumnDirective>
                                      <ColumnDirective width={100}></ColumnDirective>
                                    </ColumnsDirective>

```

```

        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  </div>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

## APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective, CellsDirective, CellDirective
} from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A1:S1');
      spreadsheet.numberFormat('h:mm AM/PM', 'C1:S1');
      spreadsheet.cellFormat({ verticalAlign: 'middle' }, 'A1:S11');
      // Merging the `K4:M4` cells using method
      spreadsheet.merge('K4:M4');
      // Merging the 5th and 6th row cells across 11th, 12th and 13th
column
      spreadsheet.merge('K5:M6', 'Vertically');
      // Merging the 18th and 19th column cells across 2nd, 3rd and
4th row
      spreadsheet.merge('N4:O6', 'Horizontally');
    }
  }, []);
  return (
    <div>
      <SpreadsheetComponent ref={spreadsheetRef}
showFormulaBar={false}>
        <SheetsDirective>
          <SheetDirective name={"Merge Cells"}>
            <RowsDirective>
              <RowDirective height={35}></RowDirective>
              <RowDirective height={35}>
                <CellsDirective>
                  <CellDirective index={1}
rowSpan={2}></CellDirective>
                  <CellDirective
colSpan={2}></CellDirective>

```

```

colSpan={3}></CellDirective>
colSpan={3}></CellDirective>
colSpan={2}></CellDirective>
colSpan={2}></CellDirective>
    </CellsDirective>
  </RowDirective>
  <RowDirective height={35}>
    <CellsDirective>
      <CellDirective index={3}>
colSpan={3}></CellDirective>
      <CellDirective index={6}>
colSpan={4}></CellDirective>
      <CellDirective index={13}>
colSpan={3}></CellDirective>
      <CellDirective index={17}>
colSpan={2}></CellDirective>
    </CellsDirective>
  </RowDirective>
  <RowDirective height={35}>
    <CellsDirective>
      <CellDirective index={2}>
colSpan={3}></CellDirective>
      <CellDirective index={5}>
colSpan={2}></CellDirective>
      <CellDirective index={7}>
colSpan={3}></CellDirective>
      <CellDirective index={15}>
colSpan={2}></CellDirective>
      <CellDirective index={17}>
colSpan={2}></CellDirective>
    </CellsDirective>
  </RowDirective>
  <RowDirective height={35}>
    <CellsDirective>
      <CellDirective index={2}>
colSpan={3}></CellDirective>
      <CellDirective index={6}>
colSpan={4}></CellDirective>
      <CellDirective index={16}>
colSpan={2}></CellDirective>
    </CellsDirective>
  </RowDirective>
  <RowDirective height={35}>
    <CellsDirective>
      <CellDirective index={2}>
colSpan={4}></CellDirective>
      <CellDirective index={7}>
colSpan={3}></CellDirective>
      <CellDirective index={15}>
colSpan={2}></CellDirective>
      <CellDirective index={17}>
colSpan={2}></CellDirective>
    </CellsDirective>

```



```

        </RowDirective>
      </RowsDirective>
      <RangesDirective>
        <RangeDirective
dataSource={data}></RangeDirective>
      </RangesDirective>
      <ColumnsDirective>
        <ColumnDirective width={90}></ColumnDirective>
        <ColumnDirective width={150}></ColumnDirective>
        <ColumnDirective width={100}></ColumnDirective>
        <ColumnDirective width={100}></ColumnDirective>
        <ColumnDirective width={100}></ColumnDirective>
        <ColumnDirective width={100}></ColumnDirective>
        <ColumnDirective width={100}></ColumnDirective>
        <ColumnDirective width={100}></ColumnDirective>
        <ColumnDirective width={100}></ColumnDirective>
        <ColumnDirective width={100}></ColumnDirective>
        <ColumnDirective width={120}></ColumnDirective>
        <ColumnDirective width={120}></ColumnDirective>
        <ColumnDirective width={120}></ColumnDirective>
        <ColumnDirective width={120}></ColumnDirective>
        <ColumnDirective width={120}></ColumnDirective>
        <ColumnDirective width={100}></ColumnDirective>
        <ColumnDirective width={100}></ColumnDirective>
        <ColumnDirective width={100}></ColumnDirective>
        <ColumnDirective width={100}></ColumnDirective>
      </ColumnsDirective>
    </SheetDirective>
  </SheetsDirective>
</SpreadsheetComponent>
</div>
);
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

## DATASOURCE.JSX

```

/**
 * Merge cells data source
 */
export let data = [
  {
    'Employee ID': 10001,
    'Employee Name': 'Davolio',
    '9:00 AM': 'Analysis Tasks',
    '9:30 AM': 'Analysis Tasks',
    '10:00 AM': 'Team Meeting',
    '10:30 AM': 'Testing',
    '11:00 AM': 'Development',
    '11:30 AM': 'Development',
    '12:00 PM': 'Development',
    '12:30 PM': 'Support',
  }
]

```

```

      '1:00 PM': 'Lunch Break',
      '1:30 PM': 'Lunch Break',
      '2:00 PM': 'Lunch Break',
      '2:30 PM': 'Testing',
      '3:00 PM': 'Testing',
      '3:30 PM': 'Development',
      '4:00 PM': 'Conference',
      '4:30 PM': 'Team Meeting',
      '5:00 PM': 'Team Meeting'
    },
    {
      'Employee ID': 10002,
      'Employee Name': 'Buchanan',
      '9:00 AM': 'Task Assign',
      '9:30 AM': 'Support',
      '10:00 AM': 'Support',
      '10:30 AM': 'Support',
      '11:00 AM': 'Testing',
      '11:30 AM': 'Testing',
      '12:00 PM': 'Testing',
      '12:30 PM': 'Testing',
      '1:00 PM': 'Lunch Break',
      '1:30 PM': 'Lunch Break',
      '2:00 PM': 'Lunch Break',
      '2:30 PM': 'Development',
      '3:00 PM': 'Development',
      '3:30 PM': 'Check Mail',
      '4:00 PM': 'Check Mail',
      '4:30 PM': 'Team Meeting',
      '5:00 PM': 'Team Meeting'
    },
    {
      'Employee ID': 10003,
      'Employee Name': 'Fuller',
      '9:00 AM': 'Check Mail',
      '9:30 AM': 'Check Mail',
      '10:00 AM': 'Check Mail',
      '10:30 AM': 'Analysis Tasks',
      '11:00 AM': 'Analysis Tasks',
      '11:30 AM': 'Support',
      '12:00 PM': 'Support',
      '12:30 PM': 'Support',
      '1:00 PM': 'Lunch Break',
      '1:30 PM': 'Lunch Break',
      '2:00 PM': 'Lunch Break',
      '2:30 PM': 'Development',
      '3:00 PM': 'Development',
      '3:30 PM': 'Team Meeting',
      '4:00 PM': 'Team Meeting',
      '4:30 PM': 'Development',
      '5:00 PM': 'Development'
    },
    {
      'Employee ID': 10004,
      'Employee Name': 'Leverling',
      '9:00 AM': 'Testing',
      '9:30 AM': 'Check Mail',

```

```

    '10:00 AM': 'Check Mail',
    '10:30 AM': 'Support',
    '11:00 AM': 'Testing',
    '11:30 AM': 'Testing',
    '12:00 PM': 'Testing',
    '12:30 PM': 'Testing',
    '1:00 PM': 'Lunch Break',
    '1:30 PM': 'Lunch Break',
    '2:00 PM': 'Lunch Break',
    '2:30 PM': 'Development',
    '3:00 PM': 'Development',
    '3:30 PM': 'Check Mail',
    '4:00 PM': 'Conference',
    '4:30 PM': 'Conference',
    '5:00 PM': 'Team Meeting'
  },
  {
    'Employee ID': 10005,
    'Employee Name': 'Peacock',
    '9:00 AM': 'Task Assign',
    '9:30 AM': 'Task Assign',
    '10:00 AM': 'Task Assign',
    '10:30 AM': 'Task Assign',
    '11:00 AM': 'Check Mail',
    '11:30 AM': 'Support',
    '12:00 PM': 'Support',
    '12:30 PM': 'Support',
    '1:00 PM': 'Lunch Break',
    '1:30 PM': 'Lunch Break',
    '2:00 PM': 'Lunch Break',
    '2:30 PM': 'Development',
    '3:00 PM': 'Development',
    '3:30 PM': 'Team Meeting',
    '4:00 PM': 'Team Meeting',
    '4:30 PM': 'Testing',
    '5:00 PM': 'Testing'
  }
];

```

### DATASOURCE.TSX

```

/**
 * Merge cells data source
 */
export let data: Object[] = [
  {
    'Employee ID': 10001,
    'Employee Name': 'Davolio',
    '9:00 AM': 'Analysis Tasks',
    '9:30 AM': 'Analysis Tasks',
    '10:00 AM': 'Team Meeting',
    '10:30 AM': 'Testing',
    '11:00 AM': 'Development',
    '11:30 AM': 'Development',
    '12:00 PM': 'Development',
    '12:30 PM': 'Support',

```

```

      '1:00 PM': 'Lunch Break',
      '1:30 PM': 'Lunch Break',
      '2:00 PM': 'Lunch Break',
      '2:30 PM': 'Testing',
      '3:00 PM': 'Testing',
      '3:30 PM': 'Development',
      '4:00 PM': 'Conference',
      '4:30 PM': 'Team Meeting',
      '5:00 PM': 'Team Meeting'
    },
    {
      'Employee ID': 10002,
      'Employee Name': 'Buchanan',
      '9:00 AM': 'Task Assign',
      '9:30 AM': 'Support',
      '10:00 AM': 'Support',
      '10:30 AM': 'Support',
      '11:00 AM': 'Testing',
      '11:30 AM': 'Testing',
      '12:00 PM': 'Testing',
      '12:30 PM': 'Testing',
      '1:00 PM': 'Lunch Break',
      '1:30 PM': 'Lunch Break',
      '2:00 PM': 'Lunch Break',
      '2:30 PM': 'Development',
      '3:00 PM': 'Development',
      '3:30 PM': 'Check Mail',
      '4:00 PM': 'Check Mail',
      '4:30 PM': 'Team Meeting',
      '5:00 PM': 'Team Meeting'
    },
    {
      'Employee ID': 10003,
      'Employee Name': 'Fuller',
      '9:00 AM': 'Check Mail',
      '9:30 AM': 'Check Mail',
      '10:00 AM': 'Check Mail',
      '10:30 AM': 'Analysis Tasks',
      '11:00 AM': 'Analysis Tasks',
      '11:30 AM': 'Support',
      '12:00 PM': 'Support',
      '12:30 PM': 'Support',
      '1:00 PM': 'Lunch Break',
      '1:30 PM': 'Lunch Break',
      '2:00 PM': 'Lunch Break',
      '2:30 PM': 'Development',
      '3:00 PM': 'Development',
      '3:30 PM': 'Team Meeting',
      '4:00 PM': 'Team Meeting',
      '4:30 PM': 'Development',
      '5:00 PM': 'Development'
    },
    {
      'Employee ID': 10004,
      'Employee Name': 'Leverling',
      '9:00 AM': 'Testing',
      '9:30 AM': 'Check Mail',

```

```

    '10:00 AM': 'Check Mail',
    '10:30 AM': 'Support',
    '11:00 AM': 'Testing',
    '11:30 AM': 'Testing',
    '12:00 PM': 'Testing',
    '12:30 PM': 'Testing',
    '1:00 PM': 'Lunch Break',
    '1:30 PM': 'Lunch Break',
    '2:00 PM': 'Lunch Break',
    '2:30 PM': 'Development',
    '3:00 PM': 'Development',
    '3:30 PM': 'Check Mail',
    '4:00 PM': 'Conference',
    '4:30 PM': 'Conference',
    '5:00 PM': 'Team Meeting'
  },
  {
    'Employee ID': 10005,
    'Employee Name': 'Peacock',
    '9:00 AM': 'Task Assign',
    '9:30 AM': 'Task Assign',
    '10:00 AM': 'Task Assign',
    '10:30 AM': 'Task Assign',
    '11:00 AM': 'Check Mail',
    '11:30 AM': 'Support',
    '12:00 PM': 'Support',
    '12:30 PM': 'Support',
    '1:00 PM': 'Lunch Break',
    '1:30 PM': 'Lunch Break',
    '2:00 PM': 'Lunch Break',
    '2:30 PM': 'Development',
    '3:00 PM': 'Development',
    '3:30 PM': 'Team Meeting',
    '4:00 PM': 'Team Meeting',
    '4:30 PM': 'Testing',
    '5:00 PM': 'Testing'
  }
],
];

```

### Limitations of Merge

The following features have some limitations in Merge:

- Merge with filter.
- Merge with wrap text.

### Data Validation

Data Validation is used to restrict the user from entering the invalid data. You can use the [allowDataValidation](#) property to enable or disable data validation.

\* The default value for `allowDataValidation` property is `true`.

### Apply Validation

You can apply data validation to restrict the type of data or the values that users enter into a cell.

You can apply data validation by using one of the following ways,

- Select the Data tab in the Ribbon toolbar, and then choose the Data Validation item.
- Use the [addDataValidation\(\)](#) method programmatically.

#### *Clear Validation*

Clear validation feature is used to remove data validations from the specified ranges or the whole worksheet.

You can clear data validation rule by one of the following ways,

- Select the Data tab in the Ribbon toolbar, and then choose the Clear Validation item.
- Use the [removeDataValidation\(\)](#) method programmatically.

#### *Highlight Invalid Data*

Highlight invalid data feature is used to highlight the previously entered invalid values.

You can highlight an invalid data by using one of the following ways,

- Select the Data tab in the Ribbon toolbar, and then choose the Highlight Invalid Data item.
- Use the [addInvalidHighlight\(\)](#) method programmatically.

#### *Clear Highlighted Invalid Data*

Clear highlight feature is used to remove the highlight from invalid cells.

You can clear the highlighted invalid data by using the following ways,

- Select the Data tab in the Ribbon toolbar, and then choose the Clear Highlight item.
- Use the [removeInvalidHighlight\(\)](#) method programmatically.

### **APP.JSX**

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RowsDirective, RowDirective, CellsDirective, CellDirective,
ColumnsDirective, ColumnDirective } from '@syncfusion/ej2-react-
spreadsheet';
function App() {
  const spreadsheetRef = React.useRef(null);
  const boldCenter = { fontWeight: 'bold', textAlign: 'center' };
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      //Add Data validation to range.
      spreadsheet.addDataValidation({ type: 'TextLength', operator:
'LessThanOrEqualTo', value1: '4' }, 'A2:A5');
      spreadsheet.addDataValidation({ type: 'WholeNumber', operator:
'NotEqualTo', value1: '1' }, 'B2:B5');
      spreadsheet.addDataValidation({ type: 'Date', operator:
'NotEqualTo', value1: '04/11/2019' }, 'F2:F5');
    }
  });
}
```

```

        spreadsheet.addDataValidation({ type: 'Time', operator:
'Between', value1: '10:00:00 AM', value2: '11:00:00 AM' }, 'G2:G5');
        spreadsheet.addDataValidation({ type: 'Decimal', operator:
'LessThan', value1: '100000.00' }, 'H2:H5');
        //Highlight Invalid Data.
        spreadsheet.addInvalidHighlight('A1:H5');
    }
}, []);
return (
    <div> <SpreadsheetComponent ref={spreadsheetRef}
showFormulaBar={false}>
        <SheetsDirective>
            <SheetDirective name={'PriceDetails'}>
                <RowsDirective>
                    <RowDirective index={0}>
                        <CellsDirective>
                            <CellDirective index={0} value={'Seller
Name'} style={boldCenter}></CellDirective>
                            <CellDirective index={1} value={'Customer
Id'} style={boldCenter}></CellDirective>
                            <CellDirective index={2} value={'Customer
Name'} style={boldCenter}></CellDirective>
                            <CellDirective index={3} value={'Product
Name'} style={boldCenter}></CellDirective>
                            <CellDirective index={4} value={'Product
Price'} style={boldCenter}></CellDirective>
                            <CellDirective index={5} value={'Sales
Date'} style={boldCenter}></CellDirective>
                            <CellDirective index={6} value={'Billing
Time'} style={boldCenter}></CellDirective>
                            <CellDirective index={7} value={'Total
Price'} style={boldCenter}></CellDirective>
                        </CellsDirective>
                    </RowDirective>
                    <RowDirective index={1}>
                        <CellsDirective>
                            <CellDirective index={0}
value={'John'}></CellDirective>
                            <CellDirective index={1} value={'1'}
validation={{ type: 'WholeNumber', operator: 'NotEqualTo', value1: '1'
}}></CellDirective>
                            <CellDirective index={2}
value={'Nash'}></CellDirective>
                            <CellDirective index={3} value={'Digger'}
validation={{ type: 'List', value1: 'Digger, Digger, Cherrypicker'
}}></CellDirective>
                            <CellDirective index={4} value={'50000'}
validation={{ type: 'List', value1: '50000,50000,45000' }}></CellDirective>
                            <CellDirective index={5}
value={'04/11/2019'}></CellDirective>
                            <CellDirective index={6} value={'11:34:32
AM'}></CellDirective>
                            <CellDirective index={7}
value={'1,45,000.00'}></CellDirective>
                        </CellsDirective>
                    </RowDirective>
                </RowDirective index={2}>

```

```

        <CellsDirective>
          <CellDirective index={0}
value={'Mike'}></CellDirective>
          <CellDirective index={1} value={'2'}
validation={{ type: 'WholeNumber', operator: 'NotEqualTo', value1: '1'
}}></CellDirective>
          <CellDirective index={2}
value={'Jim'}></CellDirective>
          <CellDirective index={3}
value={'Cherrypicker'} validation={{ type: 'List', value1: 'Cherrypicker,
JCB, Wheelbarrow' }}></CellDirective>
          <CellDirective index={4} value={'45000'}
validation={{ type: 'List', value1: '45000,90000,40' }}></CellDirective>
          <CellDirective index={5}
value={'04/11/2019'}></CellDirective>
          <CellDirective index={6} value={'11:34:32
AM'}></CellDirective>
          <CellDirective index={7}
value={'1,45,000.00'}></CellDirective>
        </CellsDirective>
      </RowDirective>
    <RowDirective index={3}>
      <CellsDirective>
        <CellDirective index={0}
value={'shane'}></CellDirective>
        <CellDirective index={1} value={'3'}
validation={{ type: 'WholeNumber', operator: 'NotEqualTo', value1: '1'
}}></CellDirective>
        <CellDirective index={2}
value={'Sean'}></CellDirective>
        <CellDirective index={3} value={'Kango'}
validation={{ type: 'List', value1: 'Kango, Ropes' }}></CellDirective>
        <CellDirective index={4} value={'450'}
validation={{ type: 'List', value1: '450, 95' }}></CellDirective>
        <CellDirective index={5}
value={'06/25/2019'}></CellDirective>
        <CellDirective index={6} value={'01:30:11
PM'}></CellDirective>
        <CellDirective index={7}
value={'545.00'}></CellDirective>
      </CellsDirective>
    </RowDirective>
    <RowDirective index={4}>
      <CellsDirective>
        <CellDirective index={0}
value={'John'}></CellDirective>
        <CellDirective index={1} value={'1'}
validation={{ type: 'WholeNumber', operator: 'NotEqualTo', value1: '1'
}}></CellDirective>
        <CellDirective index={2}
value={'Nash'}></CellDirective>
        <CellDirective index={3} value={'JCB'}
validation={{ type: 'List', value1: 'JCB, Ropes, scaffolding'
}}></CellDirective>
        <CellDirective index={4} value={'90000'}
validation={{ type: 'List', value1: '90000, 95, 10000' }}></CellDirective>

```



```

                                <CellDirective index={5}
value={ '09/22/2019' }></CellDirective>
                                <CellDirective index={6} value={ '12:30:02
PM' }></CellDirective>
                                <CellDirective index={7}
value={ '1,00,095.00' }></CellDirective>
                                </CellsDirective>
                            </RowDirective>
                        </RowsDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={88}></ColumnDirective>
                        <ColumnDirective width={88}></ColumnDirective>
                        <ColumnDirective width={106}></ColumnDirective>
                        <ColumnDirective width={98}></ColumnDirective>
                        <ColumnDirective width={88}></ColumnDirective>
                        <ColumnDirective width={86}></ColumnDirective>
                        <ColumnDirective width={107}></ColumnDirective>
                        <ColumnDirective width={81}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
            </SheetsDirective>
        </SpreadsheetComponent>
    </div>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

## APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RowsDirective, RowDirective, CellsDirective, CellDirective,
ColumnsDirective, ColumnDirective } from '@syncfusion/ej2-react-
spreadsheet';
import { CellStyleModel } from '@syncfusion/ej2-react-spreadsheet';
function App() {
    const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
    const boldCenter: CellStyleModel = { fontWeight: 'bold', textAlign:
'center' };
    React.useEffect(() => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            //Add Data validation to range.
            spreadsheet.addDataValidation({ type: 'TextLength', operator:
'LessThanOrEqualTo', value1: '4' }, 'A2:A5');
            spreadsheet.addDataValidation({ type: 'WholeNumber', operator:
'NotEqualTo', value1: '1' }, 'B2:B5');
            spreadsheet.addDataValidation({ type: 'Date', operator:
'NotEqualTo', value1: '04/11/2019' }, 'F2:F5');
            spreadsheet.addDataValidation({ type: 'Time', operator:
'Between', value1: '10:00:00 AM', value2: '11:00:00 AM' }, 'G2:G5');

```

```

        spreadsheet.addDataValidation({ type: 'Decimal', operator:
'LessThan', value1: '100000.00' }, 'H2:H5');
        //Highlight Invalid Data.
        spreadsheet.addInvalidHighlight('A1:H5');
    }
}, []);
return (
    <div> <SpreadsheetComponent ref={spreadsheetRef}
showFormulaBar={false}>
        <SheetsDirective>
            <SheetDirective name={'PriceDetails'}>
                <RowsDirective>
                    <RowDirective index={0}>
                        <CellsDirective>
                            <CellDirective index={0} value={'Seller
Name'} style={boldCenter}></CellDirective>
                            <CellDirective index={1} value={'Customer
Id'} style={boldCenter}></CellDirective>
                            <CellDirective index={2} value={'Customer
Name'} style={boldCenter}></CellDirective>
                            <CellDirective index={3} value={'Product
Name'} style={boldCenter}></CellDirective>
                            <CellDirective index={4} value={'Product
Price'} style={boldCenter}></CellDirective>
                            <CellDirective index={5} value={'Sales
Date'} style={boldCenter}></CellDirective>
                            <CellDirective index={6} value={'Billing
Time'} style={boldCenter}></CellDirective>
                            <CellDirective index={7} value={'Total
Price'} style={boldCenter}></CellDirective>
                        </CellsDirective>
                    </RowDirective>
                    <RowDirective index={1}>
                        <CellsDirective>
                            <CellDirective index={0}
value={'John'}></CellDirective>
                            <CellDirective index={1} value={'1'}
validation={{ type: 'WholeNumber', operator: 'NotEqualTo', value1: '1'
}}></CellDirective>
                            <CellDirective index={2}
value={'Nash'}></CellDirective>
                            <CellDirective index={3} value={'Digger'}
validation={{ type: 'List', value1: 'Digger, Digger, Cherrypicker'
}}></CellDirective>
                            <CellDirective index={4} value={'50000'}
validation={{ type: 'List', value1: '50000,50000,45000' }}></CellDirective>
                            <CellDirective index={5}
value={'04/11/2019'}></CellDirective>
                            <CellDirective index={6} value={'11:34:32
AM'}></CellDirective>
                            <CellDirective index={7}
value={'1,45,000.00'}></CellDirective>
                        </CellsDirective>
                    </RowDirective>
                    <RowDirective index={2}>
                        <CellsDirective>

```

```

                                <CellDirective index={0}
value={'Mike'}></CellDirective>
                                <CellDirective index={1} value={'2'}
validation={{ type: 'WholeNumber', operator: 'NotEqualTo', value1: '1'
}}></CellDirective>
                                <CellDirective index={2}
value={'Jim'}></CellDirective>
                                <CellDirective index={3}
value={'Cherrypicker'} validation={{ type: 'List', value1: 'Cherrypicker,
JCB, Wheelbarrow' }}></CellDirective>
                                <CellDirective index={4} value={'45000'}
validation={{ type: 'List', value1: '45000,90000,40' }}></CellDirective>
                                <CellDirective index={5}
value={'04/11/2019'}></CellDirective>
                                <CellDirective index={6} value={'11:34:32
AM'}></CellDirective>
                                <CellDirective index={7}
value={'1,45,000.00'}></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective index={3}>
                                <CellsDirective>
                                <CellDirective index={0}
value={'shane'}></CellDirective>
                                <CellDirective index={1} value={'3'}
validation={{ type: 'WholeNumber', operator: 'NotEqualTo', value1: '1'
}}></CellDirective>
                                <CellDirective index={2}
value={'Sean'}></CellDirective>
                                <CellDirective index={3} value={'Kango'}
validation={{ type: 'List', value1: 'Kango, Ropes' }}></CellDirective>
                                <CellDirective index={4} value={'450'}
validation={{ type: 'List', value1: '450, 95' }}></CellDirective>
                                <CellDirective index={5}
value={'06/25/2019'}></CellDirective>
                                <CellDirective index={6} value={'01:30:11
PM'}></CellDirective>
                                <CellDirective index={7}
value={'545.00'}></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective index={4}>
                                <CellsDirective>
                                <CellDirective index={0}
value={'John'}></CellDirective>
                                <CellDirective index={1} value={'1'}
validation={{ type: 'WholeNumber', operator: 'NotEqualTo', value1: '1'
}}></CellDirective>
                                <CellDirective index={2}
value={'Nash'}></CellDirective>
                                <CellDirective index={3} value={'JCB'}
validation={{ type: 'List', value1: 'JCB, Ropes, scaffolding'
}}></CellDirective>
                                <CellDirective index={4} value={'90000'}
validation={{ type: 'List', value1: '90000, 95, 10000' }}></CellDirective>
                                <CellDirective index={5}
value={'09/22/2019'}></CellDirective>

```

```

                                <CellDirective index={6} value={'12:30:02
PM'}></CellDirective>
                                <CellDirective index={7}
value={'1,00,095.00'}></CellDirective>
                                </CellsDirective>
                            </RowDirective>
                        </RowsDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={88}></ColumnDirective>
                        <ColumnDirective width={88}></ColumnDirective>
                        <ColumnDirective width={106}></ColumnDirective>
                        <ColumnDirective width={98}></ColumnDirective>
                        <ColumnDirective width={88}></ColumnDirective>
                        <ColumnDirective width={86}></ColumnDirective>
                        <ColumnDirective width={107}></ColumnDirective>
                        <ColumnDirective width={81}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
            </SheetsDirective>
        </SpreadsheetComponent>
    </div>

    );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### Limitations of Data validation

The following features have some limitations in Data Validation:

- Entire row data validation.
- Insert row between the data validation.
- Copy/paste with data validation.
- Delete cells between data validation applied range.

### Auto Fill

Auto Fill is used to fill the cells with data based on adjacent cells. It also follows a pattern from adjacent cells if available. There is no need to enter the repeated data manually. You can use `allowAutoFill` property to enable/disable the auto fill support. You can also use `showFillOptions` property to enable/disable the fill option and `fillType` property to change the default auto fill option which is available in `autoFillSettings`.

You can do this by one of the following ways,

- Using “AutoFillOptions” menu which is open, while drag and drop the cell using fill handle element.
- Use the `autoFill()` method programmatically.

The available parameters in `autoFill()` method are,

Parameter	Type	Description
-----------	------	-------------

|----|----|----|

| fillRange | string | Specifies the fill range. |

| dataRange | string | Specifies the data range. |

| direction | AutoFillDirection | Specifies the direction("Up","Right","Down","Left")to be filled. |

| fillType | AutoFillType | Specifies the fill type("CopyCells","FillSeries","FillFormattingOnly","FillWithoutFormatting") for autofill action. |

In Auto Fill we have following options,

- Copy Cells
- Fill Series
- Fill Formatting Only
- Fill Without Formatting

\* The default auto fill option is “FillSeries” which can be referred from fillType property.

### Copy Cells

To copy the selected cell content to the adjacent cells. You can do this by one of the following ways,

- Using fill handle to select the adjacent cell range and “Copy Cells” option in “AutoFillOptions” menu to fill the adjacent cells.
- Using “CopyCells” as fill type in autoFill method to fill the adjacent cells.

### Fill Series

To fill the series of numbers, characters, or dates based on selected cell content to the adjacent cells with their formats.

You can do this by one of the following ways,

- Using fill handle to select the adjacent cell range and “Fill Series” option in “AutoFillOptions” menu to fill the adjacent cells.
- Using “FillSeries” as fill type in autoFill method to fill the adjacent cells.

### Fill Formatting Only

To fill the cell style and number formatting based on the selected cell content to the adjacent cells without their content.

You can do this by one of the following ways,

- Using fill handle to select the adjacent cell range and “Fill Formatting Only” option in “AutoFillOptions” menu to fill the adjacent cells.
- Using “FillFormattingOnly” as fill type in autoFill method to fill the adjacent cells.

### Fill Without Formatting

To fill series of numbers, characters, or dates based on the selected cells to the adjacent cells without their formats.

You can do this by one of the following ways,

- Using fill handle to select the adjacent cell range and “Fill Without Formatting” option in “AutoFillOptions” menu to fill the adjacent cells.
- Using “FillWithoutFormatting” as fill type in `autoFill` method to fill the adjacent cells.

In the following sample, you can enable/disable the fill option on the button click event by using the `showFillOptions` property in `autoFillSettings`.

#### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
    const spreadsheetRef = React.useRef(null);
    const btnClick = () => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            spreadsheet.autoFillSettings.showFillOptions =
!spreadsheet.autoFillSettings.showFillOptions; //To change whether fill
options need to be shown or not.
        }
    };
    React.useEffect(() => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            spreadsheet.cellFormat({ backgroundColor: '#357cd2', color:
'#fff', fontWeight: 'bold', textAlign: 'center' }, 'A1:H1');
            spreadsheet.autoFill('D4:D11', 'D2:D3', 'Down', 'CopyCells');
            spreadsheet.autoFill('E4:E11', 'E2:E3', 'Down', 'FillSeries');
            spreadsheet.autoFill('B4:B11', 'B2:B3', 'Down',
'FillFormattingOnly');
            spreadsheet.autoFill('C4:C11', 'C2:C3', 'Down',
'FillWithoutFormatting');
        }
    }, []);
    return (
        <div>
            <button className='e-btn' onClick={btnClick}>Change
ShowFillOptions</button>
            <SpreadsheetComponent ref={spreadsheetRef}>
                <SheetsDirective>
                    <SheetDirective>
                        <RangesDirective>
                            <RangeDirective
dataSource={data}></RangeDirective>
                        </RangesDirective>
                        <ColumnsDirective>
                            <ColumnDirective width={130}></ColumnDirective>
                            <ColumnDirective width={100}></ColumnDirective>
                            <ColumnDirective width={100}></ColumnDirective>
                        </ColumnsDirective>
                    </SheetDirective>
                </SheetsDirective>
            </SpreadsheetComponent>
        </div>
    );
}
```

```

        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  </div>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  const btnClick = (): void => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.autoFillSettings.showFillOptions =
!spreadsheet.autoFillSettings.showFillOptions; //To change whether fill
options need to be shown or not.
    }
  };
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ backgroundColor: '#357cd2', color:
'#fff', fontWeight: 'bold', textAlign: 'center' }, 'A1:H1');
      spreadsheet.autoFill('D4:D11', 'D2:D3', 'Down', 'CopyCells');
      spreadsheet.autoFill('E4:E11', 'E2:E3', 'Down', 'FillSeries');
      spreadsheet.autoFill('B4:B11', 'B2:B3', 'Down',
'FillFormattingOnly');
      spreadsheet.autoFill('C4:C11', 'C2:C3', 'Down',
'FillWithoutFormatting');
    }
  }, []);
  return (
    <div>
      <button className='e-btn' onClick={btnClick}>Change
ShowFillOptions</button>
      <SpreadsheetComponent ref={spreadsheetRef}>
        <SheetsDirective>
          <SheetDirective>
            <RangesDirective>
              <RangeDirective
dataSource={data}></RangeDirective>
            </RangesDirective>
          </ColumnsDirective>

```

```

        <ColumnDirective width={130}></ColumnDirective>
        <ColumnDirective width={100}></ColumnDirective>
        <ColumnDirective width={100}></ColumnDirective>
      </ColumnsDirective>
    </SheetDirective>
  </SheetsDirective>
</SpreadsheetComponent>
</div>

);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### DATASOURCE.JSX

```

export let data = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
    Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
  { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
    Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
  { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
  { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
'12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
70 },
  { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
    Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
  { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
    Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
  { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
    Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
  { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
    Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
  { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
    Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];

```

### DATASOURCE.TSX

```

export let data: Object[] = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
    Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
  { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
    Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
  { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
  { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
'12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
70 },
];

```



```
{ 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
  Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
{ 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
  Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
{ 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
  Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
{ 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
  Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
{ 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
  Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
```

### Limitations of Autofill

The following features have some limitations in Autofill:

- Flash Fill option in Autofill feature.
- Fill with Conditional Formatting applied cells.

### Clear

Clear feature helps you to clear the cell contents (formulas and data), formats (including number formats, conditional formats, and borders) in a spreadsheet. When you apply clear all, both the contents and the formats will be cleared simultaneously.

### Apply Clear Feature

You can apply clear feature by using one of the following ways,

- Select the clear icon in the Ribbon toolbar under the Home Tab.
- Using the [clear\(\)](#) method to clear the values.

Clear has the following types in the spreadsheet,

| Options | Uses |

|-----|-----|

| **Clear All** | Used to clear all contents, formats, and hyperlinks. |

| **Clear Formats** | Used to clear the formats (including number formats, conditional formats, and borders) in a cell. |

| **Clear Contents** | Used to clear the contents (formulas and data) in a cell. |

| **Clear Hyperlinks** | Used to clear the hyperlink in a cell. |

### Methods

Clear the cell contents and formats in the Spreadsheet document by using the [clear](#) method. The [clear](#) method has **type** and **range** as parameters. The following code example shows how to clear the cell contents and formats in the button click event.

### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
```

```

import { DropDownButtonComponent } from '@syncfusion/ej2-react-splitbuttons';
import { SpreadsheetComponent, SheetsDirective, SheetDirective, RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from '@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
    const spreadsheetRef = React.useRef(null);
    const items = [
        { text: "Clear All" },
        { text: "Clear Formats" },
        { text: "Clear Contents" },
        { text: "Clear Hyperlinks" }
    ];
    const handleItemSelect = (args) => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            if (args.item.text === 'Clear All')
                spreadsheet.clear({ type: 'Clear All', range: 'D1:D10' });
            // Clear the content, formats and hyperlinks applied in the provided range.
            if (args.item.text === 'Clear Formats')
                spreadsheet.clear({ type: 'Clear Formats', range: 'B1:B10' });
            // Clear the formats applied in the provided range
            if (args.item.text === 'Clear Contents')
                spreadsheet.clear({ type: 'Clear Contents', range: 'A1:A10' });
            // Clear the content in the provided range
            if (args.item.text === 'Clear Hyperlinks')
                spreadsheet.clear({ type: 'Clear Hyperlinks', range: 'F2:F6' });
            // Clear the hyperlinks applied in the provided range
        }
    };
    React.useEffect(() => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            spreadsheet.cellFormat({ fontWeight: 'bold', fontSize: '12pt' }, 'A1:F1');
            spreadsheet.cellFormat({ color: '#10c469' }, 'B1:B10');
        }
    }, []);
    return (
        <div>
            <DropDownButtonComponent items={items}
select={handleItemSelect}> Clear </DropDownButtonComponent>
            <SpreadsheetComponent ref={spreadsheetRef}>
                <SheetsDirective>
                    <SheetDirective>
                        <RangesDirective>
                            <RangeDirective dataSource={data} />
                        </RangesDirective>
                        <ColumnsDirective>
                            <ColumnDirective width={100} />
                            <ColumnDirective width={110} />
                            <ColumnDirective width={100} />
                            <ColumnDirective width={180} />
                            <ColumnDirective width={130} />
                            <ColumnDirective width={130} />
                        </ColumnsDirective>
                    </SheetDirective>
                </SheetsDirective>
            </SpreadsheetComponent>
        </div>
    );
}

```

```

        </ColumnsDirective>
      </SheetDirective>
    </SheetsDirective>
  </SpreadsheetComponent>
</div>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { DropDownButtonComponent, ItemModel } from '@syncfusion/ej2-react-splitbuttons';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, MenuSelectEventArgs } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from '@syncfusion/ej2-react-spreadsheet';
import { data } from '../datasource';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  const items: ItemModel[] = [
    { text: "Clear All" },
    { text: "Clear Formats" },
    { text: "Clear Contents" },
    { text: "Clear Hyperlinks" }
  ];
  const handleItemSelect = (args: MenuSelectEventArgs): void => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      if (args.item.text === 'Clear All')
        spreadsheet.clear({ type: 'Clear All', range: 'D1:D10' });
      // Clear the content, formats and hyperlinks applied in the provided range.
      if (args.item.text === 'Clear Formats')
        spreadsheet.clear({ type: 'Clear Formats', range: 'B1:B10' });
      // Clear the formats applied in the provided range
      if (args.item.text === 'Clear Contents')
        spreadsheet.clear({ type: 'Clear Contents', range: 'A1:A10' });
      // Clear the content in the provided range
      if (args.item.text === 'Clear Hyperlinks')
        spreadsheet.clear({ type: 'Clear Hyperlinks', range: 'F2:F6' });
      // Clear the hyperlinks applied in the provided range
    }
  };
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', fontSize: '12pt' }, 'A1:F1');
      spreadsheet.cellFormat({ color: '#10c469' }, 'B1:B10');
    }
  });
}

```

```

    }
    }, []);
    return (
      <div>
        <DropDownButtonComponent items={items}
        select={handleItemSelect}> Clear </DropDownButtonComponent>
        <SpreadsheetComponent ref={spreadsheetRef}>
          <SheetsDirective>
            <SheetDirective>
              <RangesDirective>
                <RangeDirective dataSource={data} />
              </RangesDirective>
              <ColumnsDirective>
                <ColumnDirective width={100} />
                <ColumnDirective width={110} />
                <ColumnDirective width={100} />
                <ColumnDirective width={180} />
                <ColumnDirective width={130} />
                <ColumnDirective width={130} />
              </ColumnsDirective>
            </SheetDirective>
          </SheetsDirective>
        </SpreadsheetComponent>
      </div>
    );
  };
  export default App;
  const root = createRoot(document.getElementById('root')!);
  root.render(<App />);
  {% endraw %}

```

## DATASOURCE.JSX

```

/**
 * Default data source
 */
export let data = [{
  OrderID: 10248,
  CustomerID: 'VINET',
  EmployeeID: 5,
  ShipName: 'Vins et alcools Chevalier',
  ShipCity: 'Reims',
  Website: 'https://www.amazon.com/'
},
{
  OrderID: 10249,
  CustomerID: 'TOMSP',
  EmployeeID: 6,
  ShipName: 'Toms Spezialitäten',
  ShipCity: 'Münster',
  Website: 'https://www.overstock.com/'
},
{
  OrderID: 10250,
  CustomerID: 'HANAR',
  EmployeeID: 4,

```

```

      ShipName: 'Hanari Carnes',
      ShipCity: 'Rio de Janeiro',
      Website: 'https://www.aliexpress.com/'
    },
    {
      OrderID: 10251,
      CustomerID: 'VICTE',
      EmployeeID: 3,
      ShipName: 'Victuailles en stock',
      ShipCity: 'Lyon',
      Website: 'http://www.alibaba.com/'
    },
    {
      OrderID: 10252,
      CustomerID: 'SUPRD',
      EmployeeID: 4,
      ShipName: 'Suprêmes délices',
      ShipCity: 'Charleroi',
      Website: 'https://taobao.com/'
    }
  ]];

```

### DATASOURCE.TSX

```

/**
 * Default data source
 */
export let data: Object[] = [{
  OrderID: 10248,
  CustomerID: 'VINET',
  EmployeeID: 5,
  ShipName: 'Vins et alcools Chevalier',
  ShipCity: 'Reims',
  Website: 'https://www.amazon.com/'
},
{
  OrderID: 10249,
  CustomerID: 'TOMSP',
  EmployeeID: 6,
  ShipName: 'Toms Spezialitäten',
  ShipCity: 'Münster',
  Website: 'https://www.overstock.com/'
},
{
  OrderID: 10250,
  CustomerID: 'HANAR',
  EmployeeID: 4,
  ShipName: 'Hanari Carnes',
  ShipCity: 'Rio de Janeiro',
  Website: 'https://www.aliexpress.com/'
},
{
  OrderID: 10251,
  CustomerID: 'VICTE',
  EmployeeID: 3,
  ShipName: 'Victuailles en stock',
  ShipCity: 'Lyon',

```

```
Website: 'http://www.alibaba.com/'
},
{
  OrderID: 10252,
  CustomerID: 'SUPRD',
  EmployeeID: 4,
  ShipName: 'Suprêmes délices',
  ShipCity: 'Charleroi',
  Website: 'https://taobao.com/'
}
];
```

### Note

You can refer to our [React Spreadsheet](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Spreadsheet example](#) to know how to present and manipulate data.

### See Also

- [Rows and columns](#)
- [Formatting](#)
- [Hyperlink](#)
- [Sorting](#)
- [Filtering](#)

## Editing in React Spreadsheet component

You can edit the contents of a cell directly in the cell or by typing in the formula bar. By default, the editing feature is enabled in the spreadsheet. Use the [Link to the Video](#) property to enable or disable the editing feature.

To get started quickly with Editing, you can check on this video:

### Edit cell

You can start editing by one of the following ways,

- Double click a cell to start the edit mode.
- Press F2 key to edit the active cell.
- Use formula bar to perform editing.
- Use BACKSPACE or SPACE key to clear the cell content and start the edit mode.
- Using the [startEdit](#) method.

### Save cell

If the cell is in editable state, you can save the edited cell by one of the following ways,

- Perform mouse click on any other cell rather than the current editing cell.
- Press Enter or Tab keys to save the edited cell content.
- Using the [endEdit](#) method.

### Cancel editing

To cancel the editing without saving the changes, you can use one of the following ways,

- Press **ESCAPE** key, this will remove the editable state and update the unchanged cell content.
- Using the [closeEdit](#) method.

The following sample shows how to prevent the editing and cell save. Here **E** column prevent the editing by using cancel argument as true in [cellEdit](#) event. In **D** column, prevent saving the edited changes by using cancel argument as true in [beforeCellSave](#) and use [closeEdit](#) method in spreadsheet.

### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective, CellDirective, CellsDirective
} from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { data } from '../datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A1:E1');
      spreadsheet.cellFormat({ textAlign: 'center' }, 'A2:A10');
      spreadsheet.cellFormat({ textAlign: 'center' }, 'C2:C10');
      spreadsheet.numberFormat('$#,##0.00', 'D2:D10');
      spreadsheet.numberFormat('$#,##0.00', 'E2:E11');
    }
  }, []);
  // Triggers before going to the editing mode.
  const onCellEdit = (args) => {
    // Preventing the editing in 5th(Amount) column.
    if (args.address.includes('E')) { args.cancel = true; }
  };
  // Trigger before saving the edited cell content.
  const onBeforeCellSave = (args) => {
    // Prevent saving the edited changes in 4th(Rate) column.
    if (args.address.includes('D')) {
      args.cancel = true;
      // Manually removes the editable state without saving the
changes. Use `endEdit` method if you want to save the changes.
      let spreadsheet = spreadsheetRef.current;
      if (spreadsheet) {
        spreadsheet.closeEdit();
      }
    }
  };
  return (
    <div>
      <SpreadsheetComponent ref={spreadsheetRef} showSheetTabs={false}
cellEdit={onCellEdit} beforeCellSave={onBeforeCellSave} >
        <SheetsDirective>
          <SheetDirective selectedRange={"C7"}>
            <RowsDirective>
```

```

        <RowDirective index={10}>
          <CellsDirective>
            <CellDirective index={3} value={'Total
Amount'} style={{ fontWeight: 'bold' }}></CellDirective>
            <CellDirective formula={'=SUM(E2:E10)'}>
            </CellDirective>
          </CellsDirective>
        </RowDirective>
      </RowsDirective>
    </RangesDirective>
    <RangeDirective
dataSource={data}></RangeDirective>
  </RangesDirective>
  <ColumnsDirective>
    <ColumnDirective width={120}
index={1}></ColumnDirective>
    <ColumnDirective width={180}></ColumnDirective>
    <ColumnDirective width={100}></ColumnDirective>
    <ColumnDirective width={120}></ColumnDirective>
    <ColumnDirective width={120}></ColumnDirective>
  </ColumnsDirective>
</SheetDirective>
</SheetsDirective>
</SpreadsheetComponent>
</div>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective, CellDirective, CellsDirective,
CellEditEventArgs } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A1:E1');
      spreadsheet.cellFormat({ textAlign: 'center' }, 'A2:A10');
      spreadsheet.cellFormat({ textAlign: 'center' }, 'C2:C10');
      spreadsheet.numberFormat('$#,##0.00', 'D2:D10');
      spreadsheet.numberFormat('$#,##0.00', 'E2:E11');
    }
  }, []);

```



```

// Triggers before going to the editing mode.
const onCellEdit = (args: CellEditEventArgs): void => {
  // Preventing the editing in 5th(Amount) column.
  if (args.address.includes('E')) { args.cancel = true; }
};
// Trigger before saving the edited cell content.
const onBeforeCellSave = (args: CellEditEventArgs): void => {
  // Prevent saving the edited changes in 4th(Rate) column.
  if (args.address.includes('D')) {
    args.cancel = true;
    // Manually removes the editable state without saving the
    changes. Use `endEdit` method if you want to save the changes.
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.closeEdit();
    }
  }
};
return (
  <div>
    <SpreadsheetComponent ref={spreadsheetRef} showSheetTabs={false}
    cellEdit={onCellEdit} beforeCellSave={onBeforeCellSave} >
      <SheetsDirective>
        <SheetDirective selectedRange={"C7"}>
          <RowsDirective>
            <RowDirective index={10}>
              <CellsDirective>
                <CellDirective index={3} value={'Total
Amount'} style={{ fontWeight: 'bold' }}></CellDirective>
                <CellDirective formula={'=SUM(E2:E10)'}>
                  </CellDirective>
                </CellsDirective>
              </RowDirective>
            </RowsDirective>
            <RangesDirective>
              <RangeDirective
dataSource={data}></RangeDirective>
            </RangesDirective>
            <ColumnsDirective>
              <ColumnDirective width={120}
index={1}></ColumnDirective>
              <ColumnDirective width={180}></ColumnDirective>
              <ColumnDirective width={100}></ColumnDirective>
              <ColumnDirective width={120}></ColumnDirective>
              <ColumnDirective width={120}></ColumnDirective>
            </ColumnsDirective>
          </SheetDirective>
        </SheetsDirective>
      </SpreadsheetComponent>
    </div>
  );
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% enddraw %}

```

**DATASOURCE.JSX**

```

/**
 * Editing data source
 */
export let data = [
  {
    'Item Code': 'I231',
    'Item Name': 'Chinese Combo Noodle',
    'Quantity': 2,
    'Rate': 125,
    'Amount': '=PRODUCT(C2,D2) '
  },
  {
    'Item Code': 'I245',
    'Item Name': 'Chinese Combo Rice',
    'Quantity': 3,
    'Rate': 125,
    'Amount': '=PRODUCT(C3,D3) '
  },
  {
    'Item Code': 'I237',
    'Item Name': 'Amritsari Chola',
    'Quantity': 2,
    'Rate': 225,
    'Amount': '=PRODUCT(C4,D4) '
  },
  {
    'Item Code': 'I291',
    'Item Name': 'Asian Mixed Entree Platt',
    'Quantity': 3,
    'Rate': 165,
    'Amount': '=PRODUCT(C5,D5) '
  },
  {
    'Item Code': 'I268',
    'Item Name': 'Chinese Combo Chicken',
    'Quantity': 3,
    'Rate': 125,
    'Amount': '=PRODUCT(C6,D6) '
  },
  {
    'Item Code': 'I251',
    'Item Name': 'Chivas Regal',
    'Quantity': 1,
    'Rate': 325,
    'Amount': '=PRODUCT(C7,D7) '
  },
  {
    'Item Code': 'I256',
    'Item Name': 'Chicken Drumsticks',
    'Quantity': 2,
    'Rate': 180,
    'Amount': '=PRODUCT(C8,D8) '
  },
],

```

```

    {
      'Item Code': 'I232',
      'Item Name': 'Manchow Soup',
      'Quantity': 2,
      'Rate': 160,
      'Amount': '=PRODUCT(C9,D9)'
    },
    {
      'Item Code': 'I290',
      'Item Name': 'Schezuan Chicken',
      'Quantity': 3,
      'Rate': 180,
      'Amount': '=PRODUCT(C10,D10)'
    }
  ];

```

**DATASOURCE.TSX**

```

/**
 * Editing data source
 */
export let data: Object[] = [
  {
    'Item Code': 'I231',
    'Item Name': 'Chinese Combo Noodle',
    'Quantity': 2,
    'Rate': 125,
    'Amount': '=PRODUCT(C2,D2)'
  },
  {
    'Item Code': 'I245',
    'Item Name': 'Chinese Combo Rice',
    'Quantity': 3,
    'Rate': 125,
    'Amount': '=PRODUCT(C3,D3)'
  },
  {
    'Item Code': 'I237',
    'Item Name': 'Amritsari Chola',
    'Quantity': 2,
    'Rate': 225,
    'Amount': '=PRODUCT(C4,D4)'
  },
  {
    'Item Code': 'I291',
    'Item Name': 'Asian Mixed Entree Platt',
    'Quantity': 3,
    'Rate': 165,
    'Amount': '=PRODUCT(C5,D5)'
  },
  {
    'Item Code': 'I268',
    'Item Name': 'Chinese Combo Chicken',
    'Quantity': 3,
    'Rate': 125,
    'Amount': '=PRODUCT(C6,D6)'
  }
];

```

```

    },
    {
      'Item Code': 'I251',
      'Item Name': 'Chivas Regal',
      'Quantity': 1,
      'Rate': 325,
      'Amount': '=PRODUCT(C7,D7)'
    },
    {
      'Item Code': 'I256',
      'Item Name': 'Chicken Drumsticks',
      'Quantity': 2,
      'Rate': 180,
      'Amount': '=PRODUCT(C8,D8)'
    },
    {
      'Item Code': 'I232',
      'Item Name': 'Manchow Soup',
      'Quantity': 2,
      'Rate': 160,
      'Amount': '=PRODUCT(C9,D9)'
    },
    {
      'Item Code': 'I290',
      'Item Name': 'Schezuan Chicken',
      'Quantity': 3,
      'Rate': 180,
      'Amount': '=PRODUCT(C10,D10)'
    }
  ],
];

```

### Limitations

- Text overflow in cells is not supported in Editing.

### Note

You can refer to our [React Spreadsheet](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Spreadsheet example](#) to know how to present and manipulate data.

### See Also

- [Cell range](#)
- [Formatting](#)
- [Hyperlink](#)
- [Undo and Redo](#)
- [Unlock the particular cells in the protected sheet](#)

### Context menu in React Spreadsheet component

Context Menu is used to improve user interaction with Spreadsheet using the popup menu. This will open when right-clicking on Cell/Column Header/Row Header/ Pager in the Spreadsheet. You can use [enableContextMenu](#) property to enable/disable context menu.

The default value for the `enableContextMenu` property is `true`.

#### Context Menu Items in Row Cell

Please find the table below for default context menu items and their actions.

Context Menu items	Action
----- -----	
<a href="#">Cut</a>	Cut the selected cells data to the clipboard, you can select a cell where you want to move the data.
<a href="#">Copy</a>	Copy the selected cells data to the clipboard, so that you can paste it to somewhere else.
<a href="#">Paste</a>	Paste the data from clipboard to spreadsheet.
<a href="#">Paste Special</a>	<a href="#">Values</a> - Paste the data values from clipboard to spreadsheet. <a href="#">Formats</a> - Paste the data formats from   clipboard to spreadsheet.
<a href="#">Filter</a>	Perform filtering to the selected cells based on an active cell's value.
<a href="#">Sort</a>	Perform sorting to the selected range of cells by ascending or descending.
<a href="#">Hyperlink</a>	Create a link in the spreadsheet to navigate to web links or cell reference within the sheet or other sheets   in the Spreadsheet.

#### Context Menu Items in Row Header / Column Header

Please find the table below for default context menu items and their actions.

Context Menu items	Action
----- -----	
<a href="#">Cut</a>	Cut the selected row/column header data to the clipboard, you can select a cell where you want to move the data.
<a href="#">Copy</a>	Copy the selected row/column header data to the clipboard, so that you can paste it to somewhere else.
<a href="#">Paste</a>	Paste the data from clipboard to spreadsheet.
<a href="#">Paste Special</a>	<a href="#">Values</a> - Paste the data values from clipboard to spreadsheet. <a href="#">Formats</a> - Paste the data formats from   clipboard to spreadsheet.
<a href="#">Insert Columns</a>	Insert new rows or columns into the worksheet.
<a href="#">Delete Columns</a>	Delete existing rows or columns from the worksheet.
<a href="#">Hide Columns</a>	Hide the rows and columns.
<a href="#">UnHide Columns</a>	Show the hidden rows and columns.

#### Context Menu Items in Pager

Please find the table below for default context menu items and their actions.

Context Menu items	Action
----- -----	
<a href="#">Insert</a>	Insert a new worksheet in front of an existing worksheet in the spreadsheet.

- | **Delete** | Delete the selected worksheet from the spreadsheet. |
- | **Rename** | Rename the selected worksheet. |
- | [Protect Sheet](#) | Prevent unwanted changes from others by limiting their ability to edit. |
- | [Hide](#) | Hide the selected worksheet. |

### Context Menu Customization

You can perform the following context menu customization options in the spreadsheet

- Add Context Menu Items
- Remove Context Menu Items
- Enable/Disable Context Menu Items

#### Add Context Menu Items

You can add the custom items in context menu using the [addContextMenuItems](#) in `contextmenuBeforeOpen` event

In this demo, Custom Item is added after the Paste item in the context menu.

#### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
function App() {
  const spreadsheetRef = React.useRef(null);
  const onContextMenuBeforeOpen = (args) => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet && args.element.id === spreadsheet.element.id +
    '_contextmenu') {
      spreadsheet.addContextMenuItems([{ text: 'Custom Item' }],
      'Paste Special', false); //To pass the items, Item before / after that the
      element to be inserted, Set false if the items need to be inserted before
      the text.
    }
  };
  return (<SpreadsheetComponent ref={spreadsheetRef}
  contextMenuBeforeOpen={onContextMenuBeforeOpen} />);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}
```

#### APP.TSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
import { BeforeOpenCloseMenuEventArgs } from '@syncfusion/ej2-react-splitbuttons';
function App() {
```

```

const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
const onContextMenuBeforeOpen = (args: BeforeOpenCloseMenuEventArgs) =>
{
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet && args.element.id === spreadsheet.element.id +
'_contextmenu') {
        spreadsheet.addContextMenuItems([
            { text: 'Custom Item' } ],
            'Paste Special', false); //To pass the items, Item before / after that the
element to be inserted, Set false if the items need to be inserted before
the text.
    }
};
return (<SpreadsheetComponent ref={spreadsheetRef}
contextMenuBeforeOpen={onContextMenuBeforeOpen} />);
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

### Remove Context Menu Items

You can remove the items in context menu using the [removeContextMenuItems](#) in `contextmenuBeforeOpen` event

In this demo, Insert Column item has been removed from the row/column header context menu.

### APP.JSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
function App() {
    const spreadsheetRef = React.useRef(null);
    const onContextMenuBeforeOpen = () => {
        let spreadsheet = spreadsheetRef.current;
        // To remove existing context menu items.
        if (spreadsheet) {
            spreadsheet.removeContextMenuItems(['Insert Column'], false);
        }
    };
    return (<SpreadsheetComponent ref={spreadsheetRef}
contextMenuBeforeOpen={onContextMenuBeforeOpen} />);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
function App() {

```

```

const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
const onContextMenuBeforeOpen = (): void => {
  let spreadsheet = spreadsheetRef.current;
  // To remove existing context menu items.
  if (spreadsheet) {
    spreadsheet.removeContextMenuItems(['Insert Column'], false);
  }
};
return (<SpreadsheetComponent ref={spreadsheetRef}
contextMenuBeforeOpen={onContextMenuBeforeOpen} />);
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

### Enable/Disable Context Menu Items

You can enable/disable the items in context menu using the [enableContextMenuItems](#) in `contextmenuBeforeOpen` event

In this demo, Rename item is disabled in the pager context menu.

### APP.JSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
function App() {
  const spreadsheetRef = React.useRef(null);
  const onContextMenuBeforeOpen = (args) => {
    let spreadsheet = spreadsheetRef.current;
    //To enable / disable context menu items.
    if (spreadsheet && args.element.id === spreadsheet.element.id +
    '_contextmenu') {
      spreadsheet.enableContextMenuItems(['Rename'], false, false); //
      Contextmenu Items that needs to be enabled / disabled, Set true / false to
      enable / disable the menu items, Set true if the given text is a unique id.
    }
  };
  return (<SpreadsheetComponent ref={spreadsheetRef}
contextMenuBeforeOpen={onContextMenuBeforeOpen} />);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
import { BeforeOpenCloseMenuEventArgs } from '@syncfusion/ej2-react-
splitbuttons';

```



```
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  const onContextMenuBeforeOpen = (args: BeforeOpenCloseMenuEventArgs) =>
  {
    let spreadsheet = spreadsheetRef.current;
    //To enable / disable context menu items.
    if (spreadsheet && args.element.id === spreadsheet.element.id +
    '_contextmenu') {
      spreadsheet.enableContextMenuItems(['Rename'], false, false); //
      Contextmenu Items that needs to be enabled / disabled, Set true / false to
      enable / disable the menu items, Set true if the given text is a unique id.
    }
  };
  return (<SpreadsheetComponent ref={spreadsheetRef}
  contextMenuBeforeOpen={onContextMenuBeforeOpen} />);
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}
```

### Note

You can refer to our [React Spreadsheet](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Spreadsheet example](#) to know how to present and manipulate data.

### See Also

- [Worksheet](#)
- [Rows and columns](#)

## Formulas in React Spreadsheet component

Formulas are used for calculating the data in a worksheet. You can refer the cell reference from same sheet or from different sheets.

### Usage

You can set formula for a cell in the following ways,

- Using the `formula` property from `cell`, you can set the formula or expression to each cell at initial load.
- Set the formula or expression through data binding.
- You can set formula for a cell by [editing](#).
- Using the [updateCell](#) method, you can set or update the cell formula.

### Culture-Based Argument Separator

Previously, although you could import culture-based Excel files into the Spreadsheet component, the formulas wouldn't calculate correctly. This was due to the absence of culture-based argument separators and support for culture-based formatted numeric values as arguments. However, starting from version 25.1.35, you can now import culture-based Excel files into the Spreadsheet component.

Before importing culture-based Excel files, ensure that the Spreadsheet component is rendered with the corresponding culture. Additionally, launch the import/export services with the same culture to ensure compatibility.

When loading spreadsheet data with culture-based formula argument separators using cell data binding, local/remote data, or JSON, ensure to set the [listSeparator](#) property value as the culture-based list separator from your end. Additionally, note that when importing an Excel file, the [listSeparator](#) property will be updated based on the culture of the launched import/export service.

In the example below, the Spreadsheet component is rendered with the **German culture (de)**. Additionally, you can find references on how to set the culture-based argument separator and culture-based formatted numeric value as arguments to the formulas.

### APP.JSX

```
import * as React from 'react';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective, CellDirective, CellsDirective
} from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective,
getFormatFromType } from '@syncfusion/ej2-react-spreadsheet';
import { L10n, loadCldr, setCurrencyCode, setCulture } from
 '@syncfusion/ej2-base';
import deDELocalization from './locale.json';
import cagregorian from './ca-gregorian.json';
import currencies from './currencies.json';
import numbers from './numbers.json';
import timeZoneNames from './timeZoneNames.json';
import numberingSystems from './numberingSystems.json';
import { data } from './datasource';
L10n.load(deDELocalization);
function App() {
  const spreadsheetRef = React.useRef(null);
  setCulture('de');
  setCurrencyCode('EUR');
  loadCldr(cagregorian, currencies, numbers, timeZoneNames,
numberingSystems);
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ textAlign: 'center', fontWeight: 'bold' },
'A1:E1');
      spreadsheet.numberFormat(getFormatFromType('Currency'), 'D2:E12');
      spreadsheet.numberFormat(getFormatFromType('Currency'), 'E13:E15');
    }
  }, []);
  return (
    <div>
      <SpreadsheetComponent ref={spreadsheetRef} locale='de'
listSeparator=';' showSheetTabs={false} showRibbon={false}>
        <SheetsDirective>
          <SheetDirective selectedRange='E14'>
            <RangesDirective>
              <RangeDirective dataSource={data}></RangeDirective>
            </RangesDirective>
          </SheetDirective>
        </SheetsDirective>
      </SpreadsheetComponent>
    </div>
  );
}
```

```

        <RowDirective index={12}>
          <CellsDirective>
            <CellDirective index={3}
value='Subtotal:'></CellDirective>
            <CellDirective
formula='=SUBTOTAL(9;E2:E12)'></CellDirective>
          </CellsDirective>
        </RowDirective>
        <RowDirective>
          <CellsDirective>
            <CellDirective index={3} value='Discount
(8,5%):'></CellDirective>
            <CellDirective
formula='=PRODUCT(8,5;E13)/100'></CellDirective>
          </CellsDirective>
        </RowDirective>
        <RowDirective>
          <CellsDirective>
            <CellDirective index={3} value='Total
Amount:'></CellDirective>
            <CellDirective formula='=E13-E14'></CellDirective>
          </CellsDirective>
        </RowDirective>
      </RowsDirective>
    </ColumnsDirective>
    <ColumnDirective width={120}></ColumnDirective>
    <ColumnDirective width={180}></ColumnDirective>
    <ColumnDirective width={100}></ColumnDirective>
    <ColumnDirective width={120}></ColumnDirective>
    <ColumnDirective width={120}></ColumnDirective>
  </ColumnsDirective>
</SheetDirective>
</SheetsDirective>
</SpreadsheetComponent>
</div>
)
};
export default App;

```

## APP.TSX

```

import * as React from 'react';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective, CellDirective, CellsDirective
} from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective,
getFormatFromType } from '@syncfusion/ej2-react-spreadsheet';
import { L10n, loadCldr, setCurrencyCode, setCulture } from
'@syncfusion/ej2-base';
import deDELocalization from './locale.json';
import cagregorian from './ca-gregorian.json';
import currencies from './currencies.json';
import numbers from './numbers.json';
import timeZoneNames from './timeZoneNames.json';
import numberingSystems from './numberingSystems.json';
import { data } from './datasource';

```

```

L10n.load(deDELocalization);
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  setCulture('de');
  setCurrencyCode('EUR');
  loadCldr(cagregorian, currencies, numbers, timeZoneNames,
numberingSystems);
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ textAlign: 'center', fontWeight: 'bold' },
'A1:E1');
      spreadsheet.numberFormat(getFormatFromType('Currency'), 'D2:E12');
      spreadsheet.numberFormat(getFormatFromType('Currency'), 'E13:E15');
    }
  }, []);
  return (
    <div>
      <SpreadsheetComponent ref={spreadsheetRef} locale='de'
listSeparator=';' showSheetTabs={false} showRibbon={false}>
        <SheetsDirective>
          <SheetDirective selectedRange='E14'>
            <RangesDirective>
              <RangeDirective dataSource={data}></RangeDirective>
            </RangesDirective>
            <RowsDirective>
              <RowDirective index={12}>
                <CellsDirective>
                  <CellDirective index={3}
value='Subtotal:'></CellDirective>
                  <CellDirective
formula='=SUBTOTAL(9;E2:E12)'></CellDirective>
                </CellsDirective>
              </RowDirective>
              <RowDirective>
                <CellsDirective>
                  <CellDirective index={3} value='Discount
(8,5%):'></CellDirective>
                  <CellDirective
formula='=PRODUCT(8,5;E13)/100'></CellDirective>
                </CellsDirective>
              </RowDirective>
              <RowDirective>
                <CellsDirective>
                  <CellDirective index={3} value='Total
Amount:'></CellDirective>
                  <CellDirective formula='=E13-E14'></CellDirective>
                </CellsDirective>
              </RowDirective>
            </RowsDirective>
            <ColumnsDirective>
              <ColumnDirective width={120}></ColumnDirective>
              <ColumnDirective width={180}></ColumnDirective>
              <ColumnDirective width={100}></ColumnDirective>
              <ColumnDirective width={120}></ColumnDirective>
              <ColumnDirective width={120}></ColumnDirective>
            </ColumnsDirective>
          </SheetDirective>
        </SheetsDirective>
      </SpreadsheetComponent>
    </div>
  );
}

```

```

        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  </div>
)
};
export default App;

```

## LOCALE.JSON

```

{
  "de": {
    "spreadsheet": {
      "InsertingEmptyValue": "Referenzwert ist ungültig.",
      "FindValue": "Wert finden",
      "ReplaceValue": "Wert ersetzen",
      "FindReplaceTooltip": "Ersatz finden",
      "ByRow": "Nach Reihen",
      "ByColumn": "Nach Spalten",
      "MatchExactCellElements": "Gleichen Sie den gesamten Zellinhalt ab",
      "EnterCellAddress": "Zelladresse eingeben",
      "FindAndReplace": "Suchen und Ersetzen",
      "ReplaceAllEnd": "Streichhölzer ersetzt durch ",
      "FindNextBtn": "Nächstes finden",
      "FindPreviousBtn": "Zurück suchen",
      "ReplaceBtn": "Ersetzen",
      "ReplaceAllBtn": "Alles ersetzen",
      "GotoHeader": "Gehe zu",
      "Sheet": "Blatt",
      "SearchWithin": "In ... suchen",
      "SearchBy": "Suche nach",
      "Reference": "Bezug",
      "Workbook": "Arbeitsmappe",
      "NoElements": "Wir konnten nicht finden, wonach Sie gesucht haben.",
      "FindWhat": "Finde was",
      "ReplaceWith": "Ersetzen mit",
      "FileName": "Dateiname",
      "ExtendValidation": "Die Auswahl enthält einige Zellen ohne  
Datenvalidierung. Möchten Sie die Validierung auf diese Zellen ausdehnen?",
      "Yes": "Ja",
      "No": "Nein",
      "PROPER": "Konvertiert einen Text in die richtige Groß-  
/Kleinschreibung; Anfangsbuchstabe in Großbuchstaben und andere Buchstaben  
in Kleinbuchstaben.",
      "Cut": "Schneiden",
      "Copy": "Kopieren",
      "Paste": "Paste",
      "PasteSpecial": "Spezielles einfügen",
      "All": "Alle",
      "Values": "Werte",
      "Formats": "Formate",
      "Font": "Schriftart",
      "FontSize": "Schriftgröße",
      "Bold": "Fett gedruckt",
      "Italic": "Kursiv",
      "Underline": "Unterstreichen",
    }
  }
}

```

```

"Strikethrough": "Durchgestrichen",
"TextColor": "Textfarbe",
"FillColor": "Füllfarbe",
"HorizontalAlignment": "Horizontale Ausrichtung",
"AlignLeft": "Linksbündig",
"AlignCenter": "Center",
"AlignRight": "Rechts ausrichten",
"VerticalAlignment": "Vertikale Ausrichtung",
"AlignTop": "Oben ausrichten",
"AlignMiddle": "Mitte ausrichten",
"AlignBottom": "Unten ausrichten",
"MergeCells": "Zellen verbinden",
"MergeAll": "Alle zusammenführen",
"MergeHorizontally": "Horizontal zusammenführen",
"MergeVertically": "Vertikal zusammenführen",
"Unmerge": "Zusammenführung aufheben",
"UnmergeCells": "Zellen trennen",
"SelectMergeType": "Wählen Sie Zusammenführungstyp aus",
"MergeCellsAlert": "Beim Zusammenführen von Zellen wird nur der Wert ganz oben links (Oberste) beibehalten. Trotzdem zusammenführen?",
"PasteMergeAlert": "Wir können das nicht mit einer zusammengeführten Zelle machen.",
"Borders": "Grenzen",
"TopBorders": "Obere Grenzen",
"LeftBorders": "Linke Grenzen",
"RightBorders": "Rechte Grenzen",
"BottomBorders": "Untere Grenzen",
"AllBorders": "Alle Grenzen",
"HorizontalBorders": "Horizontale Grenzen",
"VerticalBorders": "Vertikale Grenzen",
"OutsideBorders": "Außerhalb der Grenzen",
"InsideBorders": "Innere Grenzen",
"NoBorders": "Keine Grenzen",
"BorderColor": "Randfarbe",
"BorderStyle": "Rahmenstil",
"InsertFunction": "Funktion einfügen",
"Insert": "Einfügung",
>Delete": "Löschen",
"DuplicateSheet": "Duplikat",
"MoveRight": "Nach rechts bewegen",
"MoveLeft": "Geh nach links",
"Rename": "Umbenennen",
"Hide": "Ausblenden",
"NameBox": "Namensfeld",
>ShowHeaders": "Kopfzeilen anzeigen",
"HideHeaders": "Kopfzeilen ausblenden",
>ShowGridLines": "Rasterlinien anzeigen",
"HideGridLines": "Gitternetzlinien ausblenden",
"FreezePanels": "Fenster einfrieren",
"FreezeRows": "Zeilen einfrieren",
"FreezeColumns": "Spalten einfrieren",
"UnfreezePanels": "Fenster freigeben",
"UnfreezeRows": "Zeilen aufheben",
"UnfreezeColumns": "Spalten freigeben",
"AddSheet": "Blatt hinzufügen",
>ListAllSheets": "Alle Blätter auflisten",
"CollapseToolbar": "Symbolleiste einklappen",

```

```

"ExpandToolbar": "Erweitern Sie die Symbolleiste",
"CollapseFormulaBar": "Formelleiste minimieren",
"ExpandFormulaBar": "Formelleiste erweitern",
"File": "Datei",
"Home": "Heim",
"Formulas": "Formeln",
"View": "Aussicht",
"New": "Neu",
"Open": "Offen",
"SaveAs": "Speichern als",
"ExcelXlsx": "Microsoft Excel",
"ExcelXls": "Microsoft Excel 97-2003",
"CSV": "Komma-getrennte Werte",
"FormulaBar": "Formelleiste",
"Sort": "Sortieren",
"SortAscending": "Aufsteigend",
"SortDescending": "Absteigend",
"CustomSort": "Benutzerdefinierte Sortierung",
"AddColumn": "Spalte hinzufügen",
"ContainsHeader": "Daten enthalten Header",
"CaseSensitive": "Groß- und Kleinschreibung beachten",
"SortBy": "Sortieren nach",
"ThenBy": "Dann vorbei",
"SelectAColumn": "Wählen Sie eine Spalte aus",
"SortEmptyFieldError": "Für alle Sortierkriterien muss eine Spalte
angegeben sein. Überprüfen Sie die ausgewählten Sortierkriterien und
versuchen Sie es erneut.",
"SortDuplicateFieldError": " mehr als einmal nach Werten sortiert
wird. Löschen Sie die doppelten Sortierkriterien und versuchen Sie es
erneut.",
"SortOutOfRangeError": "Wählen Sie eine Zelle oder einen Bereich
innerhalb des verwendeten Bereichs aus und versuchen Sie es erneut.",
"MultiRangeSortError": "Dies ist bei einer Mehrfachbereichsauswahl
nicht möglich. Wählen Sie einen einzelnen Bereich aus und versuchen Sie es
erneut.",
"HideRow": "Zeile ausblenden",
"HideRows": "Zeilen ausblenden",
"UnhideRows": "Zeilen einblenden",
"HideColumn": "Spalte ausblenden",
"HideColumns": "Spalten ausblenden",
"UnhideColumns": "Spalten einblenden",
"InsertRow": "Zeile einfügen",
"InsertRows": "Zeilen einfügen",
"Above": "Über",
"Below": "Unter",
"InsertColumn": "Spalte einfügen",
"InsertColumns": "Spalten einfügen",
"Before": "Vor",
"After": "Nach",
>DeleteRow": "Zeile löschen",
>DeleteRows": "Zeilen löschen",
>DeleteColumn": "Spalte löschen",
>DeleteColumns": "Spalten löschen",
"Ok": "OK",
"Cancel": "Absagen",
"Apply": "Sich bewerben",
"MoreColors": "Mehr Farben",

```

```

"StandardColors": "Standardfarben",
"General": "Allgemein",
"Number": "Nummer",
"Currency": "Währung",
"Accounting": "Buchhaltung",
"ShortDate": "Kurzes Date",
"LongDate": "Langes Datum",
"Time": "Zeit",
"Percentage": "Prozentsatz",
"Fraction": "Fraktion",
"Scientific": "Wissenschaftlich",
"Text": "Text",
"NumberFormat": "Zahlenformat",
"MobileFormulaBarPlaceholder": "Wert oder Formel eingeben",
"PasteAlert": "Sie können dies hier nicht einfügen, da der
Kopierbereich und der Einfügebereich nicht dieselbe Größe haben. Bitte
versuchen Sie es in einem anderen Bereich einzufügen.",
"DestroyAlert": "Möchten Sie die aktuelle Arbeitsmappe wirklich
löschen, ohne sie zu speichern, und eine neue Arbeitsmappe erstellen?",
"SheetRenameInvalidAlert": "Blattname enthält ungültige Zeichen.",
"SheetRenameEmptyAlert": "Blattname darf nicht leer sein.",
"SheetRenameAlreadyExistsAlert": "Blattname existiert bereits. Bitte
geben Sie einen anderen Namen ein.",
>DeleteSheetAlert": "Möchten Sie dieses Tabellenblatt wirklich
löschen?",
>DeleteSingleLastSheetAlert": "Eine Arbeitsmappe muss mindestens ein
sichtbares Arbeitsblatt enthalten.",
>PickACategory": "Wählen Sie eine Kategorie aus",
>Description": "Beschreibung",
>UnsupportedFile": "Nicht unterstützte Datei",
>DataLimitExceeded": "Die Dateidaten sind zu groß und die Verarbeitung
dauert länger. Möchten Sie fortfahren?",
>FileSizeLimitExceeded": "Die Datei ist zu groß und die Verarbeitung
dauert länger. Möchten Sie fortfahren?",
>InvalidUrl": "ungültige URL",
>SUM": "Fügt eine Reihe von Zahlen und/oder Zellen hinzu.",
>SUMIF": "Fügt die Zellen basierend auf der angegebenen Bedingung
hinzu.",
>SUMIFS": "Fügt die Zellen basierend auf angegebenen Bedingungen
hinzu.",
>ABS": "Gibt den Wert einer Zahl ohne Vorzeichen zurück.",
>RAND": "Gibt eine Zufallszahl zwischen 0 und 1 zurück.",
>RANDBETWEEN": "Gibt basierend auf angegebenen Werten eine zufällige
ganze Zahl zurück.",
>FLOOR": "Rundet eine Zahl auf das nächste Vielfache eines gegebenen
Faktors ab.",
>CEILING": "Rundet eine Zahl auf das nächste Vielfache eines gegebenen
Faktors auf.",
>PRODUCT": "Multipliziert eine Reihe von Zahlen und/oder Zellen.",
>AVERAGE": "Berechnet den Durchschnitt für die Reihe von Zahlen
und/oder Zellen ohne Text.",
>AVERAGEIF": "Berechnet den Durchschnitt für die Zellen basierend auf
dem angegebenen Kriterium.",
>AVERAGEIFS": "Berechnet den Durchschnitt für die Zellen basierend auf
den angegebenen Bedingungen.",
>AVERAGEA": "Berechnet den Durchschnitt für die Zellen, die TRUE als
1, Text und FALSE als 0 auswerten.",

```



```
"COUNT": "Zählt die Zellen, die numerische Werte in einem Bereich  
enthalten.",  
"COUNTIF": "Zählt die Zellen basierend auf der angegebenen  
Bedingung.",  
"COUNTIFS": "Zählt die Zellen basierend auf angegebenen Bedingungen.",  
"COUNTA": "Zählt die Zellen, die Werte in einem Bereich enthalten.",  
"MIN": "Gibt die kleinste Anzahl der angegebenen Argumente zurück.",  
"MAX": "Gibt die größte Anzahl der angegebenen Argumente zurück.",  
"DATE": "Gibt das Datum basierend auf dem angegebenen Jahr, Monat und  
Tag zurück.",  
"DAY": "Gibt den Tag aus dem angegebenen Datum zurück.",  
"DAYS": "Gibt die Anzahl der Tage zwischen zwei Daten zurück.",  
"IF": "Gibt einen Wert basierend auf dem angegebenen Ausdruck  
zurück.",  
"IFS": "Gibt einen Wert basierend auf den angegebenen mehreren  
Ausdrücken zurück.",  
"CalculateAND": "Gibt TRUE zurück, wenn alle Argumente TRUE sind,  
ansonsten FALSE.",  
"CalculateOR": "Gibt TRUE zurück, wenn eines der Argumente TRUE ist,  
ansonsten FALSE.",  
"IFERROR": "Gibt den Wert zurück, wenn kein Fehler gefunden wurde,  
sonst wird der angegebene Wert zurückgegeben.",  
"CHOOSE": "Gibt einen Wert aus einer Werteliste basierend auf der  
Indexnummer zurück.",  
"INDEX": "Gibt einen Wert der Zelle in einem bestimmten Bereich  
basierend auf der Zeilen- und Spaltennummer zurück.",  
"FIND": "Gibt die Position einer Zeichenfolge innerhalb einer anderen  
Zeichenfolge zurück, wobei die Groß-/Kleinschreibung beachtet wird.",  
"CONCATENATE": "Kombiniert zwei oder mehr Saiten miteinander.",  
"CONCAT": "Verkettet eine Liste oder einen Bereich von  
Textzeichenfolgen.",  
"SUBTOTAL": "Gibt die Zwischensumme für einen Bereich unter Verwendung  
der angegebenen Funktionsnummer zurück.",  
"RADIANS": "Konvertiert Grad in Bogenmaß.",  
"MATCH": "Gibt die relative Position eines angegebenen Werts im  
angegebenen Bereich zurück.",  
"SLOPE": "Gibt die Steigung der Linie aus der linearen Regression der  
Datenpunkte zurück.",  
"INTERCEPT": "Berechnet den Punkt der Y-Schnittlinie durch lineare  
Regression.",  
"UNIQUE": "Gibt eindeutige Werte aus einem Bereich oder Array zurück",  
"TEXT": "Konvertiert einen Wert in Text im angegebenen Zahlenformat.",  
"DefineNameExists": "Dieser Name existiert bereits, versuchen Sie es  
mit einem anderen Namen.",  
"CircularReference": "Wenn eine Formel auf einen oder mehrere  
Zirkelbezüge verweist, kann dies zu einer fehlerhaften Berechnung führen.",  
"SORT": "Sortiert einen Bereich eines Arrays",  
"T": "Überprüft, ob ein Wert Text ist oder nicht, und gibt den Text  
zurück.",  
"EXACT": "Überprüft, ob zwei Textstrings genau gleich sind und gibt  
TRUE oder FALSE zurück.",  
"LEN": "Gibt eine Anzahl von Zeichen in einer angegebenen Zeichenfolge  
zurück.",  
"MOD": "Gibt einen Rest zurück, nachdem eine Zahl durch den Divisor  
dividiert wurde.",  
"ODD": "Rundet eine positive Zahl auf und eine negative Zahl auf die  
nächste ungerade Ganzzahl ab.",
```

```

    "PI": "Gibt den Wert von PI zurück.",
    "COUNTBLANK": "Gibt die Anzahl leerer Zellen in einem angegebenen Zellbereich zurück.",
    "EVEN": "Rundet eine positive Zahl auf und eine negative Zahl auf die nächste gerade Ganzzahl ab.",
    "DECIMAL": "Konvertiert eine Textdarstellung einer Zahl mit einer bestimmten Basis in eine Dezimalzahl.",
    "ADDRESS": "Gibt einen Zellbezug als Text zurück, wenn die angegebenen Zeilen- und Spaltennummern angegeben sind.",
    "CHAR": "Gibt das Zeichen aus der angegebenen Zahl zurück.",
    "CODE": "Gibt den numerischen Code für das erste Zeichen in einer angegebenen Zeichenfolge zurück.",
    "DOLLAR": "Wandelt die Zahl in währungsformatierten Text um.",
    "SMALL": "Gibt den k-ten kleinsten Wert in einem gegebenen Array zurück.",
    "LARGE": "Gibt den k-größten Wert in einem gegebenen Array zurück.",
    "TIME": "Konvertiert Stunden, Minuten, Sekunden in den zeitformatierten Text.",
    "DEGREES": "Konvertiert Bogenmaß in Grad.",
    "FACT": "Gibt die Fakultät einer Zahl zurück.",
    "MEDIAN": "Gibt den Median der gegebenen Zahlenmenge zurück.",
    "EDATE": "Gibt ein Datum mit der angegebenen Anzahl von Monaten vor oder nach dem angegebenen Datum zurück.",
    "DATEVALUE": "Konvertiert eine Datumszeichenfolge in einen Datumswert.",
    "NOW": "Gibt das aktuelle Datum und die Uhrzeit zurück.",
    "HOUR": "Gibt die Anzahl der Stunden in einer angegebenen Zeitzeichenfolge zurück.",
    "MINUTE": "Gibt die Anzahl der Minuten in einer angegebenen Zeitzeichenfolge zurück.",
    "SECOND": "Gibt die Anzahl der Sekunden in einer angegebenen Zeitzeichenfolge zurück.",
    "MONTH": "Gibt die Anzahl der Monate in einer angegebenen Datumszeichenfolge zurück.",
    "OR": "ODER",
    "AND": "UND",
    "CustomFilterDatePlaceholder": "Wählen Sie ein Datum",
    "CustomFilterPlaceholder": "Geben Sie den Wert ein",
    "CustomFilter": "Benutzerdefinierte Filter",
    "Between": "Zwischen",
    "MatchCase": "Streichholzsachtel",
    "DateTimeFilter": "DateTime-Filter",
    "Undo": "Rückgängig machen",
    "Redo": "Wiederholen",
    "ClearAllFilter": "Klar",
    "ReapplyFilter": "Bewerben Sie sich erneut",
    "DateFilter": "Datumsfilter",
    "TextFilter": "Textfilter",
    "NumberFilter": "Zahlenfilter",
    "ClearFilter": "Filter löschen",
    "NoResult": "Keine Treffer gefunden",
    "FilterFalse": "FALSCH",
    "FilterTrue": "WAHR",
    "Blanks": "Leerzeichen",
    "SelectAll": "Wählen Sie Alle",
    "GreaterThanOrEqual": "Größer als oder gleich",
    "GreaterThan": "Größer als",

```

```

"LessThanOrEqual": "Weniger als oder gleich",
"LessThan": "Weniger als",
"NotEqual": "Nicht gleich",
"Equal": "Gleich",
"Contains": "Enthält",
"NotContains": "Enthält nicht",
"EndsWith": "Endet mit",
"NotEndsWith": "Endet nicht mit",
"StartsWith": "Beginnt mit",
"NotStartsWith": "Beginnt nicht mit",
"Like": "Wie",
"IsNull": "Null",
"NotNull": "Nicht null",
"IsEmpty": "Leer",
"IsNotEmpty": "Nicht leer",
"ClearButton": "Klar",
"FilterButton": "Filter",
"CancelButton": "Absagen",
"OKButton": "OK",
"Search": "Suche",
"DataValidation": "Datenvalidierung",
"CellRange": "Zellreichweite",
"Allow": "Erlauben",
>Data": "Daten",
"Minimum": "Minimum",
"Maximum": "Maximal",
"IgnoreBlank": "Leerzeichen ignorieren",
"WholeNumber": "Ganze Zahl",
"Decimal": "Dezimal",
>Date": "Datum",
"TextLength": "Textlänge",
>List": "Aufführen",
"NotBetween": "Nicht zwischen",
"EqualTo": "Gleicht",
"NotEqualTo": "Nicht gleichzusetzen mit",
"GreaterThanOrEqualTo": "Größer als oder gleich wie",
"LessThanOrEqualTo": "Weniger als oder gleich",
>InCellDropDown": "In-Cell-Dropdown",
>Sources": "Sources",
>Value": "Wert",
>Retry": "Wiederholen",
>DialogError": "Die Listenquelle muss ein Verweis auf eine einzelne
Zeile oder Spalte sein.",
>MinMaxError": "Das Maximum muss größer oder gleich dem Minimum
sein.",
>Spreadsheet": "Kalkulationstabelle",
>MoreValidation": "Diese Auswahl enthält mehr als eine Validierung. \n
Aktuelle Einstellungen löschen und fortfahren?",
>FileNameError": "Ein Dateiname darf keine Zeichen wie \\ / : * ? \" <
> [ ] |",
>ValidationError": "Dieser Wert entspricht nicht den für die Zelle
definierten Datenüberprüfungseinschränkungen.",
>ListLengthError": "Die Listenwerte erlauben nur bis zu 256 Zeichen",
>ProtectSheet": "Schutzblatt",
>UnprotectSheet": "Blattschutz aufheben",
>SelectCells": "Wählen Sie gesperrte Zellen aus",
>SelectUnlockedCells": "Wählen Sie entspernte Zellen aus",

```

```

"Save": "Speichern",
"EmptyFileName": "Der Dateiname darf nicht leer sein.",
"LargeName": "Der Name ist zu lang.",
"FormatCells": "Zellen formatieren",
"FormatRows": "Zeilen formatieren",
"FormatColumns": "Spalten formatieren",
"InsertLinks": "Verknüpfungen einfügen",
"ProtectContent": "Schützen Sie den Inhalt gesperrter Zellen",
"ProtectAllowUser": "Allen Benutzern dieses Arbeitsblatts Folgendes
erlauben:",
"EditAlert": "Die Zelle, die Sie ändern möchten, ist geschützt. Heben
Sie den Schutz des Blatts auf, um Änderungen vorzunehmen.",
"ClearValidation": "Klare Bestätigung",
"ISNUMBER": "Gibt true zurück, wenn der Wert als numerischer Wert
analysiert wird.",
"ROUND": "Rundet eine Zahl auf eine angegebene Anzahl von Ziffern.",
"GEOMEAN": "Gibt den geometrischen Mittelwert eines Arrays oder
Bereichs positiver Daten zurück.",
"POWER": "Gibt das Ergebnis einer potenzierten Zahl zurück",
"LOG": "Gibt den Logarithmus einer Zahl zur angegebenen Basis
zurück.",
"TRUNC": "Gibt den abgeschnittenen Wert einer Zahl auf eine angegebene
Anzahl von Dezimalstellen zurück.",
"EXP": "Gibt e potenziert mit der gegebenen Zahl zurück.",
"HighlightCellsRules": "Regeln für Zellen hervorheben",
"CFEqualTo": "Gleicht",
"TextThatContains": "Text, der enthält",
"ADateOccuring": "Ein auftretendes Datum",
"DuplicateValues": "Doppelte Werte",
"TopBottomRules": "Top/Bottom-Regeln",
"Top10Items": "Top 10 Artikel",
"Top10": "Top 10",
"Bottom10Items": "Unterste 10 Artikel",
"Bottom10": "Untere 10",
"AboveAverage": "Überdurchschnittlich",
"BelowAverage": "Unterdurchschnittlich",
"FormatCellsGreaterThan": "Zellen formatieren, die größer sind als:",
"FormatCellsLessThan": "Zellen formatieren, die kleiner sind als:",
"FormatCellsBetween": "Zellen formatieren zwischen:",
"FormatCellsEqualTo": "Zellen formatieren, die gleich sind:",
"FormatCellsThatContainTheText": "Zellen formatieren, die den Text
enthalten:",
"FormatCellsThatContainADateOccurring": "Formatieren Sie Zellen, die
ein Datum enthalten, das auftritt:",
"FormatCellsDuplicate": "Zellen formatieren, die Folgendes
enthalten:",
"FormatCellsTop": "Zellen formatieren, die ganz oben stehen:",
"FormatCellsBottom": "Zellen formatieren, die ganz unten rangieren:",
"FormatCellsAbove": "Überdurchschnittliche Zellen formatieren:",
"FormatCellsBelow": "Unterdurchschnittliche Zellen formatieren:",
"With": "mit",
"DataBars": "Datenleisten",
"ColorScales": "Farbskalen",
"IconSets": "Icon-Sets",
"ClearRules": "Klare Regeln",
"SelectedCells": "Löschen Sie Regeln aus ausgewählten Zellen",
"EntireSheet": "Klare Regeln aus dem gesamten Blatt",

```

```

"LightRedFillWithDarkRedText": "Hellrote Füllung mit dunkelrotem
Text",
"YellowFillWithDarkYellowText": "Gelbe Füllung mit dunkelgelbem Text",
"GreenFillWithDarkGreenText": "Grüne Füllung mit dunkelgrünem Text",
"RedFill": "Rote Füllung",
"RedText": "Roter Text",
"Duplicate": "Duplikat",
"Unique": "Einzigartig",
"And": "und",
"WebPage": "Website",
"ThisDocument": "Dieses Dokument",
"DisplayText": "Text anzeigen",
"Url": "URL",
"CellReference": "Zellbezug",
"DefinedNames": "Definierte Namen",
"EnterTheTextToDisplay": "Geben Sie den anzuzeigenden Text ein",
"EnterTheUrl": "Geben Sie die URL ein",
"INT": "Gibt eine Zahl bis zur nächsten ganzen Zahl zurück.",
"SUMPRODUCT": "Gibt die Summe des Produkts gegebener Bereiche von
Arrays zurück.",
"TODAY": "Gibt das aktuelle Datum als Datumswert zurück.",
"ROUNDUP": "Rundet eine Zahl von Null weg.",
"LOOKUP": "Sucht nach einem Wert in einem einzeiligen oder
einspaltigen Bereich und gibt dann einen Wert von derselben Position in
einem zweiten einzeiligen oder einspaltigen Bereich zurück.",
"HLOOKUP": "Sucht nach einem Wert in der obersten Zeile des
Wertearrays und gibt dann einen Wert in derselben Spalte aus einer von Ihnen
angegebenen Zeile im Array zurück.",
"VLOOKUP": "Sucht nach einem bestimmten Wert in der ersten Spalte
eines Suchbereichs und gibt einen entsprechenden Wert aus einer anderen
Spalte innerhalb derselben Zeile zurück.",
"NOT": "Gibt die Umkehrung eines bestimmten logischen Ausdrucks
zurück.",
"EOMONTH": "Gibt den letzten Tag des Monats zurück, der eine
angegebene Anzahl von Monaten vor oder nach einem ursprünglich angegebenen
Startdatum liegt.",
"SQRT": "Gibt die Quadratwurzel einer positiven Zahl zurück.",
"ROUNDDOWN": "Rundet eine Zahl ab, Richtung Null.",
"RSQ": "Gibt das Quadrat des Korrelationskoeffizienten des Pearson-
Produktmoments basierend auf Datenpunkten in bekannten y- und x-Werten
zurück.",
"Link": "Verknüpfung",
"Hyperlink": "Hyperlinks",
"EditHyperlink": "Hyperlink bearbeiten",
"OpenHyperlink": "Hyperlinks öffnen",
"RemoveHyperlink": "Hyperlinks entfernen",
"InvalidHyperlinkAlert": "Die Adresse dieser Website ist ungültig.
Überprüfen Sie die Adresse und versuchen Sie es erneut.",
"InsertLink": "Verknüpfung einfügen",
"EditLink": "Verknüpfung bearbeiten",
"WrapText": "Zeilenumbruch",
"Update": "Aktualisieren",
"SortAndFilter": "Sortieren & Filtern",
"Filter": "Filter",
"FilterCellValue": "Nach Wert der ausgewählten Zelle filtern",
"FilterOutOfRangeError": "Wählen Sie eine Zelle oder einen Bereich
innerhalb des verwendeten Bereichs aus und versuchen Sie es erneut.",

```

```

"ClearFilterFrom": "Filter löschen aus ",
"LN": "Gibt den natürlichen Logarithmus einer Zahl zurück.",
"DefineNameInvalid": "Der eingegebene Name ist ungültig.",
"EmptyError": "Sie müssen einen Wert eingeben",
"ClearHighlight": "Klare Hervorhebung",
"HighlightInvalidData": "Markieren Sie ungültige Daten",
"Clear": "Klar",
"ClearContents": "Klarer Inhalt",
"ClearAll": "Alles löschen",
"ClearFormats": "Klare Formate",
"ClearHyperlinks": "Hyperlinks löschen",
"Image": "Bild",
"ConditionalFormatting": "Bedingte Formatierung",
"BlueDataBar": "Blaue Datenleiste",
"GreenDataBar": "Grüne Datenleiste",
"RedDataBar": "Rote Datenleiste",
"OrangeDataBar": "Orange Datenleiste",
"LightBlueDataBar": "Hellblauer Datenbalken",
"PurpleDataBar": "Lila Datenleiste",
"GYRColorScale": "Farbskala Grün - Gelb - Rot",
"RYGColorScale": "Farbskala Rot - Gelb - Grün",
"GWRColorScale": "Farbskala Grün - Weiß - Rot",
"RWGColorScale": "Farbskala Rot - Weiß - Grün",
"BWRColorScale": "Farbskala Blau - Weiß - Rot",
"RWBColorScale": "Farbskala Rot - Weiß - Blau",
"WRColorScale": "Farbskala Weiß - Rot",
"RWColorScale": "Farbskala Rot - Weiß",
"GWColorScale": "Farbskala Grün - Weiß",
"WGColorScale": "Weiß - Grüne Farbskala",
"GYColorScale": "Farbskala Grün - Gelb",
"YGColorScale": "Farbskala Gelb - Grün",
"ThreeArrowsColor": "3 Pfeile (farbig)",
"ThreeArrowsGray": "3 Pfeile (grau)",
"ThreeTriangles": "3 Dreiecke",
"FourArrowsColor": "4 Pfeile (grau)",
"FourArrowsGray": "4 Pfeile (farbig)",
"FiveArrowsColor": "5 Pfeile (grau)",
"FiveArrowsGray": "5 Pfeile (farbig)",
"ThreeTrafficLights1": "3 Ampeln (ohne Rahmen)",
"ThreeTrafficLights2": "3 Ampeln (umrandet)",
"ThreeSigns": "3 Zeichen",
"FourTrafficLights": "4 Ampeln",
"RedToBlack": "Rot zu Schwarz",
"ThreeSymbols1": "3 Symbole (eingekreist)",
"ThreeSymbols2": "3 Symbole (nicht eingekreist)",
"ThreeFlags": "3 Fahnen",
"ThreeStars": "3 Sterne",
"FourRatings": "4 Bewertungen",
"FiveQuarters": "5 Quartale",
"FiveRatings": "5 Bewertungen",
"FiveBoxes": "5 Kisten",
"Chart": "Diagramm",
"Column": "Spalte",
"Bar": "Bar",
"Area": "Bereich",
"Pie": "Kuchen",
"Doughnut": "Krapfen",

```

```

"PieAndDoughnut": "Kuchen/Donut",
"Line": "Linie",
"Radar": "Radar",
"Scatter": "Streuen",
"ChartDesign": "Diagrammdesign",
"ClusteredColumn": "Gruppierte Spalte",
"StackedColumn": "Gestapelte Spalte",
"StackedColumn100": "100 % gestapelte Spalte",
"ClusteredBar": "Geclusterte Bar",
"StackedBar": "Gestapelte Leiste",
"StackedBar100": "100 % gestapelter Balken",
"StackedArea": "Gestapelter Bereich",
"StackedArea100": "100 % gestapelter Bereich",
"StackedLine": "Gestapelte Linie",
"StackedLine100": "100 % gestapelte Linie",
"LineMarker": "Linie mit Markierungen",
"StackedLineMarker": "Gestapelte Linie mit Markierungen",
"StackedLine100Marker": "100 % gestapelte Linie mit Markierungen",
"AddChartElement": "Diagrammelement hinzufügen",
"Axes": "Achsen",
"AxisTitle": "Achsentitel",
"ChartTitle": "Diagrammtitel",
"DataLabels": "Datenaufkleber",
"Gridlines": "Gitternetzlinien",
"Legends": "Legenden",
"PrimaryHorizontal": "Primäre Horizontale",
"PrimaryVertical": "Primäre Vertikale",
"None": "Keiner",
"AboveChart": "Oben Diagramm",
"Center": "Center",
"InsideEnd": "Inneres Ende",
"InsideBase": "Basis innen",
"OutsideEnd": "Äußeres Ende",
"PrimaryMajorHorizontal": "Grundlegende große horizontale",
"PrimaryMajorVertical": "Primäre Hauptvertikale",
"PrimaryMinorHorizontal": "Primäre Minor horizontal",
"PrimaryMinorVertical": "Primäre kleine Vertikale",
"Right": "Recht",
"Left": "Links",
"Bottom": "Unterseite",
"Top": "oben",
"SwitchRowColumn": "Zeile/Spalte wechseln",
"ChartTheme": "Diagrammthema",
"ChartType": "Diagramm Typ",
"Material": "Material",
"Fabric": "Stoff",
"Bootstrap": "Bootstrap",
"HighContrastLight": "Kontrastreiches Licht",
"MaterialDark": "Material dunkel",
"FabricDark": "Stoff dunkel",
"HighContrast": "Hoher Kontrast",
"BootstrapDark": "Bootstrap dunkel",
"Bootstrap4": "Bootstrap4",
"Bootstrap5Dark": "Bootstrap5 Dunkel",
"Bootstrap5": "Bootstrap5",
"Tailwind": "Rückenwind",
"TailwindDark": "Rückenwind Dunkel",

```

```

"VerticalAxisTitle": "Titel der vertikalen Achse",
"HorizontalAxisTitle": "Titel der horizontalen Achse",
"EnterTitle": "Titel eingeben",
"UnprotectWorksheet": "Blattschutz aufheben",
"ReEnterPassword": "Geben Sie das Passwort erneut ein, um
fortzufahren",
"SheetPassword": "Passwort zum Aufheben des Blattschutzes:",
"ProtectWorkbook": "Arbeitsmappe schützen",
"Password": "Passwort (optional):",
"EnterThePassword": "Geben Sie das Passwort ein",
"ConfirmPassword": "Passwort bestätigen",
"EnterTheConfirmPassword": "Wiederhole die Eingabe deines Passwortes",
"PasswordAlert": "Bestätigungspasswort ist nicht identisch",
"UnprotectWorkbook": "Schutz der Arbeitsmappe aufheben",
"UnprotectPasswordAlert": "Das eingegebene Passwort ist nicht
korrekt.",
"IncorrectPassword": "Die Datei oder das Arbeitsblatt kann mit dem
angegebenen Passwort nicht geöffnet werden",
"PasswordAlertMsg": "Bitte geben Sie das Passwort ein",
"ConfirmPasswordAlertMsg": "Bitte geben Sie das Bestätigungspasswort
ein",
"IsProtected": "ist geschützt",
"PDF": "PDF Document",
"AutoFillMergeAlertMsg": "Dazu müssen alle verbundenen Zellen dieselbe
Größe haben.",
"Fluent": "Fließend",
"FluentDark": "Fließendes Dunkel",
"Custom": "Brauch",
"WEEKDAY": "Gibt den Wochentag zurück, der einem Datum entspricht.",
"FillSeries": "Serie füllen",
"CopyCells": "Zellen kopieren",
"FillFormattingOnly": "Nur Formatierung ausfüllen",
"FillWithoutFormatting": "Ohne Formatierung ausfüllen",
"CustomFormat": "Benutzerdefinierte Zahlenformate",
"CustomFormatPlaceholder": "Geben Sie ein benutzerdefiniertes Format
ein oder wählen Sie es aus",
"CustomFormatTypeList": "Typ",
"CellReferenceTypoError": "Wir haben einen Tippfehler in Ihrem
Zellbezug gefunden. Möchten Sie diesen Verweis wie folgt korrigieren?",
"Close": "Schließen",
"MoreOptions": "Mehr Optionen",
"AddCurrentSelection": "Aktuelle Auswahl zum Filter hinzufügen",
"ExternalWorkbook": "Eine importierte Excel-Datei enthält eine externe
Arbeitsmappenreferenz. Möchten Sie diese Datei importieren?",
"Directional": "Gerichtet",
"Shapes": "Formen",
"Indicators": "Indikatoren",
"Ratings": "Bewertungen",
"Material3": "Material 3",
"Material3Dark": "Material 3 Dunkel",
"InvalidFormulaError": "Wir haben festgestellt, dass Sie eine
ungültige Formel eingegeben haben.",
"InvalidArguments": "Wir haben festgestellt, dass Sie eine Formel mit
ungültigen Argumenten eingegeben haben.",
"EmptyExpression": "Wir haben festgestellt, dass Sie eine Formel mit
einem leeren Ausdruck eingegeben haben.",

```



```

    "MismatchedParenthesis": "Wir haben festgestellt, dass Sie eine Formel
    eingegeben haben, bei der eine oder mehrere öffnende oder schließende
    Klammern fehlten.",
    "ImproperFormula": "Wir haben festgestellt, dass Sie eine Formel
    eingegeben haben, die falsch ist.",
    "WrongNumberOfArguments": "Wir haben festgestellt, dass Sie eine
    Formel mit einer falschen Anzahl von Argumenten eingegeben haben.",
    "Requires3Arguments": "Wir haben festgestellt, dass Sie eine Formel
    eingegeben haben, die drei Argumente erfordert.",
    "MismatchedStringQuotes": "Wir haben festgestellt, dass Sie eine
    Formel mit nicht übereinstimmenden Anführungszeichen eingegeben haben.",
    "FormulaCircularRef": "Wir haben festgestellt, dass Sie eine Formel
    mit einem Zirkelbezug eingegeben haben."
  }
}
}

```

### DATASOURCE.JSX

```

export let data = [
  {
    "Item Code": "I231",
    "Item Name": "Chinese Combo Noodle",
    "Quantity": 2,
    "Rate": 125,
    "Amount": "=PRODUCT(C2;D2) "
  },
  {
    "Item Code": "I245",
    "Item Name": "Chinese Combo Rice",
    "Quantity": 3,
    "Rate": 125,
    "Amount": "=PRODUCT(C3;D3) "
  },
  {
    "Item Code": "I237",
    "Item Name": "Amritsari Chola",
    "Quantity": 2,
    "Rate": 225,
    "Amount": "=PRODUCT(C4;D4) "
  },
  {
    "Item Code": "I291",
    "Item Name": "Asian Mixed Entree Platt",
    "Quantity": 3,
    "Rate": 165,
    "Amount": "=PRODUCT(C5;D5) "
  },
  {
    "Item Code": "I268",
    "Item Name": "Chinese Combo Chicken",
    "Quantity": 3,
    "Rate": 125,
    "Amount": "=PRODUCT(C6;D6) "
  },
  {

```

```

    "Item Code": "I251",
    "Item Name": "Chivas Regal",
    "Quantity": 1,
    "Rate": 325,
    "Amount": "=PRODUCT(C7;D7) "
  },
  {
    "Item Code": "I256",
    "Item Name": "Chicken Drumsticks",
    "Quantity": 2,
    "Rate": 180,
    "Amount": "=PRODUCT(C8;D8) "
  },
  {
    "Item Code": "I232",
    "Item Name": "Manchow Soup",
    "Quantity": 2,
    "Rate": 160,
    "Amount": "=PRODUCT(C9;D9) "
  },
  {
    "Item Code": "I290",
    "Item Name": "Schezuan Chicken",
    "Quantity": 3,
    "Rate": 180,
    "Amount": "=PRODUCT(C10;D10) "
  },
  {
    "Item Code": "I229",
    "Item Name": "Manchow Soup",
    "Quantity": 2,
    "Rate": 125,
    "Amount": "=PRODUCT(C11;D11) "
  },
  {
    "Item Code": "I239",
    "Item Name": "Jw Black Lable",
    "Quantity": 2,
    "Rate": 175,
    "Amount": "=PRODUCT(C12;D12) "
  }
];

```

**DATASOURCE.TSX**

```

export let data: Object[] = [
  {
    "Item Code": "I231",
    "Item Name": "Chinese Combo Noodle",
    "Quantity": 2,
    "Rate": 125,
    "Amount": "=PRODUCT(C2;D2) "
  },
  {
    "Item Code": "I245",
    "Item Name": "Chinese Combo Rice",

```

```

    "Quantity": 3,
    "Rate": 125,
    "Amount": "=PRODUCT(C3;D3) "
  },
  {
    "Item Code": "I237",
    "Item Name": "Amritsari Chola",
    "Quantity": 2,
    "Rate": 225,
    "Amount": "=PRODUCT(C4;D4) "
  },
  {
    "Item Code": "I291",
    "Item Name": "Asian Mixed Entree Platt",
    "Quantity": 3,
    "Rate": 165,
    "Amount": "=PRODUCT(C5;D5) "
  },
  {
    "Item Code": "I268",
    "Item Name": "Chinese Combo Chicken",
    "Quantity": 3,
    "Rate": 125,
    "Amount": "=PRODUCT(C6;D6) "
  },
  {
    "Item Code": "I251",
    "Item Name": "Chivas Regal",
    "Quantity": 1,
    "Rate": 325,
    "Amount": "=PRODUCT(C7;D7) "
  },
  {
    "Item Code": "I256",
    "Item Name": "Chicken Drumsticks",
    "Quantity": 2,
    "Rate": 180,
    "Amount": "=PRODUCT(C8;D8) "
  },
  {
    "Item Code": "I232",
    "Item Name": "Manchow Soup",
    "Quantity": 2,
    "Rate": 160,
    "Amount": "=PRODUCT(C9;D9) "
  },
  {
    "Item Code": "I290",
    "Item Name": "Schezuan Chicken",
    "Quantity": 3,
    "Rate": 180,
    "Amount": "=PRODUCT(C10;D10) "
  },
  {
    "Item Code": "I229",
    "Item Name": "Manchow Soup",
    "Quantity": 2,

```

```

      "Rate": 125,
      "Amount": "=PRODUCT(C11;D11) "
    },
    {
      "Item Code": "I239",
      "Item Name": "Jw Black Lable",
      "Quantity": 2,
      "Rate": 175,
      "Amount": "=PRODUCT(C12;D12) "
    }
  ];

```

### Create User Defined Functions / Custom Functions

The Spreadsheet includes a number of built-in formulas. For your convenience, a list of supported formulas can be found [here](#).

You can define and use an unsupported formula, i.e. a user defined/custom formula, in the spreadsheet by using the [addCustomFunction](#) function. Meanwhile, remember that you should define a user defined/custom formula whose results should only return a single value. If a user-defined/custom formula returns an array, it will be time-consuming to update adjacent cell values.

The following code example shows an unsupported formula in the spreadsheet.

### APP.JSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective, CellDirective, CellsDirective
} from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  const styles = { textAlign: 'center', fontWeight: 'bold', verticalAlign:
'middle', fontStyle: 'italic', fontSize: '15pt' };
  const cellStyle = { fontStyle: 'italic', fontWeight: 'bold' };
  const fontStyle = { fontWeight: 'bold', textAlign: 'right' };
  // Custom function to calculate percentage between two cell values.
  const calculatePercentage = (firstCell, secondCell) => {
    return Number(firstCell) / Number(secondCell);
  };
  // Custom function to calculate round down for values.
  const roundDownHandler = (value, digit) => {
    let multiplier = Math.pow(10, digit);
    return Math.floor(value * multiplier) / multiplier;
  }
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A2:F2');
      spreadsheet.numberFormat('$#,##0', 'B3:D12');
      spreadsheet.numberFormat('0%', 'E3:E12');
    }
  });
}

```

```

        // Adding custom function for calculating the percentage between
        two cells.
        spreadsheet.addCustomFunction(calculatePercentage,
        'PERCENTAGE');
        // Adding custom function for calculating round down for the
        value.
        spreadsheet.addCustomFunction(roundDownHandler, 'ROUNDDOWN');
        // Calculate percentage using custom added formula in E12 cell.
        spreadsheet.updateCell({ formula: '=PERCENTAGE(C12,D12)' },
        'E12');
        // Calculate round down for average values using custom added
        formula in F12 cell.
        spreadsheet.updateCell({ formula: '=ROUNDDOWN(F11,1)' }, 'F12');
    }
    }, []);
    return (<div>
        <SpreadsheetComponent ref={spreadsheetRef} showSheetTabs={false}
        showRibbon={false}>
            <SheetsDirective>
                <SheetDirective>
                    <RowsDirective>
                        <RowDirective height={40} customHeight={true}>
                            <CellsDirective>
                                <CellDirective value={'Monthly Expense'}
                                style={styles} colSpan={5}></CellDirective>
                            </CellsDirective>
                        </RowDirective>
                        <RowDirective height={30}></RowDirective>
                        <RowDirective index={11}>
                            <CellsDirective>
                                <CellDirective value={'Totals'}
                                style={cellStyle}></CellDirective>
                                <CellDirective formula={'=SUM(B3:B11)'}
                                ></CellDirective>
                                <CellDirective
                                formula={'=SUM(C3:C11)'}></CellDirective>
                                <CellDirective
                                formula={'=SUM(D3:D11)'}></CellDirective>
                            </CellsDirective>
                        </RowDirective>
                        <RowDirective>
                            <CellsDirective>
                                <CellDirective index={1} value={'Number of
                                Categories'} style={fontStyle} colSpan={2}></CellDirective>
                                <CellDirective formula={'=COUNTA(A3:A11)'}
                                index={3}></CellDirective>
                            </CellsDirective>
                        </RowDirective>
                        <RowDirective>
                            <CellsDirective>
                                <CellDirective index={1} value={'Average
                                Spend'} style={fontStyle} colSpan={2}></CellDirective>
                                <CellDirective formula={'=AVERAGE(B3:B11)'}
                                index={3} format={'$#,##0'}></CellDirective>
                            </CellsDirective>
                        </RowDirective>
                    </RowDirective>
                </SheetDirective>
            </SheetsDirective>
        </SpreadsheetComponent>
    </div>);

```

```

        <CellsDirective>
            <CellDirective index={1} value={'Min Spend'}
style={fontStyle} colSpan={2}></CellDirective>
            <CellDirective formula={"=MIN(B3:B11)"}
index={3} format={'$#,##0'}></CellDirective>
        </CellsDirective>
    </RowDirective>
</RowDirective>
    <CellsDirective>
        <CellDirective index={1} value={'Max Spend'}
style={fontStyle} colSpan={2}></CellDirective>
        <CellDirective formula={"=MAX(B3:B11)"}
index={3} format={'$#,##0'}></CellDirective>
    </CellsDirective>
</RowDirective>
</RowsDirective>
<RangesDirective>
    <RangeDirective dataSource={data}
startCell={"A2"}></RangeDirective>
</RangesDirective>
<ColumnsDirective>
    <ColumnDirective width={150}></ColumnDirective>
    <ColumnDirective width={120}></ColumnDirective>
    <ColumnDirective width={120}></ColumnDirective>
    <ColumnDirective width={120}></ColumnDirective>
    <ColumnDirective width={140}></ColumnDirective>
    <ColumnDirective width={150}></ColumnDirective>
</ColumnsDirective>
</SheetDirective>
</SheetsDirective>
</SpreadsheetComponent> </div>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

## APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective, CellDirective, CellsDirective
} from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { CellStyleModel } from '@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
    const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
    const styles: CellStyleModel = { textAlign: 'center', fontWeight:
'bold', verticalAlign: 'middle', fontStyle: 'italic', fontSize: '15pt' };
    const cellStyle: CellStyleModel = { fontStyle: 'italic', fontWeight:
'bold' };

```

```

    const fontStyle: CellStyleModel = { fontWeight: 'bold', textAlign:
    'right' };
    // Custom function to calculate percentage between two cell values.
    const calculatePercentage = (firstCell: string, secondCell: string):
    number => {
        return Number(firstCell) / Number(secondCell);
    };
    // Custom function to calculate round down for values.
    const roundDownHandler = (value: number, digit: number) => {
        let multiplier: number = Math.pow(10, digit);
        return Math.floor(value * multiplier) / multiplier;
    }
    React.useEffect(() => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
            }, 'A2:F2');
            spreadsheet.numberFormat('$#,##0', 'B3:D12');
            spreadsheet.numberFormat('0%', 'E3:E12');
            // Adding custom function for calculating the percentage between
            two cells.
            spreadsheet.addCustomFunction(calculatePercentage,
            'PERCENTAGE');
            // Adding custom function for calculating round down for the
            value.
            spreadsheet.addCustomFunction(roundDownHandler, 'ROUNDDOWN');
            // Calculate percentage using custom added formula in E12 cell.
            spreadsheet.updateCell({ formula: '=PERCENTAGE(C12,D12)' },
            'E12');
            // Calculate round down for average values using custom added
            formula in F12 cell.
            spreadsheet.updateCell({ formula: '=ROUNDDOWN(F11,1)' }, 'F12');
        }
    }, []);
    return (<div>
        <SpreadsheetComponent ref={spreadsheetRef} showSheetTabs={false}
        showRibbon={false}>
            <SheetsDirective>
                <SheetDirective>
                    <RowsDirective>
                        <RowDirective height={40} customHeight={true}>
                            <CellsDirective>
                                <CellDirective value={'Monthly Expense'}
                                style={styles} colSpan={5}></CellDirective>
                            </CellsDirective>
                        </RowDirective>
                        <RowDirective height={30}></RowDirective>
                        <RowDirective index={11}>
                            <CellsDirective>
                                <CellDirective value={'Totals'}
                                style={cellStyle}></CellDirective>
                                <CellDirective formula={'=SUM(B3:B11)'}
                                ></CellDirective>
                                <CellDirective
                                formula={'=SUM(C3:C11)'}></CellDirective>
                                <CellDirective
                                formula={'=SUM(D3:D11)'}></CellDirective>

```

```

        </CellsDirective>
      </RowDirective>
      <RowDirective>
        <CellsDirective>
          <CellDirective index={1} value={'Number of
Categories'} style={fontStyle} colSpan={2}></CellDirective>
          <CellDirective formula={'=COUNTA (A3:A11) '}
index={3}></CellDirective>
        </CellsDirective>
      </RowDirective>
      <RowDirective>
        <CellsDirective>
          <CellDirective index={1} value={'Average
Spend'} style={fontStyle} colSpan={2}></CellDirective>
          <CellDirective formula={'=AVERAGE (B3:B11) '}
index={3} format={'$#,##0'}></CellDirective>
        </CellsDirective>
      </RowDirective>
      <RowDirective>
        <CellsDirective>
          <CellDirective index={1} value={'Min Spend'}
style={fontStyle} colSpan={2}></CellDirective>
          <CellDirective formula={'=MIN (B3:B11) '}
index={3} format={'$#,##0'}></CellDirective>
        </CellsDirective>
      </RowDirective>
      <RowDirective>
        <CellsDirective>
          <CellDirective index={1} value={'Max Spend'}
style={fontStyle} colSpan={2}></CellDirective>
          <CellDirective formula={'=MAX (B3:B11) '}
index={3} format={'$#,##0'}></CellDirective>
        </CellsDirective>
      </RowDirective>
    </RowsDirective>
    <RangesDirective>
      <RangeDirective dataSource={data}
startCell={'A2'}></RangeDirective>
    </RangesDirective>
    <ColumnsDirective>
      <ColumnDirective width={150}></ColumnDirective>
      <ColumnDirective width={120}></ColumnDirective>
      <ColumnDirective width={120}></ColumnDirective>
      <ColumnDirective width={120}></ColumnDirective>
      <ColumnDirective width={140}></ColumnDirective>
      <ColumnDirective width={150}></ColumnDirective>
    </ColumnsDirective>
  </SheetDirective>
</SheetsDirective>
</SpreadsheetComponent> </div>
);
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```



**DATASOURCE.JSX**

```

/**
 * Formula data source
 */
export let data = [
  {
    'Category': 'Household Utilities',
    'Monthly Spend': '=C3/12',
    'Annual Spend': 3000,
    'Last Year Spend': 3000,
    'Percentage Change': '=C3/D3', // You can set the expression or
    formula as string
    'Average Change': '=7.9/E3',
  },
  {
    'Category': 'Food',
    'Monthly Spend': '=C4/12',
    'Annual Spend': 2500,
    'Last Year Spend': 2250,
    'Percentage Change': { formula: '=C4/D4' }, // You can also set as
    object with formula field
    'Average Change': '=7.9/E4',
  },
  {
    'Category': 'Gasoline',
    'Monthly Spend': '=C5/12',
    'Annual Spend': 1500,
    'Last Year Spend': 1200,
    'Percentage Change': { formula: '=C5/D5' },
    'Average Change': '=7.9/E5',
  },
  {
    'Category': 'Clothes',
    'Monthly Spend': '=C6/12',
    'Annual Spend': 1200,
    'Last Year Spend': 1000,
    'Percentage Change': '=C6/D6',
    'Average Change': '=7.9/E6',
  },
  {
    'Category': 'Insurance',
    'Monthly Spend': '=C7/12',
    'Annual Spend': 1500,
    'Last Year Spend': 1500,
    'Percentage Change': '=C7/D7',
    'Average Change': '=7.9/E7',
  },
  {
    'Category': 'Taxes',
    'Monthly Spend': '=C8/12',
    'Annual Spend': 3500,
    'Last Year Spend': 3500,
    'Percentage Change': '=C8/D8',
    'Average Change': '=7.9/E8',
  },
]

```

```

    },
    {
      'Category': 'Entertainment',
      'Monthly Spend': '=C9/12',
      'Annual Spend': 2000,
      'Last Year Spend': 2250,
      'Percentage Change': '=C9/D9',
      'Average Change': '=7.9/E9',
    },
    {
      'Category': 'Vacation',
      'Monthly Spend': '=C10/12',
      'Annual Spend': 1500,
      'Last Year Spend': 2000,
      'Percentage Change': '=C10/D10',
      'Average Change': '=7.9/E10',
    },
    {
      'Category': 'Miscellaneous',
      'Monthly Spend': '=C11/12',
      'Annual Spend': 1250,
      'Last Year Spend': 1558,
      'Percentage Change': '=C11/D11',
      'Average Change': '=7.9/E11',
    }
  ]
};

```

### DATASOURCE.TSX

```

/**
 * Formula data source
 */
export let data: Object[] = [
  {
    'Category': 'Household Utilities',
    'Monthly Spend': '=C3/12', // Setting formula through data binding
    'Annual Spend': 3000,
    'Last Year Spend': 3000,
    'Percentage Change': '=C3/D3', // You can set the expression or
    formula as string
    'Average Change': '=7.9/E3',
  },
  {
    'Category': 'Food',
    'Monthly Spend': '=C4/12',
    'Annual Spend': 2500,
    'Last Year Spend': 2250,
    'Percentage Change': { formula: '=C4/D4' }, // You can also set as
    object with formula field
    'Average Change': '=7.9/E4',
  },
  {
    'Category': 'Gasoline',
    'Monthly Spend': '=C5/12',
    'Annual Spend': 1500,
    'Last Year Spend': 1200,
  }
];

```

```

    'Percentage Change': { formula: '=C5/D5' },
    'Average Change': '=7.9/E5',
  },
  {
    'Category': 'Clothes',
    'Monthly Spend': '=C6/12',
    'Annual Spend': 1200,
    'Last Year Spend': 1000,
    'Percentage Change': '=C6/D6',
    'Average Change': '=7.9/E6',
  },
  {
    'Category': 'Insurance',
    'Monthly Spend': '=C7/12',
    'Annual Spend': 1500,
    'Last Year Spend': 1500,
    'Percentage Change': '=C7/D7',
    'Average Change': '=7.9/E7',
  },
  {
    'Category': 'Taxes',
    'Monthly Spend': '=C8/12',
    'Annual Spend': 3500,
    'Last Year Spend': 3500,
    'Percentage Change': '=C8/D8',
    'Average Change': '=7.9/E8',
  },
  {
    'Category': 'Entertainment',
    'Monthly Spend': '=C9/12',
    'Annual Spend': 2000,
    'Last Year Spend': 2250,
    'Percentage Change': '=C9/D9',
    'Average Change': '=7.9/E9',
  },
  {
    'Category': 'Vacation',
    'Monthly Spend': '=C10/12',
    'Annual Spend': 1500,
    'Last Year Spend': 2000,
    'Percentage Change': '=C10/D10',
    'Average Change': '=7.9/E10',
  },
  {
    'Category': 'Miscellaneous',
    'Monthly Spend': '=C11/12',
    'Annual Spend': 1250,
    'Last Year Spend': 1558,
    'Percentage Change': '=C11/D11',
    'Average Change': '=7.9/E11',
  },
],
];

```

Second, if you want to directly compute any formula or expression, you can use the [computeExpression](#) method. This method will work for both built-in and used-defined/custom formula.

The following code example shows how to use `computeExpression` method in the spreadsheet.

#### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective, CellDirective, CellsDirective
} from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
    const spreadsheetRef = React.useRef(null);
    const styles = { textAlign: 'center', fontWeight: 'bold', verticalAlign:
'middle', fontStyle: 'italic', fontSize: '15pt' };
    const cellStyle = { fontStyle: 'italic', fontWeight: 'bold' };
    const fontStyle = { fontWeight: 'bold', textAlign: 'right' };
    // Custom function to calculate percentage between two cell values.
    const calculatePercentage = (firstCell, secondCell) => {
        return Number(firstCell) / Number(secondCell);
    };
    React.useEffect(() => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A2:E2');
            spreadsheet.numberFormat('$#,##0', 'B3:D12');
            spreadsheet.numberFormat('0%', 'E3:E12');
            // Adding custom function for calculating the percentage between
two cells.
            spreadsheet.addCustomFunction(calculatePercentage,
'PERCENTAGE');
            // Calculate percentage using custom added formula in E11 cell.
            spreadsheet.updateCell({ formula: '=PERCENTAGE(C11,D11)' },
'E11');
            // Calculate expressions using computeExpression in E10 cell.
            spreadsheet.updateCell({ value:
spreadsheet.computeExpression('C10/D10') }, 'E10');
            // Calculate custom formula values using computeExpression in
E12 cell.
            spreadsheet.updateCell({ value:
spreadsheet.computeExpression('=PERCENTAGE(C12,D12)') }, 'E12');
            // Calculate SUM (built-in) formula values using
computeExpression in D12 cell.
            spreadsheet.updateCell({ value:
spreadsheet.computeExpression('=SUM(D3:D11)') }, 'D12');
        }
    }, []);
    return (<div>
        <SpreadsheetComponent ref={spreadsheetRef} showSheetTabs={false}
showRibbon={false}>
            <SheetsDirective>
                <SheetDirective>
                    <RowsDirective>
                        <RowDirective height={40} customHeight={true}>
```

```

        <CellsDirective>
          <CellDirective value={'Monthly Expense'}
style={styles} colSpan={5}></CellDirective>
        </CellsDirective>
      </RowDirective>
      <RowDirective height={30}></RowDirective>
      <RowDirective index={11}>
        <CellsDirective>
          <CellDirective value={'Totals'}
style={cellStyle}></CellDirective>
          <CellDirective formula={'=SUM(B3:B11)'}
></CellDirective>
          <CellDirective
formula={'=SUM(C3:C11)'}></CellDirective>
          <CellDirective
formula={'=SUM(D3:D11)'}></CellDirective>
        </CellsDirective>
      </RowDirective>
      <RowDirective>
        <CellsDirective>
          <CellDirective index={1} value={'Number of
Categories'} style={fontStyle} colSpan={2}></CellDirective>
          <CellDirective formula={'=COUNTA(A3:A11)'}
index={3}></CellDirective>
        </CellsDirective>
      </RowDirective>
      <RowDirective>
        <CellsDirective>
          <CellDirective index={1} value={'Average
Spend'} style={fontStyle} colSpan={2}></CellDirective>
          <CellDirective formula={'=AVERAGE(B3:B11)'}
index={3} format={'$#,##0'}></CellDirective>
        </CellsDirective>
      </RowDirective>
      <RowDirective>
        <CellsDirective>
          <CellDirective index={1} value={'Min Spend'}
style={fontStyle} colSpan={2}></CellDirective>
          <CellDirective formula={'=MIN(B3:B11)'}
index={3} format={'$#,##0'}></CellDirective>
        </CellsDirective>
      </RowDirective>
      <RowDirective>
        <CellsDirective>
          <CellDirective index={1} value={'Max Spend'}
style={fontStyle} colSpan={2}></CellDirective>
          <CellDirective formula={'=MAX(B3:B11)'}
index={3} format={'$#,##0'}></CellDirective>
        </CellsDirective>
      </RowDirective>
    </RowsDirective>
    <RangesDirective>
      <RangeDirective dataSource={data}
startCell={"A2"}></RangeDirective>
    </RangesDirective>
    <ColumnsDirective>
      <ColumnDirective width={150}></ColumnDirective>
    </ColumnsDirective>
  </Table>

```

```

        <ColumnDirective width={120}></ColumnDirective>
        <ColumnDirective width={120}></ColumnDirective>
        <ColumnDirective width={120}></ColumnDirective>
        <ColumnDirective width={120}></ColumnDirective>
        <ColumnDirective width={120}></ColumnDirective>
      </ColumnsDirective>
    </SheetDirective>
  </SheetsDirective>
</SpreadsheetComponent> </div>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective, CellDirective, CellsDirective
} from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { CellStyleModel } from '@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  const styles: CellStyleModel = { textAlign: 'center', fontWeight:
'bold', verticalAlign: 'middle', fontStyle: 'italic', fontSize: '15pt' };
  const cellStyle: CellStyleModel = { fontStyle: 'italic', fontWeight:
'bold' };
  const fontStyle: CellStyleModel = { fontWeight: 'bold', textAlign:
'right' };
  // Custom function to calculate percentage between two cell values.
  const calculatePercentage = (firstCell: string, secondCell: string):
number => {
    return Number(firstCell) / Number(secondCell);
  };
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A2:E2');
      spreadsheet.numberFormat('$#,##0', 'B3:D12');
      spreadsheet.numberFormat('0%', 'E3:E12');
      // Adding custom function for calculating the percentage between
two cells.
      spreadsheet.addCustomFunction(calculatePercentage,
'PERCENTAGE');
      // Calculate percentage using custom added formula in E11 cell.
      spreadsheet.updateCell({ formula: '=PERCENTAGE(C11,D11)' },
'E11');
      // Calculate expressions using computeExpression in E10 cell.

```

```

        spreadsheet.updateCell({ value:
spreadsheet.computeExpression('C10/D10') as string }, 'E10');
        // Calculate custom formula values using computeExpression in
E12 cell.
        spreadsheet.updateCell({ value:
spreadsheet.computeExpression('=PERCENTAGE(C12,D12)') as string }, 'E12');
        // Calculate SUM (built-in) formula values using
computeExpression in D12 cell.
        spreadsheet.updateCell({ value:
spreadsheet.computeExpression('=SUM(D3:D11)') as string }, 'D12');
    }, []);
    return (<div>
        <SpreadsheetComponent ref={spreadsheetRef} showSheetTabs={false}
showRibbon={false}>
            <SheetsDirective>
                <SheetDirective>
                    <RowsDirective>
                        <RowDirective height={40} customHeight={true}>
                            <CellsDirective>
                                <CellDirective value={'Monthly Expense'}
style={styles} colSpan={5}></CellDirective>
                            </CellsDirective>
                        </RowDirective>
                        <RowDirective height={30}></RowDirective>
                        <RowDirective index={11}>
                            <CellsDirective>
                                <CellDirective value={'Totals'}
style={cellStyle}></CellDirective>
                                <CellDirective formula={'=SUM(B3:B11)'}
></CellDirective>
                                <CellDirective
formula={'=SUM(C3:C11)'}></CellDirective>
                                <CellDirective
formula={'=SUM(D3:D11)'}></CellDirective>
                            </CellsDirective>
                        </RowDirective>
                        <RowDirective>
                            <CellsDirective>
                                <CellDirective index={1} value={'Number of
Categories'} style={fontStyle} colSpan={2}></CellDirective>
                                <CellDirective formula={'=COUNTA(A3:A11)'}
index={3}></CellDirective>
                            </CellsDirective>
                        </RowDirective>
                        <RowDirective>
                            <CellsDirective>
                                <CellDirective index={1} value={'Average
Spend'} style={fontStyle} colSpan={2}></CellDirective>
                                <CellDirective formula={'=AVERAGE(B3:B11)'}
index={3} format={'$#,##0'}></CellDirective>
                            </CellsDirective>
                        </RowDirective>
                        <RowDirective>
                            <CellsDirective>
                                <CellDirective index={1} value={'Min Spend'}
style={fontStyle} colSpan={2}></CellDirective>

```

```

                                <CellDirective formula={"=MIN (B3:B11) "}
index={3} format={'$#,##0'}></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective>
                                    <CellsDirective>
                                        <CellDirective index={1} value={'Max Spend'}
style={fontStyle} colSpan={2}></CellDirective>
                                        <CellDirective formula={"=MAX (B3:B11) "}
index={3} format={'$#,##0'}></CellDirective>
                                        </CellsDirective>
                                    </RowDirective>
                                </RowsDirective>
                                <RangesDirective>
                                    <RangeDirective dataSource={data}
startCell={"A2"}></RangeDirective>
                                </RangesDirective>
                                <ColumnsDirective>
                                    <ColumnDirective width={150}></ColumnDirective>
                                    <ColumnDirective width={120}></ColumnDirective>
                                    <ColumnDirective width={120}></ColumnDirective>
                                    <ColumnDirective width={120}></ColumnDirective>
                                    <ColumnDirective width={120}></ColumnDirective>
                                    <ColumnDirective width={120}></ColumnDirective>
                                </ColumnsDirective>
                            </SheetDirective>
                        </SheetsDirective>
                    </SpreadsheetComponent> </div>
                );
    };
    export default App;
    const root = createRoot(document.getElementById('root')!);
    root.render(<App />);
    {% endraw %}

```

### DATASOURCE.JSX

```

/**
 * Formula data source
 */
export let data = [
    {
        'Category': 'Household Utilities',
        'Monthly Spend': '=C3/12',
        'Annual Spend': 3000,
        'Last Year Spend': 3000,
        'Percentage Change': '=C3/D3' // You can set the expression or
        formula as string
    },
    {
        'Category': 'Food',
        'Monthly Spend': '=C4/12',
        'Annual Spend': 2500,
        'Last Year Spend': 2250,
        'Percentage Change': { formula: '=C4/D4' } // You can also set as
        object with formula field
    }
]

```



```

    },
    {
      'Category': 'Gasoline',
      'Monthly Spend': '=C5/12',
      'Annual Spend': 1500,
      'Last Year Spend': 1200,
      'Percentage Change': { formula: '=C5/D5' }
    },
    {
      'Category': 'Clothes',
      'Monthly Spend': '=C6/12',
      'Annual Spend': 1200,
      'Last Year Spend': 1000,
      'Percentage Change': '=C6/D6'
    },
    {
      'Category': 'Insurance',
      'Monthly Spend': '=C7/12',
      'Annual Spend': 1500,
      'Last Year Spend': 1500,
      'Percentage Change': '=C7/D7'
    },
    {
      'Category': 'Taxes',
      'Monthly Spend': '=C8/12',
      'Annual Spend': 3500,
      'Last Year Spend': 3500,
      'Percentage Change': '=C8/D8'
    },
    {
      'Category': 'Entertainment',
      'Monthly Spend': '=C9/12',
      'Annual Spend': 2000,
      'Last Year Spend': 2250,
      'Percentage Change': '=C9/D9'
    },
    {
      'Category': 'Vacation',
      'Monthly Spend': '=C10/12',
      'Annual Spend': 1500,
      'Last Year Spend': 2000,
      'Percentage Change': '=C10/D10'
    },
    {
      'Category': 'Miscellaneous',
      'Monthly Spend': '=C11/12',
      'Annual Spend': 1250,
      'Last Year Spend': 1558,
      'Percentage Change': '=C11/D11'
    }
  ];

```

**DATASOURCE.TSX**

```

/**
 * Formula data source

```

```

*/
export let data: Object[] = [
  {
    'Category': 'Household Utilities',
    'Monthly Spend': '=C3/12', // Setting formula through data binding
    'Annual Spend': 3000,
    'Last Year Spend': 3000,
    'Percentage Change': '=C3/D3' // You can set the expression or
    formula as string
  },
  {
    'Category': 'Food',
    'Monthly Spend': '=C4/12',
    'Annual Spend': 2500,
    'Last Year Spend': 2250,
    'Percentage Change': { formula: '=C4/D4' } // You can also set as
    object with formula field
  },
  {
    'Category': 'Gasoline',
    'Monthly Spend': '=C5/12',
    'Annual Spend': 1500,
    'Last Year Spend': 1200,
    'Percentage Change': { formula: '=C5/D5' }
  },
  {
    'Category': 'Clothes',
    'Monthly Spend': '=C6/12',
    'Annual Spend': 1200,
    'Last Year Spend': 1000,
    'Percentage Change': '=C6/D6'
  },
  {
    'Category': 'Insurance',
    'Monthly Spend': '=C7/12',
    'Annual Spend': 1500,
    'Last Year Spend': 1500,
    'Percentage Change': '=C7/D7'
  },
  {
    'Category': 'Taxes',
    'Monthly Spend': '=C8/12',
    'Annual Spend': 3500,
    'Last Year Spend': 3500,
    'Percentage Change': '=C8/D8'
  },
  {
    'Category': 'Entertainment',
    'Monthly Spend': '=C9/12',
    'Annual Spend': 2000,
    'Last Year Spend': 2250,
    'Percentage Change': '=C9/D9'
  },
  {
    'Category': 'Vacation',
    'Monthly Spend': '=C10/12',
    'Annual Spend': 1500,

```

```

      'Last Year Spend': 2000,
      'Percentage Change': '=C10/D10'
    },
    {
      'Category': 'Miscellaneous',
      'Monthly Spend': '=C11/12',
      'Annual Spend': 1250,
      'Last Year Spend': 1558,
      'Percentage Change': '=C11/D11'
    }
  ],
];

```

### Formula bar

Formula bar is used to edit or enter cell data in much easier way. By default, the formula bar is enabled in the spreadsheet. Use the [showFormulaBar](#) property to enable or disable the formula bar.

### Named Ranges

You can define a meaningful name for a cell range and use it in the formula for calculation. It makes your formula much easier to understand and maintain. You can add named ranges to the Spreadsheet in the following ways,

- Using the [definedNames](#) collection, you can add multiple named ranges at initial load.
- Use the [addDefinedName](#) method to add a named range dynamically.
- You can remove an added named range dynamically using the [removeDefinedName](#) method.
- Select the range of cells, and then enter the name for the selected range in the name box.

The following code example shows the usage of named ranges support.

### APP.JSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective, CellDirective, CellsDirective
} from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective,
DefinedNamesDirective, DefinedNameDirective } from '@syncfusion/ej2-react-
spreadsheet';
import { getComponent } from '@syncfusion/ej2-base';
import { data } from '../datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  const styles = { textAlign: 'center', fontWeight: 'bold', verticalAlign:
'middle', fontStyle: 'italic', fontSize: '15pt' };
  const cellStyle = { fontStyle: 'italic', fontWeight: 'bold' };
  const fontStyle = { fontWeight: 'bold', textAlign: 'right' };
  const beforeDataBound = () => {
    let spreadsheetObj =
getComponent(document.getElementById("spreadsheet"), "spreadsheet");
    // Adding name dynamically for `last year spending` and `percentage
change` ranges.
    spreadsheetObj.addDefinedName({ name: 'LastYearSpending', refersTo:
'='D3:D11' });
  };

```

```

    spreadsheetObj.addDefinedName({ name: 'PercentageChange', refersTo:
    '=E3:E11' });
  };
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      // Removing the unwanted `PercentageChange` named range
      spreadsheet.removeDefinedName('PercentageChange', '');
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
    }, 'A2:E2');
      spreadsheet.numberFormat('$#,##0', 'B3:D12');
      spreadsheet.numberFormat('0%', 'E3:E12');
    }
  }, []);
  return (
    <div>
      <SpreadsheetComponent id='spreadsheet' ref={spreadsheetRef}
beforeDataBound={beforeDataBound} showSheetTabs={false} showRibbon={false}>
        <SheetsDirective>
          <SheetDirective name={"Budget Details"}>
            <RowsDirective>
              <RowDirective height={40} customHeight={true}>
                <CellsDirective>
                  <CellDirective value={'Monthly Expense'}
style={styles} colSpan={5}></CellDirective>
                </CellsDirective>
              </RowDirective>
              <RowDirective height={30}></RowDirective>
              <RowDirective index={11}>
                <CellsDirective>
                  <CellDirective value={'Totals'}
style={cellStyle}></CellDirective>
                  <CellDirective
formula={'=SUM(MonthlySpending)'} ></CellDirective>
                  <CellDirective
formula={'=SUM(AnnualSpending)'} ></CellDirective>
                  <CellDirective
formula={'=SUM(LastYearSpending)'} ></CellDirective>
                  <CellDirective
formula={'=C12/D12'} ></CellDirective>
                </CellsDirective>
              </RowDirective>
              <RowDirective>
                <CellsDirective>
                  <CellDirective index={1} value={'Number
of Categories'} style={fontStyle} colSpan={2}></CellDirective>
                  <CellDirective
formula={'=COUNTA(Categories)'} index={3}></CellDirective>
                </CellsDirective>
              </RowDirective>
              <RowDirective>
                <CellsDirective>
                  <CellDirective index={1} value={'Average
Spend'} style={fontStyle} colSpan={2}></CellDirective>
                  <CellDirective
formula={'=AVERAGE(MonthlySpending)'} index={3}
format={'$#,##0'} ></CellDirective>

```

```

        </CellsDirective>
      </RowDirective>
    </RowDirective>
    <CellsDirective>
      <CellDirective index={1} value={'Min
Spend'} style={fontStyle} colSpan={2}></CellDirective>
      <CellDirective
formula={"=MIN(MonthlySpending)" } index={3}
format={'$#,##0'}></CellDirective>
    </CellsDirective>
  </RowDirective>
  </RowDirective>
  <CellsDirective>
    <CellDirective index={1} value={'Max
Spend'} style={fontStyle} colSpan={2}></CellDirective>
    <CellDirective
formula={"=MAX(MonthlySpending)" } index={3}
format={'$#,##0'}></CellDirective>
  </CellsDirective>
</RowDirective>
</RowsDirective>
<RangesDirective>
  <RangeDirective dataSource={data}
startCell={"A2"}></RangeDirective>
</RangesDirective>
<ColumnsDirective>
  <ColumnDirective width={150}></ColumnDirective>
  <ColumnDirective width={120}></ColumnDirective>
  <ColumnDirective width={120}></ColumnDirective>
  <ColumnDirective width={120}></ColumnDirective>
  <ColumnDirective width={120}></ColumnDirective>
</ColumnsDirective>
</SheetDirective>
</SheetsDirective>
<DefinedNamesDirective>
  <DefinedNameDirective name={'Categories'}
refersTo={"=Budget Details!A3:A11"}> </DefinedNameDirective>
  <DefinedNameDirective name={'MonthlySpending'}
refersTo={"=Budget Details!B3:B11"}> </DefinedNameDirective>
  <DefinedNameDirective name={'AnnualSpending'}
refersTo={"=Budget Details!C3:C11"}> </DefinedNameDirective>
</DefinedNamesDirective>
</SpreadsheetComponent>
</div>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

**APP.TSX**

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';

```

```

import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective, CellDirective, CellsDirective
} from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective, CellStyleModel,
DefinedNamesDirective, DefinedNameDirective } from '@syncfusion/ej2-react-
spreadsheet';
import { getComponent } from '@syncfusion/ej2-base';
import { data } from '../datasource';
function App() {
    const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
    const styles: CellStyleModel = { textAlign: 'center', fontWeight:
'bold', verticalAlign: 'middle', fontStyle: 'italic', fontSize: '15pt' };
    const cellStyle: CellStyleModel = { fontStyle: 'italic', fontWeight:
'bold' };
    const fontStyle: CellStyleModel = { fontWeight: 'bold', textAlign:
'right' };
    const beforeDataBound = (): void => {
        let spreadsheetObj: SpreadsheetComponent =
getComponent(document.getElementById("spreadsheet") as HTMLElement,
"spreadsheet");
        // Adding name dynamically for `last year spending` and `percentage
change` ranges.
        spreadsheetObj.addDefinedName({ name: 'LastYearSpending', refersTo:
'=D3:D11' });
        spreadsheetObj.addDefinedName({ name: 'PercentageChange', refersTo:
'=E3:E11' });
    };
    React.useEffect(() => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            // Removing the unwanted `PercentageChange` named range
            spreadsheet.removeDefinedName('PercentageChange', '');
            spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A2:E2');
            spreadsheet.numberFormat('$#,##0', 'B3:D12');
            spreadsheet.numberFormat('0%', 'E3:E12');
        }
    }, []);
    return (
        <div>
            <SpreadsheetComponent id='spreadsheet' ref={spreadsheetRef}
beforeDataBound={beforeDataBound} showSheetTabs={false} showRibbon={false}>
                <SheetsDirective>
                    <SheetDirective name={"Budget Details"}>
                        <RowsDirective>
                            <RowDirective height={40} customHeight={true}>
                                <CellsDirective>
                                    <CellDirective value={'Monthly Expense'}
style={styles} colSpan={5}></CellDirective>
                                </CellsDirective>
                            </RowDirective>
                            <RowDirective height={30}></RowDirective>
                            <RowDirective index={11}>
                                <CellsDirective>
                                    <CellDirective value={'Totals'}
style={cellStyle}></CellDirective>

```

```

                                <CellDirective
formula={ '=SUM(MonthlySpending)' } ></CellDirective>
                                <CellDirective
formula={ '=SUM(AnnualSpending)' }></CellDirective>
                                <CellDirective
formula={ '=SUM(LastYearSpending)' }></CellDirective>
                                <CellDirective
formula={ '=C12/D12' }></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective>
                                <CellsDirective>
                                <CellDirective index={1} value={ 'Number
of Categories' } style={fontStyle} colSpan={2}></CellDirective>
                                <CellDirective
formula={ '=COUNTA(Categories)' } index={3}></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective>
                                <CellsDirective>
                                <CellDirective index={1} value={ 'Average
Spend' } style={fontStyle} colSpan={2}></CellDirective>
                                <CellDirective
formula={ '=AVERAGE(MonthlySpending)' } index={3}
format={ '$#,##0' }></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective>
                                <CellsDirective>
                                <CellDirective index={1} value={ 'Min
Spend' } style={fontStyle} colSpan={2}></CellDirective>
                                <CellDirective
formula={ '=MIN(MonthlySpending)' } index={3}
format={ '$#,##0' }></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective>
                                <CellsDirective>
                                <CellDirective index={1} value={ 'Max
Spend' } style={fontStyle} colSpan={2}></CellDirective>
                                <CellDirective
formula={ '=MAX(MonthlySpending)' } index={3}
format={ '$#,##0' }></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                </RowsDirective>
                                <RangesDirective>
                                <RangeDirective dataSource={data}
startCell={ "A2" }></RangeDirective>
                                </RangesDirective>
                                <ColumnsDirective>
                                <ColumnDirective width={150}></ColumnDirective>
                                <ColumnDirective width={120}></ColumnDirective>
                                <ColumnDirective width={120}></ColumnDirective>
                                <ColumnDirective width={120}></ColumnDirective>
                                <ColumnDirective width={120}></ColumnDirective>
                                </ColumnsDirective>

```

```

        </SheetDirective>
      </SheetsDirective>
      <DefinedNamesDirective>
        <DefinedNameDirective name={'Categories'}
        refersTo={"=Budget Details!A3:A11"}> </DefinedNameDirective>
        <DefinedNameDirective name={'MonthlySpending'}
        refersTo={"=Budget Details!B3:B11"}> </DefinedNameDirective>
        <DefinedNameDirective name={'AnnualSpending'}
        refersTo={"=Budget Details!C3:C11"}> </DefinedNameDirective>
      </DefinedNamesDirective>
    </SpreadsheetComponent>
  </div>
);
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

### DATASOURCE.JSX

```

/**
 * Formula data source
 */
export let data = [
  {
    'Category': 'Household Utilities',
    'Monthly Spend': '=C3/12',
    'Annual Spend': 3000,
    'Last Year Spend': 3000,
    'Percentage Change': '=C3/D3' // You can set the expression or
    formula as string
  },
  {
    'Category': 'Food',
    'Monthly Spend': '=C4/12',
    'Annual Spend': 2500,
    'Last Year Spend': 2250,
    'Percentage Change': { formula: '=C4/D4' } // You can also set as
    object with formula field
  },
  {
    'Category': 'Gasoline',
    'Monthly Spend': '=C5/12',
    'Annual Spend': 1500,
    'Last Year Spend': 1200,
    'Percentage Change': { formula: '=C5/D5' }
  },
  {
    'Category': 'Clothes',
    'Monthly Spend': '=C6/12',
    'Annual Spend': 1200,
    'Last Year Spend': 1000,
    'Percentage Change': '=C6/D6'
  },
  {

```



```

    'Category': 'Insurance',
    'Monthly Spend': '=C7/12',
    'Annual Spend': 1500,
    'Last Year Spend': 1500,
    'Percentage Change': '=C7/D7'
  },
  {
    'Category': 'Taxes',
    'Monthly Spend': '=C8/12',
    'Annual Spend': 3500,
    'Last Year Spend': 3500,
    'Percentage Change': '=C8/D8'
  },
  {
    'Category': 'Entertainment',
    'Monthly Spend': '=C9/12',
    'Annual Spend': 2000,
    'Last Year Spend': 2250,
    'Percentage Change': '=C9/D9'
  },
  {
    'Category': 'Vacation',
    'Monthly Spend': '=C10/12',
    'Annual Spend': 1500,
    'Last Year Spend': 2000,
    'Percentage Change': '=C10/D10'
  },
  {
    'Category': 'Miscellaneous',
    'Monthly Spend': '=C11/12',
    'Annual Spend': 1250,
    'Last Year Spend': 1558,
    'Percentage Change': '=C11/D11'
  }
];

```

### DATASOURCE.TSX

```

/**
 * Formula data source
 */
export let data: Object[] = [
  {
    'Category': 'Household Utilities',
    'Monthly Spend': '=C3/12', // Setting formula through data binding
    'Annual Spend': 3000,
    'Last Year Spend': 3000,
    'Percentage Change': '=C3/D3' // You can set the expression or
    formula as string
  },
  {
    'Category': 'Food',
    'Monthly Spend': '=C4/12',
    'Annual Spend': 2500,
    'Last Year Spend': 2250,

```

```

    'Percentage Change': { formula: '=C4/D4' } // You can also set as
    object with formula field
  },
  {
    'Category': 'Gasoline',
    'Monthly Spend': '=C5/12',
    'Annual Spend': 1500,
    'Last Year Spend': 1200,
    'Percentage Change': { formula: '=C5/D5' }
  },
  {
    'Category': 'Clothes',
    'Monthly Spend': '=C6/12',
    'Annual Spend': 1200,
    'Last Year Spend': 1000,
    'Percentage Change': '=C6/D6'
  },
  {
    'Category': 'Insurance',
    'Monthly Spend': '=C7/12',
    'Annual Spend': 1500,
    'Last Year Spend': 1500,
    'Percentage Change': '=C7/D7'
  },
  {
    'Category': 'Taxes',
    'Monthly Spend': '=C8/12',
    'Annual Spend': 3500,
    'Last Year Spend': 3500,
    'Percentage Change': '=C8/D8'
  },
  {
    'Category': 'Entertainment',
    'Monthly Spend': '=C9/12',
    'Annual Spend': 2000,
    'Last Year Spend': 2250,
    'Percentage Change': '=C9/D9'
  },
  {
    'Category': 'Vacation',
    'Monthly Spend': '=C10/12',
    'Annual Spend': 1500,
    'Last Year Spend': 2000,
    'Percentage Change': '=C10/D10'
  },
  {
    'Category': 'Miscellaneous',
    'Monthly Spend': '=C11/12',
    'Annual Spend': 1250,
    'Last Year Spend': 1558,
    'Percentage Change': '=C11/D11'
  }
];

```

### Supported Formulas

The following are the list of formulas supported in spreadsheet,

Formula	Description
-----	-----
ABS	Returns the value of a number without its sign.
ADDRESS	Returns a cell reference as text, given specified row and column numbers.
AND	Returns TRUE if all the arguments are TRUE, otherwise returns FALSE.
AVERAGE	Calculates average for the series of numbers and/or cells excluding text.
AVERAGEA	Calculates the average for the cells evaluating TRUE as 1, text and FALSE as 0.
AVERAGEIF	Clears content of the active cell and enables edit mode.
AVERAGEIFS	Calculates average for the cells based on specified conditions.
CEILING	Rounds a number up to the nearest multiple of a given factor.
CHOOSE	Returns a value from list of values, based on index number.
CHAR	Returns the character from the specified number.
CODE	Returns the numeric code for the first character in a given string.
CONCAT	Concatenates a list or a range of text strings.
CONCATENATE	Combines two or more strings together.
COUNT	Counts the cells that contain numeric values in a range.
COUNTA	Counts the cells that contains values in a range.
COUNTBLANK	Returns the number of empty cells in a specified range of cells.
COUNTIF	Counts the cells based on specified condition.
COUNTIFS	Counts the cells based on specified conditions.
DATE	Returns the date based on given year, month, and day.
DATEVALUE	Converts a date string into date value.
DAY	Returns the day from the given date.
DAYS	Returns the number of days between two dates.
DECIMAL	Converts a text representation of a number in a given base into a decimal number.
DEGREES	Converts radians to degrees.
DOLLAR	Converts the number to currency formatted text.
EDATE	Returns a date with given number of months before or after the specified date.
EOMONTH	Returns the last day of the month that is a specified number of months before or after an initially supplied start date.
EVEN	Rounds a positive number up and negative number down to the nearest even integer.
EXACT	Checks whether a two text strings are exactly same and returns TRUE or FALSE.

| EXP | Returns e raised to the power of the given number. |

| FACT | Returns the factorial of a number. |

| FIND | Returns the position of a string within another string, which is case sensitive. |

| FLOOR | Rounds a number down to the nearest multiple of a given factor. |

| HLOOKUP | Looks for a value in the top row of the array of values and then returns a value in the same column from a row in the array that you specify. |

| HOUR | Returns the number of hours in a specified time string. |

| IF | Returns value based on the given expression. |

| IFERROR | Returns value if no error found else it will return specified value. |

| IFS | Returns value based on the given multiple expressions. |

| INDEX | Returns a value of the cell in a given range based on row and column number. |

| INT | Rounds a number down to the nearest integer. |

| INTERCEPT | Calculates the point of the Y-intercept line via linear regression. |

| ISNUMBER | Returns true when the value parses as a numeric value. |

| LARGE | Returns the **k-th** largest value in a given array. |

| LEN | Returns a number of characters in a given string. |

| LN | Returns the natural logarithm of a number. |

| LOG | Returns the logarithm of a number to the base that you specify. |

| LOOKUP | Looks for a value in a one-row or one-column range, then returns a value from the same position in a second one-row or one-column range. |

| MATCH | Returns the relative position of a specified value in given range. |

| MAX | Returns the largest number of the given arguments. |

| MEDIAN | Returns the median of the given set of numbers. |

| MINUTE | Returns the number of minutes in a specified time string. |

| MIN | Returns the smallest number of the given arguments. |

| MOD | Returns a remainder after a number is divided by divisor. |

| MONTH | Returns the number of months in a specified date string. |

| NOT | Returns the inverse of a given logical expression. |

| NOW | Returns the current date and time. |

| ODD | Rounds a positive number up and negative number down to the nearest odd integer. |

| OR | Returns TRUE if any of the arguments are TRUE, otherwise returns FALSE. |

| PI | Returns the value of pi. |

| POWER | Returns the result of a number raised to power. |

| PRODUCT | Multiplies a series of numbers and/or cells. |

| RADIANS | Converts degrees into radians. |

| RAND | Returns a random number between 0 and 1. |

| RANDBETWEEN | Returns a random integer based on specified values. |

| ROUND | Rounds a number to the specified number of digits. |

| ROUNDDOWN | Rounds a number down, toward zero. |

| ROUNDUP | Rounds a number up, away from zero. |

| RSQ | Returns the square of the Pearson product moment correlation coefficient based on data points in *knowny's* and *knownx's*. |

| SECOND | Returns the number of seconds in a specified time string. |

| SMALL | Returns the **k-th** smallest value in a given array. |

| SLOPE | Returns the slope of the line from linear regression of the data points. |

| SORT | Sorts the contents of a column, range, or array in ascending or descending order. |

| SQRT | Returns the square root of a positive number. |

| SUBTOTAL | Returns subtotal for a range using the given function number. |

| SUM | Adds a series of numbers and/or cells. |

| SUMIF | Adds the cells based on specified condition. |

| SUMIFS | Adds the cells based on specified conditions. |

| SUMPRODUCT | Returns the sum of the products of the corresponding array in given arrays. |

| T | Checks whether a value is text or not and returns the text. |

| TEXT | Converts the supplied value into text by using the user-specified format. |

| TIME | Converts hours, minutes, seconds to the time formatted text. |

| TODAY | Returns the current date. |

| TRUNC | Truncates a supplied number to a specified number of decimal places. |

| UNIQUE | Returns a unique values from a range or array. |

| VLOOKUP | Looks for a specific value in the first column of a lookup range and returns a corresponding value from a different column within the same row. |

### Formula Error Dialog

If you enter an invalid formula in a cell, an error dialog with an error message will appear. For instance, a formula with the incorrect number of arguments, a formula without parenthesis, etc.

| Error Message | Reason |

|-----|-----|

| We found that you typed a formula with an invalid arguments | Occurs when passing an argument even though it wasn't needed. |

| We found that you typed a formula with an empty expression | Occurs when passing an empty expression in the argument. |

| We found that you typed a formula with one or more missing opening or closing parenthesis | Occurs when an open parenthesis or a close parenthesis is missing. |

| We found that you typed a formula which is improper | Occurs when passing a single reference but a range was needed. |

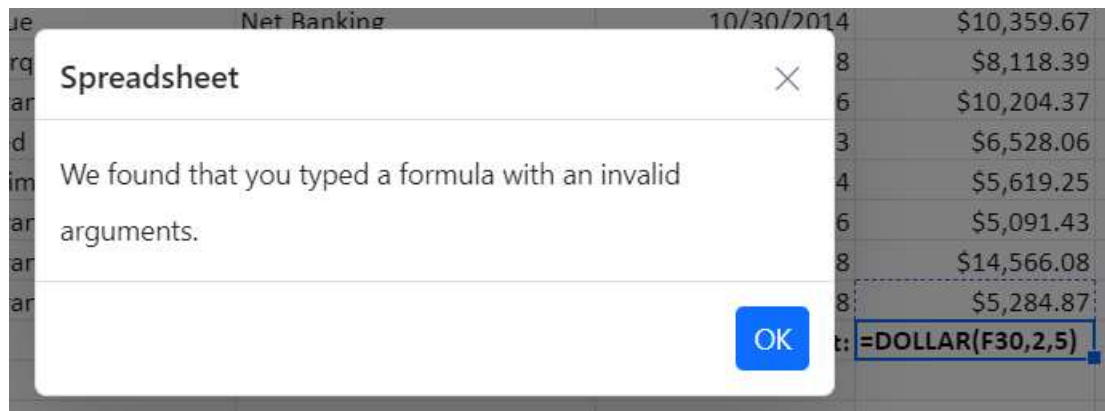
| We found that you typed a formula with a wrong number of arguments | Occurs when the required arguments were not passed. |

| We found that you typed a formula which requires 3 arguments | Occurs when the required 3 arguments were not passed. |

| We found that you typed a formula with a mismatched quotes | Occurs when passing an argument with mismatched quotes. |

| We found that you typed a formula with a circular reference | Occurs when passing a formula with circular cell reference. |

| We found that you typed a formula which is invalid | Except in the cases mentioned above, all other errors will fall into this broad category. |



#### Note

You can refer to our [React Spreadsheet](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Spreadsheet example](#) to know how to present and manipulate data.

#### See Also

- [Editing](#)
- [Formatting](#)
- [Open](#)
- [SaveLink to the Video](#)

#### Formatting in React Spreadsheet component

Formatting options make your data easier to view and understand. The different types of formatting options in the Spreadsheet are,

- Number Formatting
- Text Formatting
- Cell Formatting

To get start quickly with Formatting, you can check on this video:

### Number Formatting

Number formatting provides a type for your data in the Spreadsheet. Use the [allowNumberFormatting](#) property to enable or disable the number formatting option in the Spreadsheet. The different types of number formatting supported in Spreadsheet are,

Types	Format
-----	-----
General(default)	NA
Number	0.00
Currency	\$#,##0.00
Accounting	(\$ #,##0.00);(\$ (#,##0.00);(\$* "-"??);(@_)
ShortDate	mm-dd-yyyy
LongDate	dddd, mmmm dd, yyyy
Time	h:mm:ss AM/PM
Percentage	0.00%
Fraction	# ?/?
Scientific	0.00E+00
Text	@

Number formatting can be applied in following ways,

- Using the `format` property in `cell`, you can set the desired format to each cell at initial load.
- Using the [numberFormat](#) method, you can set the number format to a cell or range of cells.
- Selecting the number format option from ribbon toolbar.

### Custom Number Formatting

Spreadsheet supports custom number formats to display your data as numbers, dates, times, percentages, and currency values. If the pre-defined number formats do not meet your needs, you can set your own custom formats using custom number formats dialog or `numberFormat` method.

The different types of custom number formatting supported in Spreadsheet are,

Types	Format
-----	-----
General(default)	NA
Number	0
Number	0.00
Number	#,##0
Number	#,##0.00

Number	#,##0\_);(#,##0)
Number	`#,##0\_);Red`
Number	#,##0.00\_);(#,##0.00)
Number	`#,##0.00\_);Red`
Currency	\$#,##0\_);(\$#,##0)
Currency	`\$#,##0\_);Red`
Currency	\$#,##0.00\_);(\$#,##0.00)
Currency	`\$#,##0.00\_);Red`
Percentage	0%
Percentage	0.00%
Scientific	0.00E+00
Scientific	##0.0E+0
Fraction	# ?/?
Fraction	# ??/??
ShortDate	dd-mm-yy
Custom	dd-mmm-yy
Custom	dd-mmm
Custom	mmm-yy
Custom	h:mm AM/PM
Custom	h:mm:ss AM/PM
Custom	h:mm
Custom	h:mm:ss
Custom	dd-mm-yy h:mm
Custom	mm:ss
Custom	mm:ss.0
Text	@
Custom	[h]:mm:ss
Accounting	(\$ #,##0);(\$ (#,##0);(\$\* "-");(@\_)
Accounting	( #,##0);( (#,##0);(\* "-");(@\_)
Accounting	(\$ #,##0.00);(\$ (#,##0.00);(\$\* "-"?);(@\_)
Accounting	( #,##0.00);( (#,##0.00);(\* "-"?);(@\_)



Custom Number formatting can be applied in following ways,

- Using the [numberFormat](#) method, you can set your own custom number format to a cell or range of cells.
- Selecting the custom number format option from custom number formats dialog or type your own format in dialog input and then click apply button. It will apply the custom format for selected cells.

The following code example shows the number formatting in cell data.

#### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective, CellDirective, CellsDirective,
getFormatFromType } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  const styles = {
    verticalAlign: 'middle', textAlign: 'center', fontSize: '16pt',
fontWeight: 'bold',
    border: '1px solid #e0e0e0', backgroundColor: '#EEEEEE', color:
'#279377'
  };
  const cellStyle = { fontWeight: 'bold', fontStyle: 'italic' };
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', fontSize: '12pt',
backgroundColor: '#279377', textAlign: 'center', color: 'ffffff',
borderBottom: '1px solid #e0e0e0' }, 'A2:F2');
      spreadsheet.cellFormat({ borderTop: '1px solid #e0e0e0',
backgroundColor: '#EEEEEE' }, 'A11:F11');
      spreadsheet.setBorder({ border: '1px solid #e0e0e0' }, 'A2:F11',
'Outer');
      // Applied Accounting format to the cells from C3 to E10 range.
      spreadsheet.numberFormat('_($* #,##0.00_);_($* (#,##0.00);_($*
"-"?_);_(@_)', 'C3:E10');
      // Applied Percentage format to the cells from C3 to E11 range.
      spreadsheet.numberFormat('0%', 'F3:F10');
      // applied the custom number format for cell form D3 to D10
range
      spreadsheet.numberFormat('[Red] [<=2000] $#,##0.00; [Blue] [>2000] $#,##0.00',
'D3:D10');
      // applied the custom number format for cell from F3 to F10
range
      spreadsheet.numberFormat('#,##0.00_); [Red] (#,##0.00)',
'F3:F10');
    }
  });
}
```

```

    }, []);
    return (
      <div>
        <SpreadsheetComponent ref={spreadsheetRef} showSheetTabs={false}
showFormulaBar={false} allowInsert={false} allowDelete={false}>
          <SheetsDirective>
            <SheetDirective selectedRange={"U15"}
showGridLines={false}>
              <RowsDirective>
                <RowDirective height={35} customHeight={true}>
                  <CellsDirective>
                    <CellDirective value={'Sales Team
Summary'} style={styles} colSpan={6}></CellDirective>
                  </CellsDirective>
                </RowDirective>
                <RowDirective index={10}>
                  <CellsDirective>
                    <CellDirective index={1}
value={'Total: '} style={cellStyle}></CellDirective>
                    <CellDirective formula={'=SUM(C3:C10)'}
format={getFormatFromType('Accounting')}></CellDirective>
                    <CellDirective formula={'=SUM(D3:D10)'}
format={'_($* #,##0.00_);_($* (#,##0.00);_($* "-
"??_);_(@_)'}></CellDirective>
                    <CellDirective formula={'=SUM(E3:E10)'}
format={'_($* #,##0.00_);_($* (#,##0.00);_($* "-
"??_);_(@_)'}></CellDirective>
                  </CellsDirective>
                </RowDirective>
              </RowsDirective>
            <RangesDirective>
              <RangeDirective dataSource={data}
startCell={"A2"}></RangeDirective>
            </RangesDirective>
            <ColumnsDirective>
              <ColumnDirective width={140}></ColumnDirective>
              <ColumnDirective width={140}></ColumnDirective>
              <ColumnDirective width={160}></ColumnDirective>
              <ColumnDirective width={160}></ColumnDirective>
              <ColumnDirective width={160}></ColumnDirective>
              <ColumnDirective width={120}></ColumnDirective>
            </ColumnsDirective>
          </SheetDirective>
        </SheetsDirective>
      </SpreadsheetComponent>
    </div>
  );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

## APP.TSX

```
{% raw %}
```

```

import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective, CellDirective, CellsDirective,
getFormatFromType } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { CellStyleModel } from '@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
    const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
    const styles: CellStyleModel = {
        verticalAlign: 'middle', textAlign: 'center', fontSize: '16pt',
fontWeight: 'bold',
        border: '1px solid #e0e0e0', backgroundColor: '#EEEEEE', color:
'#279377'
    };
    const cellStyle: CellStyleModel = { fontWeight: 'bold', fontStyle:
'italic' };
    React.useEffect(() => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            spreadsheet.cellFormat({ fontWeight: 'bold', fontSize: '12pt',
backgroundColor: '#279377', textAlign: 'center', color: '#ffffff',
borderBottom: '1px solid #e0e0e0' }, 'A2:F2');
            spreadsheet.cellFormat({ borderTop: '1px solid #e0e0e0',
backgroundColor: '#EEEEEE' }, 'A11:F11');
            spreadsheet.setBorder({ border: '1px solid #e0e0e0' }, 'A2:F11',
'Outer');
            // Applied Accounting format to the cells from C3 to E10 range.
            spreadsheet.numberFormat('_($* #,##0.00_);_($* (#,##0.00);_($*
"--"??_);_(@_)', 'C3:E10');
            // Applied Percentage format to the cells from C3 to E11 range.
            spreadsheet.numberFormat('0%', 'F3:F10');
            // applied the custom number format for cell form D3 to D10
range
            spreadsheet.numberFormat('[Red] [<=2000] $#,##0.00; [Blue] [>2000] $#,##0.00',
'D3:D10');
            // applied the custom number format for cell from F3 to F10
range
            spreadsheet.numberFormat('#,##0.00_); [Red] (#,##0.00)',
'F3:F10');
        }
    }, []);
    return (
        <div>
            <SpreadsheetComponent ref={spreadsheetRef} showSheetTabs={false}
showFormulaBar={false} allowInsert={false} allowDelete={false}>
                <SheetsDirective>
                    <SheetDirective selectedRange={"U15"}
showGridLines={false}>
                        <RowsDirective>
                            <RowDirective height={35} customHeight={true}>
                                <CellsDirective>
                                    <CellDirective value={'Sales Team
Summary'} style={styles} colSpan={6}></CellDirective>

```

```

        </CellsDirective>
      </RowDirective>
      <RowDirective index={10}>
        <CellsDirective>
          <CellDirective index={1}
value={'Total: '} style={cellStyle}></CellDirective>
          <CellDirective formula={'=SUM(C3:C10)'}
format={getFormatFromType('Accounting')}></CellDirective>
          <CellDirective formula={'=SUM(D3:D10)'}
format={'_($* #,##0.00_);_($* (#,##0.00);_($* "-
"??_);_(@_)'}></CellDirective>
          <CellDirective formula={'=SUM(E3:E10)'}
format={'_($* #,##0.00_);_($* (#,##0.00);_($* "-
"??_);_(@_)'}></CellDirective>
        </CellsDirective>
      </RowDirective>
    </RowsDirective>
    <RangesDirective>
      <RangeDirective dataSource={data}
startCell={"A2"}></RangeDirective>
    </RangesDirective>
    <ColumnsDirective>
      <ColumnDirective width={140}></ColumnDirective>
      <ColumnDirective width={140}></ColumnDirective>
      <ColumnDirective width={160}></ColumnDirective>
      <ColumnDirective width={160}></ColumnDirective>
      <ColumnDirective width={160}></ColumnDirective>
      <ColumnDirective width={120}></ColumnDirective>
    </ColumnsDirective>
  </SheetDirective>
</SheetsDirective>
</SpreadsheetComponent>
</div>
);
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

## DATASOURCE.JSX

```

/**
 * Number formatting data source
 */
export let data = [
  { 'Salesperson': 'Jeffrey Burke', 'Region Covered': 'Oklahoma',
'February 2019 Sales': '28000', 'Cost of Sales': '2460', 'January 2019
Sales': '21238', 'Percent Change': '.32' },
  { 'Salesperson': 'Amy Fernandez', 'Region Covered': 'North Carolina',
'February 2019 Sales': '23138', 'Cost of Sales': '1521', 'January 2019
Sales': '23212', 'Percent Change': '0' },
  { 'Salesperson': 'Mark Hayes', 'Region Covered': 'Massachusetts',
'February 2019 Sales': '25092', 'Cost of Sales': '1530', 'January 2019
Sales': '20454', 'Percent Change': '.23' },

```

```

    { 'Salesperson': 'Judith Ray', 'Region Covered': 'California', 'February
2019 Sales': '21839', 'Cost of Sales': '1923', 'January 2019 Sales':
'24619', 'Percent Change': '-.11' },
    { 'Salesperson': 'Rany Graham', 'Region Covered': 'South Carolina',
'February 2019 Sales': '23342', 'Cost of Sales': '2397', 'January 2019
Sales': '20045', 'Percent Change': '.16' },
    { 'Salesperson': 'Christina Foster', 'Region Covered': 'Delaware',
'February 2019 Sales': '23368', 'Cost of Sales': '1500', 'January 2019
Sales': '17537', 'Percent Change': '.33' },
    { 'Salesperson': 'Judy Green', 'Region Covered': 'Texas', 'February 2019
Sales': '21510', 'Cost of Sales': '1657', 'January 2019 Sales': '24951',
'Percent Change': '-.14' },
    { 'Salesperson': 'Paula Hall', 'Region Covered': 'Virginia', 'February
2019 Sales': '21314', 'Cost of Sales': '2418', 'January 2019 Sales':
'18082', 'Percent Change': '.18' }
];

```

### DATASOURCE.TSX

```

/**
 * Number formatting data source
 */
export let data: Object[] = [
  { 'Salesperson': 'Jeffrey Burke', 'Region Covered': 'Oklahoma',
'February 2019 Sales': '28000', 'Cost of Sales': '2460', 'January 2019
Sales': '21238', 'Percent Change': '.32' },
  { 'Salesperson': 'Amy Fernandez', 'Region Covered': 'North Carolina',
'February 2019 Sales': '23138', 'Cost of Sales': '1521', 'January 2019
Sales': '23212', 'Percent Change': '0' },
  { 'Salesperson': 'Mark Hayes', 'Region Covered': 'Massachusetts',
'February 2019 Sales': '25092', 'Cost of Sales': '1530', 'January 2019
Sales': '20454', 'Percent Change': '.23' },
  { 'Salesperson': 'Judith Ray', 'Region Covered': 'California', 'February
2019 Sales': '21839', 'Cost of Sales': '1923', 'January 2019 Sales':
'24619', 'Percent Change': '-.11' },
  { 'Salesperson': 'Rany Graham', 'Region Covered': 'South Carolina',
'February 2019 Sales': '23342', 'Cost of Sales': '2397', 'January 2019
Sales': '20045', 'Percent Change': '.16' },
  { 'Salesperson': 'Christina Foster', 'Region Covered': 'Delaware',
'February 2019 Sales': '23368', 'Cost of Sales': '1500', 'January 2019
Sales': '17537', 'Percent Change': '.33' },
  { 'Salesperson': 'Judy Green', 'Region Covered': 'Texas', 'February 2019
Sales': '21510', 'Cost of Sales': '1657', 'January 2019 Sales': '24951',
'Percent Change': '-.14' },
  { 'Salesperson': 'Paula Hall', 'Region Covered': 'Virginia', 'February
2019 Sales': '21314', 'Cost of Sales': '2418', 'January 2019 Sales':
'18082', 'Percent Change': '.18' }
];

```

### Text and cell formatting

Text and cell formatting enhances the look and feel of your cell. It helps to highlight a particular cell or range of cells from a whole workbook. You can apply formats like font size, font family, font color, text alignment, border etc. to a cell or range of cells. Use the [allowCellFormatting](#) property to enable or disable the text and cell formatting option in Spreadsheet. You can set the formats in following ways,

- Using the `style` property, you can set formats to each cell at initial load.
- Using the `cellFormat` method, you can set formats to a cell or range of cells.
- You can also apply by clicking the desired format option from the ribbon toolbar.

### Fonts

Various font formats supported in the spreadsheet are font-family, font-size, bold, italic, strike-through, underline and font color.

### Text Alignment

You can align text in a cell either vertically or horizontally using the `textAlign` and `verticalAlign` property.

### Indents

To enhance the appearance of text in a cell, you can change the indentation of a cell content using `textIndent` property.

### Fill color

To highlight cell or range of cells from whole workbook you can apply background color for a cell using `backgroundColor` property.

### Borders

You can add borders around a cell or range of cells to define a section of worksheet or a table. The different types of border options available in the spreadsheet are,

| Types | Actions |

|-----|-----|

| Top Border | Specifies the top border of a cell or range of cells. |

| Left Border | Specifies the left border of a cell or range of cells. |

| Right Border | Specifies the right border of a cell or range of cells. |

| Bottom Border | Specifies the bottom border of a cell or range of cells. |

| No Border | Used to clear the border from a cell or range of cells. |

| All Border | Specifies all border of a cell or range of cells. |

| Horizontal Border | Specifies the top and bottom border of a cell or range of cells. |

| Vertical Border | Specifies the left and right border of a cell or range of cells. |

| Outside Border | Specifies the outside border of a range of cells. |

| Inside Border | Specifies the inside border of a range of cells. |

You can also change the color, size, and style of the border. The size and style supported in the spreadsheet are,

| Types | Actions |

|-----|-----|

| Thin | Specifies the `1px` border size (default). |

| Medium | Specifies the `2px` border size. |

- | Thick | Specifies the **3px** border size.
- | Solid | Used to create the **solid** border (default).
- | Dashed | Used to create the **dashed** border.
- | Dotted | Used to create the **dotted** border.
- | Double | Used to create the **double** border.

Borders can be applied in the following ways,

- Using the **border**, **borderLeft**, **borderRight**, **borderBottom** properties, you can set the desired border to each cell at initial load.
- Using the **setBorder** method, you can set various border options to a cell or range of cells.
- Selecting the border options from ribbon toolbar.

The following code example shows the style formatting in text and cells of the spreadsheet.

#### **APP.JSX**

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective, CellDirective, CellsDirective
} from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  const styles = { fontFamily: 'Axettac Demo', verticalAlign: 'middle',
textAlign: 'center', fontSize: '18pt', fontWeight: 'bold', color: '#279377',
border: '1px solid #e0e0e0' };
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      // Setting common styles to table header cells
      spreadsheet.cellFormat({ fontWeight: 'bold', fontSize: '12pt',
backgroundColor: '#279377', color: '#ffffff' }, 'A2:E2');
      // Setting common styles to whole table cells
      spreadsheet.cellFormat({ verticalAlign: 'middle', fontFamily:
'Axettac Demo' }, 'A2:E12');
      // Column wise styles setting
      spreadsheet.cellFormat({ textAlign: 'center' }, 'A2:A12');
      // Setting text-indent to 2 and 4 column
      let style = { textAlign: 'left', textIndent: '8pt' };
      spreadsheet.cellFormat(style, 'B2:B12');
      spreadsheet.cellFormat(style, 'D2:D12');
      spreadsheet.cellFormat({ fontStyle: 'italic', textAlign: 'right'
}, 'C3:C12');
      spreadsheet.cellFormat({ textAlign: 'center' }, 'E2:E12');
      // Applied border to range of cells using 'setBorder' method
      spreadsheet.setBorder({ borderLeft: '1px solid #e0e0e0',
borderRight: '1px solid #e0e0e0' }, 'A2:E2');
```

```

        spreadsheet.setBorder({ border: '1px solid #e0e0e0' }, 'A4:E11',
        'Horizontal');
        spreadsheet.setBorder({ border: '1px solid #e0e0e0' }, 'A3:E12',
        'Outer');
        spreadsheet.cellFormat({ color: '#10c469', textDecoration:
        'line-through' }, 'E3:E4');
        spreadsheet.cellFormat({ color: '#10c469', textDecoration:
        'line-through' }, 'E9');
        spreadsheet.cellFormat({ color: '#10c469', textDecoration:
        'line-through' }, 'E12');
        spreadsheet.cellFormat({ color: '#FFC107', textDecoration:
        'underline' }, 'E5');
        spreadsheet.cellFormat({ color: '#FFC107', textDecoration:
        'underline' }, 'E8');
        spreadsheet.cellFormat({ color: '#FFC107', textDecoration:
        'underline' }, 'E11');
        spreadsheet.cellFormat({ color: '#62c9e8' }, 'E6');
        spreadsheet.cellFormat({ color: '#62c9e8' }, 'E10');
        spreadsheet.cellFormat({ color: '#ff5b5b' }, 'E7');
    }
    }, []);
    return (
    <div>
        <SpreadsheetComponent ref={spreadsheetRef} showRibbon={false}
        showFormulaBar={false} showSheetTabs={false} allowEditing={false}
        allowDelete={false} allowInsert={false} >
            <SheetsDirective>
                <SheetDirective selectedRange={"U15"}
                showGridLines={false}>
                    <RowsDirective>
                        <RowDirective height={40} customHeight={true}>
                            <CellsDirective>
                                <CellDirective colSpan={5} value={'Order
                                Summary'} style={styles}></CellDirective>
                            </CellsDirective>
                        </RowDirective>
                    </RowsDirective>
                    <RangesDirective>
                        <RangeDirective dataSource={data}
                        startCell={'A2'}></RangeDirective>
                    </RangesDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={100}></ColumnDirective>
                        <ColumnDirective width={200}></ColumnDirective>
                        <ColumnDirective width={110}></ColumnDirective>
                        <ColumnDirective width={140}></ColumnDirective>
                        <ColumnDirective width={90}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
            </SheetsDirective>
        </SpreadsheetComponent>
    </div>
    );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);

```



```
{% endraw %}
```

**APP.TSX**

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective, CellDirective, CellsDirective
} from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { CellStyleModel } from '@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
    const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
    const styles: CellStyleModel = { fontFamily: 'Axettac Demo',
verticalAlign: 'middle', textAlign: 'center', fontSize: '18pt', fontWeight:
'bold', color: '#279377', border: '1px solid #e0e0e0' };
    React.useEffect(() => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            // Setting common styles to table header cells
            spreadsheet.cellFormat({ fontWeight: 'bold', fontSize: '12pt',
backgroundColor: '#279377', color: '#ffffff' }, 'A2:E2');
            // Setting common styles to whole table cells
            spreadsheet.cellFormat({ verticalAlign: 'middle', fontFamily:
'Axettac Demo' }, 'A2:E12');
            // Column wise styles setting
            spreadsheet.cellFormat({ textAlign: 'center' }, 'A2:A12');
            // Setting text-indent to 2 and 4 column
            let style: CellStyleModel = { textAlign: 'left', textIndent:
'8pt' };
            spreadsheet.cellFormat(style, 'B2:B12');
            spreadsheet.cellFormat(style, 'D2:D12');
            spreadsheet.cellFormat({ fontStyle: 'italic', textAlign: 'right'
}, 'C3:C12');
            spreadsheet.cellFormat({ textAlign: 'center' }, 'E2:E12');
            // Applied border to range of cells using 'setBorder' method
            spreadsheet.setBorder({ borderLeft: '1px solid #e0e0e0',
borderRight: '1px solid #e0e0e0' }, 'A2:E2');
            spreadsheet.setBorder({ border: '1px solid #e0e0e0' }, 'A4:E11',
'Horizontal');
            spreadsheet.setBorder({ border: '1px solid #e0e0e0' }, 'A3:E12',
'Outer');
            spreadsheet.cellFormat({ color: '#10c469', textDecoration:
'line-through' }, 'E3:E4');
            spreadsheet.cellFormat({ color: '#10c469', textDecoration:
'line-through' }, 'E9');
            spreadsheet.cellFormat({ color: '#10c469', textDecoration:
'line-through' }, 'E12');
            spreadsheet.cellFormat({ color: '#FFC107', textDecoration:
'underline' }, 'E5');
            spreadsheet.cellFormat({ color: '#FFC107', textDecoration:
'underline' }, 'E8');
        }
    });
}
```

```

        spreadsheet.cellFormat({ color: '#FFC107', textDecoration:
'underline' }, 'E11');
        spreadsheet.cellFormat({ color: '#62c9e8' }, 'E6');
        spreadsheet.cellFormat({ color: '#62c9e8' }, 'E10');
        spreadsheet.cellFormat({ color: '#ff5b5b' }, 'E7');
    }
}, []);
return (
    <div>
        <SpreadsheetComponent ref={spreadsheetRef} showRibbon={false}
showFormulaBar={false} showSheetTabs={false} allowEditing={false}
allowDelete={false} allowInsert={false} >
            <SheetsDirective>
                <SheetDirective selectedRange={"U15"}
showGridLines={false}>
                    <RowsDirective>
                        <RowDirective height={40} customHeight={true}>
                            <CellsDirective>
                                <CellDirective colSpan={5} value={'Order
Summary'} style={styles}></CellDirective>
                            </CellsDirective>
                        </RowDirective>
                    </RowsDirective>
                    <RangesDirective>
                        <RangeDirective dataSource={data}
startCell={'A2'}></RangeDirective>
                    </RangesDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={100}></ColumnDirective>
                        <ColumnDirective width={200}></ColumnDirective>
                        <ColumnDirective width={110}></ColumnDirective>
                        <ColumnDirective width={140}></ColumnDirective>
                        <ColumnDirective width={90}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
            </SheetsDirective>
        </SpreadsheetComponent>
    </div>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### DATASOURCE.JSX

```

/**
 * Cell formatting data source
 */
export let data = [
    { 'Order Id': 'SF1001', 'Product': 'Laptop Backpack (Blue)', 'Ordered
Date': '02/14/2014', 'Ordered By': 'Rahul Sharma', 'Shipment': 'Delivered'
},

```

```
{ 'Order Id': 'SF1002', 'Product': 'Oppo F1 S mobile back cover',
'Ordered Date': '06/11/2014', 'Ordered By': 'Adi Pathak', 'Shipment':
'Delivered' },
{ 'Order Id': 'SF1003', 'Product': 'Tupperware 4 bottle set', 'Ordered
Date': '07/27/2014', 'Ordered By': 'Himani Arora', 'Shipment': 'Pending' },
{ 'Order Id': 'SF1004', 'Product': 'Tupperware Lunch box', 'Ordered
Date': '11/21/2014', 'Ordered By': 'Samuel Samson', 'Shipment': 'Shipped' },
{ 'Order Id': 'SF1005', 'Product': 'Panasonic Hair Dryer', 'Ordered
Date': '06/23/2014', 'Ordered By': 'Neha', 'Shipment': 'Cancelled' },
{ 'Order Id': 'SF1006', 'Product': 'Philips LED 2 bulb set', 'Ordered
Date': '07/22/2014', 'Ordered By': 'Christine J', 'Shipment': 'Pending' },
{ 'Order Id': 'SF1007', 'Product': 'Moto G4 plus headphone', 'Ordered
Date': '02/04/2014', 'Ordered By': 'Shiv Nagar', 'Shipment': 'Delivered' },
{ 'Order Id': 'SF1008', 'Product': 'Lakme Eyeliner Pencil', 'Ordered
Date': '11/30/2014', 'Ordered By': 'Cherry', 'Shipment': 'Shipped' },
{ 'Order Id': 'SF1009', 'Product': 'Listerine mouthwash', 'Ordered
Date': '07/09/2014', 'Ordered By': 'Siddartha Mishra', 'Shipment': 'Pending'
},
{ 'Order Id': 'SF1010', 'Product': 'Protinex original', 'Ordered Date':
'10/31/2014', 'Ordered By': 'Ravi Chugh', 'Shipment': 'Delivered' },
];
```

### DATASOURCE.TSX

```
/**
 * Cell formatting data source
 */
export let data: Object[] = [
  { 'Order Id': 'SF1001', 'Product': 'Laptop Backpack (Blue)', 'Ordered
Date': '02/14/2014', 'Ordered By': 'Rahul Sharma', 'Shipment': 'Delivered'
},
  { 'Order Id': 'SF1002', 'Product': 'Oppo F1 S mobile back cover',
'Ordered Date': '06/11/2014', 'Ordered By': 'Adi Pathak', 'Shipment':
'Delivered' },
  { 'Order Id': 'SF1003', 'Product': 'Tupperware 4 bottle set', 'Ordered
Date': '07/27/2014', 'Ordered By': 'Himani Arora', 'Shipment': 'Pending' },
  { 'Order Id': 'SF1004', 'Product': 'Tupperware Lunch box', 'Ordered
Date': '11/21/2014', 'Ordered By': 'Samuel Samson', 'Shipment': 'Shipped' },
  { 'Order Id': 'SF1005', 'Product': 'Panasonic Hair Dryer', 'Ordered
Date': '06/23/2014', 'Ordered By': 'Neha', 'Shipment': 'Cancelled' },
  { 'Order Id': 'SF1006', 'Product': 'Philips LED 2 bulb set', 'Ordered
Date': '07/22/2014', 'Ordered By': 'Christine J', 'Shipment': 'Pending' },
  { 'Order Id': 'SF1007', 'Product': 'Moto G4 plus headphone', 'Ordered
Date': '02/04/2014', 'Ordered By': 'Shiv Nagar', 'Shipment': 'Delivered' },
  { 'Order Id': 'SF1008', 'Product': 'Lakme Eyeliner Pencil', 'Ordered
Date': '11/30/2014', 'Ordered By': 'Cherry', 'Shipment': 'Shipped' },
  { 'Order Id': 'SF1009', 'Product': 'Listerine mouthwash', 'Ordered
Date': '07/09/2014', 'Ordered By': 'Siddartha Mishra', 'Shipment': 'Pending'
},
  { 'Order Id': 'SF1010', 'Product': 'Protinex original', 'Ordered Date':
'10/31/2014', 'Ordered By': 'Ravi Chugh', 'Shipment': 'Delivered' },
];
```

### *Limitations of Formatting*

The following features are not supported in Formatting:

- Insert row/column between the formatting applied cells.
- Formatting support for row/column.

### Conditional Formatting

Conditional formatting helps you to format a cell or range of cells based on the conditions applied. You can enable or disable conditional formats by using the [allowConditionalFormat](#) property.

\* The default value for the `allowConditionalFormat` property is `true`.

### Apply Conditional Formatting

You can apply conditional formatting by using one of the following ways,

- Select the conditional formatting icon in the Ribbon toolbar under the Home Tab.
- Using the [conditionalFormat\(\)](#) method to define the condition.
- Using the `conditionalFormats` in sheets model.

Conditional formatting has the following types in the spreadsheet,

### Highlight cells rules

Highlight cells rules option in the conditional formatting enables you to highlight cells with a preset color depending on the cell's value.

The following options can be given for the highlight cells rules as type,

\* 'GreaterThan', 'LessThan', 'Between', 'EqualTo', 'ContainsText', 'DateOccur', 'Duplicate', 'Unique'.

The following preset colors can be used for formatting styles,

- \* `"RedFT"` - Light Red Fill with Dark Red Text,
- \* `"YellowFT"` - Yellow Fill with Dark Yellow Text,
- \* `"GreenFT"` - Green Fill with Dark Green Text,
- \* `"RedF"` - Red Fill,
- \* `"RedT"` - Red Text.

### Top bottom rules

Top bottom rules option in the conditional formatting allows you to apply formatting to the cells that satisfy a statistical condition with other cells in the range.

The following options can be given for the top bottom rules as type,

\* 'Top10Items', 'Bottom10Items', 'Top10Percentage', 'Bottom10Percentage', 'BelowAverage', 'AboveAverage'.

### Data Bars

You can apply data bars to represent the data graphically inside a cell. The longest bar represents the highest value and the shorter bars represent the smaller values.

The following options can be given for the data bars as type,

\* 'BlueDataBar', 'GreenDataBar', 'RedDataBar', 'OrangeDataBar', 'LightBlueDataBar', 'PurpleDataBar'.

### Color Scales

Using color scales, you can format your cells with two or three colors, where different color shades represent the different cell values. In the Green-Yellow-Red(GYR) Color Scale, the cell that holds the minimum value is colored as red. The cell that holds the median is colored as yellow, and the cell that holds the maximum value is colored as green. All other cells are colored proportionally.

The following options can be given for the color scales as type,

\* 'GYRColorScale', 'RYGColorScale', 'GWRColorScale', 'RWGColorScale', 'BWRColorScale', 'RWBColorScale', 'WRCColorScale', 'RWColorScale', 'GWColorScale', 'WGColorScale', 'GYColorScale', 'YGColorScale'.

### Icon Sets

Icon sets will help you to visually represent your data with icons. Every icon represents a range of values. In the Three Arrows(colored) icon, the green arrow icon represents the values greater than 67%, the yellow arrow icon represents the values between 33% to 67%, and the red arrow icon represents the values less than 33%.

The following options can be given for the icon sets as type,

\* 'ThreeArrows', 'ThreeArrowsGray', 'FourArrowsGray', 'FourArrows', 'FiveArrowsGray', 'FiveArrows', 'ThreeTrafficLights1', 'ThreeTrafficLights2', 'ThreeSigns', 'FourTrafficLights', 'FourRedToBlack', 'ThreeSymbols', 'ThreeSymbols2', 'ThreeFlags', 'FourRating', 'FiveQuarters', 'FiveRating', 'ThreeTriangles', 'ThreeStars', 'FiveBoxes'.

### Custom Format

Using the custom format for conditional formatting you can set cell styles like color, background color, font style, font weight, and underline.

In the MAY and JUN columns, we have applied conditional formatting custom format.

\* In the Conditional format, custom format supported for Highlight cell rules and Top bottom rules.

### Clear Rules

You can clear the defined rules by using one of the following ways,

- Using the “Clear Rules” option in the Conditional Formatting button of HOME Tab in the ribbon to clear the rule from selected cells.
- Using the [clearConditionalFormat\(\)](#) method to clear the defined rules.

### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective,
ConditionalFormatsDirective, ConditionalFormatDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { conditionalFormatData } from './datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
```

```

        if (spreadsheet) {
            spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
        }, 'A1:N1');
            spreadsheet.conditionalFormat({ type: 'RYGColorScale', range:
        'E2:E9' });
            spreadsheet.conditionalFormat({ type: 'ThreeArrows', range:
        'H2:H9' });
            spreadsheet.conditionalFormat({
                type: 'Top10Items', value: '1',
                format: { style: { color: '#ffffff', backgroundColor:
        '#009999', fontWeight: 'bold' } }, range: 'F2:F9'
            });
            spreadsheet.conditionalFormat({
                type: 'Bottom10Items', value: '1',
                format: { style: { color: '#ffffff', backgroundColor:
        '#c68d53', fontWeight: 'bold' } }, range: 'G2:G9'
            });
        }
    }, []);
    return (
        <div>
            <SpreadsheetComponent ref={spreadsheetRef}
        showFormulaBar={false}>
                <SheetsDirective>
                    <SheetDirective name='Car Sales Record'>
                        <RangesDirective>
                            <RangeDirective
        dataSource={conditionalFormatData}></RangeDirective>
                        </RangesDirective>
                        <ConditionalFormatsDirective>
                            <ConditionalFormatDirective type='GreaterThan'
        cFColor='RedFT' value='700' range='B2:B9'></ConditionalFormatDirective>
                            <ConditionalFormatDirective type='Bottom10Items'
        cFColor='YellowFT' value='4' range='C2:C9'></ConditionalFormatDirective>
                            <ConditionalFormatDirective type='BlueDataBar'
        range='D2:D9'></ConditionalFormatDirective>
                        </ConditionalFormatsDirective>
                        <ColumnsDirective>
                            <ColumnDirective width={120}></ColumnDirective>
                        </ColumnsDirective>
                    </SheetDirective>
                </SheetsDirective>
            </SpreadsheetComponent>
        </div>
    );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

**APP.TSX**

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';

```

```

import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective,
ConditionalFormatsDirective, ConditionalFormatDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { conditionalFormatData } from './datasource';
function App() {
    const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
    React.useEffect(() => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A1:N1');
            spreadsheet.conditionalFormat({ type: 'RYGColorScale', range:
'E2:E9' });
            spreadsheet.conditionalFormat({ type: 'ThreeArrows', range:
'H2:H9' });
            spreadsheet.conditionalFormat({
                type: 'Top10Items', value: '1',
                format: { style: { color: 'ffffff', backgroundColor:
'#009999', fontWeight: 'bold' } }, range: 'F2:F9'
            });
            spreadsheet.conditionalFormat({
                type: 'Bottom10Items', value: '1',
                format: { style: { color: 'ffffff', backgroundColor:
'#c68d53', fontWeight: 'bold' } }, range: 'G2:G9'
            });
        }
    }, []);
    return (
        <div>
            <SpreadsheetComponent ref={spreadsheetRef}
showFormulaBar={false}>
                <SheetsDirective>
                    <SheetDirective name='Car Sales Record'>
                        <RangesDirective>
                            <RangeDirective
dataSource={conditionalFormatData}></RangeDirective>
                        </RangesDirective>
                        <ConditionalFormatsDirective>
                            <ConditionalFormatDirective type='GreaterThan'
cFColor='RedFT' value='700' range='B2:B9'></ConditionalFormatDirective>
                            <ConditionalFormatDirective type='Bottom10Items'
cFColor='YellowFT' value='4' range='C2:C9'></ConditionalFormatDirective>
                            <ConditionalFormatDirective type='BlueDataBar'
range='D2:D9'></ConditionalFormatDirective>
                        </ConditionalFormatsDirective>
                        <ColumnsDirective>
                            <ColumnDirective width={120}></ColumnDirective>
                        </ColumnsDirective>
                    </SheetDirective>
                </SheetsDirective>
            </SpreadsheetComponent>
        </div>
    );
};
export default App;

```

```
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}
```

### DATASOURCE.JSX

```
/**
 * Conditional formatting data source
 */
export let conditionalFormatData = [
  { 'EV Model': 'BMW I3', 'JAN': 1224, 'FEB': 423, 'MAR': 585, 'APR': 367,
    'MAY': 729, 'JUN': 733, 'TOTAL': '=SUM(B2:G2)' },
  { 'EV Model': 'Tesla Model S', 'JAN': 975, 'FEB': 763, 'MAR': 723,
    'APR': 483, 'MAY': 983, 'JUN': 589, 'TOTAL': '=SUM(B3:G3)' },
  { 'EV Model': 'Chevrolet Volt', 'JAN': 113, 'FEB': 289, 'MAR': 675,
    'APR': 458, 'MAY': 391, 'JUN': 198, 'TOTAL': '=SUM(B4:G4)' },
  { 'EV Model': 'Jaguar I-PACE', 'JAN': 78, 'FEB': 177, 'MAR': 244, 'APR':
    99, 'MAY': 312, 'JUN': 129, 'TOTAL': '=SUM(B5:G5)' },
  { 'EV Model': 'Tesla Model X', 'JAN': 978, 'FEB': 1108, 'MAR': 1604,
    'APR': 879, 'MAY': 1070, 'JUN': 1001, 'TOTAL': '=SUM(B6:G6)' },
  { 'EV Model': 'Nissan LEAF', 'JAN': 229, 'FEB': 978, 'MAR': 1202, 'APR':
    822, 'MAY': 135, 'JUN': 878, 'TOTAL': '=SUM(B7:G7)' },
  { 'EV Model': 'Honda Clarity EV', 'JAN': 671, 'FEB': 1302, 'MAR': 466,
    'APR': 989, 'MAY': 679, 'JUN': 891, 'TOTAL': '=SUM(B8:G8)' },
  { 'EV Model': 'Toyota Prius Prime', 'JAN': 978, 'FEB': 1362, 'MAR':
    1872, 'APR': 678, 'MAY': 900, 'JUN': 867, 'TOTAL': '=SUM(B9:G9)' }
];
```

### DATASOURCE.TSX

```
/**
 * Conditional formatting data source
 */
export let conditionalFormatData: Object[] = [
  { 'EV Model': 'BMW I3', 'JAN': 1224, 'FEB': 423, 'MAR': 585, 'APR': 367,
    'MAY': 729, 'JUN': 733, 'TOTAL': '=SUM(B2:G2)' },
  { 'EV Model': 'Tesla Model S', 'JAN': 975, 'FEB': 763, 'MAR': 723,
    'APR': 483, 'MAY': 983, 'JUN': 589, 'TOTAL': '=SUM(B3:G3)' },
  { 'EV Model': 'Chevrolet Volt', 'JAN': 113, 'FEB': 289, 'MAR': 675,
    'APR': 458, 'MAY': 391, 'JUN': 198, 'TOTAL': '=SUM(B4:G4)' },
  { 'EV Model': 'Jaguar I-PACE', 'JAN': 78, 'FEB': 177, 'MAR': 244, 'APR':
    99, 'MAY': 312, 'JUN': 129, 'TOTAL': '=SUM(B5:G5)' },
  { 'EV Model': 'Tesla Model X', 'JAN': 978, 'FEB': 1108, 'MAR': 1604,
    'APR': 879, 'MAY': 1070, 'JUN': 1001, 'TOTAL': '=SUM(B6:G6)' },
  { 'EV Model': 'Nissan LEAF', 'JAN': 229, 'FEB': 978, 'MAR': 1202, 'APR':
    822, 'MAY': 135, 'JUN': 878, 'TOTAL': '=SUM(B7:G7)' },
  { 'EV Model': 'Honda Clarity EV', 'JAN': 671, 'FEB': 1302, 'MAR': 466,
    'APR': 989, 'MAY': 679, 'JUN': 891, 'TOTAL': '=SUM(B8:G8)' },
  { 'EV Model': 'Toyota Prius Prime', 'JAN': 978, 'FEB': 1362, 'MAR':
    1872, 'APR': 678, 'MAY': 900, 'JUN': 867, 'TOTAL': '=SUM(B9:G9)' }
];
```



### Limitations of Conditional formatting

The following features have some limitations in Conditional Formatting:

- Insert row/column between the conditional formatting.
- Conditional formatting with formula support.
- Copy and paste the conditional formatting applied cells.
- Custom rule support.

### Note

You can refer to our [React Spreadsheet](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Spreadsheet example](#) to know how to present and manipulate data.

### See Also

- [Rows and columns](#)
- [Hyperlink](#)
- [Sorting](#)
- [Filtering](#)
- [Ribbon customization](#)

## Illustrations in React Spreadsheet component

Illustrations help you to insert an image, shapes, and graphic objects in the Essential JS 2 spreadsheet.

### Image

Adding images to a spreadsheet can enhance the visual appeal and help to convey information more clearly.

\* The default value for the `allowImage` property is `true`.

### Insert Image

You can insert the image by using one of the following ways,

- Selecting the Insert tab in the Ribbon toolbar, and then choose the Image tab.
- Use the `insertImage()` method programmatically.

The available parameters in `insertImage()` method are,

Parameter	Type	Description
images	<code>ImageModel</code>	Specifies the options to insert image in spreadsheet.
range(optional)	<code>string</code>	Specifies the range in spreadsheet.

The available arguments in `ImageModel` are:

- `src`: Specifies the image source.
- `id`: Specifies the image element id.
- `height`: Specifies the height of the image.

- width: Specifies the width of the image.
- top: Specifies the top position of the image.
- left: Specifies the left side of the image.

\* In a spreadsheet, you can add many types of image files, including IMAGE, JPG, PNG, GIF, and JPEG files.

#### Delete Image

- If you want to delete the image, just select the image, and then press the Delete key.
- Use the `deleteImage()` method programmatically.

The available parameters in `deleteImage()` method are,

Parameter	Type	Description
id	string	Specifies the id of the image element to be deleted.
range(optional)	string	Specifies the range in spreadsheet.

#### Image Customization

Image feature allows you to view and insert an image in a spreadsheet, and you can change the height and width of the image by resizing and moving it to another position.

#### Height and Width

- You can change the height and width of the image by resizing.
- Use the `height` and `width` property in the `insertImage()` method programmatically.

#### Top and Left

- You can change the position of the image by drag and drop.
- Use the `top` and `left` property in the `insertImage()` method programmatically.

#### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SheetsDirective, SheetDirective, ColumnsDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { RowsDirective, RowDirective, CellsDirective, CellDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { ColumnDirective, SpreadsheetComponent } from '@syncfusion/ej2-
react-spreadsheet';
import { imageData1, imageData2 } from './datasource'
function App() {
  const spreadsheetRef = React.useRef(null);
  const bottomStyle = { verticalAlign: 'bottom' };
  const topStyle = { verticalAlign: 'top' };
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
```

```

    if (spreadsheet) {
      spreadsheet.merge('B2:D2');
      spreadsheet.merge('B11:D11');
      spreadsheet.cellFormat({ fontWeight: 'bold', verticalAlign:
'middle', backgroundColor: '#1167b1', color: '#ffffff' }, 'B2');
      spreadsheet.cellFormat({ fontWeight: 'bold', verticalAlign:
'middle', backgroundColor: '#1167b1', color: '#ffffff' }, 'B11');
      spreadsheet.cellFormat({ fontWeight: 'bold' }, 'C3:C9');
      spreadsheet.cellFormat({ fontWeight: 'bold' }, 'C12:C18');
      spreadsheet.setBorder({ border: '1px solid #1167b1' }, 'B2:D9',
'Outer');
      spreadsheet.setBorder({ border: '1px solid #1167b1' },
'B11:D18', 'Outer');
    }
  }, []);
  return (
    <div>
      <SpreadsheetComponent ref={spreadsheetRef} showRibbon={false}
showFormulaBar={false}>
        <SheetsDirective>
          <SheetDirective selectedRange='B2' name='Employee
Deatils' showGridLines={false}>
            <RowsDirective>
              <RowDirective index={1} height={30}>
                <CellsDirective>
                  <CellDirective index={1} value='
Mark'></CellDirective>
                </CellsDirective>
              </RowDirective>
              <RowDirective index={2} height={40}>
                <CellsDirective>
                  <CellDirective index={2} value='Id'
style={bottomStyle}></CellDirective>
                  <CellDirective index={3} value=': 1001'
style={bottomStyle}></CellDirective>
                </CellsDirective>
              </RowDirective>
              <RowDirective index={3}>
                <CellsDirective>
                  <CellDirective index={1}
image={imageData1}></CellDirective>
                  <CellDirective index={2}
value='Gender'></CellDirective>
                  <CellDirective index={3} value=':
Male'></CellDirective>
                </CellsDirective>
              </RowDirective>
              <RowDirective index={4}>
                <CellsDirective>
                  <CellDirective index={2} value='Contact
Preference'></CellDirective>
                  <CellDirective index={3} value=':
Email'></CellDirective>
                </CellsDirective>
              </RowDirective>
              <RowDirective index={5}>
                <CellsDirective>

```

```

                                <CellDirective index={2}
value='Email'></CellDirective>
                                <CellDirective index={3} value=':
mark@gmail.com'></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective index={6}>
                                <CellsDirective>
                                <CellDirective index={2} value='Date of
Birth'></CellDirective>
                                <CellDirective index={3} value=': Jan 3,
1985'></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective index={7}>
                                <CellsDirective>
                                <CellDirective index={2}
value='Department'></CellDirective>
                                <CellDirective index={3} value=':
IT'></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective index={8} height={40}>
                                <CellsDirective>
                                <CellDirective index={2}
value='IsActive' style={topStyle}></CellDirective>
                                <CellDirective index={3} value=': True'
style={topStyle}></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective index={10} height={30}>
                                <CellsDirective>
                                <CellDirective index={1} value='
Mary'></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective index={11} height={40}>
                                <CellsDirective>
                                <CellDirective index={2} value='Id'
style={bottomStyle}></CellDirective>
                                <CellDirective index={3} value=': 1002'
style={bottomStyle}></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective index={12}>
                                <CellsDirective>
                                <CellDirective index={1}
image={imageData2}></CellDirective>
                                <CellDirective index={2}
value='Gender'></CellDirective>
                                <CellDirective index={3} value=':
Female'></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective index={13}>
                                <CellsDirective>

```

```

        <CellDirective index={2} value='Contact
Preference'></CellDirective>
        <CellDirective index={3} value=':
Phone'></CellDirective>
      </CellsDirective>
    </RowDirective>
    <RowDirective index={14}>
      <CellsDirective>
        <CellDirective index={2}
value='Email'></CellDirective>
        <CellDirective index={3} value=':
mary@gmail.com'></CellDirective>
      </CellsDirective>
    </RowDirective>
    <RowDirective index={15}>
      <CellsDirective>
        <CellDirective index={2} value='Date of
Birth'></CellDirective>
        <CellDirective index={3} value=': Dec 8,
1989'></CellDirective>
      </CellsDirective>
    </RowDirective>
    <RowDirective index={16}>
      <CellsDirective>
        <CellDirective index={2}
value='Department'></CellDirective>
        <CellDirective index={3} value=':
HR'></CellDirective>
      </CellsDirective>
    </RowDirective>
    <RowDirective index={17} height={40}>
      <CellsDirective>
        <CellDirective index={2}
value='IsActive' style={topStyle}></CellDirective>
        <CellDirective index={3} value=': True'
style={topStyle}></CellDirective>
      </CellsDirective>
    </RowDirective>
  </RowsDirective>
  <ColumnsDirective>
    <ColumnDirective width={20}></ColumnDirective>
    <ColumnDirective width={280}></ColumnDirective>
    <ColumnDirective width={172}></ColumnDirective>
    <ColumnDirective width={160}></ColumnDirective>
  </ColumnsDirective>
</SheetDirective>
</SheetsDirective>
</SpreadsheetComponent>
</div>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

**APP.TSX**

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SheetsDirective, SheetDirective, ColumnsDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { RowsDirective, RowDirective, CellsDirective, CellDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { ColumnDirective, SpreadsheetComponent, CellStyleModel } from
'@syncfusion/ej2-react-spreadsheet';
import { imageData1, imageData2 } from './datasource'
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  const bottomStyle: CellStyleModel = { verticalAlign: 'bottom' };
  const topStyle: CellStyleModel = { verticalAlign: 'top' };
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.merge('B2:D2');
      spreadsheet.merge('B11:D11');
      spreadsheet.cellFormat({ fontWeight: 'bold', verticalAlign:
'middle', backgroundColor: '#1167b1', color: '#ffffff' }, 'B2');
      spreadsheet.cellFormat({ fontWeight: 'bold', verticalAlign:
'middle', backgroundColor: '#1167b1', color: '#ffffff' }, 'B11');
      spreadsheet.cellFormat({ fontWeight: 'bold' }, 'C3:C9');
      spreadsheet.cellFormat({ fontWeight: 'bold' }, 'C12:C18');
      spreadsheet.setBorder({ border: '1px solid #1167b1' }, 'B2:D9',
'Outer');
      spreadsheet.setBorder({ border: '1px solid #1167b1' },
'B11:D18', 'Outer');
    }
  }, []);
  return (
    <div>
      <SpreadsheetComponent ref={spreadsheetRef} showRibbon={false}
showFormulaBar={false}>
        <SheetsDirective>
          <SheetDirective selectedRange='B2' name='Employee
Deatils' showGridLines={false}>
            <RowsDirective>
              <RowDirective index={1} height={30}>
                <CellsDirective>
                  <CellDirective index={1} value='
Mark'></CellDirective>
                </CellsDirective>
              </RowDirective>
              <RowDirective index={2} height={40}>
                <CellsDirective>
                  <CellDirective index={2} value='Id'
style={bottomStyle}></CellDirective>
                  <CellDirective index={3} value=': 1001'
style={bottomStyle}></CellDirective>
                </CellsDirective>
              </RowDirective>
              <RowDirective index={3}>
                <CellsDirective>
```

```

                                <CellDirective index={1}
image={imageData1}></CellDirective>
                                <CellDirective index={2}
value='Gender'></CellDirective>
                                <CellDirective index={3} value=':
Male'></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective index={4}>
                                <CellsDirective>
                                <CellDirective index={2} value='Contact
Preference'></CellDirective>
                                <CellDirective index={3} value=':
Email'></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective index={5}>
                                <CellsDirective>
                                <CellDirective index={2}
value='Email'></CellDirective>
                                <CellDirective index={3} value=':
mark@gmail.com'></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective index={6}>
                                <CellsDirective>
                                <CellDirective index={2} value='Date of
Birth'></CellDirective>
                                <CellDirective index={3} value=': Jan 3,
1985'></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective index={7}>
                                <CellsDirective>
                                <CellDirective index={2}
value='Department'></CellDirective>
                                <CellDirective index={3} value=':
IT'></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective index={8} height={40}>
                                <CellsDirective>
                                <CellDirective index={2}
value='IsActive' style={topStyle}></CellDirective>
                                <CellDirective index={3} value=': True'
style={topStyle}></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective index={10} height={30}>
                                <CellsDirective>
                                <CellDirective index={1} value='
Mary'></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective index={11} height={40}>
                                <CellsDirective>

```

```

                                <CellDirective index={2} value='Id'
style={bottomStyle}></CellDirective>
                                <CellDirective index={3} value=': 1002'
style={bottomStyle}></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective index={12}>
                                <CellsDirective>
                                <CellDirective index={1}
image={imageData2}></CellDirective>
                                <CellDirective index={2}
value='Gender'></CellDirective>
                                <CellDirective index={3} value=':
Female'></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective index={13}>
                                <CellsDirective>
                                <CellDirective index={2} value='Contact
Preference'></CellDirective>
                                <CellDirective index={3} value=':
Phone'></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective index={14}>
                                <CellsDirective>
                                <CellDirective index={2}
value='Email'></CellDirective>
                                <CellDirective index={3} value=':
mary@gmail.com'></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective index={15}>
                                <CellsDirective>
                                <CellDirective index={2} value='Date of
Birth'></CellDirective>
                                <CellDirective index={3} value=': Dec 8,
1989'></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective index={16}>
                                <CellsDirective>
                                <CellDirective index={2}
value='Department'></CellDirective>
                                <CellDirective index={3} value=':
HR'></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective index={17} height={40}>
                                <CellsDirective>
                                <CellDirective index={2}
value='IsActive' style={topStyle}></CellDirective>
                                <CellDirective index={3} value=': True'
style={topStyle}></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                </RowsDirective>

```



```

        <ColumnsDirective>
          <ColumnDirective width={20}></ColumnDirective>
          <ColumnDirective width={280}></ColumnDirective>
          <ColumnDirective width={172}></ColumnDirective>
          <ColumnDirective width={160}></ColumnDirective>
        </ColumnsDirective>
      </SheetDirective>
    </SheetsDirective>
  </SpreadsheetComponent>
</div>

);
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

### DATASOURCE.JSX

```

/**
 * Default data source
 */
export let defaultData = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
    Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
  { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
    Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
  { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
  { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
'12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
70 },
  { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
    Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
  { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
    Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
  { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
    Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
  { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
    Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
  { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
    Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
/**
 * Image Source
 */
export let imageData1 = [{ src:
'data:image/png;base64,iVBORw0KGGoAAAANSUHEUgAAAOsAAADSCAYAAACrfzewAAAAGXRFW
HRTb2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAD10RVh0QUxUVGFuAFBvcnRyYWl0IG9mI
GEgeW91bmCGYnVzaW5lc3NtYW4gc3RhbmRpbmCGY29tZm9ydGFibHkuIDt0SWEAAAAVdEVYdENvc
HlyaWdodABVbnNwZWNPZmllZIRxMzUAAABGaVRYdERlc2NyaXB0aW9uAAAAAABQb3J0cmFpdCBvZ
iBhIHlvdW5nIGJlc2luZXNzbWFnIHNOYW5kaW5nIGNvbWZvcnRhYmx5LiBae8yiAAAAGWlUWHRDb
3B5cmllnaHQAAAAAFVuc3BlY2lmaWVkozTnUQAAs/pJREFUeNrsVQmQpOlZHvj89593ZlVlVXVXX
9MzrRlpDh1IYgUWEgYLIQKwWBlsrwPbBBActoNDDlgbFrBgl3vZ5XRgmzA2axtYlhAYWFhsLiGB0

```

aBbmqt+n+qyuM+//PvZ93i97wBuxEev1tDQj/19HRWVXZf6ZWfk93/Oez2vVdY1mNatZL/51N3+CZ  
jWrAWuzmtWsBqzNalYD1mY1q1kv0uU2f4JP7fX0b36/53Q3bdfD7L57XGdTOu6yq+GfqvKsqi8/  
4u+p2z+Si+NZTXR4E+ddeN3f9TJ4t1Z2/L/xalnPr4brWZekRa+I/ZTZ3Mz8NvtYOP8FdRpliYn1  
1N+9k4QrqzaXsrD/60XhD9x5Yv+0Wnz12zA2qx7sI7f+1P90/3r/2AVR29bLeajfLXstwZb/eF4D  
yFxb2L/1jVMP3N0hwn0B114nQ78wMH44kX4pY3KqtDpbaBMLmkazeZFli9c13/c84OvuPy2b180f  
+EGRM36L1zP/Mq3+5ODm78gQHxDLRXj1SrGbJEgSQqURY3xpbN45HVvxOTGTaR5ghvPPAvHLQHKA  
e4WnP4QvW4H5/ZGsgkq2K6HOM2RphlqeXyymh31h72n272tL7zyBd920vzFG7A26z9zPfl/fJs1P  
d7/1jt3Dr/q4OadywhbsCoLVZzjvspwPjNk9wie28alhy7gjW/+bDz34cfr2drD/PZN5AJIOxC7u  
Cwwny+xtbOB+y6fQ7c/QmlZyOMYleWJ129hemcfYa911B/tfjRP48975B3fkzafQAPWzv1/WL/6r  
s9/3XKR/Yskdy/Np7OgSEukAjrba6PIMyzTWpgxRLEuLiSLdVg5UfP43M++y344Ac+gjpPkUZTd  
MMQSZYhXixRi0PbG/i4eH6M7TO7iISdHa+FoqzQG23ptWoL8MPOTRTJPxGW/a7mk2jA2qz/1/Wz7  
3yzs4qc33Zd+9VpVvVn0yXKvEKrE2Jre4y6KiFYFQCuxJQtcXwyw+3DY7hejcUyxuWHz+PtX/BWv  
P/xj+HGULfx0COX4ZQebty4gc1egGWWYOPMCA9dPouWMKoYxfCCEGVZoi0+bibfa6sWEH+ktL109L  
gB+84Nf+D9kzSfTgLVZf27983e+bSNKkt+x0XpUpzDHghAdOq1Q/uMijieaLa8EffkWIctrrOYzZ  
EUptwtcP5hhKWbyX/j0y3j0wdfjTz/6BK5efRpvfu1DuH59hm7HRTQ5hNXy8fDDZ7Ax2hWTeoVKG  
HWZxHJtHzvnL2B5eiSmsSes66PX613tdDfedukt3/JE8wk1YG2WrN/47ne8qXTb/9J1vXO2fEYOE  
8AKPAQCMcW6QiQ+al7FSFYF8rzEbDEV0OZYJSUqYd5FnOHwZI7YcmBXfb70i98goOzjPX/yFF5+6  
QzmixzT4yOxbifCyEC35+CNb3wdotNbqMUMzsXEqzAbuBJICA+Pb6Bnd09pGIX2/nyaLy99y2Pf  
tn3/UzzSTVg/a96/f6Pfc13L+eLdzphN+i0Q4zGbThuIIxXCpEKuwo048WRgB0oUaKQj3E+XWCvZ  
iiiCPNZiuduHWjwKa5cJBUQRxN8+X/3NnzoY0fi6gyj4TaefvI5rGYnaMkhMESiPHL1LB577DHsP  
/GnqFxxYLuN2XyCs1ceRLpYCdF6AoRJs3EPrlfLx97p2v+/L/9aebT6wB63916/Gf/Wbr9Oj2b  
1ZZ+sagvxGMds4IyuYCVB/wWji69SyS5QRB0AI/uzQuUMSx+JSu3I6ETRMs10v5f4V2u4f9wykmy  
wROKOarsOp4s8L5vZfhieuHeOchl+FP/uMz+PDHPojz4y2sqgKjfoDP/YuvQ1BUuPhkR+C0AuReg  
N3ds7AFvHFSi3+8gN8aYnrrKoabg/lgy/Odb/6af9IA9h4v5zu/8zubv8KLZP3JP/1G6/pzz747m  
q7elJalm2dCm0WK5WK02dEhbj37NCIXsDckzeu38bt/TlOZ2L6+i0NAFXwIPawMJ6Pk5MINw+PB  
NAONns9TKIaTwrTHp8s8MjDl5AsIhQtB+PRNj760edwspyJqSzXqCv4dYHzF86LmR0hShhHsnE4W  
SDojNafbeLw6lUMtvm6zo+CBzHftP+k++5cd9rPv9DzafYgPVTfn3sF7/Dmq1W7+4PN9/AAgXbE  
mNXfEMC1lHZsNVGuzWALX5rQaYU47fTG8D3Brhx+xDz0yUKR0wluU8rFN9W7pGmBSJxRrO6RMdz5  
MvH8STGYrXEW5d2MTm4jb37ruD0dIUb+7fhCXNmRY5ep4Xx1kB8YxeTo4UCeB5VcOQQAmlzzmen8  
IMQd/YPUcjh0061AzjBG+e33/8rey9/U1NacY9W03XzIlhXf/0HrWUUVxtl8YysjQQkFnbPbKLTB  
soqQsYAu318KubsXMsDu+MORru76IUu7HKO+85uYHT7GxkrmOZLXNs/gtvvYbC1gXa3j8PjBU6nc  
2R5jNGgjY88fVsA08VqKX5vvMTFvT1sDjfUte7tAPvHclY/cwxXQ09128KgM3QGffGJhckP72C4c  
w6LKcuJfUymJSyrhenJdGzVxa997Dd/uNV8ovdmNV03L4JV2PZ/GJ49+4ZBXgvYTPeLKFenM2G10  
Y4EpFkOHMxmwmBbKCOBXOAJyASYrO0VJnSE+cJ2C47jIYlW4lsCp2K2dgdDVGL2joQNT6czJGJSb  
2xsCVtXeN/Hb+Khc7s4uX4V/eEluUaJtLIxGPZR2h5u3JniwkZfrulimZbIj4R5+2MBEcaV10XFY  
95t4XgpprIcMEXuYLGcXW53B78tv/mM5lNtwPoptz76y9/zA3m8ehMZMRXgMapbxiFIEwGhfDqbw  
w6WkxN8fgdpKSZslGD/QFiPJqv4mHMBZy4PKsTw7Xe7GPxbYrYWCjJSWUwR9jpYLBZa3FDGHm7eO  
cIwsIV9Z7i8tYl0dYozulewt3ceTzx7Dflxgn07Y8xOTwSkY/i+J8/VldxunWeo5MuSbWOVlvzOV  
TN7NV2ZqPE0uuaocM37/vfv/qP+2+/rQk4NWD91FlP/eb//Bbb9f9ukdEknQozRQIAH92Nbfmew  
fFWqKsputsdJGKWxsqVYmFF2w8wEXCvkhi2bSv7rZyPdg4WiIsCAwHsroC822vBrSwMh0M8deMQm  
50QyyTA0XwuJq+H/ZMTnN25gOXJbQV91aXwwhb2j5Z45ct2xFcOFMxJUSLodnB85wYuJ8RvRkt+J  
gcHzePeEGIQIFsl4sseI6/uD5K8+MF/92Pf7Be081Nf/LXf3/TLNmB9aa8//flvf3h1fPVfdnauB  
EHbx8WHHhV7WEzRpZi9YuIKLDEa9bGluyM/S4SxNgQMEzVvZ2Ia91q+mMhTAW8qzFpgazBCu5Vjv  
lwiK2sczlJMfymGalzmYUetLk5WC7Q8F6elgzitcelgCd+6hQuDECd39vWgCD0B8+EpPO+CmMUZV  
lktLFphNV9hvmJHjyPm+amY3V2w0yfpC5zI74b9DaHeQF4j00hFfzWf/NCzdybf+sQ3fPEzD1y58  
IVf8vU/2rTb/ReuJhr8SVi/+1Nf+Vo7q3/r8GQynh3egtseIIummc8mwlqp+JoD+I6DZHaCZHkIa  
LYP2xG/NLQx6ngYDgJ44p+2hQXFWRQz1dNC/rq2kGziEPsOytwS0liWmMPTYdJQOGRJNRdxJKCS3  
lCFFvMMZ8Ybwt65XK+FWbJCkWWotTrY3RbGzOR6qTD2/iFy8Y9dJ0Cv18aJHBbiImO2WOLA9ROM5  
PW6bil+rC/XdcSELzCbz93r12/1w1b70mjQ+8qnP/BbX3D74+/5N/c99t1fSwMasL4klu//2FffX  
1XVbyzixZ198RGjQMbR0QJpAiznOW7dmmD/1nWxeSyty12sxE8UxrpzbR+3j09wuD/BVMzXduiiJ  
2Ztf9hG4ISohD9dgadd1/DF9E3zFB1h0zovxm9kF04OX+4TCRtmwsTMxdq2g9AuBYQVqqKcttFUN  
vZ2t4TVu7j2zDPaQXN6MkEl4N7Z2YJlQ3zUCI7cEFzjudv7GPQ8efliC1eV+Nhy8KQRjk6m4J0Hg  
z6mp6ed08Pppdr2v/raR9490Hr2P/6HC6/4C81m+M9cTQXTJ3C97+e/c5CvZo+Xln355PZNDMZjp  
DOCKUYRz5QZmVNdzieYzGbY3D2H2ekcz159BqONTYibiTqFmL0pyqzUIJMtaEwFSLblCmE6ci3IA  
ZCIaRoh158lCVveXLgCTJuQthws4xh+K0SS5mg5OcbDF1rdEVj/kAoCd89dwP6xsOlygU4Y4AYBK  
ex56b5zKASQ85MVNrf7ODhNcfXmdVy+uIvldIKHHnoZ2p1AWfhATOmWvObjoyM5CFI8cN8ZvPxVn  
47Do3nabvtPnj1/319649u/9qDZFY3P+qJbT/z691nx9PD3b9189nJ/cwe9jV1c+/AH8KBs4HxmY

UnAJhXKtMRCfNG2HyLav40zF3ew2X0M+zdvCZhqCBxRlZb4iAMNCNm2i1Zdivmai/9awPPbaA8Dn  
B0PsVhk2myeMoLrOvJpO5gu5+gFHQGUhTpLxFzOBcRtOHmpihH3338Fd25ew/LoUIDaEb90Jc8hp  
FlE4rvWqOIV2nJoHO2f4Fj84/vPbsIu5YRwGC6Wg0cYdrHKxaxf4HSxUvB2O0N0e1uYTg6QrZbBw  
Y3l0888+fEPzWenX/EFf+sf/mqzOxpmfVGt9//Lb/6Fk+PFOyYnRwjE9/TaLraGFxDJBu5tnUUmD  
LgSgJQCrFL8xCpfIRQzd7ZIMReW7ThMzzji16Yqu6Jtci57TwVkj1zPDeD4ngZ8LPFT7fLPRB0y2  
qt2C5H4pKxgXEaxAL9CIvd14f/OlgBbu2cUzIe3b2E2maIUBk7kMLAsSx6fYndrLOzbUskYgl8cY  
1Se+MVViePTEwx3dmAlMfqDDpYZD4eOppPGoz4u3ncB8XKFqMiIeq0xptmOyppfeuD+/+mLv/aHv  
7fZIQ1YXxTrqX/3vW+ZTQ5+ZXY4CeaysdvdDbQ8W0xR8enGe0jiHFU6U6mVdHogJq4AsnIxOThCv  
EpkW/vwB2NhNTEtHUuYtQQLC23PR9jqwGu11YdkUUQtLFemS/lgC60rLitbQFeothLb7LJKuFn8U  
gKpLM+elb9yDqgZ04tibbonXjSuU5srLS1JB1lep7iq0tZnWq+V09BOS+sZwabhDIAVJqZnQx18LKJ  
18sg9HWCp3+UNUrevlQAJuImW3hdHqCcBCi5bbR2xnMz++ee+dbv/L7mrxsA9ZP7rr+h/+MluuTe  
bS8TBMWUXQJvRRJypeRiZMVvwxV+192lp4QNMtAqoiIgu1Ndob2gIcq9D2uDAMEbYD9Vld1zdN4  
YxHCTOyha7U5nHhrqJQH5j443XJlLYwYikALARwhbA5Ga5MxGwVBuUhwZyqGNMCxFyDU7ka1PUCN  
YV5/VzAmqtWk7xuAl0YfSGWAEs1/HZHbsfY2hjJ6+vAdhx9PvrJq2iCC+f20GrLoWK38eyl4ijH  
Ifinz/46MPz17/hL37da9/6t3+u2TGNz/pJW7c+/O6fWS6iyyhjBH4HLTBdt9Nj4FTLCeMoUpEyI  
S8BBzQgxN+Refu9noCwFCZuiVkbaw7UcwkeH05gC601FUGsN2Qu1RL4ZOJb2gK02q01guvw/qWLW  
gDuC4hrDTPx921huVjYN0MRpnpofL48L0lmvyXAFp+2FKA2HTGJbQ1QVQXN2wC1PGcSjPrPrS0Pg  
2FPCyOsThcbNK2zSCPMY9lKbicYboyxMx5jKe/t9sExzo83cP+lHTmgOhgf3MLHP/ihvhv4P3Hzu  
Y8c/OWv+cH/q9k1DbN+4ln117+7fe36jeeWR3fG6eREfb0zZx9ANDvE8Z0DbI66AlTx4VFZ7M5o  
mWs5u3m5hgdMZhd2kKvI2auANelMew5FOUWJnU0YOSIz2q5tpq3tQCzyJbyVQpI1SPUfteayBcmt  
ARURelqKkY4kz8SgNUU/Jb/CZOmKwEs00Qucna002/2PO360VM9EB80TfSapfE3kYvZTWE1MLIsQ  
IwzMchLIKGTfZdLiPM5TDqy3sYbg1xZm8Xi2ksrL3ASMzqjhwO4XCE/dtTPPnMTXQGrZv3PfTKh  
/7aN/0vq2b3NGD9hK4P/Nw7f2dysP8mT9AUJTU6smkFAQI8oda6w+RwH7bYqKuTifirgh4BQ19My  
F5godPvC0g9BAIEulzKbd+AJhTzUkDiOMKO/F6L2VskmmXNk7kWQVCWlGas0K4wo6WFExCwEsQl8  
66VYeJS/FsW5cMKFISZmOcZwS3sWzKPY1WocgEeJdQERAxspdFC2dZiHphNW14bsZj3Jf3egjKnX  
ZRiky/kEGIQaraKsJBDIJAD58zuNrqnviqv4uXsjrhZjNAmQ7GOHq9UO8//EPveef/d71phGgA  
esnb139tXf1ju7cfiadz8dVnNwHIIjH/HkGO1Q/M2whXwWwRKztczFRBbTtSWs2ml54td5mgf1q  
bvkHWAYyPaMmj59RHVQawd3PzurFACKuZnHwqxlJj6qQFFAXzilgrJe6wor0wrgSjFTa/qthQFuR  
dwJSzMqxiIBULWYv87j8R5CXVS6vdyA+bixgjbR2wg3bch+BsbYeQvzjWA4ayxfTWg6mWA6MsNPG6  
dGR5n6PFgnmrwsddwc9LBzflNeq/i/0VRNfAbGHDHbxtbtuH0wSg9PJ5//TT/+W/+h2UWNz/oJW  
fOT/3+38mR89NzTApUIuy8kSqliM8XgbZx97JZaHd9AR5ownU+xevh9uOkdbfMEgdOB1u8qcHlnOZ  
fBoAKclMKDyffVvLdujbarsKDt7f7i8gc9gNU8JhRFi+PDGNS0EWEZSAEBpFaSdyuxaTuBJT2NGGE  
rWWFJAapBKflswppOnaofrYTiEHg7A3U7V14CroLddVU9sucn181+8jErBbYilUWYFITOkN8VOjx  
RJdVxhcLjidzcRbdhFFN3Dm3I48R4DjwylaHQ9nzuxpWgl7qxecPbv9k3Lph5pd1DDrvWfV3/gBJ  
4/nzz37+B+cW05OkYgpOT2YYDzs4uL9F9Huj8AQ7er60xjuPQA/LDWPYkJ6v90T01L8UktAJ0x1O  
T5smqlUjhCTl1f6iTRkX+qQCZitfqknJv1bWBzNdQ5alymEhDMpVrc9kMjJEJy0LI9XCQgdGhOWZN  
GpcMnpMVhVmpn/Kw6BYp21o+rJsUV6UXEH8W8vV4gzuh/qoFGmr2m0dwbGKhYH18Bhf2MN8wmaDG  
ofHU+SfHSiP2FGPwLdx9twenEoYt9fHAIxMLMx79uJ5jIab6eaZvTc9/NZv+KNmNzXMek9XNDt6y  
2p2PPbEv/SCWMO8586IvxaUCJ0cgVMhPtra5u4WOpsj+GWivaF+KD4mTWSCVUBBhrPJoAiwi2kdY  
Vn6pBodEqa0S6rl9xhJkvsJMxKADA4J00F8WccLFmjUP2RKx+M1GVxizEh8Zj1DFCB2wbQQ1uY11  
R/kYBDWL3hfpmzkGjSbeV/mXCthZDK4K4dJHC01+uyIOVsvp3BrT30++TLF6eEBRhvbIE+PMJb3e  
XByir4TalEGK6+W85VGvct6hW6PcqcjeQ+++OsDliwy7/pYs5v+bDWyLvdgZdHiXbZtB57nqILDe  
HuEjS0Pg15HfFHxPdNCNuYQw/FZbXVrt30xf9vw2n0BbR920FaT1hZwsYJIA0U0g+Q2VQ5rDTZVa  
n5CGddRs5i/s/xQ/NuOmKm+YFZui8/p2CYgZckhYlTm+zhaRWQKKVytgKpdgtHX3/P5XAGV5nBp7  
trEMOuPK30NLJTgl9ZUBb7Kk3pMKzmsomJreqVm93IRYSVsPN7agGeL9RC4WnG1MeoJs7K4I8XB0  
cT4u5api66LFVJhe7/d3f3Ib/zEqNlNDbPes/Xsv/9pxy7TnXgxEVYrVXTMEdoKepvoCRMxIuzI5  
nRtH63RBGLWEHmbsuEFaPQR5b41Uy0V86YCUbsnP69UfV9uqGlqkQbJusKOFX9cGO1gjdIyIMV0i  
t8RQmGerV6WC38zFsoSwZrRYQMm2OHEZ4eamY40pGoqwab5X71t5jqodMsZEYcqb83SX56aAmv6qY  
AGkPmcrLWbzSg4R+V+aaRTbFVN7Oo2xcE9Rb7TV7N3c7CE/ZvljIn8H+Vu0Heyd2wHDY/+KK2Sj41  
N9t6GApLbQnGMX3r01Ka2stu6v6vxarmwhYYUGhQibyqq1JJG3C5sSMPI71gwz+BO04AhQKwJMS  
wYzAVJLWfdoiiodhCgKVdjCtZov+sbBtLf/nczHYdNccJiOT8UC9RLIfTWcq/NdkT0t9XHYNYc38f  
B8uKyTA/tsWtk/nKpMaBLX21+6MO+LT+vpYX6yOd18AGsiLkefsjzFFKhFmTS0tDNnZ2f7cZkc1Z  
vA9W8Vq91201+37QReV00VSeuKLDgSsPTF3ZWOKyel0QtmgBTjtDmRHKyNVjr8+Phnl7cumb5kca  
e0oE1peS6uHjMMJbY0jMMmoHv1SoToNFhJIntvlfDWNazV/qazvmWsLuGxeQ/xgW8xkzq4hoyvQ5  
KsSELHmWM1mmt0Ege3qbTK+s76v+r6WqjeJrkv1YT1hd1ZU8XriAyAU05qjPhZRJP5pJAC3MZsuk  
OcpEmoSxymOnruNo2efRRJliJcn8LtdhD3xXeXAOTk+7PzTf/hXnGZXNcx6b8LrcF5Visk6W1ISx

cXuxh78OkZ7OFbTEGLmscDB0ZoCAQZV0dz2uvEbWoigPqTtqy+qaRSatRoBEojQIQ4Lkz4RgBAAP  
Z2uMVyp1rD+XMND9GULFfpWosW6MF9t50y7eO76oHXto3S0Onlt8lZqxtPcJn27lY2cT87nsNkfK  
0zJHlq5D3O6Fa8hh46ViWluV4aRmUMua402V/I+CwFpSw6pilVXDFSJCR7HMQZnLooZPMZikeL04  
ONaVly5A7Q6nX6vJ3844E6zsqwvuDr4HDaycs2jhcVdnY3ZUP04fV7yoJax2tTgTlSc9e2Q1YXK  
ADtNWNy3KJdGXOSJqajBf68f2WYk+kVAasGnpRRy7WfSbNVwGPl8vtKywGpcghjNBuWpK8qPnJts  
yoKcJiuqY1PWmnlkmFma82ayqJ3o8laUWzr9Up9kPyPKR45RDTAxfOAuod+iMApdxkxp9PCahZhF  
rF6sCNgl8c4K2FuV3O6rcEIVn+Ak9lCGPdUTXDmks92N9DvsZd2EYin/Tny4KbAvwHrC7t+4wftt  
pPk8NPkFJujbbRaQ6TzCcJOH36rRyNZWC3XTWm1XGVPCvDSH61L06PKnCqZy5i0tqntJd8JK7P+1  
+CkWr8ifF02qIzOYv6y0hY5vd8a+mreKuBLyYorqgpToM/UD69NNmb6BbWauPx8rq+BeVzHpGkUk  
Lwi72tXpqeWbXrMs9KcV8vyws5OUWlnULRciPnf0TxxzMAT2FMbwfNN51BeThG0+3KI2EiFnaIGw  
drmyeQUErYzuhlYRfWLDVgbsL7wwaUk77Q63SCPlgiENE8P9rG7t4uwK74pm8HddUTVdZ73Aa2aB  
Qqe5lBpddJlRnW8tLSMUEFZrs1jPkZzq5YBER8rq2o4WJjYMkxM01qY22JahiBms3euFb8G+OWap  
Wlea2Co5uVnHVR9d2NYyPj4dceNtkBnBFmB/+fMfprRAtSCQsJXvCf1150UnrzPQg6BtvjohYCU1  
wjE9z44mQmIK4w3PRzvH8AXk5pCa0cnSy1JpO+73Ojg7O4Y7Za31+ysJsD0wv8xHZxfLiaBK+yyF  
P8riiL0ugNNgdhrs5FJS7OhTX5Tc6g0NS2TdhGHVoFk07wlSzl3yCiusmZhghk76ZLXW/LJh/e7Hq  
PzJIgYn0OvVAoBSdcyqtS8qP5NDoyqNWLe5LtSn5aX+rJqtUnPc+LOuif7S12WBxvNvVolzE3le+  
718BfafZ3SyNNv0PBNV5vXZ0cPUzu3TFY5nS218ZzeS59uYnkxQ5rb49xu4c3CC2zfuaCCqWQ1YX  
/CVxNmF46M74pi2cXR0gr5sOqvOtdhAfUzZuJZd6+anOUqQ8rtubJJalekUOJbzEaTq165NX1Spm  
tC2sJOyNDtp6kQZVwNBPATWwNH/q3dzqI9LM5blhxVLCHK5K1GwNLGkqmEiJm2q4OXiYzW9w/PDd  
tbXLjRXzCIKoxyhlf9m89QGqD57a8nCDosu2l0wQraCPI6NBaxVJoOn8hp6/RYG/RAH0wU8ObSy1  
JQtnjt3DlmeYDmfYVfciP07xzg6nfjNzmrA+oKv+elszKojNm7HwniBW5lyPfqZtqkQIQMSCIZtb  
TUzK6yzE/RlCV4GfKdyDgoiuy40r2mpnZyrr8qmcQPoDBZ7UqvcBJkqcxgY4JlAEKJP/Dnrhym/w  
gHLGoAqDKgLBnaoRCr/1wNE+2ANYKHBpPo/2TJkdGtdJPE80/NwcB31jTWAJH5AVcR6/yJLhNQDV  
VmKFjFLIvuczSN/m+sHhXhv9BFHKfIoRtBp6yyeQn7XH44xn2dZs7MasL7gq6iq+XhUms6X6GgdL  
RQg6jfwl1rPba+LHEwrmqZvbCPpolKhbA0XRlLz0w617UyL5+kH+i1TzWDbz4OeZqZ1NxnZmXwrq  
rVparkmUmyZ3lNNCVHL1Paen3kry36+TU5NYdyNDGMNWE3w6GHBi2mk2VpHp0sTyGIQiqDma/Y45  
DLZahmjz9tsMKDbTb9WQMr3TOXDPEkUsAR2FC/FN22JRyDMXBrNi8ViqamdoN0MpWvAeg+WbMQZS  
/BS2Xy65cQ3y6K1yiD9Q5qYhRb1CfqcZTj107nhafqS0Zjaod9KELH21w3MdZGtKwKYhRG2EejWV  
jlhrHoNYGU75kUZWbY82CxNpL/phNoIoIUQTNUoLkmn3201X19fx11GVfeTzrpxTRW8bKkr1oGpq  
qiNOqEcBlrPyy/9uZjUaarBKVfYlJIytgavasOulaOPCm2stKY7qH46RPx7VlHzZJHpn4mpzNc3  
z/Fmk5QpA2xNtHgewFWx741n840tTKdTrGadNENhRhphzJnCWPimiKEWjtZbMtam6u+Aq1em8a2K  
qC1FLR2Fa6LiBjgSU0+NaVZXrqfkiDSgn9X/U2NImuAKHYUTdlHGkXlOdt0QVJRqYcG+RpUzTb84H  
pUkHBXxrtTnJUqhLI+cdcXrPK4WZJj64bK+2wS0NoVpwjNwxuCTU2pQqSVmrTVJkDKnFdZrcx3aX  
EAwZnmNQDBMFmb7nJlFmuaZrMQ/LxPs7D7YbXZWA9YX3kwp8VxaIs3zNGCDd8m8I3OUjqvstLZGh  
YGoUWiZnlQ+0PGf75xR8DKaK0xKlQYW9GvFD/1a9VVjUxRBM5i6vgw++fJ7YWayaeEUE8VAQwFU8I  
CoT0a2Dgfi2Ky2GYIcNkVgyGMQGcwGIVXkajLLVbi5M5JlPSb9WLYJSI81Wve7S0dk6plBCSw/hP  
D8WkoeVrdfythKK0V/LRKy0qqpi0E1eayZnSkvM3/lypjKmPLAOpqcIggB+nOfkkg1wo871D7/Hu  
/DIG/LGDG7WC7YsL7xNBlsIY5A5okWENmJjdW6KDZhCYWECZVg0mks27KHiAs3DWm7DH8IK+oAnt  
70WyrBv/FYmbtmf6rXhBF35WVfVI2q5DW0EaGmxvsXbNKXlWioDSi1hBnXYtbbqow578r0D12lpA  
wH0MX011Wu237FVzg31MIBGq01Aqa5sDVSZwJcJoQmKIq2BtSmv2k61Ke7gZAAFOn12z0JHQJkKA  
EcDeX9yC1BbqpD7ruTvM6fekzyqFXAYdAnfC7QRn4x9cnLSn995atzsrgasL+hq9YZTL/DTQPxBF  
u1HSHY9/X0tatCILLtWCFABhiolWKZ8jxHYgr6nMq9rNH8d46/asnEpDapsK76e7GS5LYD0uqqbp  
L2rBLGYzAwoMXhEUFcsA2R/Kx1Qb/18ch0W79dMrXQobOZpfTCft+TcHCfQQcmmespjttSkmdy1P  
0ybl4dMYyOr7qaJrPVBdDd1RD/XrdiWV2tFkqVWvoe0FNOWiVXf1TY5pnqWSSpsb6lg+Hy1NK13Y  
ppTS2ogJvRyFgVHT69eanZXYwa/oKvdHSXR6WGqKQzxOa5pBU7VBD010x1jMmrABBGY2uYlu/5K  
jtak01VBeL/ceF12Z+CIPchKGhK07xWhmYetFJJF/qfJpZbaSdPrSauq2Y3/dWa/a6MBivrwUR2e  
WbzNTGqy5RQsbY42XYnZrXNEZHiH9vymIqlyRrBni9Sre676R4+N0FbZdrzqqY4AWoz/1rr6A2+X  
t8KtNWuqii96qNsC6OK+TyZxFr1xJpkzqj1WraYyT0c33mmqWJqmPWFXX/1W39YCMpdMcxDxfoki  
5Em4utVJkp7t2pJI71iouZppvUO7EopqJsvOVf5Kl2T9yQAlhVMFRXyEZhraAl/ZdQHFZYMVhkgV  
sKcPBCokaQdpPQbyZLaeSM/810Fa81Zqowui3laesYEJvNCTGUVYmP8uLQ0VcMvmm52AFXRLcOih  
1Ib4kMDeEtfS1720EOFOePS5HzL3IyTpD+eyTnAie25uAKuMHacVGpZTGaJMG+lxM2JBLG8NdYRM  
y874/jIZjXM+kIv2d9LyqssV5GqA3JjctZMO/A0faKtZILQbDZR4bPCq5ThsjxCcXjT6Ckp+9YaT  
aXMCLmsjgDD6Y7UF1T/WBvASwVtncZwvJbpXaXaYgm6emqX/T2FppHyxUxV+jMxL5nvzMXUTgUMe  
VabiLCYxnkiZmiSaA2xoy1tGVJ5HC0FT9NALprkWom1zsfWMM3xrFwq6kKB7ZbQSiT10VlwIddhA  
Drk4KzSjPQotY3PuiZ14Gn/dImNXkeLPhI5xOaLBL3toQ5zDhzrVrOzGrC+8H9Q2/43wq6PpFkVZ  
Mul2MaB1r6WTtdULMmmnd58WnyyWoM/QoeYnk6wiiNQDCJOBVCUWmEUVcxECooNN4fyOHdFchiH7  
e0LApY2ydlqGsVSv1RZNqEJG6npyRRJVVH401ZRs3g5U7kUDp7KxFy+feOqHCicIperEr8tjNsK5



HBZztUE3djsY9Traamj4wtHL1bqs/YGfTHt28bkxZ8pH6qKPxvrs8oo9pP0eXCoNrExlwOPFoS8T  
Dko5quVstxQ54k+albwdeVotzrCqLGyLv3ZsmyneVmdNjurAesLv/zWP4WX/H3B0phiYrFsFDIrg  
y40dVdH11XVj3NtCvHd5tMj7XNtt8Za+RTFc51gzjEafleYJVmpYj4DMpUvPu7JbYSjXfjKpH+us  
J7+ITt4yIDMt1pmHCPByRGsY7jQ6qWy9BctSsS5MCLLF90KSTJdtqrl+QVAcjjcd+khnZR+49YhF  
skcw56HXuBpoCk5OUW3m8mB0VoPxjKdPZVZaaqm9MT0zep1wYZtqpZqUxvshj3MxY+P5VDJK0+HP  
qsvJodNnifP+7xU+GcLXSLvubacVN7DfrOxGrC+4Ct0wsmyxkI28ZjscCpMpWMr5HeLySGOnnkCv  
d2Xo70xh97WrjFp2QXD1IrY0OlqidnhHcwPbynj+Ny0nHMjgLNdMQ9nc5UntYYbykrMoVZ2qfpJK  
rhtJV02yMcybZSkC6TCZE7oCmN6qswW8zVHLWCLjx7hd2dswhabYzPnFXVRZff94spZqdn9LCh8  
He8OMViOZXnPxWwJ2jJ8w+6woyBKbSo67W2E31ciqrJz1k66NqZF1TkBOFYpXW/mRxePqPaBY3pV  
JjzFeujXOs7iWUhr7cdCNN7Pf15kgqY183OasD6gq+/8T/+s/qHvvrzDwQK192ixqA/guWx0dtFd  
LSPzb2XYXj5FVoo4LgsqrelZJBVS2QWP2yhNbqI3Ys7SJYLM91NzOksilFGU/19V1vcNPhEv5Ump  
oC01GYBqKYv1fMJIA6bUnEJMSspE5pWC238Ho83EXS68MXs7Q4vot8bod1tmz5bWraUZgl6GLQ98  
VlXynDV8LyAfl1k2Ynx+LzRlityjV/KQZXr6uYLnJWB0W6TvK8404FauZj1acUG9Fi1jAlK1hhHn  
IHj+8LiprmAPIylm8SV0MfXJmi1+qof/Hdls7MasN6T1e8M/17ghr99uH+t3+n0NPdJP3K0e1EYa  
UtM3MIEh8RcZN5TRRY0n5QpENMWlwsgbaeEK8Bz2mxB68iVe2paOr4pbmf7ndYQsTopj7Vggaka+  
pnaKM70iPikpfiopvHbR6vlaI1xd7PWua9UGGTfKevlB0dFZUxXDRUsQg+B1Rar3hQ8hCwNdAv02  
jtYRQtMJ8da96t1xgAr3PJlgoxulrCqJIXAthETN+VfPUZsRZAp3IgJGLiO+2hmMXJ8yMlQ6+vY  
0RcFV+z0PJbTXFWa9Z7tzqDrT9tT2/Oq43Nvh8GWqzAQVRVRbHt0ohl26YUjyMu2DJGny6lnEmSG  
zYVPzW+c0vM4p1Kg/Y3B+hvbIrZ2UXN8j0BsSWmoq15U1dLEAl+9VWTCBmVGTjJPDN+YVWaua9Ka  
/QTBZgc9pgsQ2HZjhYwOL4DThHQFjoeApp+LdUvzeNigz4dRrAJ/HakwPd1viwDVzTJTWueAN5nW  
WgtLXE8XJhfzVI5gNajOFbim1MBkQOt2LyeC3B5X0a/3drGuVaASE6asCfs3u82keAGrPdu/fXv/  
NHyd777K95rxYt3ZIGw199WFUNOKFc1Q8voG1XCeNnpIRYHh8JiY1ib54Egw/GtG3jm2rM4vLUPh  
72fgp3ek/tyCDi4+Oij6GohhK0leXVoChq0cohqgvMj5LJnZLmrQErEFF0tZwIMMqCjOdt4dh+hs  
GN/74r4ziMxT12NImdHEyC+CZuHRbwQ0AXoDQMdhux0B/KcpmiBBRYWSyoJLGf4LeytGsIgg2haA  
FHm60ixGYLFx5Xynr115FiLNKp6XTNtyUuz0BO/OU8T3Lc9MLIy/d5cyPzvNzuqAes9XZc2B19hZ  
+M33YzzMUdo2BxJ4VqmNJDleeK/pbefxOEzt9G98hosV1M8+0e/jj/6wNP4WDRE99Kr8Ht/+D6Mt  
vq4b2sXrwtiPCIG4fU//mOcf+1rhAF74sPONcBTCwvb0SmqxQnKkzsoxa9M40rMyxTz2UquzZGRF  
pbTE9y+Psfe2UtoDzq4cXwNUZThg8/cQNEb48Yz78PZURcPjhw8sjuEn/s4eWKCKnkCFx59EMOtL  
e080FpgP9TUjK9T7tQmV23imlpNDHWzp1ZY1nOMIJvvMohUmW139F+zGC3HU7Zm/tWxCGRLGwOYm  
1dF5fP78z3o/yJZjc1YL2na3dn/E2R0+nPDm6paUm1B21hE/PTUpWWBU6evYrMbePd//ZnMukdX  
HrT2/HoF34hkoMFWhsPIpd9+Kpa9fwps/7Uoycqficc5RPvc+My2DXjTAzlhPTCBATUS0XKOK/+  
ZKc2bH41SWmSzE/5YBYUt+o8LC50RHfcopcrv/6t34ZThfCbrcjPP3sBL/zS7+Mhx57BivwIp74k  
8fxuu0Qoz0lHsAfn8MxHn8bOmWpsnD+vHTYlFwqqQoh5b6W1qirSBC7K4vkItapW1EYnzhWWThZHA  
sptVfY3LYK18XPfz091Oyx+wBk5BEKfg7kEsMkqe9X99xfNbmraes9W/Yf/2gqRf/mZM/cF+9M7p  
oMEsWk2Fz+tXC2wuvpRLE4j3L79NAZ+F3ubI1w6fxbhA6/CH/zzX1K1v0cfeQjnL34uqniF85f7G  
Nohjm/sYPax5zC+chk2e8zWA5MhZis46DirldQtFgvxXYVdBbC5tULOpvjpkUt/0X7tG/CKz3kre  
ruckbrCwfs/KuYnh2Vt4+yVV2Jw7iGczXaxOLmK937gTzDqOdjptDE9mcITP7k7HGnkmC5ona+M/  
KnWE5suHR23sVaiYM2zJ+zJlgFPm+rF0m+1VD2CE+FZK8w2917oYStoYVPAT62nthui2D/tDax6Q  
y5z0uyqBqz3Z1VVR+y8HuxUWEXYxRd/1W1g6VzAFSA/uQW/NcL1N96P+xiUSUskwqz1ZIBsA7+Nv  
7FX4kZ0B488tCEMFICe+KGDp2pE02NsJPZw5uFPE58xkksRHB0d/VhqY7mZxVpbRta0EnOYxbiZI  
JRDkY+eu4MxpwEsV1hc/7gGfzqdHQzyCNMqwtD9y7fi9p1jJNcex5U9MdfTmUqkTk6XSIWpz9Vd+  
ELrtTx3Z2sEry9+rLB1qRVatja9c54s576atrPcvzi73eNsG46XpLiatZnTWr47aNmVgNfHdqeFv  
W4LroC3x7Y++ZtU01X/aHH4d64A39Vsqqas92RZjttm+1vgpGgHoUZaFULUOKKiys55BP6mdtuUL  
LAXbgmFJRkdLecR9vwWztQJiihAMSnl17q7S3gPPITelcfEHBaWWh6aJnVHA4FvAttmavrUxXgZ  
o8GGSqkmLcANCLKPPKZr9fKJ7/OsHzfH60+/qze542Vi0e8GLOrH8ejVYXWZg5b/F+nb2MYbAsYB  
0Y2xnfgkA3ltVariH0BRuBN3oKXZsjEnrDqdSE/c75sqJfDyrfMNL1xv68MSx83cBz0A08Y1EFP/  
O77BbADOQh63VDnxZYpyyftQEYML0QD1gas92wVmW0VVEcENS6c24NNKROXozBCo7jPQchagkeY5  
qYotzue4R/Hf+iHaoSVmrGx+29Ts2htb8kFtQ1xgmOOUKXjHXyHLtr8XTtGF0XZRwtX+VQRsvRztNB  
WAtOfNcUzYejHjrdDQEx87VyoDBAJOzbFnb7sVGLoWQytmqNpQzpmMmq8v9syZV92JvqSpXOCn90  
8Kkn1hVte4UstnOp5Fft8zmEC1vgaEwr6s1Eo4Gkgj8jXaAUF7XhXaIoVejGzjyesQjFrchWcnhJ  
N9Dz21kXRqW3ksZ2MosH2mwsYlhbywbPINFGRQ2mlumWgeuY73NSK6ClHpMqqxfaleLMnAQGrkUF  
jjYLLpuIaw407yn3e0JILt6LYv9oeXCPDV9R3vdrdPKhfE89ScDYa9c7pNNbiORawFDLR15UdRyH  
5edcT3UdlTMErdhZ5KxOhgKblfkkSwc04DKLXDx+EoyXXOVqfkFCbHSz03LO/+GSrN1YZi8sZyf  
nCq3DLT3nNhVF8jzxd3N9EqU4xoAut1jcB5Ke9B5V2ymddsKLOaftZ7EWDqbyb15hiV10coYAVb0  
GCUAnV2DTdzYR31yhICEiNSn7xvDQpy/JYSOH12mgPehj2LRWKcCjvwtGNLORflxxWqm/mmpRK7  
awlQistVagF1Ghoax6z7dnYsDL46TEQHavsqRY+6PS6zEye055UaN1uWa2b0hnNdRzV+aW/qa17j  
mWmBFBreA302qrWsqrlluFcnl/Yv+u10Rf2bMkb4MHRYhWW3Gtva4jVnZsqQWqH4Xouj7xuYVSPJ

jfLKuWa6a/+pNXsqoZZ7w1Yu4PUSkfisPbUv3NL0/9pfm0g9Vslc0I3ey+So6xCF61VLhhhW2pr  
sCRjSij9YhIo85ghS2Vf9FRkMjMVDot9y9N2aKCzN0crsf6HIY+BV1ewtlc/qsYTaFe8Wqy6QzW  
i0YUTZWQrEdll0yLFqQnwXaNysvQwBU8EaeG6lglwh2QxUL5fWKOfH8SCzL1r53pmzcVqCRa1oOv  
l2i3Wlr3+ula0eojo9RrirkWzF6/S46HqVsxE3o+KhuzeH35fFnrzSbqgHrPQowtQdVvQgye6lqq  
CM0WAhhrrws6fBh0yCu3zk5nDkVFuaTFFN0Le9ptj7TJHWZGC1SVXpwjL5vbTR562ypEViL+RCVJ  
HXMfHRCtcxfr/JNs7eYrSxS4H04R8crXTiLU1UxpAgbm+ZtMT0VospqUC1glvZyTo+tc1oFtDsp5  
fWWYr46qlZYa9634DR2birKyfC1UZzcrxC2WqjkeeJtjkIPg3YXWVpjtljCF30cJnayXMGX1xW25  
E4C7tZohOVqAe/MEM5eA9YGrPcKrHuvqK2DD+UoVmpSejqkyTaCZESGbWmld2rZ1PGpafCmlKhOc  
qtVzVAVIHScRq36vwRyxfSIqt/LtfJSFSeoGUxd4loAzhk2qiCjahSpajdVrCryxcSkJm+5nla+1  
ksCr+GLWUtyXM1NdZLcl8qMLHioa107az+valip/jFHZtjMHWtt87qs8G7Em/50YJsxHcLYvJLnr  
RAGnuoBt+VnoZgJ3XZbyzCz6QTFItVIOVVtcaZukli7VOC1d3u5di5WDe7qqHrvYkv3X7SEmbTC  
gCXWSUV2A5UhZ9kSb0jzmVFlawlgwtKtcQmMc9Rv5MOrlq+9HfTI/9p66/Wc3DE57XySmVOHQ157  
shGKikSzL4nQDKT4jjFjRPlwKYCmtvskmEUuvZUTI1yMGqkKyt7erCYca6Vmb70+10VKHE6hN9V  
duMxOADXctFvm421yZ4ywjGqaMp74cjMDgcmeoWfstBR4AoZiQOuAmOWABJvJLXWiBzU/WrzeDoC  
p4wePu+B5uumwas9xKtJQtiFF5562qlCvVki1UldGqjhe+/TNFoK0Ppz2rL+K62u07FGLPWUj8yV  
ZDquAuLvmJm9JrkvkUcoyzkKd223CdWRqOCTpE+rXQQsxuyf7ZnxLfZzE69RYKUDMmCfM/Rulwdt  
qxtbnK/wgxPpiYUu4KIPmt9Pb40PRA4FitRqlqPGGwXCr6sB0LTD/ZaHvp5ICbuQPXRH115eaeFi  
4yHkJEoaFs+P9XHVM6lqTM7MI7/+DvNRuqAeu9M4OzyNLJa6WtKhAsflB2Za2sHWiVkuVVT2HM0C  
KQTzBmQUcFSGF/v7rxU9XHFHxVLSU5XQmqseqIZ6pgJF/RxyaZ301h8Yd6kUj1PSrpQaJvpF6ozh  
K1a2MxMbyNjm5mrtoqw8f6aC7bN0CyrSLV5nKMzFHReV4CfmRpgdg4xWMX0DQXEYQS/qu5Bi8EOW  
ipVylY9erE6sKrTEZ+0Da/bhy/XDpMC6WSuqaLuqIs8i4wU6nooVhFFCHutI+/s5e9odlQD1nsXD  
S5K26LagjVdTzh3dU6MynlyYBTNTHafUFq0Xk8fV9PXMekQYD2MuFqPrjBqEWRYDq3SqiHVATbsy  
YI11TyiKqGYkDRzHQGNaxlgJ3miQ5fTYiFg6KK3MRQT09EiBU3dWMY0pknOtAwfo9FfRodtoxVc8  
WeZEffWQBhb4taT47QvlyYzudU183oqmuL6fkqNcOt1VUFCDi45FOYntxHNJnJQEPiuNie4muItT  
Z6Vush9zqL9GW9v6oIbsN5DKziZV9p+zWqgOjWsynrdu0OcdECra+allqY7hWWHhCALCZy1VjDql  
WkW52Ap2pcCJlvV9WHm2IShAdmWH1vawG2tVR5YQlirGLcZgcE2NkaiWcEUzxba6035Y7jtttYUm  
cCQPH+rszZ1LZ1NQ1+6QKJAtel/B20dMukLnwcFI9f8XqsZvbYkmDJaP29pmQILBqvYu2q3Qo1CH  
x0nWE1Zb7wy8284KY9DgrIU/U5f/27V7BS91539V81uasB6b1eRV3UWA1sXUTGXyT+zZabDqc/HW  
l7Ls15hXjE/aQpTEpTtYXVxZKwRYd34jKkSpIGjZX+aRvEEOGebViI+sTCmK2ZkkVLiRUBH97KwV  
L4FHLUYBhr5pYJwJqCJaX7q6A5hyvYAYWbm2+g4SUZi81wa2hYLINym8NrONbdqq2w4hb4ZJfbU3  
2agS4cmq7iEq2M7N1qsw7EYdXa0HNIiy2tQYn5wSklYfte6KLXCbGkwHe0QLcXaBEHLWE5fm4OX  
vWZ/6jZTA1Y7+0f9TWfX9W/9kMro4JPVs30T20rg63FslXdyR1IKVP1R2sCbzET/3JlJq11emKud  
pWB+cU5rVSOKMUndDojVbzH2sS2MwF4tBR2bAmjZardyy4YT56TqZrKNXnPqhRGo+L9nKxW6AhHD  
LbhsTVNCy3kkWRZF1PQDHbNvDudoUwTHpmpdaZZTPJn+SG1fzl2gwEym+LdvlEUZpCpMvNptceVZ  
YwC+u2xg5EcNq1Wg054A7ndxnL/CIG4AOzMKQ/voDfqPd7/6/+giQQ3YP0E+K2tUUb20/LZ2mgTw  
b77567VfLU0QCSgFBMWq4myZDE/Rh4L83SF8eyBMYc9YaewtX6kY/xEnfgGY4qKD1gIeAiaomyhp  
txLGRsf1DYawhaDT57Rd6qXefIoEmALQ0e8m4Aw8IxfyfQOWbMiK6cmkERWlwoFvuvdemAzSb3Wg  
6Is1kOW+ZwVB2eF5iDCOjilnr5DQ79YyiOzHJ3NLoLWAFZ3KM8bwJuLRbEQ6+DgUADrXB0/9qp3N  
LuoAesnBqxb53/PquJHGQ0uBUisFqo1FwKNqDIXqkOLWXifiskrpm8+OUI6vYmq3UHQ34EdCqvmp  
vCBYy0YPWYqxGYDQDTRa7CKqIqXGpwqWp5GX4tsgXytWQtTfIfp8pHHe21XzMwT/T81R0vflqefq  
R/LgcYaCCPA89hMXrWcdVpmPWjKWk+N4/zWNRgZvGLJoSZhqclEg1sVEo3/69QUdMtRimuQLacop  
2JBjKH42Zb6u27XV3+7PD2A59ZpZzz+kc1v/Mm82UUNWD8hy7K/76upl8qt8ZaqM8aXu2QWZcIE  
nTMnbIbR/y1YnqI5OAZFJ0Wwt2L8Hp9IeKwQdlzXCcIT9ZKWT6g76qJLJgnxVE1HMQbA+FFSKKp  
ohXKx2GtTo90aomipr5/aEKm62OD1S5wm+ZaK3DSgoBWrI61WBTZ/uippMcYdNSyxmZvsn0UKFEa  
qF+aLWOpJU6V4emdGkmaZlgFS2JIjVlizraMlUrIpmttA9W5zXzfSfy90g46uMUBRZNHXhQrXT5+  
JUf/uUfbXZQA9ZP2LJe/dYF3v+rvytO4Ds0/WLyMWaoU22qeyoOfhEftTg5RLR/FVVo33+1Qh7m  
xqV5SRyaz2e0aXf63ng/JxMy/FCBWuZJsgpBi5gyZIFksUCq+kE0WSpA2Y4prEWXzeZ3oLJzghAx  
Q9lr6mqE9q2KYZg4CmawZ3dQntwDqZ9vXy+7a3SAgtaA6let6jNYcF501owwcCZagfTdf7PmnXMa  
2eNiAbWqIgoB4NjWlqGaKXy+OUMCDvCrh46l+9/Ynuj/+Zm9zRg/cSbwjv3/VX79uI52dDnTOGvm  
Xr0iFpNulzWB6cRsukNRLMDtF/xBgRhX4Gqpqjg/BzRMTJoesUdCANxYltp6gdyAYBWEZHRhH0du  
R2KX2m3QxQc20Ef1a4FnK4mYxnpJfOxgMKkGeGoNErGJHC0f0eIYQSCvITCC4qW22wlHOpXpEKJvy  
shSYUoZy8JMjjOlS8XzZrGWMmpzOuJrM+1UMg0F06Prsl5Ynlcn4xX6PwzPXbi5+zlvf7N9+dVNU  
KkB6yfbFD7zcFnMb06wuH5084xM+HOza6BmPQpxeozVwT7cnctoiQnqaBuckYTXyK+2vlRi4lqqh  
6QupBtq3tOpfDNA2fc0SOUJQHLHvr3fenGCilrJ7VLzq/QN2WZXeZZeA6q9b2rzK4psK7gyDURNT  
m5iNL6oEV+Hmr/CljSJWcGkQ5qz1Y7rUF/aNk3RbM8rtRLLTbtgugl+W8DZkdckZnch/rc1gLU8R  
sgRkXIgVGWY6G1hlmvNnYdf+7lnL7/6TrNrGrB+0lbl+v+nbOZHuaHZskYz0tJ+0wB2TF/1SPzMC  
DuXHxM266jL56zVily6gGLC0pzUfCvNSgZ7KIDG4U9ppe3rlhbUV5pjpQpKLuY0MmHX3NZifHbRw

DHF/axwuqtEyABXqWAPxLSW68ap5lzzZKKyMb3N86b0kAGlfKX3NTNZmb6xVGWClVe19ubWKjF6t5+VaSO/1RVzvQ+3PUApvvPRcx8U9zfGwTLFz/3hb8l7qmCJSb473nrq57722xt94Aasn9zl+L0fEsx9tWXVfZbnGYkWX/3RIksQH95E96FPh7dxdt2YnpKwtORPSw4JLgKGYmK2EioyB2cWIAyOUQ200KhW5etdYF/CWclh8cIsyfmbpIqfZasLGJBvxuqD6t5WNYO81yglj67a+xAQFVq6mZ+cgft3obWGL0Jnw3wHLrMActkeaZgjX6LY3Rdm0JhAI30zQg2fWq2BbqtdSdOgVx84lJM8eu3D3Ayj8XclgOh38H4zAW/2SkNWD/5gSY/PCjK8th26j4tWkdnqpr2TEKQE8/HV14p9wsUDKzyoe+nTels3mkFGkV22PsqjOgmppoInLfQD4GehTLNtAywFNMzIFOKH+gGG6gCAWWQKAuXdqXBKAwpE2gtL3Fm5RRzETBxqh0nzngr+46pPFRob60/c86kgKxyPaC5U109XI9S5mvGegKeFkXRZEeuYyMtLY2s9F+2SHSzMWA1EQb3fVeHKfdaHQw3Bg1YG7C+CPzWnUdq+/ofrbTvUxjGWqc32OFC0LpdMRUD4c3kSCuAIL4czVZ1+xgtF1BW8oV4ohVI1nKh6Rg7rrUogv4v6ZHlep4qA7Z0lnP0nESLR8sCaY2Cwd9VYpgg7unmiuO9sA62kkTGPCqOLIKx20jXUZg7Ih/7NTrmma28BWmaorMz9Y/Y9xrt07JCiZ2AF1GjYLN9lofzVLH+bGavJwOV8hBxGZ0Ff8OfHmJjeh+A9YXC2Dr6tfqsnyU5Xq1YxQObbutPa2tUV9F0OKDJ7CazDC49BrNXRYHApTUN8CmacruFVf83YxRWduoEGqRP9XwbWR3YuSOsCdZpJYAIq8FFI6KZXNwchlVOSWts11XLD8P2MvKlRqWFlpUppNWDgALXneEQF5nfHCEUiVSHA20KjVfXGtJMCuZSqZ9GG1mdFsVHqo1kG019bX9T8m21vSSbXraMRcLoS0Wg+/5+rNWF7uJADdgfZGYwrX9A8I0XyFkM6Zagsq6UBisNYTf62mNLpvCs6M7OFh+EPWqgCfbtZs6Q09VUZJlsoJRIHNVo0kL+WmSCTnL8Ryl/DgVE9trdlGFXRSJSZWUXTbvLDW9wrJAFjfk8wiFb0YuVrZcKxADuiVmtl8jaIfae8rGAuZjdeodI9REGSPa60HHxd0mBOo2sWuIgafKdA3ZLlrw26p2oe2B4osXeayHEl8CUzauV2nwTCx97F081zSYN2B9cay6tXlqpceLEuXYUVUFSyVOOFZDxdAKYc7BDvoXPRx95Cr++Jf+EMvaw3G6wETA+tfe9nnYee1jCFem6N8dnkMtGLNYvne8QLmaa4G81x5R8AlFukSaLNj8o89R5gnSNFWTlB07TOVE8QpJnuF0PkGWZRhQ8rTrYoFKGQSttpjZcxVbU7mZWhMz6+aD9cBk9qZaplywVtaljLoF3VvxVa2wQykJY46zk4idOL0OqpUcDnIwDPltrYRKk9nRycnJtza7pAhriyMi/NhfrvP3/uMdB85lqAIKJVky2bwjOMKCZXQHTncXvjDS1kM+Xv+XHHZ4PR9HlBb40+eewrVOgMuvfwOCOEU1OYSzfQ7e1WuoOqGA6VXIrrj6B6onnUAsj5r74f4GN9CDCcj5FIOyc1Svkk5UWKgT3X0Idh5i//204eTrDnzx7HdfjJT7rvvvxsp0hXv7pW8/7tZ3tcyZtIz4yga8i5X5o0kf0UdeiT9RiqjV95Imf7KuiPvOrWsko7znPI6MDRYZu9TEYDnEkblL2oI+ds4/97je96yeXzS5pwPriYdcSf6+2qt+W3d83KRozC8YZXTSfDbLJ3VAYdmuIvTe+Cq2NHl6/CvGWZ2/jaL5CuRcZdnMbmZjM7a0zSFZH8v+LcNt92HtbCDBGYvKK77h3RqVc/I99CP2rz8F5+BXCPjHif/8HcC6dh//gK7B84sM4fe/jKPPd3IiWgr+O5mHvf+wSws2esj5zr35/Q86UCHlipqdrjznTNqXkWk8xrwjidYGVBan2YV9FSG3tBDErr6Ya68sI94QtmYU2EaA7e3u7z76mke+tNkdDVhfBkbnwyI9ngsK+ty0zLcy2GP1NwVMiQfB2gDeEp9QNVbGIw8h/+BTGLzyZbgsjGr95m+j2N5AuS2m7s6DSMXkbQ/GRs61NUJ+7gwsAbGrq8cT5NvbOHjmw7hw+VHYJ7dwbMfY3T2HZHKW499GBMxQVLeKT7z7B5eeWkPF15+FuMrOypoxqHIZEItN5wvdVwk28kp4k26rEoj5EafWaPETAixSELHGNCvDk1JYmXeEwNqXrcFdAZ6jd3zezxQfuUbfuRffXGzMXqvwuhW8Jp3lOm7f/wZx6rOsXtFixy40dvCQIVrCudlklc6pyagZtr5lzi9S7h+B1YqQHjmnqWPPonszlsjvtvmn0awMtbChEGC13JYOaa4L5kFbmJfyLMcnSD78AbRvCEPu/yay1QL9lQoPCAu6/QH6D17C5uUx/G6oqRSaAASp5bWlGydP5siU+R1VntBRG3zl1fM5aM0MoMnkMix8o0P2pMi40zI+LvWjVuJ5t3roDncwizNceuDKv/iir/mOL292RQPWFy+7wvmbYkJ+sPasfq06RaUKfZPJVEObA6VKS8vwWKzvXbwgDHSdEvioerZqLnni53oHU7QXS5QfehZZR1jw3HlVIyWHT8ISsxbLGZznbuLSM9eQfvAnxReuMBA/MxdTmjnRmqR4Ax+t7b6Y1G0V5aa00/Pra5ZrCJvmsayFFCz2t7XHNdcKqVonnxemfnndZKC6SewUEoa1OVyZOVlh1+Tkthw+Lux2G3VviMXhR99nhd2/2eyGBqvw6hV+5tdcy/7gx6/Jjn7U8n0TiaUig9dRitIIsWX8WUtNZGG382dh3zw2RfwbsuHJshLfkKhC2nc+ExFy4UaXpnfrgOip24UQz1IncZzBAQCehjmdkVA7EtFDRE+Z2+iH8LfEdKT9Kk5eMycooYWQ2j6MSJmW0WfzVSpXe7LWWkouc0jLstsnXBRI0hClFyoZ3NpsLu9I8pmJhIq/Jbnlm8p3vX905f/9nfNaX/dlGYb8B64t/VZb1E3aZ/UiVVoG15qKjZmLCfXbZ2OvADYpEfteFvTnSan4cnJpRGa5R12frmQLSE3OYmknDkQZ+qISIO5mRM/VrLfYvOJeGLB46ci648He7yFiCyEBSZRhV5WPW+sYshGQ5YiGPzZny0WFXRqFQ/VQxe7Xpnf9niaLfUsa2nbYZ1dHqy20P2eIEabaE43R4IKR+b/A3PusL/lZTAPH/czUjHz/Rq7P9j4vSeq6oUtMPyolstTGBHRX/No3aOkFcTFvtGN8SIN5/Xr53tAXNctedMx6L92t1LZEZrZtWbhdGmSynMr3ubIGT5/HEjO6fHyI8t2nEtIVq2f4ml4DHIBCF3Sj3T1EzBSRZVb7qdfFDbQ6RUKxehfVysnBkRsE6Y6s2SorUceRISvlFthJfd3GkQmzsuAmGO+99+We9/T3NBmjA+tIxhV/1DnEB3XeVeZWWRaJCZDqMSQlubQYZR6ItcSzmN2k2dMUK3hrCum8L9aaYmWPxVUcCypGpui3gKlbad8qJco7gpQ7N4/0zLXRefh7BhTOWaVb76wlv8sUZsB7HeAhrmhBXZUTGswz5aqVdQTR/c2XVSospqE+c0z/mqB7mWlXQey0K7tHU7QmYYy2HLBanS0dzeVxy1N+99Lbm02/M4Jfc6r/p635u/ns/9vVuUb2BDdlSfaPmEaVTCBysmUz1jzT14WuxPmuJ0Wan8AD1sCVg6oldGpneUmevqxyo4j7SBB5/nvdheZZqIdGXVTYW05aFRn6rZcCq0iuxaXmT3+uwKWF9FjLkearmLlUtaDKbemPzRYUllhIWJoytTfLt/1hTNVXOxoKZnEmr5IslNh5+9fvu/6y3R80n34D1JbnEtHxzXlUf86rqsIWggJfq5mfAhsEmBphM8UGuMqI6/KkyI8bzj1qpFrCnpX0c11iHXTVjjeIglfoHKqRNMDNCzOJ9h6xLm5uMK0C12mLNkY0+f dRCxchtqleUFFBbIk1zZPJ/BoqMGGKmLXSFTouU5wq4fSxj/rYGWr2U0WdmS9zJLUSHRwgc5yi8cOGvNp94Ywa/ZFfvjV+flZb1rUWF1D6dKjCU5uy06vLPQM3bZaq+qPqz/NCEOTmf1cpqI0jGbh5tb



XPVJOW11HzOBajxStvaHLnNPnKmQb2QNb71GvCcPWVSLdSGYoQ3ExYmMHMGkuQFZmIap+L35jo9X  
QDM1M1aL7jUCeJc/BytkbO0UNg0WSCenKKUL28wuPPqL/nGwfoJN2B9af/xg94v1rCfY0FQzTEXb  
BC3tR7IBI0IVqoFMtBUZFr1pKWKjLgKM2owqTASoJSCMXCVL52vmum4C8vKdXyO3eDKl/vY8kVzm  
vNrQh5fZcbk5s8AFQHP41Snzilg08xEkzmQw4CttkVlRnGXHEciN1WEXCKRRCKLMqID25iceMGw  
sEo7Vx44H9rPukGrC99dv1vvqKubPfbZcenOn6CudY0NeqAZl6FKeUrVgLmFaokEjDIbQqH63yZX  
L98t9KgsV2TTeV+2UL90Dqbm5GTiMT1zeR+1rrpXUNJWtDAFE29VsYvGOUlK9J3zQjaTGt/S+3Uy  
ZRhMx3g7JiGdsekM0oqXghI83iJeDbF0VNPo0uRtO7GvHf54Z9uPunGZ/2UWK3+uV9cLfe/16+ty  
xTTdjJTpjbN3VBB8Hit+XkJSH3SsWCb4yxijdpYfIYWNLiGnet4TooTMzLSMJrp6YX2meoBICydm  
4da+2bWK1nSMbNgqfpfaMom1S8GmNRVhplAlwtY89pDRp0nKhPmleozuRT0Ztuc/P7o6SfRk9t+u  
43wwqWFv7132nzKDVg/JZb3qi+p8Qc/k3MMhg51okqh7WtwSEdFgtHaREv9tAc8Z5wpWo9YFBCSi  
dnMHqWGdcnAZOgyU1Y2DJqZQJCOLiRTWpQW4ZQ6V/xNNodrKaGYy3lqel+ZoiF7UhyNvm2eE+C2i  
onrNHXkZjCWPf3OTFPtYnrjDoK4Qm97DGe0gdaFK/B7/eZDbszgT53lu15WWJ6mTTTQxIIIXwSLW  
CPMyema0kljWOIX2qmYt0VkBi2zjG9dkKAaTUFLtZyoMKEFEamAsLCNKUvFBmFU9qYwZbJuKjCAz  
uTosQLVpG9iYfNUvhdq+jK4ZKkZrL2rRam+bJpE+j1Pcty+eg10EmNja1NzucG5y3C6o71bt259W  
fMJN8z6KbOYT7WdjoAhUgCFNfOWuQ6Foi6v5XNdNvEmTBauixqqUkHsVutKJ5jIcF3QzzWTxzVCb  
JnAEf3STBg2zUotxs8t1kvJASBAPDxoIT9LolwHGscCOoJUhl0RmMq6tgKVvUtlacS8KQwu0MZs/  
wRd8Xc3drZVUcLZ3JSvXazyNLDC9nfK0//r5lNuwPqpsWo78+zSFB1kBQRPYpeZdrFw6LI2qbLTx  
cq15pYlSgSsQyG00pi6zMVC856ULPWlU4MBRHAnciJAJkAOSuKtYgoDftaZlZNLprobxIJu7IIG  
uDntXPzHPRXCyMyrNkmHCXJ+68Op+jJix2NBur/Vv0+yvEZlPI990JcOHP+0o2bT/+v8+ce+IXmg  
27A+pJfTK2UrgdfTN9IWCyRr7YOSS5VeLSwlnXDIRy/NOWAlrWenM7SwJgFusrClcWg0VIHWHBMu  
XO3hrfWEY1xlqsUSw4zyoMMzLwqgUjRxFWSGV+V8qZ2rdFhgpURYz5OGZ2HBa+dpJjeOck408Wo1  
9PUDwv6i/4AwfZ5JG4be2fO06wP4mj5LnmbDVgbn/Wlv8QMzRgk4qCogLlWAVYkDEt5Fg0SeS2dh  
M5a3Mpb15CVJiBURyDdkfXKcqWKDbkOmKw1rZLT75R/SbVOSWA90KqokCSJHg6RgHhBabXEFEMw2  
suCiEx+lwlbZqzL0Ppfy4B3FWOxf4AzwwFGG0Ntf4stIBqNMHj1m+D0d7C9dxGe620+XMC13Eu39  
p/+ouaTbpj1pe+zZulMFfmFHSmMzfa4LGBvaVv80lzn0YIqg6X8rLKNlArb4eibJpGWJNJwLcU0r  
dzcfJMVwGuKhtVHwoTMnSYcUMVywsooHFZkXbkffl1foOI9M00ZkXlRGSdgyYwQAHh5yvWg6hZ/kO  
L85RtjrCOPWmHMwlv2PYucVr0ULJvrOuQvaIHAynWA42sRyuWQ74PflC/6V5tNuwPqSXsKl7xM4f  
LZlewF9z1D+Q9VfyoXaOvqNBQtT7XVNOAiK6oLEk84+TU1UN/2/2XsTKNvO6jxwn3PuUPnc9WZNS  
MgSqJkkJIEGJDBgY7Ad2+nEjtOdWUk67nYn3e6O3Z1e9ortFa/EdjdZnSwbD01jQIGYCWFLQkggR  
gmBxBAQAiSB9MaaXs13PkP/37f//Z9zn9zdiWPj9+S6WqWqd+vWvWf497/3/va3v911Bud8qjOqg  
cs54Q37ntvbhVZwv+e8qU6U6w8UNGJtFd0WZ/PZ6KDPsiABgG3qM7hU+hrzTs96WzvyszYmCwdP  
cCump57j92RUR15ze2y5Iy16z7jwLGLpFEbka2NdZmeX5C2M+yRbiYERJecPpnMO44evXzfYPeN9  
QL2rLXoq6QOJqnU4FmdgTSjPwmnkWxtrsg4BLMR7sLgOGpRSzMwMBp7X9XyAQShPxaG20XTuft3z  
4WtPRAaXNg7gMvNYvdcQY+IEhHApYxT7fwmHkydqWpdgrD4Hj3sYK8tdfc+Lzl6SBojTfaz7rnPW  
B9fkK03vElmD13CSXiHL7nIHx9dVtdXZGFhgZRF5FljYxOQRW2OTjT3veu+sV7gAFMyfgpKDGAb  
ctGEqk3R6SrtqXtDG53b8sZbxLqsCAtpH7IFoiBGRUchIvDG/dP7pAceERC62PKpqbmJQFGQIGM  
5t4ojqjXM7ZSTOkEkdQh8C00BeKp84zLkxNyOzCPP8GzKYzeU3Sy2+Uq268ReojU9Lp7MrRyy5nW  
H52bU1mIjNkdH2RWZfH7u5uClQxtjbb1/zLX/3f/sdf/Oe/8pv7d33fWC/Mx8jkRtFe6z1/2QS/L  
46dAUWTUnciecztyV6rJVsg+oOBRJZRSjBoYKUBWHmU+NmpqlCIJ6nkgHzUWWVfB3FQnBvtDpyqH  
ns1w8L1vuASo2kaA4xo7LnTe2ZLp0XFZuPiYy52bnLu67sLqztz3ycyVr5JLLn+p89B9absN4fClV  
9Drb7h8dn5h0eXbPXKn4F13dnZJm6y7Y//iY48177rrw7/2R3901/YTX3ny9/Zv/L6xXnCPuDayh  
dIIZFkgjo2eVlALayMjtCeEyNvOWJETUofJhbxod4s5KI6hSzN4JFSmaVgGxtkW2KX3DZyTI2LC  
T4BIk4jwlFSOC+c9NCv2nGeu0bD39nYFsxfPHrxxTI6Me28ciotZ/DtuUslPnyVXhRrPTK7uOTC2  
h0ypo5dcowh807etizOLblj7HO48twMgKUWc+N6vSmf+8zn5F2/9/t4bbMeJ//nK1/xstNf+eqT9  
+3f/X1jvaAexchoBlqiFBHE/yDlENXJ20X/60j0pDPcSNrbLdnb2pReq8tBxiA+JL45PXZ/h7INx  
80AgIpEEeJcVR5qIEikXfc1kjqz6LpvwYME6SBH+LpBPaeLjh2Wqel5ajrt9TLZmlYs8ZfdyHGR8  
wePyMhYUzbPbsvEzIXMT09Ly3lOzMxZmD8gbWecSaMuUzPTsr21Ta+eOIP+xCcekve85z2y5Twvl  
KayetF0h/qR66595bVffPwrX9lfaF+r2IYJYe0//vIeg6c/O5ae+fpzSVRbjdAlDmqBtRpb5nKIo  
EGWFHlpyvM9hL+7W9Le3ZHO3q70Kb8CYKmnYBGKq84SGB4DHUZgDFUH56ohjaoRk7FXfRi0erJ8e  
pn/PnrskD0yOYknpmsQNuRxs4Xgr7hVJO5dIWtnN2Tp4EEyqs6unZXDR464Qxyr1dVl1mimxidkc  
2dbRkdHZWxiVNbX16glVXOG+sJnPiO/+7u/K9vbuwzBGS1gqFVSx+vXFpbmrv7UJx9Z318F+8Z6Q  
TzS579UH3z3se8kEh3lTFNoCkNUM5hPVxJ2xghb0NDbSlTYeckUjeH4GV0y3ZaCS/Cg6HcF4Z6Dm  
1N6XOgvISRft830+oasn1qGfoRcdPFhmXFesJE+yybyVecR+wdfIQeuvVF2yW4qZMbln13nNdvd  
hw4cpi1142NLZkEc6lWl01nnBg21XTHvbJyhHEBvO3jj31e7rzzTl1b3eZYS21ScNGA24hqids86  
omMjIw8euDQgTc890Cn9iVK9431AvCsJ7+eFM989r1I8qPssuHsDJ1KDM+ZNEYPqMa+VEqLghLY5  
3gLtnhDY3inw17VnGiwB55AwnfGmbW7zhNvsjF888yKM/ZUDlx6iUzOOQMDIypusnyzVxXi4xW3y  
dzSpXJ6ZdkZVV0mXUi7ub1Nr7iweIA9rlubmzI7t0Dj21g/K4sHllgGOnnihPOWTZe/tuWxzz8id



999t5w9u6klJeeV6y42j2MlzYH1hC945vHx0Y9c/tLLf+yDd969PwJ931jP70e+9mzU/8aDz8SSX  
8Z5NxxgORT3gSBvIm6MSj0wwdCQiDCX83IW90B7u9dhEz1ILSjWDVD0r1B76bem78LSLSH19Vbrdv  
ow545ucX5Jms85wVIq69KQpq5PfJ7NXvVZGJ6fkzKnTMjElztB5e2NTxqammZ/u7e1Jp9OS6dk56  
XW6BJa0ntqXkyePy5gLg9ec1/7CY5+Tu+66S3Z3W/oZ4pzb4VXpUZuS+HAYxopjmZgYf/9bf/AHf  
vqXf+nX8v0VsW+s528YfPqbUfqth56Ja43LEFZChBtEfRgdbg9GZ9Qm5t1zTZBUCBFLlOmupRvA  
Wsp7+45I24R1AEpApPSe6096bu8lvNqID86MqaDkqnNFJE73Jq9QvpLr5DxQxcxTN5yOenU/Cx/f  
3Z1jchvs9l0Ye8GDQt56e72Do97enaGBry+tiLlxoisLJ+Wxx57jB6111dWFHjMRLYh2kjkPKvTId  
ci+NGuav8bOYEcSF1ZPvOvRzz/xD/dXxD4afP7ehMPfVwy++SB4RaK9FRDxzRVUUrDERYGIIItqYW  
+N+D20lYd8rJpP3qKTP3RciZvBmNp280ZT6OLxyzE47GAae740f1M3Jl8rY0StlZmJK1jY3ZG93V  
+YOHGBYvbO1K4eOXcT3PHPMDD1rw+WkZ8+e1XFn8KPjYzTg1t6OjI6NyYkTp5xHfUQ+9MEPSRvMq  
rzghlPDdACpa7N7riEwIgSgWqQ8uvOFNhQkYvb2Wv/glpTv3P70Zx79n/dXxb5nPS8fj/9f/2Lkq  
iMTx2ujo4uYjwpeLhhNQIMzEulHODc1Hp1gwz1Ek6Jc5Usxi7Xo7kjW2aQkS0EyfqqaLqqBlFrw  
LgzNvbuct/0otfJdv2ATM8vStIckfX1dY7yGBkd11arxfddnF8g6QLeEmR8vM+my1Xn0VjN0z15  
WUaPb50HH90Pu9y1Lvu+jBpjcaisgnpeA2MlOOWCx3PEXlvi6FYsX8fbAbIeWfmZn7xU5985Nf3V  
8a+Zz3vHjtPf/XObPGVi7VGXafF2S+gd0+8HEZjAKKJ3WKHtHANrXIIZslUGmi4jLGMzk6nEzOC  
arC9/PDrCJn7IO5K6S78HIZOONfOnCYRrO6vEivCbBn8+yGM54apVm6Lifd2dmSOWfQvV5HWrttW  
VpaoiGe8EASqI4nT56WRx/9rHzorntcTqzgFoy0gfNIICEFWiPKSzg0hMFJULPyFKRN3fOo71J1g  
iMlacz/8vbbbhr5xMof/eX91bFvrOfN4+17/2D+2e+evPEllZ0gS+OTUi9UhZBeC2R7yJH2B+xEz  
aIuJ4sX0A6GFKifkUoPzJGRluwFCQJ27kLNOmLIjZpkjXEpLn+T7GTOQzfHZG52lnRFkBRm5+eI1  
gLVBY83adZ1c22TjQFTUzPOYHcIIAHxhdddPX1GRicnXMjadv/uyJcef0I+8pF7GaqrJx1XpkCE  
qh5AmUosrEa7rhdHEADh8HD46JzhyGxKE+5n/e5aeDcV1bWf+n617764tdc95q/++/7e/uh3/7x  
noegEubZx5o99LF9TMrsnDwYuf1G1KHgdabKteC0Tf1Gr2oMn8HJOeLywOh04RyDbWGgQRT4iVi7  
ROGQNRh3EsluvS1srPblcmZKWmOjsiuy01hrAhp8X3XedCDhw8RAFo+uSqNkURGgXOytbFN8j4QX  
xj26uqqTiiYNM28FovuFx56QO+58r3Q6HW4UFnKPJE0CWgzTUBaBAiqMN9IG+ziJw++hBwXkG1PXY  
ah4D2wO/Lts8F8/+r1HZ2+6+fof/+xnHvsrX9bZV4r4S3ycfuz+l288/8wVhw4dkme+/YxkUC4s0  
jBekfPLQFDxcw6VhrYyzC5JDvRGqMPmAYVGZMoRJRBUr0s2dZFkV71D6lfdLp1+JNOLi0SCAQrhM  
Tc3J51Wl+WcxaXDbGdbObMsUzOjLnccoUGCWgjA4x0Z2sbBAZZWVslIvypT39a7vjAe8iYCuqKb  
BzISfC3kBYIMr7MY+IY0eyO3yH0rfnaK9r58DeZL9xod9HAefa9Hz55/ORXrr325XP7xrr/+Et7t  
NdOvj/KulPI3x7/5mlpb60SiSlyP3qx0DCYnThwsQCdBl0aLaiHRactmcsni4GfmG7ixMyiFJfcL  
PWr3yLx7FEKoc0vLvBvz5w6KePj4zQ2kBWQ087NzbhwdtcZ8SoNuCgSWTu7Sq+Lks1zzz2nYJE7j  
LW1NZmYGDJ773/AgEr7WGdFux68YJIUzHeRqzJfZWkOpQFiI8B5wTgVRBr1a9j47sdJMLbG5iSQN  
NG/QwShrKz0Zf1edvrG177qv9k31v3H9/yx/syXf2J39dRLm+Nj8h0Xen57dVe2V1ck39smaoHgV  
PKM3pXDjP3YC/S7srba2pRBe8d52g7RX04nX7pc5Mp3S03Q1dL3t3Zyeoph6sb6JoEjhKswVBjm2  
NgYObv4/cyM87Sdnvv3thxcWqSRfPfZ79C48Xt42omJCfnYA5+Ue++9V3NRT2yAt0zzwutAZdQhV  
sMcoVe13BSbd9lWCHOLnN4WxkuvS6+c8gvn3Kw1CU6hjtS YBQDWaHZ6vX933bWv/OTNN91w1X7Ou  
v/4nj3OHv/Or4zU4ua2Cxu/8dwZObE5kGXn+WYPHZWxxqS4mJMLHDAexhF7TgH+MM+DQj/Ap7xPI  
bVofF6SS26QZOQFyerdOq+1C2FEaAXJN5KUTU5M0OoSWi4vzNAiUYvCaJBnjzyMjY/wdQmXmp+5vz  
m5uUMh7fHxS7r3vPnngox9176ceVQn5LpTF/FhsKB4JpmoiwtnMgCSVolEwSeNcM9zgNeIaX6dk/  
5hyp5BFReiNTQUGDq1k92m39uP+1296/fUPj42N/sTHHvzk3r6x7j/+wh7bqyff4QzqkjHnmrdT  
L67sStb/VxOnNqQiy5dl9rkgox0GpJORFLHOMccmktQF3RhcNyQCCWZqM/wOFp6iWSHrpW4PkEFC  
ISmE5MTDKNPnTrFhY6wc3tzi4Zs5Zft7bMyPT3JEPX0aRf2Liww7EUI3Nrdk6kppR1CxiVyXx/8w  
B/Jo4982r2+O6TfxOHLkXrSei3hd/HRABQYazJIp1FRSvshH0ckeIt14SQ0dGjAEYBqJG6DUBNvp  
fmZoJARu00BkjTdQbs5Ukve6jaMU6+78bp30+//Tx586JMvesR4Pwz+S3gM+vm/GpubbULyc7tfy  
IbLPTGU/LmVLdlcPpyD3W0KnEHqgTNrfG8qWPwQTSO8NOa87yWvl8HR10t9bIJ9r1jUCFXhSU+dO  
aldMULM0gMMCXZ6jsaML5RpAA7Boy4tLRCNXT5zh1rBMHCQHmCoe0f/P675Vof+STD5MR3y5DTW  
2+o10QLn6jhUpANVMNBLxAjKLMa90xjEqzoeT9GmrGudZfIegGoj9CYkYEaGZwNr+4MKeemo2+d  
dWh8mUqt3n+ZSLFH5uZ2fnqdtvu/nWfc+6/hzfexsb77ZGcAl03MHZO3ZWHZ6BZFeMaxPbLRLf  
WVNzheWnTcbYddL0jOu0lQqUvS31pB8/qUiM0fJbBpxxrLnjHvEeU+EmTA+1ENBD+y6kLflcr7Z2  
XmlCrqQFmQGeM2d3V0Kes/Pz9IgAR7BuGfoq6v4eVyWT6/IH/7hHfLNbz/FPJozcUB8gIzXtvyX5  
9RMfAgrGhYjdzUjo7YTyfwdHW6V5hVCf51UxIh8Z/dc5Of2uL8HpREZQIahW17cLXXvUSeqXHcRt  
pJCMFbWeeYrnZd94I233/x1d54/fP9HP35q31j3H//5Fzxp/Ea9XmuiINMrEqK8oOLvk0hWdnNZW  
d+RJReyx1M7EjtDrYNaCJAUPa4LV7uw9ypJI0Vd8UDNDHJymoaxsrJCI4Ax7u4AeOrLwYNL9KDL7  
ncgQ8Botz3nXHXCR4BODq7vsqelnZbO2kWFxfJUrr/gY/JU9/+lhTOy6McA8+IEgs4yWjXKyKNP  
EnYj9R4YajIOeE58Ry+s+wUqZfFcGd63rRg2QleFAwm1G1rdUWRMeQZ70PD9wqOaMBv4D0LfIrbF  
OqWL0dsnnef0uyn6WvcLvKNt7z5ti+7XPct93zk/t5+GLz/+DM9Wq3d6wdp76UjzkhcYiatnW0XS  
qpXGnEGu9PJZGVjj/klumXS1rZ209RnRF76NkmOXSV9gg3bzBvhQeENEX6CWQSACOWYra0d55Uj1  
l8Quq6trcgComncogalchJK89jNjQ165YmJKfasold1wYWeQIu//OWvyhNPPMHRkz0wpdxSAbDEK

escvt6k4cOb29hImmukHUOs+yLMhdi4aIkGBt/r6zxZhN3YcBAWjzRGeU51NhvoaMq689ZqqL4xg  
RPwPLML1gmRN3CMsZO51+ILYFlzdGyq2Ry51YX9z/zk3/zxX/37f+9vR/vGuv/4T364tfVv00Hu1  
tKYtNodGZmaYY42MT7KULDLnNap7V05u7Ejuy7P7OzuSb/XknTMec6RaXokGBm0jnDr4BnhCZGTj  
k9PMW/dPLtF4GhqaoIGCAQYlEJ4zc2dXdZcsTkA7QXgg1wW6g4jrLnOMRw+efKkPPTgw7Lb2mHoC  
08Jw+R8V+gIg/jQTYu0wbJ+2qyPa07pjA/vjQ0Ez+M7+3ElpwHTECNffwF8FX6cJIEkoMEQc4sUg  
KJnpZGqgBt0i5XhlEnHeX3k6FDN2Ntp8fhxbiunl4+6cP7nz57deOq/+lv/5YtidMd+GPw9ejhDm  
3X52EVxTdX/evCKS8dkY01D0NWlswwbztzuFrG62ZNYZ5Eg8K/Vjb5B89pjUcq1LIicFiITFi+Zue  
FiIaLdaAI72XNh7gF4QCxYGC8+z47wlqiYLBxZoJJs2867TjCEPnn8BI0ZIEjx48fpJT/x8U/K6  
eWTVJqgdCkogL2UzegAebSh3CO+CG2LIhgaa8LMZxN6d0OMlTwxCIBnflbnnUkxd08GD8/nnbcFK  
QLeN9HmVw6V5vt6ehPeg7XcHMOehZKO7byrITQ6izASBALj3V6z1epcudGoffAHf+BNX3fX7W988  
N9/+JL9Y91//H8+nJG82621RYAo8AJYcHlz3BlBj54QISEU81tuQU5dc4Mc/rGflXxsQVrOm+xs7  
Epe22PISIQXNVIXziJZq1E8e4eTyJeWdNKe6rrLQYH2YxVHC02AzALC05zwrhk3NzE6RbHD65CkSJ  
WC0CKkPHDggjz7YrFnCFz9PqYkYLowNizjQWctWNi/Fgu4a/F7pkIn2xIuCSon3tjBEM1Z4cfOGY  
E/r9YhGCNULMqZm51TNEaqNMDqUc2D1+I53doYeE4CKPWNKc2J6e0xwR24fa/1WCnXHaa7oZig1J  
U13/V/T6/Wf+Gs/+vYvj4yN/OD73/fv2xfaGtrvZ/0ePJwXiLrtzjNuYV5WbzZkc23Zudo9WX/6a  
7J58jvyhSe+Iu+/+08w+lh+8//+gBy65OXsmhmfGGUeeerESRLwL7vsMi7GnelNiqhRA2lji+Ewv  
Ci6YNDOBoNGSWRvtylZ81PBuKFICokVdNLAQBvO+M+sLNOIDxw6KCefOy2/8Zu/5cLwVRoMDBvSM  
FkWsem9YaMkBz0NaaMicH5NsdDAJZaoMKmutyedbp+bCZ5vNMZcy02Mtalln5OnT8jm1LSM1sYki  
1OPCoOq6DnHsXptAlZ+rVqPLL1tEjmPU2ZzSeIZXxQ/z7XZPvabiPf4+Oja24zu2NmavJ/eNfvv  
yfbz1n3H+Hx8Yfuv8Yt7AUS7QuImHXcgnNuYXxaDlxyhVx11VX0rocvvkyuee2tsuRC2aRWyDNPP  
0WE9+hFF8tFF13CnlLkqPBsyF/Xz26yxAK+L1BdLGWEyAyPnaEsLs2TWLC8fFrGJydkdEQJ+sBTg  
Rgvr6zRUJFTLp9ak/ff+QHZ2j7LXJHUP6ghZpEaZaI8XXWGPCO9X80YRzWVoUFdNY+Yg+J3bIJPC  
xIeyMBKmlpjbejx49ymJ6eY56aR1onx6GUKLOW5NrHXkiKQMPBZFI1LVRQuylRxAog3w3Lk1AMLY  
8SRHpcUCUCWke/2utni+vrGzz13/NS3f+on//rf3jfW/Qcf17321RdPjE/f67b5qXTQ4dyXbn8gK  
+tbko3NSDRzGOMpXnLpMZleXGAoCBkV5I9o/D5y5Jh6OZczguAAR7mlu+eMapekAXg2eMdRlHlqM  
T0tCRAzU9La6xDxNSVCGAdlR52HhUfrd9o0NgBRj3/pCXnqyW9wgXPcBvi7VHTI6TFhtPSabtGPj  
SiVed9RzoGHNqXCBkPZgp4bXzAQgE0oL2EjwbHhuMBBxr/BH+bcHQOWRD0gzhU/w6PmWRxKOchVA  
aQVXoiNGwg9robbMFT1K2tIjGYCC51htAid0fWDY9ja3r7s+edPvOutb7rtmy48vm4/Z/0r+nj5y  
668zq2y33fGdfHVV189hUX0za88KU888T157JGHnYdbI8jz9//bfyYvf+Nfly9++Wuysr0nuy6XB  
KHBwstedyAtK2NchNtbu2yHQ354YGLBife7W84w51UideKvFGilnozoPCmOjav5um4v10OHDnFh+4  
sRx9Z6Sq+DZ+obc8+g7XKib8z0R/sIgVEDJvVXsc1CEKt1sHC607L7tjQ9qamygavGwjYn6q3m4QS  
i8GVCGHxkYAnWEjRgB8CqCRKIkCfwtDpTcstNUu9SAWjp2G6gzY6s04P0Tf6JXXVLh4XHntv30t7B  
LugTwWes4XEeH2322610FKtW+dOstNz09MTH6w/fe9+CpfWN9kT+uv/66qNVq/fQg7f4vSR4dy  
qNi6hXXvYb54/bmWXnyqa/Id595ynnJnCUV1A0/ds8HZfQnJ+VHfvp95SMfeo+Mj40x1MXk8alpp  
RH2vVIgctN6I+HEcXhePDczOyM7uxuS9TI5fPgoPQnC3unpeU4wh3FMTc7I/MKs7060XD66xvwWd  
ERwgLGu3/ve93IBg+TP1jQsfBiNW/C5W9EwDkVwm9rqBoEzF45iQJZpJ+Hv67VmmHsDY4Pi4Qja5  
pwHBqd4fXWLS3MOHlkhJw7oitgYrKZKGWryjTnBhvv9INOG+4Z73/5AGwfoaXMJ+k5leKzDtjBRL  
yb9MSpl0IBKw7G6a8OmCE5wd965X4QcN897Uwpc9b7xuhuv//Lk5NgPPPCxhzv7YfCL7Pha61+dv  
OK/ePk/2d7aeqrXbb+ryIorB3k2NeHywptedxNf86UvPsJ6YL05RrAFdUcSG7od+eKnPy7z114mf  
+tnf4GeDvnk5PQEEVrIfgJsYq5Xj0lwGPeDl0NIjA4ZoLFLhw678LpDLwLDxeLFWNa+VGgAw9PCk  
FBLxd/0Wh2GxB+++yP8HHg3bAo5SznF6J6Bp7Pm8Z4L42G4+JmIbF3RXmws8Ias7NAGChpww/8dQ  
n+E9r1BVw4tLfK5OG7QMxK1BYoMRJegmbalppjk4/YL5rP5MAAE12rDu3rUqjjaxkTLAcdaIwL1XX  
+VkFHiIN+X5Mkx2mwAGfoGbDfpl0shdupLfurm+9fyNN7z2nd//ptuTfWN9MRjpda+qvfpV1/z63  
k7rObfgfz1Nele6S9rEbr2weIIE+KuvermcOnmcngWe0eqF+BnGMT4yzoWzvhJaZhCpshkci6nb1  
tBwbm6BwtpQuQfX9+jRixja4r3PnD5JTz7M001pFuSI0EfCdyC+kGopnNfc64iCTozK2soqFzgAp  
4999OPyrW9+U+feIPSttKxlvkxCpQeXC2fIPV2+ivWsr4dhaS4IOdGMnF0VAe8wNEXtF0a/vLpG0  
gb05eDBgzW0lSittXkqFSWaoU5roTeeG/jGeiLAhYQ+2KzIg3EmvuZrx6TnoSqKrA97nnIHQ6bdh  
qahtAvj3ebRT/Pw+4HPeUHaqDVGFgdh/nOodu/bt958y3+3HwZfqKDRta9s9NPB77nd+za3cBedF  
2ticdTjOnmtE5PTXDStziBe8pIr5PFHhGFCW/urYpkxMT9DZnnTFOOWOGR93ZrtmkJUSLswRVEBJc  
BTEz06/70H5+MMP0dsh30V9116OnF9npG4TgHEiV4VxW6gLEgR+pkcDcQGAldNYbDBCpDjhcWY6s  
B/GC9YRar6IAuAlNS9VJUMz1BrE1Dz5nyExOMO+NEWP635PUAkhcb0RCBg1L8LGzNjXZRFRaK7rj  
i9G7NsgCwpEicwbPY4ZXnu0Oeo2nxrDZLQieGHPe3wNBOWHwGnecl7mvf/3KV77yZ5uN2j987AuPf  
2rfWC8MIx1znuM0t+Be7W74UajOU6SMO3wszdFxF3JO09sg57v66qvp5RCKAQvtb+3IqFvMzYUFa  
bV3GQq3ujsu5J2huuDM1prLQWdJKgDLCB5k0RkkHiA+bIFL7N4HKOqxI4d1Y2tTnn/+eYJMKKrTp  
0/L/OIcvTYMs+Y8E8TO9jpd6bT3fGP5lnzta0/KZx/5nA99u14mVMNekPNRmxxp1k05w8CuQEYwj  
ahEAtEeHhUbFkLtVqdNfWCoVEBgTUS9RUB0VSQtCuErqytpQV6wAkU1ft1mzLoUwGbPqjfwGJdi4

chHE5h+g94djCs8n6e+Yd+F2ohiwEnGBjKG2beYe9toaqsduojabR9iRrohFDiuHifwRb6byG0MT  
WewV/Z6tQeuuuqqr4+ONv/Gl770lWf2jfX8DHen3c26x3mPq91iWgRwgvAqL7TdCyHg5NSM9wZpY  
NdccsklmjvBmBHSuecmnTGuLq/J/MwBWxEGgxaxllvotlxlRjJi14WzA5ZhYIDI/1wYFrwWapKd7  
q4ziI5bXNpzSrTXedntrRYNM3eL8/Sp59kWh+MEPREGDK+7vrYhT3/r2/LFLzwhqytnAnAVRZr70  
VzR0VOvc3EjnGQ3jDOAtO+MlueohqbC0DVHFDKSUB2BsaKeiw2jwd7VUR47Dd8LjTP81SgIqsHLK  
nCkWk9gSoHNZVkaCkg1jBUpsiC0hs2AIbH7L8sHAT0eDFQWJsbwaMi0RsIIBueD59Dq107b0i8oj  
kLpB+eoIXeJXBdpLxxfonXmpvv817jvT7zsZVd9udlovO1LX/5qa99YzwsjfeXBBDC4qzfIL3cLd  
xEghZUaIGqG/1IQcmBcCA1NksQW86jLR3cwb8YZEAXmcnRSlvNYpmYmpb2HFjZxhgpAadaFum5Rj  
U5IMxrhOAqUW+A5Wc8cneD7biGs7fTkwmFZlxOOMAdEKxsW56EDB11/3d3ZkHn3czYoZNV9fODAI  
hc/yPlYmMdPnJLHvvCoz1MVYTXvqKiv1kvJGqpZCFnQ+yLpQTtdoPgPA4ABYSFDGhUGuDDjzmWgv  
wNbiSE0qJR4v6QMf7Ex1D0x33ScOHZAe3MYTcOdO/JHk4/p9z2TCSJrmDeLz+cs2j4nu2e5gUxK/  
K81G+zk6fYyz7pKSJ2088WxF54ppeCaoltRkemc20odN6ewm/jn4il33Lf2+vl3r3n51R91G+bf/  
eznHv0Ll0rdB5j+lMeNN1x75FWvfPkjnU73W+1u90bnQRbZp5nUedMt15pxng5CYyzss6aXhoWak  
YwwtCkyjCakhgUEXk+C7pYlabvF0U8zKutj7mmnDxKakth397pEKMFOQR8n8lv8e2FxVhYXDsium  
loQ//daPRoXvObZzXVpd3Ziym9t73BD0YReVhjo8eMcjQHQ6e577uHvSi9XoqjwonqOg6GwUlvUa  
n4CQEFxcTCUzIgKJelSoXBkbMwboII2NPxCecX6OZEXW4sJVunGFsm5tFfyj0WfQx5LIw8ibYSUV  
McJOSbwdXralO122EDxWnhPSK0SSPIgmtZ/1UNj84VRmrZxof5en6dl5oHPjHPM85jUS5gNXtPvd  
Ra7vcFPn93Y+tZ11776F37h5/9ptG+s36PHzTfdCNTlpY/s7e19w91UZ6TpVM+5POzu2HEpJzJQ/  
aBZMI/qTd5sHRmRBQKATYAhb8TcUws34X2npyakM9DdGojp2sopGR2fYo6K/BahL4EdD3LASFHKi  
ZnDpRoSM7fL5ejRwx3Uc6BkU3ifc5uEaianZuUDfczyjRTc7POUHfld377d+Xs+oqWZ7go+9wcc  
G4Q56ZhpWV+aGwgXczqLUEVhFHSYL3BFOJDW7eakB8C1DLPZSUVGBxz3khC7jjpgi50PwYvMe08VA  
y98J0/d85HtmODwYm/Xqkih3hHpBNhU2BARTaDrBqAarp95TZSY0kEeQnk2Gehu5I1Wv/IwICwmG  
0v88eQ8VzEBcn7HZ/S7vct299q//NGHPv7Vf/Azf+/AvrH+RRrpLTceueH66x7Z3W092W53b3Q3b  
Mp2YeYwLo/DLk3CgLuRE2OTRD0NbFFurBfepvqgLiDUNudmp0n3Ix3PhWWTMg3a63CiOIz3zO116  
vbWaiPOo2JIcsTPAQkffOGxUQUW4GFJTOA80xGS4QtPIQSQA08Ow4SXHZ9oMi8GeWFiZlqWT56SD  
991t3zr6W+ShKDKginlQ3HslAuNmlyY2CQQsq00CUPSnDbyPZEajSAYc9bXVjjoJ7kNAqExNqhRt  
wlxWHJcDwaHDYyIsq/T5p7KiG4ZC7lX3WCJ7IzLJJRrzLgL77cgy2qGis9VlYmYhli4kDf14zuMf  
ph7Q0tzzXM1Pxf9wu9IRcw4nR0RgZWOCp8xZ/ZTXhq6bcwcWM2WvH7Tba7XfP6xL37qR3/kh+r7x  
vrn/HjTm26au/6GV3+y0+59o91u39jt9aaiXhDNC5fYxpV6QjrZMANlPW3Eg3gPpZ6gCIsbHvXb  
jgQcqI8Mj417Xb5lOAQYjDTkyq2vbbvkZPHv+u86A7J9zAetLNNT03TU8BIC7/QsKCAupIr3FciO  
9T8dzd3ZbfTdZ72KH+3urou086bwkhOfPc5Of7s8/Lxjz8kaa8LzJMLGYvdBbEyMqKbgIJAykaix  
0sk5N4KwJSelnlu7jtuEvGEib6cXj7Dv0epxsJcy+GDARSKNsMYYaD9IvMdPJGXgamzZKVSL/UgE  
2N/n/hSkG2/wvVC+AtkHohwZ9D3gFPmQaw0vD7z4T3vaX+gqheewcTni9i/zgAwbySo71ZYU/Z+d  
kx6a7Sp3t2fK5999ruPvv2HfvDPPST+Kwkwveltb5zY2t79k263d7XbcRcHXW+YUDHIFeq3Dg+UN  
ah2HyW+j1PvIPK/pYXFQBA5IktTFxWeOTjz52gMU+MTMj2zqY06yADoCulybawlvMrMj7akKy95  
wwJqhCdIPNJSU+fM3H3d8+326jTDLzYnMmmylF7/QYpgthAjrhcFoa8tdciS6jd7dHLwaO8+73vZ  
akIQmMpAJaBTnBrNrQ8k/v8FTVMhuv+PE2Yu1Ef4aYjftobyzn1ctod/g2Ko0md4m9YU/XGxfesq  
eEhIkHeH3uWEcj1wSsacR9c4KbWUw20s5+tM6es9apxsfdVLGTxkZL9tOPHCxhFSHQ6FfozBABwK  
xMxbnPO8hvBNEQL9RqPk3/mN2yytDwgp1Pw1IHCGyNaxpAwJBguc3rNiRMnPUh+9RP7nvXP+Piht  
7+lefMtN9yztrH5nVarfatbO4sD5pMqIKOymikRRtwwm9od+bCPYy0KDQfBKlrfWAti1zaYKZIyB  
4PhPfvss86YFDhCt8vExLQUUDFwoTXyz5dcdinDzrHRCff6jgqSCWlvLmRVhQWAJ5BYQVmk32tL2  
4XN4AOvrq/JkSOHKCW67gwTxnpgfk62NlschozyfV+8P2y7PjiLLhuvxUAM0on+XCUBP2aL6EY1  
VDri964BvS0ZdjPlRrC1DMrp/m7pcWDNHCEpamvPr6a5RBu07MVQuPyuZR8N5kJKv5aFq3nDl4F  
9/dY5taj55UPZt25GSLAWV6PzTSUc+Idj0Ca4XKp0a+lFRozYgGaU30FuoWvtQU27EQbk5CvozXI  
VrBd/weoFa703n7DTe89tZ9Y/1PfPzYj78jeuPtt/yq8zTPuAX/jl57sKiIbep35cJ51EI9TlqEB  
ZJ6owPDhYbIEHWUuzDKENoC1mcnh93w8uYXnNWC7hqEhfBWmOpGzV4MJC4AxrTloiNHSR1MatBUM  
vZSJhERzJ4zZoAlvb0u024gH9rrd9z3lnqcQpFLTB3HMaOeCHJF7nKvpvPYD9z/Mao+4BxQE2UoL  
3EYWWEe3IzEIgd8twc9dyIeOS7zVaK6zhHw11f4PvPzi6yvmiFs6mgyFMJaKG2tcPCqaEzgeEiQ7  
ws1qFoyEkomBi5ppCGHE8kYVBNr2AqS2XWv/h3+n8QqW5pHeZkvR3qvjJXBfxO8UG+yrYS7/WiOmu  
BQv52aDzcAjyKzBfBUwmxbn5jaK5u5u6w/e8fa3JfvG+h/5ePObb3mHW8Bptfe2fz4bpEdTD5DQC  
/X7oYaW5joxXBIJoWfPReIAvOBmwWDHnNdCyMecSfpszjbPgZtnYMRXv/YVbgiQBsUNhrfD5410j  
TkvuefCYJe7jo+yYwRGjxwSTBp4SRg2cllsCnudbedRW/R0YCPBMI3WuLgwwxIOAa25Of786U99V  
u784Ad9HVUFzvKiVHag0dVjT/XLA53QckIsRII8US0gp2YsXNTuc5fPrja6ABLDNzEJmwDKWZJ5  
mkWcsPQOQMHhdJNrrIwocRDR5cGQMtyQzM+3De90F6fGBKnud8QMh0irSykRESxnodsZ/F+enliU  
77g+3v0t4iRGsTBY2JXLTYyjftn18FQ7vK7b4wHHJXl1f1D04LLTp0//2r6x/v88fviHv//i229/3  
eOtdveDu629K10o2RwgR4p0MaYeJSRimKJ2GAUBME4u8yRzM0CrK1LusqnN1+FGoQbpQyYbzCQeS  
Tx98gznnS4uzQpUDcemJqXn3h/Nz6PjE5RrabqFLYV2nkTaY0nmERq0EQZv722w0RtoKwnyhd46G



AlAIYTLMOD5OzerOoMhnfATn/gEARdsSGiVQyJXROjrPR01TjD8KSo9KuuduXbZYGfIc9KmbW9oo  
100CBdxbOD+QvepXhsNwmcgVjCFYJnDh76e7ECAShI/X1Y1hW38hoW6KEHBWxrDKXhmv3GUaKw3F  
KK2qagckzfyJAobLX7mJpXwqNyp4J5mlFmLSpxryMxz9+EzJ7HHXi5VktbVn4q0nhz7GjJrx7ECY  
5EpiwFS2eaCddTu9H7q9tture8b6//L461vfcM7Nzd2/003M0B/IqYSk/0S86ZnbJnirYbKez7wz  
JWIwITldPA4Q2yk5sgQR5aTWiohr+2yJRBSSezx1FPf0vcstIsFYBa8339wHndhbp7h49jYVGiqx  
s2Gh4RH3dxYZVmm3+2ozOeoAj1jE+PScBsGhkWR0bS4JIShjrBXdw11WU6fWpbnn/s082qEv3CTj  
WZT64U+pEQt15rKufBE827zrJrnDXw04YkKDFVMnCq3uS5H3WdSdnRUwR5sBCTqlyJP0UvO2fRKe  
ZgAxnlvzzwalMJM29qYP3rjNxD1V5xXAibtUfVPGOpzRR74bTYN71H/ndJomWm2G/KFh7zc2L1y  
omyLsLGDSHYw+Hcb5ISyjisDRdq9FIMwsYeFeX5unM9un5287f3jfwcxzve8ea522+7+fHdndbP9  
fuDqQEL98NIItk5keoCAenFDbdd30ALEolIaXXwZOPU9Y2G4HrcXNUDKqeiRVE1PdGbC9QYRrOwt  
OiNWuuOe+2WHDx01JTE8YkpGg28N4wRpRfUVkcNJonEKNG94QwHdcseRbknnVfWcYoJPTTa8PCZ8  
NgfVcuabtc18LZWqU8gU2o6ZFU5m/GZPJyKnlRsuZC7dg9Z94XXh86w/OzC9Lw18Vea8ZUS/bdtd  
ghfDWiCoQFt1WMoF7blnXyP3I/iQJRiaoUEfoTemSLfRalbJHsaFnVcCxuEUhtliCGFXNUGOE9T  
eTNFCjsb/1heRRCaNxeOy97jbK7LayOyVlmJiRZSIXchv1m513/s3PXC65084Y3XBu5hf7G717rX  
4y5FX3w4NFx1MHQx00uymSvny/Ck+YqwgPoKuQyU2XXGEqYAgzwtDcarZHI3SUFz7Q22mTeVX3YD  
Q/cWb8w7d8hZ4098MBwOJbPf/7zcsPltyhBwBlIa2tb5haW5PChi9lvunjwIMkKlAt1HjVNMepwx  
hnkeFCIGicEk21vT/NehooM8xo0cDSSo/H83e9+t2xsraeApl6zJu2BIRWNetG1sD7wc/3iqoJJM  
ZBiX+vFA80DY86jo+WNC4i/U+DGWFow6CqQ1JjHnNazJHhVRDUkRbjnepn+PUC1xJBaboYJhdHM4  
JH7RjbdPfYzghB2x1KqQ1RGTabD95usecQ4qtIbY268WtaBV4zoiQeZhsOhX48Sj2RXG91tMyvoT  
Xn9sHb8ZiMeVU7TfHFza+dn3BO//VfCs95yy7XRdde97FdX188+s7F+9k+mZuZuXFg4eM2g27vM5  
U6XuQXrvgaLOXRkCw1RsBPnReR3t74ycoqcZRTkqFwYXtNWb2JBr4O80hqZq4Vw+1790XgSqZcLR  
iJf21a+7beffsrlq3Ua5uTsJAtRn5NrXvUqOfaSl8js4gEZmVatYISmOuhYc9FxaC85I0WbmQ4kT  
nQSm3tdo1nj+RjwBdT24Ycflme/83QAzsBKsgWK8wHryYynGppWG75Lj5H44y+BNTAaVYxtJoSui  
UfKzTA05PfoayZDZZYh2qAvDw0qnTiqwB8PbYwG9vHnQt1EWk7xsblHeAvvictatwFCBYxiT804  
y5rPlyGwxYC69+Uz4HdZEwoOcdTa5T12U3c5HK/2cUWetXd+vzHfyXC4JtuevWBvb32V3dbez+/t  
7N92cGDh5vsBW21pNUBQygjyJJVyi6W3zD0Ah/U1+5YiPehDksZqQ/LaqrgrgNw0y0p9n+pNOXeBm  
7Eqpc2rAdquberw71499vnH+drJ8RnOtmkNOnL55dfI0sGjMuXyVoS6oBaynzTvs5QD8EcR1IR9l  
ZwG3mwSDTbhMmNM4XWHXUj9hs98gV4N64MC3D40rPm5MkbXO5dVd07GY6WSEA66xQdDxWehkaAa7  
prhV9lFcSKha8UMylhL9LKIQvLII7P1oU2R3OMiDqFvIP/HWiLBRLijf/X1VPHIciVnsfsfwTuoG  
hrUbPKpoSyEex/Fobab+/CXDQMobfFLN29YzjXk12vk3i8ajsKAUTBw9wCku3YHX/2qv4y/qI315  
puvu7LT6T2+1969xnnQ5uzsIj0Gam0wQIXNZ6oMUGnBANUP+RkU731DRAn2Vkc0pbxUyhtMRbxaI  
yzoquxIQK/5f5eopC9nRIoa8neF5TFCbw05UOSaaAyHxRw7dgnRYRgchMzwPgshDJKcjT2pAItW2  
Tz6ngh98TqEnlicVkbC+Z4+fVLe9a53yZkzJyphZhyYOQh/oUNs4maWlxpvdYhCqBiG1VuwAYBiC  
ama0Enj/7bba5cbWVHmqQB+VxVUD6Svmikf3J+ayXpwfJYvq4WaYSANdHKK0OY3QaTFENGHELVI  
g5lFNTw9PniBfcxqpDzbrNHiYvhea7IPDxuNuh7MoW+T02UTFH12ASgojSUhhTwGgTPip/deUHb6  
R+/aI31DW947Vi/072v0+0eRc0RHmjMt2CpsWiI2zWzgkV+50a+2ScXNOeF7g7SACar9ZJpfyJRR  
W98I80xT08rL00VSLBFXQp0ZaGQH3k0WXD7RZdjMXJFIffffz/pa+AIH3E5Kjtp304+MzvPRT06P  
sJIYengIROtpFnarT6HVqFkg9fjmAB41V4v59889tgX/fuXIWWWK7OKM1Ch6JCVSK/VLy0crXoJM  
1Ao/rOVzhk/dImnZ+aoRhiMJo+GjJ1lFQ/4WL4cV/SALQJgrhlFQWXQrqkZHTpj8D52jxURLoE80  
AsZLkc1zVd9dGAePOAGIKP14RJTkiKR4r4U3oCDMHISpBRWekKWjLQJGzCbHzIdzsWWOd8aqABUJ  
WzWTLiCdfB4mm6d/vSL11jb7e5D7V73MjRoY5edcQs41PAi6/IX3mA8n/rcSNkkuacLeug/KvOnz  
LeokYVipOxCMt3GiT03LzGjDQuIjJo4U0OGmD0596xxEVDiD33oQ0Rz5+Y0ydLRywIoYeEceSRKM  
Kr4V2P4C4/abKixkG0EL1tvBpldG/iEQ9nc3JI73n8nmU4kaGSDwKRJ6koKYD5X03CVYWjqSfKhC  
bsYyvfmMog9QSVee9vU1AyNIYS8tSgYAY0+10jHNRikyH9KyDHJaa6UbWisaZkz270N6G8R+RqtD  
1mjmr8Hw/fbVayN5VQdr6GRjhdEGwyG2uMKgoxJyKktH7W6tIKPXvtYU19z1gHRVuYjeURsYynz/  
SLKFU22ETp53GP2huuvS150xnrj6659h7s4ryZJvtMhKUASC5Mk3ExV2tPWLdABsfPZQ1QpEUUMG  
4mqAVRBAWru5JrzIKcyRktZa0yGSjPVckDzeeHxSo8ActQv8houSFON71Pl71d+5VcIDB06ctgZ6  
tINEIR1GLGGjVFQ07ednaBSXVvjSFaIS45ut9OXz372M3Li5HfDqAnzZNqipoaKiW2sH8ZKzTojq  
VL4qnNk2FPqFhpUCXudLumSFDIb5F4su0wJlEeNDWA4bby5xTW9vt54w2CpuM5UBeg0w9MCqQsa+  
4tgsDZXSJvDDh23t08EspK8PTmqav3zDZygl+e5C+e76sEmEHoV418uS3GZsExARZF9MsSkEjo3  
02szp5Hw17buMZ87SCwqdzvpnr990CLzljzNPTXLi9tIgQhEWB8mjVT8XqySmzIeKFw/atc0tyK+  
8ZOqXvZjiINO2Z17AJ6STmuIfKUM18PjeMyjKveeJu3wsVRKPJHipkoAj1cLtB6LX7/wAMPyJv/z  
b/WDp3Egxc1bVFDvgRPZvleaL524TwaAOH5vSqDhcArJ1fkW3ffHQgag8q8VOS9rLHWtauF51SpK  
cNY7N9WI7QFz03OWRaID1BQbLqNpVoSqfJ9EdZmeZ/lG6gJVj2X1jS98cb10FJoZZ66Jz7wc3H8f  
U1NkjBIWUkRMPKSb12plyYxja2MaDxqDEaae0+8X5nW6HLPgUL71CUTz1zyYFvCk3tmC2WiRawL1  
/xaoY6fyZKkCIL3t4UQ6IiHyZQ6PFPua9ffFEZ6+tvuv4fdfv9S7hgKS2pXsUYJODaYjfUQnstl  
CqsJxIXvyorQvqgD3VpxOg/jSTA/VSPbzbCuIhq3mPT0bRNzgriFSP0xIfDJew/tOnkeuNTP0jppq

ae+Eco/qKsOWF8dpabt1PQsm85R5mm5HB3K+SgtjbtwufD83ciHZnjfO+/6I8xroReCprB1mIALZ  
KUR25iqi0vZOdrMZmE/v9cVmMPfPv/8d2V8dIw5snoVj5pXWEkcUuXC9ixVgAXpBzYaPEcv6gM+C  
9v1uyK4WablJFVuKCqpRGrkpRBCW14tkdVNff2WgnWp5qqRYheMeCI fCntJiSKTAGpxaLpRSOt+A  
/KblKH3Csp5+iCkSr0njnxEZp1YVeeQ5fnwPY/85sD7PwhKIu71t7xoJPX733TrvFtc/6u7WE0k+  
kAiVRg6DjxTekWfs+V+EreGm6myWxBKJX6SmRQVKF/Y4cH6q/cq1PIo0G+pYxn0huWeYiZDobCGm  
ekQgBG8rUmc2Feo8ensUmteP3bsIuZ/MDiUn6zli6MJ3euBrsIg0dsKsj5KNcbRRR4b+c/79reel  
Y9+9F7NsdxrEWdY+Duk+OANq5onktqH0k8+GOqEwcaIPH55ZYULH+EvJ8D53NvCRer4oq0wIM5aK  
lIpz2Ko/GOBxOHZOpSRsLnWG7HKp6CsU6vUe0OombG8GVIO7/0YHRX6d2QKZRKohmWdOA/AISmBN  
SunqcGixs65rsxBtaHeelUpI+spigYiJb6mHPJxH2X085KqKP6e21fk3yuKCr+urCFCx18Uxvobv  
/GrkfMwD7uTOkoCuhe5Aq+VVLBYvS10dvqDsteRuYbvWQy9pdT7iaGP4zsmIhq1TWIwPR8suphzR  
htB+b3KeDGjNQPVxvJiyIBNnQ830Yw2Ng9duXF462PHjnlpzMQrMyjfGHNjIO9CKdNYwzOT0IyU5  
+Y/q+AG9r+/8//QNjogtrlyVGEI5OY6T4m/pVHU60PsHTMipBHwyPCKJUNBE/R3t2UOoMGLGSCA/I  
BQMORo8nNSHcnfDEOpkPIWmcYmC1zavFXmONFQH+beehQQJUD5LjyBz0xWfHxsQZVFFHexCIMcMh  
ZFY4RvE+yWdyjjAhbYfgr0W0hWf9tjdNS9aVNKYoiG7b0LJDhGX/6sgHicGaKU+S1OBPAJUREATN  
256/Y21C95YH3zoE7/nrsM1DBkHKJoGAXsZGfd9g9EQ81ptTA4XPPck7agWclJQ7qrME441dB6Wp  
HEsHE9AYF5Wq/khWXHIT6tlyCrSaArYm4BqCHau5zXk9IrLr+Tnow1LF3NCIAdh3e2tn0nTUKew  
AkGiYJK/jORCXz0/o/JU08+SUK9Cpk5L9oYpfeG19McOKUnDuSHClNjNxcNa6PY56CivGSQH8anZ  
2TC13Uth7V81cLg3HvyLLCAsNBzJV8YH9ir7AfAjvIo+VctWu9fRMV8eE/SFIuUobFFS+UGUQKL1  
ssaVe69dT5Zy1vkaY0hVKXaWhRKfrqBWC00DrV3y3PDRmwsKkiwSjHkye3cxc+rDU0RkYTKAJvUo  
/CemMI+fUEb6y/90j+7ttPq/BSJDiiVVICBkHt16iGzgYSZMZyQcFfNNXfxfoRh2YjPnzR3UGTUD  
nTzpFaqstCsRPNisZkpVQ2hsFkUmrszwPVyIKETx4y0KPziy5mLQuuXuVSkqvHwfpSB69owMD1BA  
j9MCXmsDhIujxOPs+ub8r73vS9oQIWM7rTngaUkPFFVUEoqSPbAq+xbfZT1F/c5Z5ZXCWhtN0wOM  
ZiMimkbox3XwCv1J7w3fSViiAJNMDzb2HQwc0115Cac5aG0UxQ1h5geKNOYJGEfmxZOJDedJFEmU  
+jaqTCdJC+nxPlrEYf1YXSW965cXCWdUfPpvCgq2sV63ULdG0djQhOahm6ec0klADRDGiBZAJ0Qu  
bjfo2B+5QVrrL/zO++sP/rZz93pLmkToR3VAUXVBLW3skY6mAqY1fyks4i6u7bLhZqh606d5VqY7  
gHc8WEySzXuUVsOFESytPPjMajUHqvyosqKiYdYS4UPxXKvOGsASDUM5uL02rRoVofOMNhxCFPxW  
pAaAPQATJqZmaIXbeJ3knKEBTedrFwIDz30kKyuLTMEhndE2afwYyRU+GwYtQ2GW+Hk1i1PjeoKN  
L1/L6+tslY7PzcnI/VGMB6r7Ya5qHkZ1TDUKzQ8jTwImGeVOTUeoSUqy3psXZlmHhwsqY4KMgU0n  
fdQSQqpR9pt81Eg1QISP/uRIYYwp++dY4lmzUGBNPKdtj1o1WCi065JVwOB8aERKIMD5XtmYDq  
ahGhOhFXb904Av3oeQow706FO5HL1hjvetD93yqM8guM31X242xoJoNBWEQFquAmWeaJLqQIyn7H  
g3XrmrtEOKrTAe3V3rrKA6gh8mYUFmVWhoIBpcHgKDqYR1KGj0vyl+ABFeRZYTWR44cIQML52mKg  
ghlwYqCx0foGzAwHkX1fW1v8+cIFtedd97JcBWzWDAbZuCBKzNIRgyVsLy6kZVIsErUqL1FbCKHd  
1+cX/KMJ11oVvIJ5Qq3cvtk/cShRY1ADze+F3YnmYbyIO1oplDef0FqWG0gPRXByeecyWv9hh7Z  
XEaRRyE3DSXzk09GE3y2IDsnpsMC4P8SOvf5BYD1GskgVvMhnHPPLPPwgZEr+y55I150KISbVjpx  
m8MWsoqgTxTg9RyWy0Ate51r78gjFutb77tf+r1uq/GBSaa664Nxx1kmeYzaGwOIU8cBQie5RfPc  
NHFqPWzmvQsfX5uYbO+TYXbKX/plQWMg1qlwRwmJlDEQ0aoi0BDTuoAZTKk+yPew2uLlG5CBw4cC  
KWQyEuEaCP4GMsmIEmwzJP1BDakhdrzU71TD70R3fLqdmn1PuD8FB4FQPk1NY7WouCqkHpfTTKM  
LDLas1ad07JLYZiPwyKRhqXWkNWlZXPxRpmZGFnQqAo9/eGsqGeZUSerrunIG5E0tD3YR3V18fFp  
HOGO38ism+9YJuXvXFLYKGSaYSasRSUkXLQTVciyz8vXXxmCQqjdor01jdOqIkhIGBaQ55b4qPI  
/WeJefYk11kuDTkFpDX6ypCV08o8URF1ata6tK44Iz1p37qx6LBIP1HWVE0bRciLzMvieZYyNpHm  
FXCi6aSrKwLQRiemkZt4OemS1Wrx0HX1cgDVue0NjDbgbVnsQhk/qEOm6JEf0NXj+fCMgyKy06bO  
Cy8YdHvK664QtUm41I5r1Ef8/mkIoZ6/JEz6jlo5ild0j2PuTfvec97ArvINGMFLxrB8K0TxxhhKx  
haifKgHhxieuXQAnmh1bY3HsTA/z4VIj5+UQtyJ12giz5dyKEV4jh4/0mNO6lqnNn4uzpHKjLiHY  
6O+BpqXm51vlE+SgcngZ2aezVXT9oswoqMIm4DVRoGo15N62fKHXroxxH9GVnKTJQpAmgQ/KAH1j  
XNT4ddeVza8czi0L9vVy/w9qkYFvg4MJJ6ls6giUYt1HVUBCyospwvPWF34hTvWKFvRlHkieUlnq  
zdHgik79kpmFUPX2esYi3I1la+ZE+UjCcKHVrkvh5RICuL1lJoBwo/ERj7koUYq54R35nHsPcyrc  
BGJF0s7R+TL1cleesX3adhVCe+M4WNoY2ga6Lcwm9AtmCYN9I73vk82NtdL4TAPcFCUu+8FyXxYh  
w3PrpdTudygbfJ7e5tsUwDBBrXwRQ14sqvKtssdk0I7U7I88ilwWVj4CBdr3iAiXw4zDSL11DGbE  
YoAs0aBgKobxlSii8DMXPVrCTKQ9tCVRKtssd/1RMGT+P+U9QtmJ9M9apXNHm/HpDbOS3KI6TgWJE  
/TEE1XKN1hDks8psdionNHQuSGelBwdjTqIh91IUJedOaL/wjPweex5I3cls0j6gs9vPg7xIn4X6J  
AhnVbmrWmyOwoIsYfg8hH14P+UM971rluOSN2qfQZQ1aMTmoecJLskGeYq11GpvlYsOHgUZltRe  
assl8ZNBbf58OHD7PgZJbjk56p4aiRzSN5c5ylHpiRpuq941IZ96uSy3PORjwS1f+Sp11fJxQ9sz  
VIVPgvrRDOfzdOZRiIAXGDQEOBWhoIJ321TG0K+uSkwdHVRcckGvpTj8zqlMaURkmzTskgK/RaQ  
7pmlMQM/7c+F9aFG/P+auiah5y/mv8bgwm9pEE+VPwmJPEQH5fGFXvRtGrHi/hNOq9281TI/OoTQ  
0koDuGqD9Zxzf2GkviNw+b7BAVEkw7ytdsoKfuAq61TXh12dWECTLH895hea7o5ZMcgTGXY1gh1q  
7wyvsBAPCEASJKgxq71FvGLQskReSi5RAokidLT614ErRpqGWhgtcChdrkirpRvsrBAJQ9chMkoU  
mnTSimMBgWiDnALukNUwa/Vw0ITSUMjdeyJ53jPP/mTP6aiPkkenlppA5vs56qRGYUjixdhcE3zb

qQE9vyJE89TywnGqp9fV1JHhQfNvNHn1lGubJ6opmURiUuOsPGz+bkut8YsH2hPwUjU+Hw7otY6+  
KUPSi10SdK8HEXWRXSSB9aS7YuGHheWesRZUD0k/zgtynwW9y9TqqdNcKfB+u82PgMITpV+SnDMR  
08VNpNJsAa0Px/uyirJJ2l0lysjrLzSTldcmMY60qx/0p3QKV6EKOYNNiOpaslaUb2ksw03f5sxi  
meZSGihS3whX0LoSxqi9xwI/0JIW/Hg54a/WkwvhoS5Avul0oXCdijb8UVCbqwTySdLAEY0NBybb  
FaG+RbsymGZg+WITNZWz8of//Ef89h6qbZ4NYKhZmGMhAE4VkeFt6PH8BPSrPaIzWltfZ1GMT87x  
0iCqoqV0LacSVppso4bOkIEJTRRwC33rDIYB8PsQe4jHaY4Mj0545H5iJxfZSRpaYMMqHNRdyx+9  
hgnobtKNXyzkCbpta4OdE6DBGnh6YvmWVPfh2o19iwb1BtsEQ13UHn1Ck21Co/+qZzL9OSSV11C  
09hTQEENZOhur91ZNmGbU0Y56ZXF4yxuovj1krRwCapdTDNKcxYrQhOz5HEYWznSXWLwrWwKUzXp  
+yoSSrF+9C61XpwsZKhXObcx7DI9HA4bvWzaluYlWmqryPN0C2fSy65uOK91TFFFXyX66DuWhUcs  
3odEeAPfUhaezuhc8N4v1bwD721RRzYUyq1GfmZNPQG1+7vSdbZ9epMUwE2usAczMpZHB5HDsu  
vWx8CO9B8jziIpmqvJobWLIzcFtRjmq3jS514IsJ4rWJV6TF+FwnHPYVhEMNfPEf0+GkEoDeRYP9  
dJat0/kkVrxZAJlCXuU26cnZRmlpzn66hBqBuF9p3Qe2z1JK4eb9ws22GSAZxHXoJk3Ihi5ZcXR  
Rby15JgY8oRGecMVYQM6heksbY7nX/qjG6R8ipxLZRvqqGpkQ74eyygfLi7JXA/vffLPVhQ9ciB5  
G71gAjex0LbWlCeN25vNR8KbKVKV825xPhqTlKVsjSu6NEjF6vRjFaG0PAMC0UbuSs0Rh+KY5bN3  
Xffrc0KHi2HMfchGyEUmMLlJOWXfarYkMA7rioX4nhPnDopU86j6siOrkXhcKAgEl6XKVB1XTOIR  
BRA095fGrdnYVnzUB3jgA009ihVqu0zCgRFRYwftJGzUnY0YpqShOaKnLXiNJRqzDufq9ZhpB85S  
1juehZqkIXPeckFz/Ny0lvkJ8j5GTg1Ke9ZNceMLOY2HjX5595AcX08/lHorVRh8JoPs6MS5EIva  
856f6OysfP4BxeUsd76htdFr73+VW900/yfg9VhpQDL14aUBTHTxQTMPApsCC5rrrm+pmwWzg07x  
DRyonojeFg8gF9CrCvIVZrOTlyEsCurNDHnXgLTmo8t/z13toqFPWWXjjY5X3rppQxT4UmNomfdH  
XouacVrxPS2mKkKr9rzOkAGAMXeaMPQJ3iNQuCskuEvmrz9qMa+ns+pUyfieJqemtI6ZKgpKhoK4  
1ZxORlSyY8yLY9hM9CmhyS0ChqtECLqKNWakYV/g6G1ZaKaYtSJcrAtHzwXuUUEwVaL9kuliKwMk  
403bDVaI8iYR4MHptSK9/KRIk5MCSKWZXY7G0k1SclAKkqViaE8s1LOK3zrm1QI+8ZqKnxkwTC3k  
CoP+Bydrigg1BeMsd52+80Pdnrd424xMTtuFNxVklQ1ZDNAABc8EasOVfkCdOaG6VeDaEfPJ+WH  
PLQ1qWDMnzCGajHxm2DRx34sYLaxZ8NXVDzcvEQ8psbkZVeuQhesxhSzjsX08j96IkjR46xWwazb  
CwvDz23GBs50NGKnOTmfr+ysir33XefL/so9U9FvdIALlVJDxRXkyKEpZyJ5NXicX4AfGBMaHuDs  
Hi9VkkQYHo3o5I8HWrls0RuamCIoUnPmDhc/CxkQhITogz/AW4ZOJueeTXuyc/WA3YZHeY3rAGn  
Sv3QLIhgjynpQcU3jOmirgMVC0QRBVBCp9Xkl1lav2Jz0OjqDICwNviZbeikiCXIttGCX67DGLR  
NeAAYmq2ZyECosWFP08nExe0MgxXHGI1Lwxj/Tt/52/e1ut1b8aQKJfPNEFuyCQaah6veb2g1Oeh0  
p3N9IVUAKTDEJ3uZaWbEDYXUZijwrICB+dqQzplRHyz9EM1VKFb6EaCoPzqAQTFa4EKZihzpsK3  
fHc8Fk8CLR4YMKr36sSInR+ScEDjbKmiCS2ZeURFxx+vNfa0WjAi4+hnqg7c16CSn5MBQgaPP+ss  
mkxTNVa6/L6iswvLrCHVo2iLNPoMQ9Cw7h+1q95Wy7IPlYJPaZGmMfnZYOMQ6KRr+I9gTizEgjdK  
56vG8lQp40ahtZtLSeVMhD10ioSyBNMGxZUE1fb/nAsRpgpdF6QAYG2KVWjFst3q22Qeh8HYdp51  
TSh7W9JUIWwI4y9dE9UkWKVEBpHlVJgNGRuflbv+j031pPPH3+nM7omZVDIti98HStW4KfwLVCEs  
5kHnY1KjczfiDgyT1cvb4hR2ST2YxRqAZQCM4nQvh+HAI+bxFJpq8o8saYMo+MKcKcqFBLqmNwDn  
1ATfd3WygR4r+mZSZmcmB2qCdqEM5teluVxoKohbP3whz9MT1j1maKjRlFfDYmZn6IMFUGdic+Hh  
9k6x08+L+PNEZkYm1QedFxVZhCS67MUx1LzQ4211xRG0CBRYhAWvY2YKAXJAOANXNTQIYhDckWtN  
kS/zKVsdtCNTVTMN5pk2ZCRhWtYlmzK7qGiMmNIPBcax2R0P0QtWZo04RlUshCdXsBIBrtwosaoE  
ZgpElqtW0Ljg5H7yzEZ8XCdNi5C+mYpkK7POJhXdVK6P5f+eW+sP/bXfuTV/bS4WJN9L+UILxkNC  
5KBqokQKEeCnvnphlVH6nwzyuQwvAnV+ZMt5MGNUNcbFvwBrbQU8TlOMHqTS9R3/J4zhX6thJDN  
ZR5AVFC95bQJTI/P09Di3w4GfshSGzxK8rGawtB7733X11ZPTPkPUlehnXUenlcia97hs6YuBIiu  
+sDkGrQ68oi1P9dHhmUHga5RzTz0JNKBQV4PG349BuZoe/DfGk9rp776uqkOnQBmb5xbsqH1nnjq  
4wVvWXqG5FZ1KjYWhEPgXwmsEYjTyQwnQx4DAaQWDeWuhfWf0yyCfAMr2SZFrkfQiahUcBKNlbaG  
wpRI3MMAaVuXg7yChuMiasFjlyqY1alX83DmubVeW+sm9sb74bYMYdFeQkV9q4GsbPIy2WoLau0l  
qyzRQ2yFC8D68U4u3gei2WPodieGkWWByU7k28xbzBa94yYotR8DdIleSlNaqG2hseFJ9XHoAHav  
MwLpVwiDma2w0ZrXOTLAzh+hMa1L4saxiqKLqzcpQV33PEBffz2dT9jK5lsiqla+ZBWsc2+YTO65  
wa39rZwzeXgoWMEd8LmY3ROUSG3JDSL93UkqT92at9G1Xx8uOsIqS76aDnw2eWqeLCWHHI+CX1xV  
J2v47tf6mxqL0W6DUxkKuS9b+JLIBkJExKiK2sSj70RvOM8L1lFUTmIywgrifiIw9fBI8+tttrA20  
IG41EN16uOVLUsZw1hrbKVIQsAzrvqqFCQMlUaIoc12/Nur15vntWX/wB95ykQvvLlYtooc3apIl  
cHy2JVIXuDEsLQ4DKPgpyizJorZGUNA3NTvC9yRF3o+jfKxCkV6Xlzl1+SyId5vCqQVf00vXroY  
dRcuXhJwXDD6GVFBfAYEv/S3y8dOERwaXxkVAXeqC4Y+5YpkU67RwOE6NlnHv20nDp9vPTQFZ1LT  
laL6y8QG4emsNUeTSYTrz9z5gy9+vJYmObFRUkeSTNveUjzPGBnuW8z44AKT35AWGqgjk1rslwVG  
wKGNPc8q4oocHNMa6t+7mrqgbQ41El0AFRjZpchaq0eiP7ihdsthAw19cxrK6WVWrcn68Nzxle2V  
KpCKmX6XCqUV4bbRZi5mlTa7WKqGRYhzc+G0OHcR0Cws2pKNVybjkPZUUNrw1/w3YDIPy0S06+Mt  
dXZ+4C7gFMFu0gS38EvoTex8PzKIEEZ18PNrIahyueNwtgDvB5hGiniPuwkl9S3OGmeop0cGOrUt  
0K5DY6KhpE7vRme7bWZm3nJSWVolkVhfmhRQRNjr8VkwBmNjifqrqftaCMWKKDpdIA3v+OO+6Uf  
i/17Jw+vWmjloR+UtvUkEPmYsoZaTmGwjdxr6ysMCSdnZ7RRSL6utgrG2Jh6wjGmE004Vwk8zS/g  
hum5fs+rq+EuFnQyLISctQaKRXraYJ935xu7KPY59a8dnEUhizHceHrropZWPg4LFSnFENGy1kp0



q6YRkTuN9MEv2EO0UclC+MvrNUxDXlKmbSx5p6GGPmm9aLyczAZa+wvPFOMm4Y2SgRp17g2xGQq5  
W1prE+ft8b6A29906ILaS/Pcwn5YDlGz7yBhG4VBY+igNaGkx6SGSml04CSUuTaQ/9W2pCo1IRt2  
EyXTOF/CGdz3oqUw3jDrU2jIRqjeqzSWCQepiUWck7+6huzLWw/dOhIgPfDgvdIKWxx/Lww5+Up  
59+mh7JaqjNWp0SKDbFG+ca21jESidKWZNOSfXr9NrMU+ve63I+eIVIDtIBjtwlELwpbOJGiBrCR  
yW513wIrOfpSVIAMo/36XXb7C0mljA6wWiGERI6YsIkrKhUELRLumLxgqDtgmb8HnKsGO51tVTAS  
jiZGa7EYYq9z0J1Xfl+W1YDBmWPADDFsnp6pf5ZnVRXnEtbtJT6ShQb0PIiCZ0N1PuNV55Xpd/Z7G  
5/pP6/nzuf+89ZY2930Pe4CLCK0SSSnoaFHLchHVj0cRcsa9SEV/cJ3Qxh5YeC1eMXrxdqQGHQOo  
oIgCHdvXxPDDW42at4DRuxlLHzdjVzXvGz1stDT6mbokRWP9GmEUNIareRtZx910WXUXTIgA58Jt  
DeowBfaa4vj/9jHHmCEUCLJMGToKDHdt4gFuZasjEQQsbj3W1l1blkMHDpc8a+RhEnrCSRpGGkgd  
amIpnmDyr1GLkA+NnLXyvEWrKn6gWDACUqxdJGx5kiZv3thspiN7FJBRX0Duve+uilFYbRfDR2Ex  
o28klfGJUX0XF2sIdlnOtwstHDPPeMoOofzHVQRc4/GV4cyGzmnQrzgRlCPy+FaHtcoFurywAWu8  
oQtSuing16c5I+cl8b61re8cdQtzstNsiXzHTXGkTX0tfS6WUA+xZA9IL15FNrjkNoQJmTgKbyY  
okn++dRaELH+2g3yMB7WN+i5Zp5OU7cpt4nZe9shb+BQVSzj9JAthk4bjdVb3JEA/t0ECqQZwHE  
YEC2ZEyk8bGR0KopRrGEP9+Sp588knN8zyohMVhvakG2QBOrJzfeqjtGdUPRnDi9C1ZmJtnCIy50  
eblygWvnS+UZpHIS7RgTpC1sSG39bItUgQPfGTR8V5+grzem35lyHSTPajm+bT2raJnkSfW49+sD  
deioaFYqiLh9Y0y8Z08lWPOFUUvfdDOD5MM9xkR+Sa6Iw7BmQ4S5xn4W+UXE5vIPEXURwyRDbvye  
TVxERmOonBfiXsUeaUUUV6Y31s9bFNk5HVzcmHujoyPL56WxOq/60/CquCZEO/3gJBgd29QqDJWQe  
Pu8ohaVQ6ViLwFp4RkM1Tlk3+9ogzT0blp/J8kRqCt6aL3hpTmVKSXaqxpNzrXixnVjeULtmIlw  
yPqw/j6imZTJTj2G4fpcJBMMQkRYyCB7YNj29vtSFyvyT3orClUd4qhrXV9FCVlMOzycTyUtKCGF  
Ke4uXWW4f7s9FzZKZQbY6ePWCBOvTBvy9UrWIOEje5IvUETYYZPSTM/MjJWD42Ixx009jNM1UHW  
4UZqFIOwJYhOimGm82ViVaEYoZhFWH6XNbXULJkNVmPZNfioZa1ktmkLCJGD7nxxHWGjaU61alzm  
VeeNITZojpjSXEjztQ7s4TGNKkfrot4R6HXC+BDdXLBWE3ihVIREzMzM53zzljf/JbbE3cit5lig  
XJzc6+o74Ek8fQuu0B5FIrhdsdcpoq0Uyh224U8GSA2KzBOoKwJlWXUOSeFbowoaP4wdRgvlQA11l  
bhgRfzYS2HaRTZ9JVOE4G6ap5VWOZ8LSRZKIkVetv1NTky7P+5zqjk3CJ8fWa8qjLa125LPfeYzX  
PwWIbB+nGnd1IvEj65g+FtRL9TQqkfPsb29SdUHY8i0+wMaQZaX4Jwee0YVfIJsCa6L11ULAnK4v  
ql6X48BKDFA2xXt+M8NgY3IruLlHlyJch5dH/ai77ah7XWJbgRZmHqeqUHy3hXDUwRswrlnHenG7  
fEMU6uIPhNFrzgs1PjGcs7iyZRoUc0/y1A1H6qVF8UgzNo18j527CLPA5nGvG5uFYnQJytdTQE2s  
MwIE8BUGrV6/+JjF+fnnbH2er2fcVd3kb2bmXFOy/AqowxL37M7ywlqRh20uqf11NiOm3lpT3qGL  
K8gyL5zpbLDM0x0fwPCA0VhobgDwyG2Ua6yKxypIEUg9ds80WC4WRbGBA4JZtsksahsCrCFNDc/4  
469wfKKDdm14b02W5bN5a2WotZWziq0Bc7YPLU9pbxQcCZo5rr3OX78OZlFWGLpxMS180EWQsbcz  
+QxUArnvmPM3Aj12/hzhRWEEnn4f5rUasyrlg8Kqhlpx6xJJ+atLo4vWcGolmuN6ss2daxonjsrcYu  
SA2EhNucyUYAl9+FEih9EJcYzv+OiKmSreMXtdB4F/bZm0loKqCptVls9wAxtpQM3nqlSIElW6f  
rStMA8oc9AtRqQFWJ6X9aFhZolrsPo6PjgN3/r3xTnnbFy0nOWN4m49XtKBfNAKIEL2i0y0HDYX  
5Bzux/iKHsBxG6oo4UxNOJwg6tqCwXlOEJeldTC0GUJFzsoEsqL6IWG4mmk4RDX4KIy8TrOQ7uTN  
boTzSxeWD+DZ028qr7mTsNqE8hh77vvfpIL8HhOBjLvpeSHwdDMGG4eJkfiz2ltbY2h9uTEmC5if  
05QyI8qotbYEDPfQ6qRjlu8IJ+IV4FIyiYK26RM55eq+dRjKkLZpniBbKsNps4q0+zzqDJ+ttOd3Au  
uFhWWW5ru+nwhmbiR6RLXmZNxzdWIzpxBDeTwc0YoQOjswrOaOE8RV51cASq8czKRdqdUNF8mH0  
pnqNMFAcIgbXvU/1k3Eh3WhGaAiSIBgB2uv2WzunXddN294wy0Xux07eFCU3Q+hiwU7makRIJTN1  
R9sOkRD6uaRqvsZ8dxCDLSFqbJAWc+0KXG4fn206it3NrvIlPukmHQ21F0TiBG5D3URAprmji+dm  
AZtaDh/gTxdpddK2LrrJ6alQgCeBo8gDMR8G8vAnHpKNjbNhhku/KPtWjYxv+auVpGpeigZG0+m0Z  
Le1Iwc8S4rhZVoE4I1jGE3flzKdyLtU5zeJMt8lFJXX2o/jKic8VaIKdx1Y+8XokYpYXESNow6tY  
zElW2LfSsdCPR8SDNPxIfe5Df15FDSJFRgqQipknopcXKDBHrXXEXE654jkGk/IsL9RYE2PIegIJ  
4mXbK2pPrBvVRSCGgUtVefwZOXcVh+1WbpmzCTq3CVxuIYm5ar6wzV6VWwLRd23TT39Z7Wp2l+Us  
boLdoc74SmyfAqrWUog1GOREfTh0il3LyqFBONSmJ6TusnwEV8yGUGqVZXDihDxjaAQNJfsvIRJn  
XoAhwLZvmRiIt+6cCvUMFEPFSUVzmgcleFnJEMlhCqBw4wSYI8Bvr12jxxhzlH1Rft77ruXdd1Tn  
iDripzcGkPceq3ux1s0Qh3aPCcMcX15WQ4fPBB2cxAQPGuSebFqO1lpoySXU/Amn1Gz9/N13xQ5f  
Z30PhiUjc4wI8PiLztnJKgm6KJNFIOoJZUOqDjk6ipJg95X9dAwNpuuZx051FbCdcd5p0APhbeI  
HEatVipgiiHN6yklWpkhPVTnTNbCnhr2E9GkTe6qJ6qWogfhe0esM9XiWoX/TJq8LX4KnbJAbbp3  
1MBw9f3LT+1EDhx6xdGcmfjvOq023R/67wy1jfd/kyUto71RVmvtEVGkSy/ywLMYC+ldcSAYEmnY  
Zf8Tm2MzpNKXgBjySSIZZMD9kn3OrtrU9IisKOYInCESZMY3GQRXmUwcFHpqrGBQibsbGp2WVoJh  
yqIoge07EqGoiOpgrE9Kwvm7rPbnV2GqilntJNTY/Lvr3yNxmnbaxMDcC2I/ia6U505U4tsYKmWj  
txxLZ9ZlSnJSZCHTVY2Lj+9nOEK9ec94prQ8AdQlIj8uEV6WIirqdA596JEhmbBxBVKJ7GCTB9aF  
Yc7F7TlkKF7nvnUwpQSLD1Fih4AJFETX4Jn/Z6PFnwph17el1R47TFxzkUcvYE2jsdld1JNtdddkk  
OpGk3iwbk288Yg73yggsyDBpJ4hxU3JHVcjGVUwSnyPrp8+yCokDlPkQSJ3Xk1VD03q5biSghUMC  
Wr71QjLhngzvUZrZq0JY+65VPDr55Wx9tPe2wdFvlhkurvYBVPjaLoF09PdDeERRKt9iEtQJSstg  
ABHLAGYUINPqfOgwFMF3PFhUy0pgpejHGmiuyJC7Toc3AAK7HiYEG6epyjv+qKK0ZIGgIXlgSMTB  
DeggplVxtGYzVK9+e43kE+BggFOCI+8Zq105H7wwYfIGdbP0WNBd1ApfdLgaEvoGVE90DOeEIm0d

```
1sEeg4ePOw3oYFolaEfhhmzrOLV5hmqIdfG5av5Tpe4pAnaCF0lDGiEI77UlnMzURplhZkppR6RD
dqiTAzvTx7IF+UG4OfnRv6vc/WGpq4fRaVEDgNr686BoBvEBuojkvVvWl2aw2EJNDgN0G3gmXo7e
lE/8gJCdNkZvllKK4aSSoldMgj/R3wHTzokPqAAUaFT04Oomu+z5gUAsUfHulgaVieggMAgWjEbT
SpT7o6ONrvnlbG60POfuzNpApHTRdCh3g/xmaLkXhrjROti/XABrNivzKN6yF0093ULd9D3gEEqR
WC8GPURCQhnCEkRdhZ+fqsVqmrxm8GY766Ih8TPAm3Qz2EZeJZULXMaQvehiedaC0z9pDa8fmxyI
lDYWNZx/23t7MqnP/2pgHoavVAL6FlQxFFBxxbSk1hAz2WV8/IkaMXKTvHkygI1ORam4y9CLXkf
nFGeRa187PxPE0y8vlapJPoQq+ons/GxoYP8WohnLaiv0hlVIhJflbGXSI0xSaUBsQ49rVtExIf8
AgyyULdWHxVgDk8ucMSNLcQtiMq6BdKMFHQ0eROPcWwaLiw0yiYWRilUeQVTW0sHV/zj8MozUFFJ
UKCF2XK4sdhAGwLM29qcWjER0qVlJueSgklEDuuiOUqPR/e59ZDD32i+LPalf8jwAAZs7UsKZQag
wAAAABJRUErkJggg==', height: 150, width: 180, top: 62, left: 50 }];
export let imageData2 = [{ src:
'data:image/png;base64,iVBORw0KGgoAAAANSUHeUgAAAOsAAADSCAYAAACrfzewAAAAGXRFW
HRTb2Z0d2FyZQBmZG9iZSBjbWFnZVZlYWR5c2llPAAADAF0RVh0QUxUVGFuAEJlc2luZXNzIGNvb
mNlCHQgd2l0aCB5b3VuZyBidXNpbmVzcyBnaXJsIOHbh/oAAAA6aVRYdERlcn2NyaXB0aW9uAAAAA
ABCdXNpbmVzcyBjb25jZXB0IHdpdGggeW91bmcmYnVzaW5lc3MgZ2lybCCprqv/AADyDULEQVR42
uy9CbBseV4W+Dv7kvtY97e/qq7qqquqprt6pWwFAQNxYVHECUHEGZUhcEGdCTBicNwGBQ2JAMNw0
HFGHRxHR3AmJrBRERVZG16ofZXb7nv3ps398yznzPf9zt5X5VNd9MNJVR35anIuvfdzDx5zsnz/
X/fb/t+RlVVstle/+07/vjvkcD3xPc88VxHbMsSyzTFMEQfn3LDd8LvpSpLKctCijyXoiJ097Iq6
+eklJca+M3ADg3DFIufYTti6ucYeK3I1/wP362v+2ff+WeNsD0w2sN9Y+fousmfre7AcP3QsB33U
xxKvf/P9LlPsBV45Hzb9u745W3GFqyv//YX/puvFBCa9V8DVse28bDEVGAZnxqwrXNYv/rPfvf2y
/gc2sztJXh9t2/52i+THMDK8kLSNMMj1TTLJce/8wLguwDir8Ia+b1/+Ru2X8jn0GZvL8Hrs33zV
3+BmJat1s3YWM8LK0r79+pmKSU2N3bx02eR220L1u32K97++Fc+IwaoJzFpbJyzX+xfwNyF/zUG7
A//P/+HsVoupMwyUGdTMvx0bBeWH0dvVvKbf/c3bv2jrc/6OQTSr3gvPcYacOonmrCuDCTBwsI/t
WBpbfiqruOoD+t5LvxxPPBvx9lY2E/kw/4yfdav/Qt//xcd4z/+tj9m2I2ukZeladuWuZ5H1nRyZ
mN/FnxbyzBKC6sLXF2zSuOodH0vdxwPTxdg7FXh+mFuu045PLhcXnv86eqxpz+wwWG2YP3s2v7EV
75PLq7d6wxYEzsuP02wmgBreQFWgM8A6nz8tL3OrpGliQ2/OXAdr5FmSS9P0n5epG340YFhOW5VZ
J5RlR5e74C+VzjOHB8S5XmS4fNSYHmFz5o229lzz/Ompu8vzDyPWnsH6dW3va84vP5YGYTN7Q20p
cFvYKB+1Xs1/8L/NH0hleYjCC5y2/KCB+e/iAR/whv7l6DEfKrc/B7g4W/2N9w85+AYRthlgIN5i
+16PTdoNePlWjTzwmoeOZ2ny6u+EzZL29hLkmq31eyYludYeZ26VpE7hulbplFgsakKw7RL27QK2
7UKwnWRZbOKqM8sQz7xAmap14YnhiVee/sF37y+Pyld58NLz202rvxWBq0BhVXju3dsQXrG2L7Y
7/96Y313Dinnz5gf8mbGICl14s3G6Wh068ugOrgcYDHU3g8aTtuC4/uejW75fnBfhKtRjieKY6hC
WPZcBx315AyDH3PhMmFpZTA9hoNvKZVVV4Xdnq3jlaNcgHDCV4sZW6k6bla7otgWJa1BX+zHLNwb
a+A1Y3wWAG0sVUZM8dzT3q7hy/4zc7z8+MXp7Y4u/ly7+ihs53Lb4nwtwL72YJ2S4N/7bb/9sver
v1LzWE+AOWm0lvJZ0KJDVDi6lNTYl0BngQbftJyvHdjXwP8uw0aXGA1sLM8nwAPPgA2w/E0yiJfY
99NFHhi007btMwGcG7hmpDTiPG+LqzoQbReDZk+siyHswRoxTaSaGEkSSxJkugjx1IBUOI4XQldT
yzXEteyCs/zxfE9MasqxTGmllnOzUrO8fm3mr3hh9s7uz/T3b304db+zbtH1x9bma5XyLYAYgvWX
+3tj37pk6+mY14fwH4iHxY4s55wXPd3+u3eM1WeH5VFFhpVkrVwH7HPUVhkZ/BV3aIoI4A1gC33j
LL0+beyrHx8qg0Azg3TcG3LflsSpXJna2EpK+mEEjSajCXJYnwqebYCEBv6nB634ch0eiKrZSR5G
kuaArjwlXmsXhiIa5vSCJo4uUxZuYUDA4XdwvL8pRf4U9OoXvYbzZ/t9Pd+bOfKwz+z/+i77vb3L
q9xjcrtHbQF66/K9ke+9KlX4fdfBrBdAPaZVrvzRe1W53HLMgamaTTWi+lb4tVcBnuXfrRI4z72d
6cqMpjEvAEQtRipTZN4ty6wyPssvjAAxHW0hHW2xbBMkfJUHNFHZzuyWpzJer4Q12tI006J6zfEx
LGQsCZ4j+W6m4INGhmXkPl8rsdY5LGea7PVAWA9rAaImFhkDFjoClw/Xk4ks9IiaIRRs9Mdtdd5
xuNl093d/Z++Nj7/rQ4VufHjllumG2p8Ras/8W2b/wtnyflxur8Z/B7fQBrALA+wPAkAPvrhzs7T
wWB/5Yiid7mNlSt7KAJq9oM/eAEFmuJ3cGqZlaWRLt5lrtFWZhpLu2WWS5JFuMTSGvKVRkSRwt8n
itxvIRlXEoB6+o2OhLiYduOpFkk0XKmVtNxPAmb3bpMEQsIGLAC17ADUGELIiwkjROcRyEZfqZFr
myghfcEniM2WEEA0JdYFKbnZzKfnRZ+0F1euXb9uNVq/bzjOP/m4KFH/93D7/3NLzYH+2tTtoDdg
vV13v7kl79LCnh1haZOx1fAGgCsBcC+BYD9QKPZeQTFbb0VH/JEnmV7ZVnargeglLk0m6H02r3nR
YosL0tgcHKzyqt2imNaLKcKNm4uLGcCYfbqg7r4KFOydcXYEXgV5aLCajtFF802CisbSPswq60
DAXPirAmKwUtPxcLBTi2D6Adl+iZC2018TC4sp6DQCDJqdxLHcrhgAa2Ab0sF70h1YaljmHKA9v
38Pi0SctvuD2aVLV55rhM1/6/ruv3zsA7/1Zw8eeXqGs9/S4i1Yf+Xbt3zn5wv8wRqoFz9fH8AaG
8A+DMA+47jezX63exX0dw/s8EpVmc3Van7IKiITr97ZPZThcPCybVunwOq7l4spALNWMFQl6C2AR
7BmWaqfzd81Co2PybMEVrQlZKOYVX2cJoPKWaiBlyfrmr5Tgf/gjTLyPK/ViLZpa2CJ52J7nsxgL
fOi0oosPk/rzY+ZTkcyX8zEDzrSDFoCviCddijddhN+cUum4zOZnB8X8J+jw6OrrwyH/R8Jgsa/O
HzosR+7+Z7fNGp0d4vt3bYF63+x7Rt+89t/OYC9uL4Bfv1toJpPgB7uNnz/IcBjP0ris1EUtSpaw
PlYOrld8axKcIPP+r1OJytSmY9HoLQJqGahJRDT+RIWMQNQEz2SLE/EBYV1/RA+Z67AZfGE1KkGj
8J2H5QVIMR7aYFb7aHSXu4viWcwth4AGWPfLvYZaeMBn4efKT72yX/HaSRZYer5a0oKq0CWVXJyc
oLncqXX9G97va60m77s7++LBQs/A2hn5/ej3nDn5Nr1h3+s2Wx+f6e/80MPvfeLT/duPpVv76otW
```



H+1AWtsAft9HGCNDWafwnt+G17aanc6R6HnPYp/90F5W/PF4nJh2BKtFqCtvgRhS8Lak0GvLU3fA  
4AXa1Fp8WgJ79w9kcl8Ju1Wb9MOZ+AnQFRkAIupAMwBcIKVH+8BRPB9cRgAIADWHV4FumOcQ6HPw  
adUS9rqDSVaz6XKM1nFK0mjVNM4Ftv8YDk9gHa5XMPssYZ1JZuu2/BSfN7p+Vwi+MXM0/LawB3Hi  
YZy48plabUa2O9S5ufHaavdHR0eHv5Et7fzT8NW+K9uPv01J5cff+/Wwn4G27Yo4jPYTFgnTVAqY  
DdlEJvVrixrV8zcvBAwBZ+U9wO8v8F3fRiVxk3LMHp4j1kUuR9FqyEsagJoW5Vh2Y1mh1FVmwdrW  
KieLNaPjGdzseFPVqDkL7z4vEyma/HDhqxhaWm9CURGU32ysJ6ZqDJhLUNRAUNX4NatMaO01Rgz  
kbHWJlZ9UUTI5EGKKuJfa2WEwmbTezHhf85kMSPQJcjIf2O4T9n8FOb3YF4oMbn5+eSV6k4Bii0W  
UoDvmscrTS45fiBWvaz0UTms5XsDVTysHeIReKyOx3dHb4QJe/a3V3J0cFh8fJP/9C/xvJydvT4+  
7eA/TQ361u/9Vu3V+HT3H7ge//OxxOTj/9lU9ckLfz/t4Aavr8RNgadRvgILFELz/lVXvRLy15bd  
rDO8qKRJ41PH1TyutvFoWpDZcr52bFSUBP/Pj071QR0M83remH6juv1SvtkU1i0FazXZDLWnlkFd  
1Xq4sH64bLINDCUwGKa+NloAYzribKDEgBbRQv4wKDzFREiDPuHlcYZhM1AugBoDj83y5aSgwoHA  
OtwZ1/WyzmOZw3wNsX1TLABX0Dp9RgYkOLfC6xhq9VKMlh5tbbdHavIM286nfsqKm/6rjtN56MTX  
LG4s3dlz3NtafDrv/2x3/bOTxZ0slgAgMcRgPlb8Nhttdu7zdc/ahi2j5eDzxp7WZYP1+vFYZGXL  
sBoGY6v/meOF7QaTQlct0G6nJ1Ld/dQ0iSXk/uvaCoFUNbIrB80AL61Bn94DHUPraW+ZwEQ2aCvg  
efLoNNwi2rC2jLwJEYurWZPWNfIHG9ZbkANWmWAlMy7AkT6nka3LS7pMmgwfebZYgYKXahvGsC3H  
o/HAnZQr1LwezMsmHeOT+XkbCQVfOjAo8/rwvrm0u/0pRV6ekx5uk5NsU6Gg/YHr924/g+7vcGPX  
3nymdXR2z5/eyNuafDrTEXMC3GN8uMpMe/6ffzldwEAYaPRGPqusxOnxcjz7MtFmb91tY40AcRGk  
eU1bQZFJbja3R21YHiP+oJMj4TtjqZDlvNzBaPrteR8fKrUlBstVcT8qdFbjQIzCJRkP7cnaTyXG  
L7uOQ6mA7/R9+h7NtXKLgC6MAxVbobpmgyfXxSM/gY4lrpqKYVfm8CSko4b+N2xKtnr70iUxqDgk  
UQ4phZ8ax7vdHJeR6Gx751hTwIsJHfuHuN4SpmszuoySsMVVkrS2lpV5Zb5eu98Ur7ffP7ZcXn9+  
tj62I9/BGBNt3fXlga/rtsP/pPvkU/MhY09/O+r+Zvv+7uOY/sAcel57qNJFL01jvM+ANC4yLk24  
RsqpduUC7y3klFd/B2+Y7RaamSWOc8YFrSCVbQ9fxN5tvGw1F9uNLsAV6T1xB7AZsJHnc9GAGILv  
m2LKVeluFwImKKhNeXBLgC2uijC0bQLBdOyeIH9F+L6vu6/yBP8LdWOBM9tqL/K57hYpfESfjFUK  
LDr2njUfrUJ/zYIfZx/ILMpjqPRldlkhSvnhYUhlAWghfPDQmXFaYK1zA3i9WplFMnteHxvuf/IO  
7fW9VPFTLaX4DO3rIzA/s3v/0n9nWADIpoAwW+nV+F63iGoboK/Aa/WU/PF7PhlejUAZiwCjcXwr  
BqajO7JClSXgPEVILZaVUZaYeJAhWU1VgMgMwAmGwgjxRUtCenhLWzBRZKmrBuUjF9IvAxuwCkp  
5RXLsYseAZfMgZYF6xagvH3YJkH+9ckwd/Ox2daWsJzIMBYmRSC9nY7O9Lt7eixcbE4OXlJiVi8O  
X53rVz2Di5LujwRE5/fagSaY93f35VG6Oqx9LsNuXzpCDdXLgdHV4XiE6enI8ngc5+N6+opnE8wG  
k8ejtLqN5yffT/v3gs/3/zwB/+3rcjNFqyvI1gtU2/uP/1V76mBC5QBsF9cJ3AkSJN00et2b7q29  
b7VYvYW+KuF5YXRajXz+YrAb6lYGi3rHm7kdrsrjmtJFi2k2epJksxhcWMt5dPCDACPvhFGGv/Ot  
DAizzNN0di01NwnLFuRrtQX7Q96sFzsJINFLLbM6pQKAM0DFet/5ZILXptLu7en5nJ+fAYzHmi+lz  
8p64tX8RIHugdi220NYy45Mwx44bj8JkA7kaPLbwGdhr+9nkq5nomVL+XK9Ruy0+tisYkBYk86s  
NpSrGR3fwcgTmS5juHrTmWygF/tNyy46c3T0emTk8XyC0ej8Y3pySlne4dtwfq6W9bNTwM/H8GjB  
atY4uHt7e48ZVTp+wGOHdNtL9brrLc4u9egj0jL1+52xGbUF8j2HBsUM5A5LJwJi7pcTTWgVBq2W  
tiCZJdRwODTBCD5ZeVFBCDC/4OPKNpHa4np1FaZhfx0c334kzn8T9Jo+qrsoqGFzmDRmM5Zwm9lP  
W+I58JWV/OkO9EIf69zuibelyWrOgDFdBDMtt8aqC8bw0LbOGZ+TqsZysH+ZSwapqaL0sVYBv2uX  
IVVbcJPHmLhaDZ8jTC3sThpmkmPo5LpdIr9hG6eF8PFYv2e2Wz2ntnopPcz3/dd23tyG2B6vSyr8  
dolzpNasSEBcm7u7u68t8jza6kVpiwFXC5OqXMKw+Gu9Pq7WlTrghKGfkMKAIwVQvPFuaR5KXGyl  
nJcaMEDG1TyjZRLFK01aBMBCX6uDYAymzqWtvYGHwimJgj1aAX9kVaDV8Q1rSmzozoMuhU4vcIr  
3MdS5/nf1lbrJHIypOT0Zkso6XsDAziOQFAxYWhU183B6BD+KCMHGdxLGtjBgD2lJiFXLohy+mkj  
mLjfIBCgLYt3jLHu5kKskHFM5xjifOJcMxN/FvE1eon+K1Jcm26NJ4Zn00+4gfHrIeMt3fa1rL+i  
rf/8X//YQXsX/re/0DL2gdYdnaGw19/sL/zO6qyupFk+e7o+N7NOI4t3wvl6pXrWuubrOawVmtxg  
FZSRM9hOYRoEIfTBkZzJRgvShj5WlqhNikfiKeZWr/IdEpdNE8raRs1Fa6KBI+CuThx4Ue3O1lhV  
fBySDXEr67YypN/ZCGF0Wlhfw2uTuOkYBkiWEUL3I+mW3SOqKWvTJ9pb8RfOgsjeDL0vImspidl  
nHx6Fw6nZb0NRqM9Quv8WEGWqELV4+O5GC3I038odPpaFXTejmV2XQs08WcdNlyw1Yb1v1x00On7  
9+51f+Jf/Rt2/tyC9Zf2fZXv/4LHwD2z/++D/gA68MH+7u/CZbqPVlW7M/ns4dnk3GPlrEdenLly  
hVtJcVjPVo4lhKyRpe+JYM9k/GJRm/XALJQYQVOXJwstKuFlJolegboZRLF2qNKn7Iq6+J92xQtF  
uDrWVThAixr0Gh+TgngMmdL2kkgx/EKzwdi+a4CEJYMTBrWF1Yuy0ulyGyA195VQHwF33KGz9a2O  
3wa47T9VwaraFY2P96ea5VU2zR47GbTlM7hZgi2t0Fixj04Yu3pRU44uJ4+5227OFvgWuoK8A8M  
alwkqRa67xYzd20K17OxpN3p2n58Pj0xN3ebVuv/rK37/yGL5U/8z3/Wn//K1/3Ba7nem/b39v/c  
gDsnWleHm1m0G0XuCmt+TatWtyCHAAdQevJ+plX5/qBlZ3uCL6X34f+cShKCLoMVZtFY/tMzqj  
p9sY1ELbYWrGwW0YZ2+Kah0o9XZFEpGnBTCoMeknHVvKkx2nkjQ9LWcSi221gN7ulAkOE7Saz/sa  
IqFBRWay7Vcpb4k+kmcaWUSglUMUFXwjdkr69qhOH5HUzukygyWmDSa43cbdBrMwa4A2H2xyhjn5  
klDiyEsCUNbdvs9acHC7sOX7cPKRlhcuEAtVpFlW15ntpw9MltM37GK570f+0f/8zYyvAXrZ779n  
T/51fJN3/3/6e9//Q9/id/p9J4cDAZfWVbF+7I0vz6dnPdYJrizeyAPPfyolt95jqtAaneH8FFDW  
KYzaTRbkqdLreltNdrS7+3BR22o37ee161rjPISCARfLT9a1pVJmxJCg5VHYZT6LRR8ceW+14qC  
MOgA0DNd8FLJrv2uJYbIWrfPyu7Wi+NMBnPF7DioLTLZW2RHb9+n1HnkoGodQS/t1LLyhTOOkqUR

jM6fdFtxMoq3kbRfAT/t133L5SRHF66CV/XUMreCh3ptppYLEzpt1uytwNLO2zLpaNrGmzLsHhF0  
QpExT+YLJZvn57Prs7Pj7eR4Y+Pl2yLIj719g/+30+Rr/u2/1N//1vf9Fv9Xn/nba5lFAVcvvcla  
f7IOlr3tX8UgDo8uqW0MsbN32x1hWWI09070gYnVHTpGqyuV4tyVxmAY0kXlJYHYFgEUYL2aqURg  
M1qJDq0EcACK6X/dgGqIltpdw6rlkqdp5OKerKM8rJIooXjoLK+G2iLHMHu+dRNqi2rBT+Zl1kpL  
BYE+q5lmuvrqzLR6LRNJQtQbpZAstiB72U/rBe24K8utbMn057ZDsBaSgg/VyuxcB4sklA1DCuQP  
BpJu7+nPrQWbWDhiJbzTQlJKb3uAD8zbaUrWfWRUxcMytZryosdq84jvXC8vazydGTn7+9CbfR4  
E9v+71/8R/qz//1v/9dTnewew333XvuiHvzfHz/3XEW+a3mjtbiHu4fKHjGs3Peu6yPgK+51nrc7  
rAvnTDUDIrZYAvNaWdBQC632vLE489oftgbnOxmMtqvpD7o/satWWEtgTwqDhIXzRwWpsgUT01j  
lvlsdIplqAS87FpwrRKqCJnGfOnKnJagsiyaw9gNgFOy1WLyu6ZKF4Ak7B+YAE1/FzaV2NFeh2JH  
e7AAi/1PBqNASjvTAsquN70u0MxTbWdf+r31GaHJod+LkAftiVdDXSUKq228GnV2Deuv1KzQCS1  
RZ0kJFMF7eVNURJ4vquu49zfrwRuv8hnp+TJmy7crZg/fs3f/zn/4DV6Q13AYtn1ovlV5Uij4S+7  
/d2Ds0cN3prMICFSmSEGzgM++pDugDQ50xlnDFaRCvxhwditWrqulqOJVvPtBZXaWMMsGogsmPFx  
79bu33ZGbS0k2W2vCm3jo9lMVtK6nZruuv76lOm9FeLVP1S5l9Ju72gKYvxfQVEFDNA5aivmsJR  
pwpKcp+V1YnZWnGXJQUMSPBPZme34NfvS9pOteyx/kik8VkJL0dnHq+VDre70xKvBiBqhvah0u/u  
glqa+Fn0OiBzp6JX3XBENoK4Hw9lpY8duIwp8zzvX9yJgVWNB9+93w2keFwKNPZAoxkYQWDYXulv  
P/oeNK+vthb3MLlJ7Z34JYGf1rb93/7NxpNVjKIVd9Pkq+Ab/XuqsgHYautKvad4b4kq2md5wQwK  
lDWVqevhQxsTXOD+ib1bEN9QUZVKA+SUV8oXamiYKfpm0rrfdUPBHAZifWxz8AlZdh1rW9d6GBiP  
74Ha5izdthTqWwzqzQ5g6UOQHsplMb9NkBsXOYplJb56j4EAT4WxmryiKlWlJPywDXfHYKp3pAt  
WHxLBvEtXQX709BsAm1ALV7BFDP626aZhes4RznFWoHkkmKzRm0LFsMezjPWGm3H8A3NipNHZGm8  
zpYDpsScC2w4qVgAzYL/UG94zjVCHhVFWaz0SmTJHplT9d5dvL8T64vv+MLtzf1rL+0luz1XYBp  
rcUafzFuIHe6YeNNU52uzs4BO1b10Xt7F6B/2ema9nZu6wpluPbL8r4/EzpbduAR91+sYB2Gz1B  
W3QSQHQz1XxwThR4obCpvpqiJXWBSWDjxrCoh/2G7LTeljuHZ/I6flYPHa6aP9rHYcpcrYNlBqwa  
Tb68HenYpkt7bAhoFnc0AAop+M7sGRHwpCyRRUJALoAPQ1YoIFFx7AH0sTikIH6epahVrnAcZ2fn  
cnuYCgr7DcoAmk0Byop4zuNmsIusGCVGawlzrVYA/UVa5s9HAsDarTmebGSNhY3A9fH8Vry0ksvY  
VHAouRZsM4N7ZHFAgdWbQ+wMD0aRfFeGi3ZPLSVgNlGgz/19u/+7rdaZVHtwgK8z3SDx1vdgeu6t  
j3YBSBxI1NBQYXGDEtmJ3fvutDfm5/e0RyppmJwI7MA4PjkWEYAL4sRytLWwcr0M9lZxpmu5OVGL  
KyytftQRfWxAKScur4AOqNBTVi665f25cbVa7Lb7wL4AuU250BHMq7ACxZGitAXbtum2OFksWcb  
F5bfmQ/0MfMizrdo2AvGDAAAIM5858RKLYbdNU6skifvjVVFVcpfXAX+PalzNED4PKyTi05bhNWS  
iGz+7cAxB6etbTQosyWak3r6LEly9Etabr3sAC5cnh4pMfDa1ZqvfiuP8vKyqLhuY1rUZJdTvNkG  
xXe0aBPvf3Q3/kW3KtZiBX+KdsPnwGtfr/u6v2wPbBoUSHysF7OcJ+CZsK39BsN6Q53tJ2NdbVs+  
CYoaHVxx6qvWiYLaXd3AWpPlrOxRmtJadmpUjFC7NbC2xX8UEaDKx15YyowDAIsY+CmI2HT1b3dX  
em3o2T41s4f2aRSCEwtPwTQM3xeo9nXxvJSVffhV3oNrURqBD3sPK07ePC3OJqpB1ROQoonZ1q2y  
KCWadf+bVFpg5+kRarpKYuCTGZVS6Di+NlNxBVks718x2/qefDXj02BTBn6+Jvalj3xuBIF49oN  
lKzOZuvsNiwmmoidfi+eR4nzSB8oddp/8L81k/HB098YEuDt7D8JJTDYGKiGgBwbzOM6hHcuIUTN  
M3leCRT3GBZtJLOYE+6/V1JFmPNpzLAUwKonpWAZulrGSH7PFfrmQ4r5qMymM55UTtg2I7G4cX0W  
VlUQEvGG9+m5bJq0qPWizHcylI2mOeg3tEU/uRA+j34ss23ygvPPytrZsq9u7fUr4WZU+uaJ2O8B  
Z9JuqxSpGvQzT2JpvckCxsZs1wtTI+2pzNCWALFRYF9p1SBkz9TNBZ5kYtLB7z1Qj78GTYGWChw  
vuA0XZ7X7J4IjEWjNOT12Q42NfPzrKF7s8N+oKLBdBb6hsTsB4WkPNbH5b+1cfk+luFkOc/9hHZG  
Xbl5VduA8BDysZYrWajm+f5tbxIe2VeTGQ7N2cL1k+0/TCtqpiciW3pouP5lEt2qTI+mZ7dNlsgl6  
7UqNrBX04Jl6zaPJKesJlMpoSeu01dKqGDF7R6uAq1aokaSaYe48QtYM//B5zlhW60hb24tF6Q/C  
YpKkNJKMc3CIgfWA1M1v9m7XEuJAmgCAN68d13u3H1ZjIMDGVFkO1trjS4pMHOjKm7hhFhIGNFNp  
NHb04WB0V2WcfJlPMPT6ak0OvuymNwR12uqomG/11PR8H6nKys2xQPI7IFld411+Spnyha69WKEh  
QiAPbsne/tXVeupAAVeT08kAO11YVmpnkjr7wahNLHIze49K639h+TaI4+J9dyLSrHv3mEAWFemR  
hyvr4HR7OE9rygN2Ij1u338lhezCYrXBkhv4g4cFFl0LY3XNkdIMLHfbiXyYFPv12BRjJuxgE9ZA  
LTMCKqWkRtq2Z5ZeUppHdDAFqWJO2lYoVR5rrTaXZVlWcEq86bXqqGCyg6WxOlClQUrtXYugFOX+  
jGIZKYGwPKydhCOAFxDQUXN30uXLkk/LsU1czk95eAp9pQOJc7n4rsdjRBbsGyS134sI858HxuG4  
niqqZYGfO3l9LZ0B5d1MT0WXAVZwyp2QjYBrGQw7MkENo6tfBaOqVUm0m33tOKq1d2T1fwc51DJ+  
dmJjtZy2bkMgK4kb+22cF28/r72yxp4Pd0BLkyrk+eksXdTDq4dyTF8/cMr11R+dxV+PzC91qHje  
JfKLP/ZLV3Pusv2n7wu75ZG1dwt12pqvKpMo2/IE3iR1bLuZkniciqLXHroCWGNXASg0tIw1ykl/  
Tdb5tO5TEGVx7OVjEcjiaJakoXdk8yrmlQfZNsZKGae5QB6IrPpGXzYJQAYlfunr8hysdQbtqxY1  
G+oAv5kPJP55L7+fXI+kcnZbQAw097akgsC63RxUL7Ljh5P2/FqZYlMc5xskQv8jhgsQ/RDTa8If  
Wb4mNrNA3DRz8w5fiMBRW/Wdcy06A4WDPqoPiNawmkAlQbQmo2OLgra/cNAEUCthSAslSS1Zz1yk  
7XPLaXi68mxakmxg4eMgRSZc3hWk3vinQbS6e3IrWc/pMdH5RrPduJuq/Fss9f96PTlD0U7j7xna  
lm326sbVn4OKA0Ny36oLLL3FEVJaQfTdnwp2ZgNkKJ2fncJSxGpR6VMycprBxystR8awKobtaPQ2i  
emXdgHgcZ3W8HHDh03VLYKNpTi312hLGxaOBQZUC2w6PXwrjsqEcq6N9prCH9Q4DUDP3lKn2dS8K  
kGasFQRVtL2Pvi2iQStPbX8fN2Kwmt+VxeHjA5mmWnTOKVBQwB3MbsjVnioWk2MRPuUEGVpKmi0N  
p1tLQ2WksfFjiFOklPNKFD72WwhZ6NjUN5L2M9YQc0hVQp8ezk/U/98PJ2o0uHuwRHO3wCYezIf3

ZIAVDvsHUGyP9V0U2f3uizP7wD8Q3n8qc+T2y/fxoKQWK1uv4HLcGTYfgsL5/mb3W/dgvXjwVqUt  
mtWl/OseBvujSGoq4e/mVQ2oBq+dr9YBJ0hzbXBhp6+lcSHz+VrzmXEC+pqzO0WkBb9NG8YNW8JuT  
8v3mOapGJ3FDR/Nx1pi6APQHlUkmvDrgJwTWNfZcgrAhWqJCQKTAmiGoX4tAcNCfsvwxQKwFWwAS  
+/K4+rXchJcGYbSWIE+M5daMbhj1ur++MrzfIlzoAhaqEUCVDlU/zhj5RW01epqlZLf6MOqcvqdJ  
+MJFiGj1AasXq/LAg9DRjjn8ehcBoOBVGkpqZnK4vxYusMjTe/ce+Vn655YHMP0/uWa0mOfMUCar  
mYS7lwVA59Z4Zj91hA8z9P6arKQE7uyinTRyAv7sErWg8JxXlGzvgXrduP2r7/rj5u+bbSL0njIk  
PJGlucD+JVXqDDPcYeUXGgWb7e9tAZXZbaYSpIVWhbImzJPM62tDRxQQjvY1KzXhQtWXij15bCoP  
GYDd6k+IgsraPGMwqpf0LCjHkAew2dlCxoDWQwqkUIzL2NWdbqECmmUUzGqlC02GpAq4rHKUPCzS  
YHtg11VaeBQZZYejkHPs7LSQnv2oM4nnlYnCTFhrpQ1wD78bQK/hJXlv+nnUnyNqhKk9+ZwV61zr  
0uAD7VomOcfDnqS4/Wu3ZE5KHpn94pcfvgdcn73Rbl37442IFD1n40FJiwodY9HL/8cruNlcUCFs  
TpItlyICd/56vWHJcDxj6fHbrvT3zMca2BuawK2YH3tBlA5pWEegpo9nqXpNfiqe2dnZybFtOmns  
UghBe2NcFONP/LzSk9JCXWSG242Am13CBoautIE1XUpuULLyKQ/Bc4MqhI6msZwGG/mLbm7vQkuZ  
W05Gahqcipbp6PC3/T7SP6yMlarmAGYni2t5ACqpVFWy9iUKapYN8scAXqKm7mk4yvpFoy7OzJ/  
nAg89VclqsIQL2n0qPhZuQFG9k52ZyiaHRNK1D2YrmqVSvyRjzcUd+X4u0db1/CkJ9ryO27d+tp6  
bCQezt7ei5WEMh6fiINGcru4cNy/97L8uILH5a9gysS4pg6g0Pdrw3/dXF+W2ywiEZ/H4bVKGQ50  
o6g3u7ex6Rc3mi2uy0snDt4uG/2INNwKx+z/fjf+xazMox+GsdfWJbV71rMJ+85vnf7YM0Epu3ih  
kxkBCrYa09qlwut/fPKJSjgZfXXeAP3GUxxARptd/NUCJsbp7yxTYxBKdJYx7JUgqVky5vT0NkxF  
9+DUVflSwJLPlutAdhCW9gMUMTveqyUkoUMnueKVj7SjgKsthPUKhJqI0vtM2XNAief+lg4WF1V9  
8hmDyqlxqD1L734bD2Eym+pHx2BnjbCoYy1WL+2tgH8bAaSfOyryzJpnm+nJ45qQ03kflKUNRvTY  
fnb7b60mj78zy7oeaYW2XQ7Mp2d4W0R6PB1fHZcp6ncujmf/bhZPJPM7g0VdtPCYQ1mVSD5vJo7Q  
etv2477vUWenWRJUrz79/9Pb8qbdhsN3mznL33IqYryGqzYF65n58/E6+V+nGYejJv2m06mZ+qHc  
XIardgKFNh2murjKXI2fReWyRejTMT36Wey79TSQBbnx5A6WqWmJn9HDCqfddkrflOBnxsWGw2i  
9mb6XAui/h9T8WxTYDAeVtwW918TMA+Brwb0VlmaR+z6Yh3KEfWo/MUSUJVZvgwgBrxfpcWnqCj  
H4u03R8jsNo1ALhLMRPASj01/HYR4v3mVqoUfe5spW0g5ctG0yBfbNS4L0dHYsBe6+VTmmeaT8sF  
xDmkkn9S4AzdeJN50zPXhE76KjbYLCBgSr97L/F+WbRDMeaJWUdhlaZrOmZSc41JVVFLM0T+dFG  
pd3fvoHy8vv+OJqalnfhNuH/tnfMHBD9qL5+DcsZ2e/L43i9y6j9e500jZXUaFaQTH8TANWMEqSj  
bVraMCFDeMBrEAXlpWWI2Rxxg8tZpX06sXVOEzclQUupFWoE806tSoCiNLXqiQLWulj9QjRgRfMwX  
ApOiqTADVMFBJCocj/2YWD/1UZfiQEpf4tkg3ry7IivoZR5BT7zZIJFo+udgVRT4nuqQ8r7FLT1  
6x0wYkwc5nNZjLHApTCB9eRGCx4gN9MX5fRaVYwFZtxkt58bapQtKiI4QXwOUGP53OZzhd15Brnw  
Zpin6DH9Wm00+KfBvU3x8DpyFq1K1GDR8axQktWZVjY5Ya2eyEg6SxVtN7xjOt5HcRWWG0+H42j1o  
0WevoTdn+BMxp/s+/x1f/Rvfk7e1FuflWTLdUxocb4Tr2VePx6v19ThJw+XsPgxoKGk8kTrh3etAh  
bMr3LhLUL/F7J4Wy+8OHgcIculh9VnID3+Nf3dwk+qQC1Y6sE3MLLS9jRS4pP9JORS7qoM7paHFB  
OZG4V9LDGFB4SjzKSwMWqOba2SYmhDUAWbjeFmPJfDKnRvzSVXhIVMF/9Iotb7YJBgyoY6Wwnroc  
lZONVcli0a3p4BvUALxY/3Hk5EWbHBgc+HgbwYWGzawswOn2VHqz04jx2ppI8LFYKsu3sP3c5YOq  
XMc2VphzACuy5pmCqolezrrlWWH1G3KQcUpLdMcXgcVnsh89IpUeaUpMHby+U0TV7EC7zYWhlm9M  
4kWfX041Co9/2Tf57/5zj/84Pcv+Ka/vaXBnyvbT/yDP29kq0UjW0+fSJPoC9I4emy5mA6ADKOiz  
CcsjKtN1XP4ZbAK8K/oDLqWToPhnlTxQikr1fw4gY39oc0G86C1yqDSQXC6OtVCsDK0m4tVidYDs  
3PUMOUhyApW7f8Uje7Sz71QKbaV5laaKiJAL97Hv+mrKAau6vulpohYfEBw62JQ1bUoOJUK1Q6nG  
rVmeIzW1gIkWp09tYBS1or/jm2qnAsVEhn8MksylbpumWqJTKKEaEUSOrarAC/u4UFIKSGE64Br  
bCpDQV1940uYE49WUAL/kGRHRZLULA8r48t7B4qA0gYGeZEd8czwRh08P4IO61wXOsiy45xjjM8R  
pUy1E/9ePE/fp/ceP9v/5y4V9/04fD1cmXG60Uny4rrWZocmZYH4+nDVQplAbpGxcL17FSpMGyEr  
GFV0UpChcdg8dq9gaKSwyytJB6TKwJpqVl8T1FypSulpkCVWtjqaPke3UEFzexwR5XtsUxCAUwk  
GJfKbDsB4k3eT2NTTTaSXUKBowY0PF1Do7og9PemM5R6s1qKVhZTg5gSWKzvs/UMabPSiDTAkfLs  
cQsaFivtbGclvHSjcfk6rVr0mm2VC1Cj4MdRPRLAcxms62F+jxHVlRRPma92ugMYwHwYWGPlt2Qg  
6Mr6iIwCFQlucAT0OsJPq/FHkxpMWLN862tc6VdQg78607OVfWNp+f3gnR+/jSeCvHG0HbcPbzOx  
GOqWsqf5uOD3/51Wxr82b79y7/xjQZ8INDxnYMqi54qsnw/jtZ9FtKzbI5Bm5R5xGgBK9jUOTSd4  
QEOJLWl1lpf6wRdVQ9ktLTT6dd6Ryz1gz/GtA5vSP7NAehMo1TLSTN15KV2uJai09c0OFC5tFVrS  
QXONBDlqy+onVJ2WV4NqJqabNGrSbXSdYkEUDF4jtdTMuZfqiQdICu1VNVj0QB01WlIeB4BAKspAE  
v3vaL3S0kNNGPhaVKRS7WU03bXAFh6rklahCrdCmDwQF83DONLDN63Mb1IH2dT0+kAYvYAN3t9  
Q5gZSeSdTr1c+Pb0u4f4BquL6bDqwoFc9005Frjyda6czTwf9sp64QAXXlKrqnogDrersogxtPVfY  
2jeVEUSb2qwJvHKsGyHsgiflybpYl8vCymC+YFYwU/zfNMjapa/mXcWLgzTKZsAr1762Nqha4fP  
aVApaKC2wy1/Y0N345Tz0wlUNhwbTtGTTcBFKZXDEaNY+lzqnZo1nU05MZGVft5deCvzuFyU8ule  
RlDRcB1vAYpLUSNqddzSVSQjJpK7PSpyrq9jn5jQR/XtjYpHVuS9Uzn4BAkDB5VGWi5TaWJQheEf  
D7CMbnS37uswbAcYM5yAwsX6DUobKliZ/Br11MuU9oIYDFS7Xdq/5gCb2UNWo8qi0z19B7TJoYcn  
12XY5pgNWptkWW02rabqjDBgVyWSTnUqEgW5y07bFauF7wE+jsGzafso7PxW18dlPtpbD/47V8vX  
/SnvmcL1s/WrchSJwyCmlm0fGe8Hj/qBf00TqJOVDTBxHKj35vCV6XgdQJrRTE5jS7rZYm+GHKN  
JdYaqQ003CrjZveIlpoW2n1zPqeolH14Kc6n6qDhdXSsrWMqWJvUgWppkuMTYTY0hub3TG09DrOQ



v3amlJzFCSBa7puPXiqdOt92AWYZq2ISKte5oZKrbDrJezs6LlZLiO+ZzI7H0tnANpqDETGpwDkE  
ACe46PX0tk/kssA28ntO1Iu4OdmCSj1QJbL87qtD/tm2x2vS1kFtcmjFhUVLqKxhO098TkSCKfYC  
FpquTl4i+5FA6CNlyuJctZFT3RB8hs90Gz62+4CF7obr6bMWDzdbPUCnPOP4jvjvbkGm1AUSq8/Eu  
v6r7/hD8hu/+X/Z+qyfbds//wu/1wiw9Fd18WQSZd4FKtoyDMvCDWNR/S6L3kHFVJU+KxVkeV5bP  
KZphv2BmkNPb/jZgxY3s6wpnrmZNmdrDL1Uf7RSuIgOU6a4A2lwpb6qs61AqhRctWSOVVtR+oxmo  
T4cKSzHV7DDRfWcGNDCgkH/19rkZ2ldNW/Lfxu2l1jAymOVrbtdUIbf52St16gd+ox8OpD24rjnZ1  
eguaLElq9lIpUaX05Ek0zNpDo5k9/CSzm9lb2uxnmvdMpvTudhQxJwC3+yNpWSpXgIm9wBiQ3K4E  
KTt9KpdZqf2vbHwMBjHyDjnxbb7R+IGHY2Qa5M+pxOkScPz23Ne1/PTV2R0/6XHsXC+13YD+K3y5  
OZh6Nr2ix/mJ/o3gfgD3/6Htpblsy4MDjPlut7VeDn+dY7d2DccB+5p0mIULnNoguZAzk60tfaVR  
fekauwnZYtcWTDQcigerARVGzxcMvxfWnCp7NdW4shaCxZ9seWtNqS2nUBgAabchUwYzS3JJCBc  
i4GLRyIsKqnelCMoNZ1ow8sam0Zoa10oTAo0MRUKGlq1FjB7pQqP2oYqRb06yKCz+MefLfOr0YsO  
ZydaosbQc+8rQ5xNm1ViKDV98NdoVaZ6YA5mJGEvYH02LbHHC4WHSorUoaGPmYaL7QKi4BlgMviI  
tbc1+vEghLLOdBjyFZTYSTOwutYiklXo0rGGhOgbjFzqzTBDECDH+MEM4ddS77f18XiDKC993h/9  
yBO49WP44V35ZN34ZSvMUZ88KKuPvjX/mCdR9tals+e7Qf++jcYzU6flvQhyzRv2r5PQZbQtOyi0  
RlqITxzmdQa4saulAxWiK7mevMHni0xJU740/yy+dl9yZN6jqllbCwp0x4bn9GUGmwElKo9sAhCE  
yFGBshWbekIcpUBrSlu3WXjPJhfQ5rsmM6rU9I5F9W1tLmdAHPdWgCNLqFGnO3635xhQ4vrcn4rp  
58Hofa9EphqyZJav5izd3T4slCmtCt2gHMB35XT278g2XIqnZ0D6VNDGO9JVpMNe7A0GMUGehchu  
ghsJxStnKok6F8HO+DQq0wF2SR3tP6XqZ0yW2uqhsyCfm5B/SdWXZlGkqaxOxufBCsKhGtPbktPM  
mjzTVy7x3Gcl/AIdBF99WF+308eHl+Dx1fjcrPvr3jtf+Db/sAWrJ81dMJ1TVCwrlGV13EjNfElN  
slWbT+ccpJakqy0h5OWipFa3nQEb0AB73wtvW5b2r1DDcYYla2Wig3TLkDH+lmXtJZwJNB4hQnGB  
7W/1lby1Ms/fdaN1bxQAGRwRjIYznoWq4LV9h+AtpYz9ZSGcoAxUz4KSNBK1iBr7TEWGwaMLKZy7  
LocUuNU8KmtjSRpww8kbLZU9ZAVVoxmh+0DHYHBAozRKz8h8fw+LOSuTpJj5RGVCju7e9qsQEpoQ  
VFbc7ksO2xpsClZnEjKWbO8NjjOJSj3cjmnlRyL4GexhsEod57VQTMVfLM1wKZFFFw2sxSXXZ1yA  
WNUfjE5r0dUws/Ny6Jw/OCteZ55eHyBToF/9VFufpp4hHh8Gfz5g6LI74GcG7qQl19dgaR35Q00  
IRpLPMSTl1lMG3fTFbHAWjvzDctov6yQdnKQikqKW0S1aWALNA/uHyoliCBZQ2GR3Ln9kfxGk/9t  
0olU9qqNEGPlakdSq+oY6XFrkZd6MCHab4a9aUlpW0locMaSeYNbNWv571FC8S/aypHh1Ix7rTZB  
wDvwoQWVlkrI+qoCnw+wJpR+pQwKovFgQoRVWZu1CLgl7d6qmgYA2TMoxI4HtgC5UKZ94wWx9osX  
sIvzzMXrMKEj9nHfnNZp6kCXXO+cAG40GmTAI4rWS1UBcJt7WL/C51CkFA+pjFUoBawzhwineia  
GmjgpSJDtDCQmCAwsdhe3C2nI93GFRbLsbamJBgy7btdHfwHbELp/gkBiJE491MPEoi/Qgu2xjX7  
jnTMEtDKYUUP/BtX6uL5pf8mb+3Besb218lDEYjaQZtZ3LnuT3fb0e2G+RVkTtUfGAZHST3ArcFK  
lxp4JEWzw99abMIXs06lt5oVGfo9A9UYZ8dKYyO1qNuUr28dYVRVdf7lsZGtbCmykpz+ZsKpFkac  
LqIGNOKMn+qVYNGRhaPFoGWuVtWfg8iv/Xg4xr4Fp/0PSW4te9tcsAyrHJGmsnihswSziOvWCHkh  
RrQcpvtuqgDN28Uz6WaJSKtjgZ/fP+SjmbE8qWpG2NZSNjZkyYFzs90tbTQLgNtDmBl1t/tCfNcJ  
fZNuZqzk5dVD7i7c0noFCznM20EKBSwjCLneN0xLCarpkKxqFaRJK5ns2ExvAerveMEYbleUVYNf  
WcnfLnM0w4M7GVcyJ9+zddqbB7g6XKDri+uNkAqxlg9btVDqrmOGKWmzDQIaG0t6xt5e+Hf/mPDd  
T3T9Dv+7OzlXbJQ+HfwZbKQFGkRRbVl02o8gaEq3EybdZLoXxG/2ZHp6QvSv/xWobv3ijZxU42+H  
7JYor+ZdWqoBq7CkvNNye5oNS37AVB1U0jI3KnmR3kDUXJUKSuBaqm3WzkANKg2szeM3jJQxQWAg  
SaxNrNXWQ5obL5K0HCmZzi0mXXFtKz0J+3S0b/nDOWuqVo37sJm/XM6lUazHmrlU0YUQNzywTzU6  
QAdgI0WMVtPJZ6B5sLahr2ujoHMprFa18qqJ+Rli7k04fdz/AcHOGdYxCj2pt02bq1IQSUL5nlZ/  
cRqJvq4vr+jvbS8lizzhHs5sDvepNHonGd52sJ34jIHi/OcZ9HZE47nD9MkvvMaV44jTq7SquLKL  
ihzhZXvF2A9T8FOSgtmFSyphEtgWpZd1lF2S37kb32TvP+PfucWrG/EjaLaie9Zhuu0cEN2nKBp5  
FXmmEBFFkdmvDhXKusZjloiu3LVH/T9Pe3hJFCHw0tCpUNaYFoJh+Mn2GB0sFEQjVVDhqmNagQWg  
01i1D5pbQLqUsI6iFRplJdAntT22Gp9CVRaRHMTpNKxp0qftXZCfT/SQxOWkuWKzM2WBDYopQmAF  
bpScEayjgH+K9eckv6uznVFIQEGrjJVUCtD+ue6oKh6hSdHa3/TUGNrfa+io3Hi7sS9i/BcvbV2  
rMmmAuT22jIdHKiM2tIqzP4q+kaAAToudxY2iqIc7EDrZjSCiZSXLJfDtkKd2GIFwpUHenh1NeRl  
q8s4j0vbLl0RYubuVW5zf7BiCXEYCFGmmUjsA/OGDn43E8OGNzgcccrlKGS3UHj+hBCg0AtR313  
7EWEqx2HS/A8z/xPf+dvOvrV20L1jfa5mi+r/hjhl4btiqqpWbRUu3cppVtF6YQditbXahMHagE  
WDqENHy0EX0OPk7aMq9W8/JcrVWwHmBs9H9tTSXqpTrItKrcp+laiUxwHNxIxobCkYfUWwVS3kz  
QwKg0nqbpW5UmF711hdZBl1Djn516qm15VTas2y0nsmel1+rqv6TLIpwq/yQn1CHhuf0VZu1YUU+  
PotgET7Xynwjmjwso7TwLamKT+XDZDXSDVDjOJd4/JIERQnchIZKsXRbA4CXqek15qAFVrrMchUO1  
0TqumICBftmWWOynmiuxWQ/LFaTOF3pkCymw7oNU9M8Eto6DpKBPVjizHHcQqjFxnpsk+wYkoMp9  
86sQg6TaP3luNZt0H+KgN+nb4qfMa7ArbpoRFLNMs6mm4qa8Ej5zU3zNqyig7NMreW9Q3pr8IEu  
ABpmkaHsE4D0EEXq76xWk4MtSy0lBTHzrKN5bM0qT/cvyInt35ehpcekXg5gzVhZwjcs3SGG+1Q0  
y0X81KZgqFWU10kUfTWVesn4KSuXjLMSj9LK5lIe/k7eKnjt+vUjwLa19dpIwDBT60mrWKQ03NqS  
u3UZYQsWjfnV2vwSjXNWpmi9AAYBs5ot1MXBhc+uAe/kvTYZDyqqJUrOK+VKRJaY8qUBvBbcTxJS  
skaV6ltQTOL8ldKr3pUmbQHewDaWiL8nQtZtcPUFI4VUGRnvDyRdTSBjzysB2GBEazgf6ZRRiUGA  
ct8dEhh8Lb21urMOZ5rVbYc2zGcVv8+fNjDoiwd26ya+Iw78D0fij3fBBsa4dw58+N5POa4LiMcY

VJ3G9I/NbR5X6+xZW2KTOzNYmnK5kLKT/39Pyfv+Nq/uAXrG2kr1lOnSFb7yXp5JY/XLcPk92+o9  
dLIJqd3m5aW0DEAU4zvyWnkuQ6TsKXMSnk/Pxc1ilIMhxOduHkaUdTGLUF9TZlNemmKqnUG7ba1  
B/WzMuswbxbxby/yqUqTYX0vIr8XQCVT1kJ++HnsVqnn4ZSbm83agLfczL2pwwfgdKJBqKKWUWX6A  
pa3curfKzuHDytKjRmX0hwrP99sKaVmAYQ2gmPRYMoIlpGR24W0v2AjPP1X5lDVesOvZwqJWkyMA  
uuYj3gGF7o1YaOL5lOqas4AkuOz5nkt0zRTMTZmpY/v35Z+vyeNsA9XpTVJi6RXlYkDcFVYGHCYj  
aWVRy3Hb05B8xdpkr1YZukHLdu6h1Pg4B5H+YwhMw2mG5RarpUgzu0w6cKaaupM02ebRfCi1GlrW  
d84Wza+axpeowHf7cZyD0sagNXI4admpIm0T9rhYj1QbchlJGKpkp0UCguaOxqkiaJIilgGVjTlZ  
lNTNxrkpBVRvZJ6tdYlGIXqvB/K6kGhw4Xl1u2g2tzbBIbXmZp2uqSv763QP7ydzE5RS0BKcrG4i5  
d5ULzHwVfVOXZ+j1Yp1+SKjviLWpk2v1KFY3JOVsm8Vn+/bsI0lLG6qwbRCLXyuc1cdVmgV6aZez  
1RanTM9s5ro53KSAKpn5A6kv9QIztCLDXhRkTEIAGojr8/dqBeZ9eyuqv+bMokBPrvVuYrFb6T9u  
stlhMu9kP7B1Z5Xpj8nRXaI98HJdOZk04Zj5oblLnAuge8704b47y+ydIzzfYltEzjJOa7dkzjVn  
2Put6bCRj0fdhMB1tVSL5C5wedFK10pP/13/4y88w+/1W1RxBtiqwpW+/Wj05cfhq91UOSzb2hDd  
KVUL2kEAi4DIIPWUCOVqrQAC6Fgx2sm4zPVG+IcmChNNGmvyvvdS44sAJZ18YNj4d0aCK3kQYKIVV  
7+J7XFibJrKLfWJNDJrlK/KvBh1OkdDTvRlqf5AjSPberXnVbt3ygeLgaaHhsjEeBrQ4Ykz6qoVQ  
xx+xc4ivJ9ibByKTKFyGz4q04bUx2P6FwAk0TXLSosv2OVjOaE2t7PRvdjkiFkEwSNMQYmtLGzME  
TNyzQ4kS8sSNz25Vt20v5yNpEgLFVRPV2fi2aWWE87mIxmDqcxGt9lFgdPlfgFIZ+/qK7CM5ziup  
WFZ5/j38/F6+VyZx+cAc4XHNZz6Y+lu9ztw6l8E//QZ+KoBfpqmeUF9rRqoG6v6AKhlDVRReL/78T  
9/zzVuwvhG2dDV3osndA7s1vA6z2auKjAV+tTA2CwIoew9fi/k+jm2kxWB3CJvBg2Zb86SMAFPdk  
CkRzjItZVPkQMTf+l1tanQZrKFqA/tKgeBchwnXF1GpK2EK0KglSzbpGOPVYXGY8W0ZpdUeV2tTL  
CEXtb/GpoDiouDfUYpcWxNbBbPVj97s01Cf1tIIMoPHnCzgUoFQYwND+LCBWM2h2CHLDNvwPUNN5  
XABYvjKN1xpcYAWq7qYgokjFeom06BvSr0oIBLAndfXAm9UVQqyE6Nu9aNb4Xd62rWzWtxXmh/gG  
NqNpuwMD3BFHCL1Tjge5L9PR8RNFF1/GtWBKBUtdAPL9OyaVUDPoDnb+Ij7l3HbsHsC6xt89LCDHQ  
avzVLPX/auwor+DBRAXIP3PqO8nAeobXfD/TUODY2hmxPdf9g1xLsPSXU7W5w03aJuctulghTejR  
ANCap1A0aL5QmJ229D3JKCzWOIkAS0sZJl1nfjzrBYpq8wHVpD+IASqsoxVNwkt/NoCp/NZqelrs  
NbYEduzazExvyFhYOicGQAQtJqnqjWK1T9VS7kZTGywY8XStrz6pOpmAAKQ0WRzo0pBQJubSqJiI  
/UiF/XJtJBcGFjRRGLS26lraeGrshuA9YupZggfk7XIGcd3lEwJZeJZAahuKQkspt89lKgv38tIM  
BvrKTC+xPXM4rn68BRtmy7gix49hPMNJv+cSx7NsfDtAnd1/XWju4u/rcQFvXasfbXSnI1DOhHNJ  
lcavR249UZyMu4txsKYyi3S+N/iNP8TTmeG6/pf4Zz/AwfYZdHyGdvzn4FRTQHTGzGRDyDN1ZJ+m  
kDl1fvJv/unQIe/fQvWX7PAUpYaWR65levtpNGq5wVdl+CDDyTp/FxvcMqlUD+IFoFdN7ef/Tnp9  
br1TR/neoPfvX9PC9zDZkPScaoBFQKKlklFyiptq4cWi7nMpyuZw9dlAQIjqnrBqd/kuhtliELLY  
fb2j6Tfa9czavQbsdTX0xxhpQ5vTYMYxLJeHQReB6RMTayvzeOSHRDs1DZSP7wCtVdlwnLDImJlB  
ckqFi5WLJpgVRNHeVDLUK5HdnENQIulbsOr5V1YwuyrCBwLODY26LF2GrR/NS90X73hDbgGMWdTv  
gSdIxnfyulnp3dxHYc630blW+KpNtBn6zks6G3ZufxWpeDtsNYuVqHwepFKce06YCVtq6ralu9/K  
M/S1/HUR3AoZ0Y9Q/pFLFgTfAcT/Px/qyK/BwB/FQA6Mhg6/wyBKhdvq5/6e39a3vF1f20L118Tv  
m8rTbTz6f2G44d+kWXmGhRXUxFaLNHQsj/HaGkqY7qGhchSrUrKtGbWlFW0wO+gtUYqccQxh32NB  
mugid3ioMkrajZNZ6rgzwJ36glHcQbgOHUkeNNR42yS/znuoLPxuFrA7VZT2m08eqDdZV0UUVmbt  
rhNa11NKWvazJgK/Vc9BQaCrHoRqGhtseDw5qTVWyyNmj4/0YL6LM42Cvy1vhMDQZybetHMTqCqD  
GitvYJjn0mlyLS5ndVbKsBNOS+Rlgaj1lndZwsLuYTPORvf0hY4v7Wr7XI+3AfO/5mMz1RFoj88l  
NXyVItKdKQGh3ndf1na3T3seq202wvaGlLorEbeHuUhQSfzppFbj7uggYV8eqLcRl/BMC8xaOlK  
XTTbJicf22Yr8DaPw+gzkg7NkBVl+BTAfXpr/80tajGg1XwjceJ3zRgZaBFTzjsYeUaeSSlNDEj  
euwXQzWhpTPCwgiU86Pj6XTHUiAGzdaTjRtMgaoKqvOwL1vAmHgZ21lARqDESRR5EE7R6otaen6  
ObmOydAGIzibBkNBwF4lDdZUykwKxQY63VUS4nSysPSMv1TixJaWv1j6Y0GENPlxFmj2alxQ2VW  
asYmKslhT74XR1zmo9076jn88SSgJdhcYtFibEEjY6YrVaSlTZCxyo8qBX54CpNJEmao0zlhbcJ  
2eQqcB54DDE6ww1IMX0B9+rM1/LfVmoT8Vg4QRF0JtFPbAl17acEjj3pYdr1j1m2UzFoMJG6cocI  
G92j3SqHlwHTDnpvSySghvnphu845ZlX5e5odus/snsVwsA2v6z2u/1BzVQLU8/B5ZtncdC+HLF  
SlwbVEvgArQ8iIWGvW9CCYZF/fIRfR9A9Q3mnV986RuGHENArDWaes3d+LVclGyn9JrDnEzTvH9r  
eGjuSqqzWFRtJSNkGr4ddohA0WMVpHe1FNQ24sSwPV8ihv/+gOtJA45JqVjqoM9mnXbWy10RhBum  
6iwrSpgdu0Y06Dysc47LUFNCZIM1nAVxeJpI3ct1KZ9POwPh/WkjbU3PnJ9Z+U1Fa7qaiWNQPL9  
EH54HN+GOB8OrCajoRhUwM7lFgR09ykGtdFFyzs1M9N80TA4yUDbUBnrDKNUKe4dhYJ2xSsrR0t  
TTSTNiQ34eVdVqz05v47UL+KZt1QsuVguZTk8kqWjd3MJGQ7r9Piy7I/Fqhuv16fCnG9+3rRTVB  
VwHvzG4C0qfGMwQW845jOW8SJMnwvbwK7Dg3QDbmIH6rgDUMY79Pq7Ju6QWP/8o9ZoaLdFWoFYXQ  
H1ti9wFQH8KVvUi5bYNMP3aY1VDq1WRjcxGOE3Wk9SkEK5KiSRq6fz2roKGIt4BLGynPdSxjExrT  
E/va08lhbPj+2gt2Cqnc2c26Q4H1sTSKWYftqKZm44aWiZOoWmuMmGENKutKntWax0jQxcBgZ4pc  
5m4p+Kkko3NFMdmKWKk2r+MZ+nN6JgbvaZMo8qkvkZhaqCqHqAFCLn1kKm9nR1p9XfFDdtKo302y  
VFZMJrreTBlxdwxj/Hc+nr4qT26rqvjMQzX1FpfrQf2qI7YrvWEGVyyChWSq1ZTZSo+FoJqz4Yfe  
6w2ioUWdSugr504nlkrWLR0DOa81loGG+FwqhWodBIv1cKbQciWv6umFGe4NhOsJg2jMuEqB7ej1  
fLzglb7JaA9xtfyfhzYXUPD6mZVlNV9nOhBXRtSucDobQD0pXd+3V+JXntPfBz1/YT3zRvJur5pw

EopTbfVT6LR7XtZvDg2bW/tuEaYJCSzBVgt3GwJ5UXEUm0jjkW0qaK/6QwpK53bpgEdU+q0jMWCi  
aCxmQguGytVaDqDqzetMQM5HI14PpvJ87fu4Yr7OieHNJE6u7v9rjQ9WzWSONApMFks4GihfJaCm  
rIIIn7pFVPtXtfBcXAZ5VEvNAuM1tSCiLqSoazLKqi6waLZDCVqPKcXmyEgOSo7iqUa1x+OpTFcZH  
vAn4X+fTlB4fQGfeajiZd1GT9qhI/vDlhwOutLrgSjvNJeUxhulLnDqLzIgxY9r4mFaSbl4lwVE  
z2vpzlrYt+sqW0GV2Q1WsiqAYs8t5SOU2fYtjK4EDNJvXNcS7wW7sT43qn0hpcFFwMLQZuiTD3Dw  
upjWsvKstZes/sxnPKTEv7832AyvSLNQHFmPnJibGPFUMDGoo7wXUY0/vETbK+N+hKYb2TAvnkCT  
I1h5fWPUuvWs/eyZPmyaTmzslx3QRNddnwwwGQAMCvOLQM5QdJS+lQq4DMwo40IKd4NKvjavCrVI  
Uhx9d9MgbCURoGaQXLfKycj+dkXbsv985kss7r3lGqDaTLsTcn6lZ6oNt7vb7ch9VpOJX02h1pe  
Jb00Gk0HFV1COimMtBkMuq81uJ6y6S2kl3rL5Eos+MnzdQia5EOzsmExcT5qnWPAMrZfCnnZyOZJ  
iWIZUtuT8/1ldFU7ty/q1bWdkbS6gzk/nImPhaC507Gcrk/kpv7Pb15+Uha7bAu0ZNqow1V9+Wad  
bpXVfvDoCdxSqkbWE23peqhVLHIIk5m76obQSoewQ/mqI8sGkmjc6gtdfStvRyWPTjQNr205Mzb5  
KrfCam46/xWbZRFZcME6uWac+9Rvd9+KJuFUYRmo53VOT5K9T1xvFRZ7jcBJNMpcCf5dubanGy  
Kj6X0jXZwP0tXiJozT7mJyauMGMDivSYuRamQcdcHxF6SnOWjplEOblmtZrmOt/KG1JKAJ5H47k  
MduXAP162g+ldFd1gGw1/P4fCQfe+mu3F1VOiWcSon0h0uAPoW1FfigZ6DXZ5NzSWD5+8MjVfnn0  
GLWHdN/TtOs7rKhZiUkDNe/q9CZjt2wVYSM6GQ6RT3hqu5zptWl8gL/TT94Dd94PAKIDE+S0pIf+  
+iH5WY5lNfKBJB54nd2dTrd+ehExmenssTxrtJCZnEuy3ilY0Dqzh/RCiadMMBJeWyWp1Yyelpl0  
jq0045qTWAd02HAVw7lFAIKHAXrFQoAyPFdHGciYetQZmfPw9l0a3eM424KMLhJpQ09wCb8Nj6pc  
ECb9+HKdOGf+thHHy7HQWkUTwDI74Jb87DtNfbBSu4Cm9OK079qH1ULgO/99L+UYzwO3/GbPuH9c  
fB5XyLHP/PBT/gc/87nt2B9nbcP/5O/LGcf+Xe/6DF46wekjFYl6JYTL0aXsjThCt3ygtDkgGR2j  
KTrqQJAI7xm4UwnAB7809OX+XymAR5aKlBjcC5Mv92Qt1y9WqvJV3W9YQS/dDJdyHN3T+R+ZMgxL  
NdaA0auvPlD75VLV6+pn8wpaqMptY/6YIiW3L77CqypL11YWRUYZFoJZeClpYRuOGLhRmXURV7UI  
9bmAaPOv2rS0d6kJURre9M83USINymc+Rx/t2SN/X3kliuyd+NtMgUlH+4N5Nbdc4BpJe9819Ogw  
g25ev0h6fYHcVv1F+FnLgVLmMxXaywut8XK19IMax1t6Zj1PBubfrvOvMk0QJCKKx06RfKm/rsTy  
v37r2AxGKivW3D+Da5hGHZlvTyHi3IAKrzQjidaWfJsxgZsirlxCKE6Y/GK6/ide5YuVJaZLEaNJ  
F176XLUJmtpuJqOutF8dN1vDL4U7mu7ypNFpRUhFQ+kvPBN7wOwnwx4b2TAvgnKDY3NjWzIx77vr  
zE8kThh+5bX7v4sNX46vWVHDAktJmKaRUXJcBNSwoSDqFgiuFonWmL4ajlUoSkdvr7hufArnbqMk  
K0eRSrL+UI+9vwL8FGP5bKXXpHfDCYt3IS/82t+n7z9bW+XPJqBZjvyyNveJbdHEzG9lvzG3/o1o  
IigjM2hfPiFlznZDuC3YAFFV1Euk9VS0sLQelz6xDRbZX4hV1ppPW7ttHIWa1QPZ1LFxAbutgfgm  
lqn040zuTcay1ve8QF5y+NP4TxMufHo22W9ShTsT3/+b5RL167K715fVuzr/2v58q/43dLpBLLCQ  
kVX4NLBvdm/fF2aoMq0pjoOcjOPh3NnqUXFwg+KuvFactExcNxUYAycpvrrHRYWIwSJ7Dec7EbX  
ZULUDxZ8YnmnM26ZdXdfmkG7Q0yGds2mlgwmFM8yBo7WRh2Bc37GgqSjNZHHw9udcyDPPLAFQGM  
E61Ltl1+UPDWjtdgddKbG6xq5UpNR3BOqsD/EaobgB9anjtp7j30UViEE8v1clbqtlpNzVFSTYDC1  
AxHLOFzThdU4hetBaYwWFGkWjga71XlPjng2Nx0vrBAgeV3vHn3Ll/R6qcWLFb3MFTa2mp19IZrt  
3q6r3t3npXdjg/fMJETu7fw945cfEhRGR7ekOeff6meugYOy/LAM1YbFZVKouasRGJdcpVtvkV3M  
1WirC2b1BTacpsK1oIR4Ggps2Umz334ozLcvy7Dgz1YtUCGg3258+LH5KmnHpYbV/rysZ/7UTk/u  
S+Hlx+S45efl+d+6ofkUq8j73rrTfn8tz8qj968KoPesAYc6KoWRKhYuVfbV0bFnVovmXNgLSoc  
pIenm+EpMlp/fyFjhWLNIDIG8jy9duzhHZ3dkzGYSVYyql2oH8vJ7FyQqiI5xPdqG14whl1NQMsLF  
rzW/QStNitgIcWC1gZV/iJcjL3XAlXTNJ8kkHSxfapgoi/13i1YP9ON+rPwAQ10YW0rwapld6Xc+  
o//VxX0LieguDPXb65hbfIClsjzA1X3Y8mdYbhqRXJYI/qpSwpVa9N3rUnL3ylgxvhwu+GrdakFu  
DZi2aDRTfx7Hzf51cOBDJq+5hI/+P3/FFTvwuwcXgPNs9UK/bpnnpHL1/bl5Y9+SK5ee6t+Rq+3o  
z4mBxxfjMpgSmcJS0TB8fxCIA1+7X8W5OTAKqlV/jUHixvY4AjGeCmG38bTplw5eFhTSqvZVINg  
7ztvbowcSE5PDYU8emxUtdbH/k5+fff94/kJqzsFz7zbn6sevy5KOPyP7eobTaA/j0jlpW1YzaS  
KVe5H3pKugUdgC63RlKujxTK9lq91T5gmWQ7HEt87pjKYYlpY+u/cMZh331ZDEby/kZfOfJQs7Oz  
mS1XNfSNFstMINGEq0H8PubeWmYWbRSHznCd8VqrenpHfjct/uFEfWrg5IvBp0HMT4Ti/pgKzX8H  
I8GG6+Wpnzcnj9+VpLl2Q6++mG2HntO2NPxh5xJyim9VHRiAcvNKiU1IGSslI93hL0WL/5qm4ba  
/rBg3GNOQceEyNmBcD1lOreUHJJRqtMbu/uyC8896z8+x/4F9Ld3RW/2ZT1ZCyz8bGY8KauHF2R+  
PxY7tBvbYVYBALNh7bbhyrrSeudJ3WZIMHKiLOW/m3GGmrd+wa4Gk5hA6gTaK0v9ZamsNRj+Jsrlr  
PDI/ESmd16UE3w+LVmERerkpI4GczRiv9sWZ3hJ3vOe90jbt+Xo6EA6WJQ4tJ21zIxak2YbnOPis  
JdtbWb4bLSOTVh3Ti1oYNERR5a0ikWdj1Y5VSobWpWEZb42n2cp/HaDBRKuSqAyF8t0EJZFyeAnU  
yo1Wq2kDR8abgsAnitNpiAA9ePEHyRVofFWs1MsQLUvnFueLkfnveH+we/P1zxR42M4wOmnC8JfS  
wv65kzdXAIUbXzWi79Fs/vNdH76VGL6Dy90bzut/beAap3oWArLriuIElYVYbXmsCfRrptUrarpm  
JKu6tw6VR5c36mjm/qQuksMm7AoLH6wzFCMKJHdFqjZXgOW9GHQu6n2clpVLIPQl7f//+y9B7Rk5  
1Umuk+uUznd3LlbLbWktmVbtixsbJID9ozNYAbDM7PG5sEzPDBhHm/AngcGFmEeMCzSEJ6HgWEYY  
AwM4EV2toJlyZKVU6tz9823cjinTnr72/9fdVvGHmyjUbDukWrdvvdWnTp17r//nb79fYsnqVopU  
b0xJ6Exeo+Y8EkOLkofViJ6iE4BcG+qdpEMgstmlrYmyldiVgVmBCmA3QgSlck/877K7SPQ+LKg  
avEIBE9KDDEROHux902RdceE2Nz8yVBF/kcslarVRGJdv19XczPGgrbDPpScA/LNBCq4pJukAj9I  
G9E2Op4AFPweXiFtRGKs3fPg1ycP0uA+wg5zQSNv9hb0tek3LemuMUAZ67zblRfMumhgaQcav6



e5sC/yx1lzi63SEsWLU2+QNKPBQFEShbtBrU6e9KUEjZnXJKby5mjd+E3EHG6DJxpp+oZ712WawX  
5bGatj2kxysWvHTVBZbjb0m12DjU3L7weAFV0vFwh50SVKUcgbv1+y1hgTBd1eR3eJFBhxQzngbv2  
JT2Zbd3+H3y7L2kuAMWQoypszmk8MIIX4GG9x3RS12eq9Bio8TGURKgOopAYKAAH3GhUBYoIpgBM  
QAQg12BF/sIkDvgcBOFdFKFLJWPo7iEjoQo3mGIVhOwqUmcRDYOhNTCJMjXCoMrWjF/jTkPBLIpI  
pvf0Yh71M6Xyfy592aPZ/k5AWA4gu21tGI7zeCUSklFlqZysWUIQEL1NBTCNCypNOyLEXqgfDESv  
kcFGkwkZpEK8/b2JuUKJQq629RjI6409onBsnXRqLsj+OB6hQ2215GpHPwen2XYWeXoYiRwzkYuL  
/fEL1UBcDmThqPr02hSQXGrXKlTfzSUKaB+d8duNg5+VTzuPcQXMHpOtx6/7IvB6RU2m8JzulYax  
XNB2GqMhx0b7HwYKAcA38mZIp6MiBIV4LEA7DXtB5sh5iy7rU2Vowq9ykiKTHJ+gCVSZZSSgDxWKU  
1E0dyVsFZ1UNky7VFDMEokv3keaoJkleRzYFYSVgX8fSq6dyMQPvH3Oq0g7iaYqddn0z5fNktSqi  
KKkMuB9wY6YTgfUOT+EopyJiV5JgS1Nmp6pQpRUC/MK3OBxaA+qGduSyq7KQ7XSHxt7TB2BhiXmU  
FciCgn/HYEeSr8X87AwctCPEm9Gvk99kIdneWgKBHwRawGV+f20s3FWVPgwUDaE9aTiN0ELpzovq  
vI9ft6ouyVgilrzsAiBjXvb10e/AfLbK0guFZorHcv1zpfqc9C2ubrb2uKNELO3rka/ObzmlId2v  
lCv+oWEyM9ETvrl6Vk/z58ERaAoGNrSFQ+yHdw06PYKHodcAeriocHzBAjIfqesNeSCrAojaeCy  
0WKCI8KBXAJVagdaScIrxJYkhJZ/FNVN4RiMV+MbajwGsPjQpdiguZUFaKE/zdW4mYyMwoqBIDvI  
ZYcjQ9UBHRVPsc5o2PthvN4LakBckD+xPNlirgbhoxwOhECmkQ8NUbcTNHQQPNFzSpmxqNQ1EMx  
gXjFGZG+WCuHulTX0UAGjmxoaUjCQmpKeeTvogewAcwRKqzqBLzPQKJucteEPIeE9GHHXHovkiXz  
z0qc7LALePDJklc3gxVRR5q7jBQEK/10Yc00bZiD4zNckbocWgsdlv4pQjV+xOckgSwwzrr8PXni  
0XpTY8RKQiiZnhvXJGHPHuLR3t9Vu1Tp/BABSbwI6fcmQu1h4IOdzFqIRhc5T9IyiBW+CtjQRii  
LYELGIAYADUD4vU1syDYEjDPCcWpW0o9jyw4CNMTWUqRk3rwIv5rsdGyiEozmFqMjNcDaZbQE+qx  
ZZtWfSJGB54nCxek3kVemD76wI7C0oqXnPV11YcVjyduKpoHV8++ijgWe0KbPbcnAH9LyLgBZIBnR  
OvFY6+K/itaH5gvRZ4HCKDMtpqeyosxzoFcUo3o98TnwP8UsIJJdcDTPNAaF/AiAiNWNr/wPGEa  
nHEHhTgE6Ca5vcdp3Q0UCJbQnDOoT88Nf8tMOUDpTuT70HeKyrRLTcnYBVcm3jLKT7byctXCMMmg  
/aJaLzJYXMzm1852EW+DaPmDRgk4MVpgAwj/ULy0WebUZtfrgY6zU9FFzSaCOM720An4TidjEdgw  
7vFL8+FyL2K9UUqVGpisOEK4hBYcSshQrVB/5mlws4PAwIYQpgM2csAojdt30tYUcJQiaOtKRGaI  
UYLgWNLS2Wg6YOfkaVG4GAEEb8h4kZ7CERj/F5Fzt1A+wK2f1SghYhMyA8UZQul6vPFxq4+E7ifD  
E2mBi+OCEa8v5GqsJw9NHJpjp6pIRVTWaxCI2OaStl05F/ZAPieyb8hwYGSVYopm0Rtp3oCNZDNI  
1EhPB4IPywbCnc6vMHG4HvCyh9MOctv06izrarbiSjec4tF2QSRX4P+BQYLInCE9kgdMCyBnrNAF  
F2FH0b1XpgeDfTRMTWVfVPLiQu1/UhXLsVRUHFLFUwE2cP2ep3/QieResMA8TC+iJX0uR57xvoUH  
ZBsDERgaiTsBPCMsUgMjMc89e8cuz5pQ/kStVWw3QF4q1EihKp+mKZIRSTCjAa8o5MbIpuSw6D5  
QDuQxoRxxgJqThNeV4eThikGidwTYaKtKeIzIHgc9B5h0ML8oMm+NRNimgXilfhNxxZCR18GboMjj2  
sAq5xRiiFTICEu4spRWqwWQsakq3qamfZ1GE2qeNtKe0hfqToDtbFUpib7s5ReDjwVDBD5L24Kb  
wJS+YWXTtMZ0yqqzvBsM0ZGS8uAiOqeQ6VAHKVLa+q2juY6nkQjMtw7aydpWrzEA26APwHAp+Uc  
6UKVAKdW3hjCfCZolqAqhly+HDUESK7pN8TmhlDnfpQaOeMDV7n7uaF/d2dLeqsnpylnkWB7RRrr  
+CTeNM18qIv1Gsaxud+7BnrUwVeyv7BgpUZUF202Xj449nBV791s1Rt3GG4pUm/3aZOa4d63TYbg  
S2D2GAqGA06KtQT5v1EVT7Z8LEg0ee0Dcy1RkJflulqqRKWUgaL3BRE0JE8zpxw9pvsDYRmeHqdq  
KwCvYPwGV7RUZM88Macj7VV+Hfln0w+I2apNbuh/Fs2nFTmW5UBKa8+LSQBeK/C81jaTUKRSrGA8  
9U5VEXbAOKCaHhBDJemeljekPcSmhShN7U1+bgr4TJYErOoK2gqU79GKQ8ojihFysYm7teo017jD  
5+n1s4Fqi0clTlhDB04fa71dM3SiLFAvyiGC+9aLM9J3xi9WGCKIXDxS0spkQcYgZHwn0Gao78ju  
GLX8Sjs72CQ8BCH13P3/d4Pf1GW9qK3/5wmCn/yY89Ynypj/RyPf/Ac24tsI/mE63ldFDGHO7HWf  
4F+q6sgcbxj5/ySTK6QpbxhjH5hviJFGSzlPtBSyIXRnbdVCDxVBDdNleeiD4mwUnK/bKIWseRda  
gMQYEMui/E4Iipq6T6uYuxXxj8N7RPx/BJ+T0PfNNWftUtlmMgtM/PJScEVGxi8PPIhgMBS1zXZV  
QSQ7C9VFV4bkYKMnCVvhs+EnBZkbCbNKFBRsU5FIsQXOKG8P9pKpAbkRbEOUQvasRjN42CitXOZ8  
+cqe9hTlK80ZhVtR/JbtVkB9gkfDoMFyMLnDRQPEp1Z4J8tSsc9EdOy8yXKcRoJg/ehIqerLB4U1  
fbG/FKQDFvVSXe74rn+F21pImL1WY89Y306DXrUy4w0fshlnMex0MJA0WqimlgUkmVwW8JEW0R+H  
fGuprAfKhkMnwJp82SCGJoRjgEPq0CoZE49oKWY4E3NGqhgeakiNUOYKfqsJMz8KgpQeSM8g4R4U  
9LCRALbZUmiq9eJglHKQo+nu5B48ytBIcJ8qBFGwkU87cnairUQXhdGBuMVzwQeJ9OYFdISznWNa  
VFLRvIUOEL+bdS691WE3+JphYw8FYOGt1fQSEPaRb32Onu7Al+KR6uXTrODbdCafwYjRa6uRg8dC  
Y0xdijskXw9frH07xhRqb4kXtT3KyrvBZRSpqSQ7+ZkL8M9LlUqlPFGGo934vLDJz7sVZp/VF46c  
ikOR1/0XOvJt/20kuG84rFnrE/j8djf/AZHn8PznkOFLfbqKSOmhZxLYH6FX2L5mo1qhZyVCmVK  
ZdzBeSfJqoPC2QOpkN6Q632DfZEw9EppbObL6JqKr3URBaieiJvg16oGKSR6YhAeSQBV0BQSjiLU  
zFy5HvKUFW7Bbmtai9DwmhpxD2w3i2gaQzQEgyA0rMWjAShifyemH710LXFcWyLNHhr14kspso8  
IeEy5Lou2LECI+B/pLPnCopEtXzVZ4V0pJhZggyLIpVoQ+TRsMgorW1VTa2vNDlKGX3SCQ+sLEkb  
Gy5UpMUjJGE2A4zxvlyTb5iGseC4DXoZKSvm0uFtiZoU3312ijsbce15aMD7AK2a+9MgkHv+n/9c  
1/SELpS/tt97Bnr01owNrIkjtscCn9wfq55V2N+mcqVKpVKJSOpUPZpvVGh5sUILjSitzc+xt/Vlv  
ZoO2A0VYBw52NrWlnhkLMipJzKmHL7wKsrc1M3OVD5pai4jAVZo4aTpgkxnglaZCjE1GCFJwd000  
sihTDWT9WsVg4Sa+ELBM5ooQzOzJy8y8bCkNF/k/U1RUJ01ptV7eqoVY9g61NZs9lMRrG1Irb2qN  
GEz9oi211eUO4/1+0yjqAzNnAEIJOZaoX4AT4zwFj3mS5fPUTgOBjME9oyEzVL5LVawbvHPCzLFM

y2SoRVUqC9Lvs0hD0X9bfn8pm0HFmCOPsow3I5fKsbF5tL9o9aFu8Lh4IxfqCRf6nK59q0/vte6e  
YYtlu0t/YxN4X8olWvbeTbIUj5HtXKR1lf20/JclRarOVpcqF0dwyrbtQQg0e93BdwP5od2L5BKs  
3gzHYoKSEDU3ZSZyiIjZcjCeJhqDVcYjEheqLww0z1WoqloFRuVo4wCHOemrzOVEQ1OF6B+QP3iS  
BWHklQD+xORs5AHUFZCqqb0Uqlpb1Z7UgMYWoTEbn6X2V9LOqr+jwJjiKfWue+sZSW6s5Y0y7U8i  
MTsKqdFu0vu1XgibSTbLkqLaoRh4S+ehmKmt+wL60bS/eZefZLyMzpgleoU9DfEFAFBjDRRffK8  
3x5vJmh2ARK1wwF43U21gUadS7F9ZUTn3Zs9w/zc0f+Np1EW4f/2Q/+k6hdhBxRPx76w/fuGevTX  
Ica8wLbZk/51waFv5jLV0Psyj5aJiaIyUyhWSnyQ81kqkmWIS80AxSak5i2Oi02Xt1yEdYEVw+CK  
6kKEs+nw01BG7mqv4ogMUT15TrVvUulGSqhMryTYP/0cLneCCDHI7y/tuY/ShWvk/JgkgnrDUPzC  
gO7zDkcr3rFmZQqtbcPWXhmGrOQV8AWAB1I3uypPBXXS3pTGfeVSh2erzV75LqkVaMkKEWKMLM5L  
zYvKYjx9bV5Y2uNxtTijS5MbWqPehQmSsEAPdXxGFjelAJ+jinACDZqKLMjj8Xz+GuxeZAI5HXIY  
yXtQN84L6AOaOay3wbXqtWx1/ED/v2n0iS+jy/g4nXf91PxP3XBXPetPyF94elz1j/1/RyZqNkw  
qBwxUNFXNYgCUD/mqbGo46phrjBcACepEQEpWLPd6YyHG2IAQ16W9L7hN7NRmtD1XjESKwZHajS6  
1U9UVN4kYBKImdVWRnYfSV0R0i8p3qtMALxLLww+SJE2HhaHQZ1KwZCDeE8UsatZ1j5YVYkqNFULY  
vG4RhIqXiS0UwAPnOXTl1gqxAW5A2wWFLFO3aBABWEq3VIl1iqdaOwQYk31Zyc1NFCoa9qxErsh2mP  
EdVgdVGAz02c0WPb7Roc3BkLZftgmFLZKFCITzMeGw57QwA6hgM7vgeEC5PTYgGSD4Xw339wvG  
rpZOFZcUIa659DIQainIh361abrx2kyMeMk2owBIn6KjpPf9lMiBSpyoHvG+hTa6NQosXCmxvq52  
zo503b741H/d1c3t8PU8lORrEB1kw0BJGFDIJDCKQAPgJ8VbiClAndxY00NuOvwlUzVqM9SzQII2  
k6ZXLHFIEw9qC4ACaEjUcwJUmAhhRZCCBiNozK1An5fMWT0NyG9IZ5VDX5LZRbFMUefB+FxZkjuS  
thsmHGKjKfI+MQEuG0KjDxNcEIBSWEpfq/tsr9DM3kIOhFRd1Cup8rFWgtfiUFKVMx90/DfvRhk  
TsrfR61GYb8/ZnVNeqHIRU5chkOQhrzJYDdcDjqCvABEYsCmhD1ex2RIULJK+Tzrqjwye6XBJQr1  
cnycpRqhFUEKQ+nRJMB57eFxmACQAs/fzIelnmDdaEm8FQeCfLvaM9Yn+oi0uftsz75sBLHye102  
p3ffeTxRx85d+HSCms7PhutsB0CKQS8MHYXKLIWtIYx0gb/Npmq0MTOKPiYJYvCv7XNsXbiQFo7  
VTtyXUhSg0DiNr4FX1OzJtiwQN9BXfiyX3BZSyjawqPbEnImpM8k3Ke0KhK7jmtEOOzx5mE0/DoK  
Z8riyJpbcisK+kQWlOKqpZROqsiS3g8XR6CZjJm4f3uqo1V1AK1dxgmvLuttGPj2aYTU3c8pvYAA  
+lsyIYie0NPOeXnYeIpkUpxrGQvM3VNqAds2/IQIWpI5EQkzuGGpLIFcoy+4rCHtpVMeelguE2r  
B0Mpg8Gvdy5R++9Jo7jw2maOE/lrrrx035RPtenfuv79oz1qbNVY7cV8T97mmklMKhKpZS61vcJD  
svG3f4wG8Ng2bPAYKd4YZR2wCpoCeeQI2t1qzegbr8nJpBg8aDKC/U1LEyyrjDSTIW8WjRZMLzoD  
zqKPDuz9GKNffs+SeHJ5NzZiY9X4a9AC9kyh0q6RyqiysAvg1kQhuLYWrdVRQYwCuivIxxGeIjBd  
lMKYvreJPGTWRLi9fWYm8ptDa0Wrnq5s7aFeFhrN81Q6AGFKdYcZKLnw79b3eqKrAbuHobe0R7DZ  
paCRxkjfXx9ne62tHZQdHJsJesB0vWdzYtskjb8zHXLBAoXoa+hBKSBRQZ6SuhhnQLmXZuG54/Hs  
WkPY2PzOMPfRXf6Obtv/49Tyns6KZ3/oo8ns7jeULybfXpVkfSRf08bPbVatXS0SMHh8PR0BsHo  
YEFaa0J4V3BXAggyp2T0BrwQc/IC3k2hrAvb65Ssz4nFC/sqJWqG9Fs2kdEsLTnMAYt1G6p3FTgd  
TC8SEa3xYiNKCEHhS3xpnmyOXyYVe4nxSYwZGhgQR8wlKRVXkdmASlBQMvpwtAGCXLQAeDqRPm  
6LNgp/ZClssVWwoypkKA22YyuAMreRuGEoIS77B0LtIW8a62GSq0i/CYFSpMTKHYTHDRRC4CxvtN  
uekhkzP9DnsLeRrcjqkkoBlgjldWme+Jsyr+qU5DhZKHDaPqdftsbe+QQf2H6VaY47fljcyjmik+  
OUU2dDZa198SLiHhS7GzYwCmz3ts8cD6003mpdvInJzFft27f8Z/f9/o8GQXddNjKbz+9XGpIe4  
W907Te9+1m/iu3ng6HKOsUMJ3mQ2fwEyrqGWeI/GIR3g+Zc44HJ2iTsdrjU4YdjW/q170E49Br10  
6L0zWEzh3TxDaxwfnWDRr8qohhtmkiz1MMoU71w4W1AXD3rqWqDJWMGcsd0DbTV7MiW2Vn0FN0cq  
rEcVPIiN3SIKyAJGidImE89XSRTMmjvmHoPSvX7GFqNXQyLN5EY3tm1FUAFHgreMAYRXKYKJqWSn  
an6LgpK0wpvGitgfuooQWc58WTX00rBR1WmEX6rsNag0xdWpeeazxdEpNq3xwLNFEXxPBKuKcEim  
6plB08PwH+1foD65x+ikDeuxx67j05c/yKqNxb4I7B3HWN2dyy5e6151PqdNd7cKlApyGWWawwG2  
5wr53LslY9evHza9xc8bFaGNxbcnY0tHhEcmr8jU78ix96zqzkL3tj3aVesmiXRM24on+TWYVe5  
p+tob7juU6uXqk8cf7M6footuwk7MsfNceLe5Iqfl4QcoVBm7x8icI0oBx7wNwTDV5UmqZU43YhS  
iX5VqY2DJqhmjRtqKkkMQSVZftiXBB9Eie4I9QtHVNADmkWKSfmTAULIaRm+Sv7VhR6ObxF0UiG2  
FHkkcJQqr2rIVy9Fn5fqEoLBKRvaWIJCgtGhqKXGIuEs47ymKQ95gzGmKkqsKY6nZG0SQacacBFJ  
HzG8lqA9znM5bSZNre3pXKOW+X7iNncYW/InyegWrnKIEyIfL+gQnPkqPxyYlKHwx2h0sH9wrjB+  
dNn5Hke3+8kGQlXOSIEeGQYKsbpUt5tjCSx5+YO0r0P3m3n/LnGJDRefvdddz/8optuWkv9dN0b7  
mQA+id8bz75a9+pSOhiZbz15kF64bf92LNYLT9/+qyGqS3X+Ox4GH8zNBC7V33d2/sWW0+xmLsjT  
iYblmm1xmOM2QXSWNDFi3GQhwIQh8TA9lqSRzqYfJa21xTPVUB6cczXiQshkTv5JnuarPvV8LZ  
kNlrE7CUIWMhQcOEGFDF3X7/CMUDBESJkoSQ96fQ1rOATPounJOHbDX2j71EO08cj+NtzeKvSFAC  
higDJhzbpzzldwsqTE4AWJo+KBibdSgDgD5sytbX4kyVHjNWH+ldLfwPMnchNTNzGY5eo+vbTAAS  
2OVwmFbV7xNUQwARDLg+2QXazLoiJQCD2HMMAXhh0AKgAELjNRhU+m1W1IpdgsluX+RoKEsgYEKW  
YBqz+UaCytrc/MH2OA79mA83Me/fc19d9/zkm5o+9jYEGp/HoLwZ+0Sfh6BIRLPfyRrWpaUnvjgb  
8N8hq5t7xw/evQPHTOxhF0vVGRfWap7nuzdczrjx7xQfA6NZbFz+La+vTnrN6qyaqKhgYYYY5YpB  
TbV01DwQtXGUTOhUtjRUEUB2HN+HPa2KRix5yg0aNBalUKKPIKhUHiiTjvNeFNJRmM2RM12YgdC  
rfWhVfyWCB4vixTKShGoTBlrlZ1VWKRMady5ieJkWXGxPPNppZZX1qtKZCUYlhEmlARDoTDDf01  
BCM6/zaKhVKRY4+fIowJB7zNWNwn6/DyxXY0Y9oPBjyZmYp6tTZKFGiSM75miCvgXx6wtFEp98Tn



uHBzppuxVmq7wnjjwSAYWLwnP260WjM01xzGclCMZ4k1w3Hyeseve/eq7bHQpUh+OvPRRC+FwY//  
RWl7MmGmuq96UmeNROQg+rxZBJCsltkQ6kfPnIk12bG3asb269ib8spnkuNalWptWGW1avS5vqqF  
DuoUGTf5dNmpyv9UWkPpirPtYwpAEIYdlWOaShpSFKTSKrNI4j5VF4/NRBHeJsKusBlklDzlt5iN  
OVrYu9qZ2oiRvqvjaaihGlUyWJvZbKBCTAentBQ1y0hLobkASPUBY+pF5y2aQwz0UsDobGlveoVt  
1M2IUXRqnpPWmdSNWs1gMEWBfn1rU1anm9QO05SjCiADTSeRMJgOOb8v1ht0JCNEd65Pwxoro6KL  
nvUXEk2AdzFWnWRAG7xhxw57KyeoWNXH+dz7mej7UtrDM8CybdV8tmwu14uX0wMJzdeaNyGjmmXL  
d59L1xcm49SesV2f3j69KOP7CSHltYaJWaC6rJseNiDeSOJg/GesT7jtIU5lqOLv5/tZUXLQWWwa  
dJhJ5B3ff8FR44d7fg5f2t1fXUhtTNT2gquKbOujPMXjG+fvRm8HOQXelCL63dpsdbgO5vqPHlKw  
m2q/JOUIPC0Boy35qAlIwidVWocmJcTfeILT07maZqvA2eADkr2PV5iQoiysxyZBCK0gYSwu0sP  
wPdYwTPmEIMUzVrK2GqSHMoahjB6eO+8Pum043GFPVphTHW0cFui8aQzyNGK4+JgjeSYttA3omKM  
QLwIRsDqsCYtMnn8yoS4R0M/VXglfHVtHwJywdspIt2kz0vPxc0r+yNAXUyeKOMRgNpilk2RzqBS  
T300YscoyIzESAijCaGgSKzG/VaVtnyaqaVtYuuUTx65Ei70+qWur3RMD6AX7++tXN2cXH+o2ykf  
co1RL71SoLwvTD4GY95p9VgU0+R6MWXqUoqPIXydtmFJIn/gSPKWzjkXV9cXrq32Zw7zXlQgkQp  
VighblFORB/ma675ghVSuxTPUeqnwmHbBfXLus+oAKxT2ddSft25dmm3yeCgEKhruFNIwHiG5qex  
dDVY9sxFbMfwaRsrA4vYoyUCcLYBFiC8lCwFlqObqHo/JEi7QVTlWcKdFBhkw1wRDMq+grI4LSoN  
B2LE6ObjYKlGj+sIpR0OqgwY+LQI3YAVWi2Cni7Ab9nf8CbGBtCDuTh700RaSAvDaAZy9cWjPszw  
AgA+Jxfyp8FtDyl+hy5qE7HQ855axIOG55L584+xp56wt+3NXGaIYrpqfSmzdgrVrPMBomqWXQ8Z  
9s108HJ649zqGv5ju+eHA66r+912ke2d7btKGWziFJAEA6whiII3/Osz4LImBfHqEO33XG3cfPLT  
mZorWCnN9JAOW1ZfOM0ST8SBpM72UBv4K28US4VjFbL/e7haPxK2/UcXyZbUtq3vCBh6VonpI3V8  
4JPvbixTSevAjO/GqJGziUhsCUkPapNk6nI2xSYQKyqu0AZae5fUw+KUwaDjYn0hI3tKQ410Ji6+  
aJSII9CyT1h0ArdY+sdgT9LFKuIAgaaxbMWi+qt0m74K4MG02IYPK01I3RDSambCkYo9DMCEdLks  
0iPWFe+VW6uIolEfVK6tHqJdrbWyEUeCO90kKFMxYmpZsgcRWFCUWYL1erO9irZHXJ4jqdg15Mxz  
c0v0XDU47vEaUgdGkD8c8+nLkcwrigVDGkMoDEAERLKRgZIA6Jg5KactPPf00aw3ij4zupNN5xY7  
LS7zVEwusmwzbtzfvlcqdLspfx3SyPOiDVB+J5nfRYcIvAr4r6Uu+POBwyZCilHogiOKqymYsn6/  
X4UHF7To1PstF+yLacRw7s3/+LpVLpz3kfn1sY5I04ReaVar6plCGwsC2d1ZpdXtTVNxxGsZUo  
2WiEppLMwoW8V2KDFBVRsH+QNZMvhGhqjklRLMUy4RA9oWH11eCwwBB0H+Ldk4QKpZBy9QV21A9T  
MU+KFM8pGdNs0wXsqyZl5xtaTICNmWfiCSzQydpqxahlWKAkKuzXZbZPDIji5C2TKu8Pj5c1IQA  
rtkoTYvSuY4xuO2MP7Ds7b6Ia2ubdI25/u2V5b80ZDwOJR7mMv71FjcR7YZ0vLCCuV9DKZPOOQdC  
x46EW5gQ4pjQNoP+jlLKN15R+H74rCH9tiAy5blliuVemdpeXF89KprcnOLS4fjOJpvd/sW0gd48  
XyhyB852Posz3SzFUYFm0741KcfwsrMvfylJzEeR61WW/Im6LpEUSxTMMhP4Qj5jxyzvXuZgsC6S  
/NzD4eTSYk391dHk0lulMVSGxbVqz1+TY1cDlG7HPJdvHyeDi6t6K1QVYETkHqZu3OfmV700++Vc  
ScKlyucwZa05i2Vq1KmoYCKNxBfFmicgYyMxItaNiB30cX1pL0dikuGRjKpYSp91QEa7vtmWRGv  
Cb1XfBRAemEn0+nwQCA0KkDDBhGYcgo3G7qL3V19sDBJKIzZ86pMTa3QJ2tC0J0FoyGVKutiIBAB  
2Tq4E70oC2EiHbMPDCbLBo9fTbq1SpVYXzuMSGlDdzNBh2yags8mcEmCIUEvZ4kkp0Ijc8A9IsN  
qQ84bhZNh6acZhBqsQ3bddqr10sXbiwbnOU8oLl5QMr+cRYG3e3xiv7DsZTgvA9z/pMJrBQAcDI1  
codgUGyTb2BjJ3xooTRYi4VVC0wligMs3ASNiaTsMI79sJkEsxxrjXmhfX742H3/xn2Ox2M0YWRM  
ry8Z1Cx4JGfK9GfTYvUC0ZCGDadppkWiaZfU835KyFjSnqIPJL+KYkCQKbxu1N2B1e3izlvzdSMK  
0jWUs7dkiHneRwKpgLcT3UPVFeYMY8KD6q5kIS3MFFgfhSf5Kupyc0wApjFMxoapbGY0swazSmz4  
pPH/KazrZKno4jGz1/bWKfNrR2q1hYlPwWXMNICFIRwBJybIsqplktULBWE7BteFe2biQwcpJSvL  
tI4GMim0Vg5RpVKgVZWVvhcQfAtscXJe0GJhr3YQBTPc4KAei4nTCGnmXCSR5MQf/dcbeFAki8U7  
Hy+OGeb5jz/+dlxGJnJoGlogvA9z/pMZqrBaKBrSLSr8WY30sWNKU0oG4EZx5BvzYocnpb4+6p60  
Db/9g5e1acchxNXMz3B2/m3hVHgJQZDA4um3qgKq3+vH90FC+eoeuIkOyp9fgk9UezRGfPdkXvDL  
jAGl8qg9RQ4obRxYCI2hrMNva1VPMUZ53i8wUD5fAqsYM9lujmZDSXMiNqKlTQwdHtIHHReFnCa4  
qO7grFVvatUVXYz6fIqAxd+KOCWU8EKyWvK0mYXEJglM66ymVCiYcOWOGB8DrzD3ffdL1BAGCB4j  
0F61t5aJdNvCAAC00M1s6rYIKwiRwgTgr122ylyGz4bs0nlWkVC24Bz1jx7vWLjABvuFm+oHbp86  
QJ1W77Q8CDMJnamwEhjCMLPVxVKy8xUqJxEcgc5ZDby+Tx/5MC0rVze852CmSYub3DWmdXt5OD+/  
YkLeco9Y30mol9DdvApWPuOux7AHy1681OkqJRNjSROEgiKF9kwFnh51jNYF9EZflzm5+zAKL/5p  
/7H6K9+4pu/2/L8ZWeSvCKJogoWth1P2GBRmK6aWONrj16NRUKJV0dTVWZSNQZwesV86LXBmMi  
QILFLtQWGRp8qTQfFUMigY8UoAxtwk5hXUOKRsiszjwYx/Hq5uUDi5KL8Xgm0QvNUbos1q5bjwF6  
jcUtDDNnoiRlWbI4SqooSXCv+xqUPLJ/HjyqasKcL4G7tgiUyx+EstKLGSJ3hSIwV14YBBY/fH  
xVg/G7U2yCTjR7Keb7vcrjbFLig4ZSptXGRQ+EJlfiLHI4WqN/bJL9Yo+E4Ir9iUzTYIc+v0JFjJ  
1W4HisaHHjZOAvltaZQ00zkvmLjFHOacEBNhvW8n//SmeF5NuVcw8q5bs71PGs84vw2aBlpWDe6/  
QvZnrE+A4bKoY9SZ9Pom5tuvD771KcfTFUoKugigxeQdE6kisnxL9sqG2p2gL9vGNL7oMv8WEU3Q  
Ru2rOA3v/f98d//4jvfbNvRbY7jvCxG/qhnMhVNBv2WrTV2qIc51qmgN/VEHyiih/K27qeYvyX8  
DGTI1MMVE1iikeOxwGZiB+OvWXCxmrfqMAV9l0XG/LqBmGucEmgTxRsFHDk7Mxdnlz2NmSAXUlp  
2Eo8IVrU+Kwn8M0DxsLFjome2SYHkCLTBuipXAi8JxCb3qFoU7vr7ZVyROlK6tbVbfc8QkKY/bMk  
x5Z1ZrM5eL+otpaZyMtcuiL90I118gCqtTrlGtdpoVmhrBml6hczLORNgAVpDEbvm15Ir2R9/NS1

cZGMr94gNobF+TzeRjCdZOpTKN9I1EUtHM59AYM0cLUEiKPGD3dspzHxRgQaudGnM7XS8awH80Iw  
veM9eImXEXASS2oRAimjVkh54aTV2WqjyceK4WxarZ4/Ae0+YJpGEdVzEqP8zkuWqYZqXOk9A0//  
v7Zzvaf/Nb8Qf/wzu/wvXy9zgZvQDvF2OQnI2qUqnQ5c11Wlo5QJ7pSENDMkZDgHnC6YQdHyRsG  
RTNcX5eaKN213YubtDahbNUK9apXKiK2jg4IKAGgLYJQB4XMgtB/aqKKslwAvH0I0pEWqxXqXOn  
q2stG84vOaETnRuYinqBDQOodzkTSW/uI+vr8jPmw7Ik+4Rq01FcQ7rPHhagDLsGZULvk/RtwVkk  
Y22zXn/gw88wBtXoIbRER2gIMaGkC/4nNPNKefZki97HKaif5pEQ7rqyGGan18W0eTJJOFUoq3mh  
2NsPNACSi3dEiKXBkYHCFahSEA4LahMwueKXH/icwZB2zovl8mIleQdh0IxfE5cI9570o8t5Akk  
3FsFaqpW6yknNfwRm3w+Tp7xvp0Ht1OS8klxPFu9TWbAnCm7HxXeGEs0oTXjkFzvCavgsHygr2fv  
54H7S2mPFLdXvns4zX/128lf/3Tb39xEazvdDzvxXmnIrlXkRfpluYWtdrbrtDi/T6F+NBBDATOmN  
KF8bezp7FEoMo8oQxerTVrktZ6vzEnYF29e4vAlIYsNw7UyYUmwDTWHakMdYMJhn1fgsLjAz/cVp  
y4bqzu3rHijABaAZgyp9o1Ra1BSKFDYvzkJpK+YKcxolrflL8DgKU+YUB6VYH0gzXKhZV30fLYWCQ  
hsG9+e2j/8lhYHKU0v1fTQa7rABJfJtNwVOOSCX71FzYZk9LW8q7NurzcnNurc/RcNCjcRsshyFlu  
x3J2UvulScwdTR2lmlenM/TdCXlhlCzlGzkSxiKM71/KIQhSecCiD3nIiItCnQZR5085UF3gww8  
6ZRQy084Hs9tBw3cXKSx2Zxb5BZtrtnrE/nMRwM6R8OnGdXdnKepC0kYEDTrGWZeYR/wx6STrOxP  
paw+0KoBu+LvBZV47/+mX9Fb3jPf33Smd/w7343+fOfEouNwbj3p26ce7Pr5UxoiKJV8/iZJ6heb  
/Ki9AjJsJHTQh8F/ICZWAd8Too0DFGB5+SpcfAoh6dzNN7aYENiTwwtGDY29BkB28uGfXL9ApkFD  
u843Iu3L7GHKZBdrJDNxt1sLpJdW6D4ifugPkCcQHE4WCCjVKC0kFd6rdYiG2tBGASnE0GJsDzE4  
sGlqAVQRqqkM+Ta4VFRqBKvmgpiMUvURnb6zCP0xBNPCPughzxy3BLuJ3g95NLNWoXm2EiLeU9sP  
s/hcLm5T3qb7a1N2tle5/cHGV1HZlsdYYtgYyzVqN3uSP5pc2oh/EtCajdr6u+OJeARIMwCL59x1  
GJMghE/ryhDA5DtsH0lBDLu80eMTJt3iTAIdyyrtBWzd7XtRpaBnYLTBxCE7xnr056zfk6Fm90vs  
7lWFJcsXUZJPDxbSnfx4uvncnnlVQTgrw3zswxlennDe/979qvffvNb5haqv8HO+Dtsjm/z7L2gp  
H5p9QIdOXhMwQM1cTZG3UgDhSyINoP825rIVI/jIegdsTfSkFercxjXFfxsHPRULZcNL1ebF4+Z9  
DbJqJfFs4oGS7/HMZ5DRr9L0fYar78BmWzAKYD7RTbsakExKYP0JG8UNoR+EKKrdo4tcCsF5ECPV  
QbdrxDBUqp3maoAq3smLaf+aEB3c64Ko2+wUSZ8HuCCi3xtjVqTFpYWqZRH9zihYrEoEpsQl+7tr  
NM259ZQns+AEmGDAC+VoLLQzuGIAIYHGEPs98YCxjYvYqQx2vFPxEsWygdrLGHf8N/fM8rGJM05  
LTEgapgZoch+HL47+ZGHOWo/ULxLEcgGxSHEUC6KdpDU4LwPWN9RgzlymLirprck77XnjZNs03+o  
yH0nfDi7DiOmxpaeEYd6YyA+/Md7/rPn8x++R03f8/S8vyaOez9WNGumVhwp86coqWFA5yv2WpTk  
NIpQVAdmZ1UURNF7SIUppYwLhhZj/PZNtunSelgJNXjmoOeJeOFzqGlbURkNw/QBLOeoyF7Walzo  
hxRwoZiLy6zYZYp4felSnnO39h7QCJRCIZDATAAAG/4saCgpDdLut0jfEu4TlvDqLUE5HTXS1ONR  
kyFO+EnBuz5BA/b28Jg5zhNtK6X9izWq1RdkOF9U1vmXeZ9zcI4MeulVNiSbtKCKNjGEH465OCKZQ  
qlEo1FfvKXNoe3mliqHsgUa7/Q4rG0qKUnMu05Q1BurMuf2tn6pzuFxpnbvZuabvoD8R6GTDfodz  
h8chOfsSc0tst1TruNvm2Y0icddAKEzIQgv79szlqe1wGQ5T1I732WImDrXJ3vdKbEDP389y7LP0  
apVrZ3X/tvf+Uff+/t/55PJb3znK3/adtwemfYvYIPQhB7OPffeRjff9DVSzRTdc6EzgidjAxAwf  
aJyRkcBHsxMqXtD6dGucTjq84IfsnkFoeCJ47CjWFO2z8vFO+yBMAfjAo4Hi+HcNTMDyspFqYaar  
i3AfeUVDen3GloAy/R8AUXEWi0caQ4nxQmcaSoXVTDXFWA8B51T3jxQAV9dPUutrXUqCoNGRziW9  
++/WmCRCHJsEcebySo8IIRMmxv0JgzgU57nSbg7w5j0REqeKCx4Qe/vtCYo9G4L/WsMhuvbXo0g  
UhWash8MQAcaM1gsCEIJxK+4/Z5ji/EMi6KAbad8TWljuuY41EABXvufHYMO1ThVltft6L+qadS  
4NBN/NKjWxKEL5nrE9rNfizx+Cyf4yPVEu4fNZryPiSem7f/b5b4//0PV/9Hx996P6ak8v9SJPYz  
trWNp099ygdPnxMeJuUjkwqvSFDqtGZnmLJZtoyJsbmIFRlW6JdahQ4fE0KZMmstxpbS1Gk0uz0A  
rh3ldJ4xsaKcNkUP15TIY2EwV+Rk0nBC+0ZiSmxtYNcNSQFrdRarNOBJLIY5qwgJ2BGtJnYUDutD  
WqdfZg9Z1HGyxb37SeH/z0ebFDY3SivV6VSc14QWONBX4Ac/cFEvDDCZIO9HwptyHs9NyeFI/R80  
YPNFRZoPBxQsVynHuex6XBCIUjPefOAYDYG2NNI6eKAdQKbgzD88+snjmVYqZ5HxZKAs9TEchoZ0  
WTT8bw7eUM7xTlwEI87KQSwzFx5RhD+rF3XX54Z6xdvY7sh8T9krEQr4HU//Ltf1Pm+4z9+dDK/s  
PQzvI7+YPPSw5Fh5ujcuXPUGwwo0WGL4IEF25vNSNmmcolixFrdTSQZzUTYU5y8Q041T16jxh63Q  
naRF2genpQ9ZyFHFhu06S15jV2uYlNhgDXtK0m2kUwEyd2KNVuGIi83Zvqjmv3B1BX1qfBVR ejIU  
b0drj4hQwVoL+WqdYkcoI+KcbjmvuuOPLcP2ScNgURiQ9vaOE/tlgU2Qs7DDYfPo/RqOYcUrLKLc  
JfzcKCRiujJFssiGRKzQYF+ZczhPs4DJYFwoobGE7SPMGXKiMIQAw4IhYT6BRSSVSZqao24n4bjn  
uW693nl+duyN4yE/bN8TgznVKmCcIlbN4z1ueCiV/hWac4Xix45Fi3/uYPfNHne/uvfDCYm5v/w  
UNHr729u3U6GfGCFtBu2k0SYT4SeFzSWvQgPsNpWnNf6LCESSzuwLnlTJwnc612mwoeafefh76M  
DnN10+KISivGKvQFjBDzUBIQkyWzeps8GzTudup5k02ZYRM0E/dVVtP9OQQBr1jzjsNr0hesUo5f  
gxa29Tt7ojHxchafzimdm9IF1c36NyZs3SZo4txyFuQU1N6NuyJS8U639+a6Nlg0Ait3UJtCUgyg  
RGOo4TP2RbxKxgseqqZSFdkwt0cTgK+l1DyBTNwPYAMAx4lgFMG/Y7VjyJzGp9MSiUyk9YpcUPc  
+TySHX/NUGURnzj7ewKgnACQfiesT7HjNbnPMkrFNiT5bUxZXTLb3z/F3+D06BTRtbeubDv2M7W+  
nkKOOS8685PsMFGSr/0CoCB4IZFqNmZfBOMqRZqpom79XNBx4lwlgIKyMlJYcmBsQqxk6NG5TSAQ  
dgTxTLVYLhsRsaTk3b11RXo30in4lacT7JHB9AjBmRPG2oEkeL+tryP4HEXaQMxav6vvHCYw/CKM  
Or3OE9sdXu0021Tqzdgj5sJVzEZEaFSbkPYOQtF0ycYD4SdcAJML98FqB9sba1Rt901DNFgaAktR  
SIYMBV84+eiAgwjQ67d6WxJdp3zy2yoeXh3Y9jaMPH7eqMZNfYOXHJLcx/N4uD22M534vbF9HMQh

PPm4+8Z63PFSIH99fIFnTd+FnrISzje9ksfyZKwfbZWrf9Rc24xbm+e5hzSpwc+czuNACAxGdZkJPuOROCaVFvEpkyt5Wpoik9DK83BEwtPcCSFJSERhzq7o2ZO1aSOfrfJV1SxFcWMYi40dDEFxrmraDfNC5S2bKbhh4meFEowABD0JQw1tASIYZd4E2IDrClQfeUE9VqXKACOMWNUGM7yvqPU4Lx1zK8L2TD8AueX7ImrjTpVKzUJoXGZBQ6BPc8RoH6vPxKh6gHnrJDYxJW5/Lt8zqX5uQXK511R+XN0bxE6fCoAq8kRSCH+9KYW4pKtfnVQqX2Cf6gH+qPovMb5x+PFUG4S+2NU6qfLLIih1I32DPWZ7+hIjd1BAH0uetuX/ndv/wlnbvW3M+p2XCzXJ3LsLgS15RRvFOPPkATUp4wUYIu4lmV70OVVUmtuQqNVVRndRFFsUukqncLI9f0p8pYU6XjCg5zGZHB/ZMnujQu50yzma6NDMNrzqh41qGJhZ9YQPrRSKnaSftHLWwYbSDiUgnN7b9GcMYLh17I3nKb8808G6BFBC6zF+bn6cC+FZrjXDskE3yensz/5iyDCp5BKytLVODn5zjPTPhix+OAPR5U01NBWLkyaJ6IfEg+n6Pmwj5+fUKVEufuSBNE9Q4RvEvD7hZxPpoWCvmwUCysufnix51C/c/4nA+yoY6RDYTD7R1BOFQAQWbacTbnrE+OyzyCvjhDH8oORua4qgmYqTqc9Wf6B9Cor6g4473/SB1uh2q1eY8jolNz4H6WUZV+UPU3Vmny+eeoDBT09zGFWHQVFTrypAVA9YZKaHhKTWoCVoYyEPCoKdq5YKMcjR7g2JS1N6k5KxTy1DnK+jb0q8CrhaGSBIDfmKRYy9Vw6jOVSMRGv6p4H2mGxsGA/ZoA1o+8VLKlascjpucP+dp/5EbqFYwqVLik5sEZCdD8j001dmw6rUV8dz97VMCOVxZXqGy77DHLXEYHAIEMED15JNqskeTjwDgj7B1wnowUIWdhE8hTpcI5s8v/Zm/tuJ6Q1zm2m/E1BL1c7oKbK/ydnSu+f9jeuvfcQ5/qFQqVZGm5nkGqM0Lo/SSC8GdGd/V53bp5zf/9n5/58rI+Tj/2qGlb1tIkToqVQRHTCoaJBeh7c6rIhZvDm5tiaLuDi/cCWnokGJD1BQNivxMkZTHbJgiXcyLGARfBMJuDkGtxJpJcohtC6TR01QvjrSHppqrQD4Judi08izeNhUDn0LOCjdaIgz+hozDGRIBWBHJ6b2qIfaYF3nz+MvIbR6kkMNxG4DBUpUyTiUmwy4NO5dkHE4KUJzHjgdtGuycYsPNU6M8T8VKTYyabHAIZxw5jAJpDnngOmYPHuvidQJQBV+OX5mXgXTPL1B14YAQpTU5x071h8I6mUZBWihWB36hyPlG91HDzf8Nh94PbF1+olMsN5NGfpytb3TIYcP8un/7vufUurZp7/gnHb/67TcDuTT7/vd+4HXC1geeJtMwXMD2XsTmcFM8GW8U8qVxqdQUWYiw35JV6JUq1GcvmzQPUEfxT5IHPHnQugjUz1KjbzJrqhgvUEBStElq8FzC4ikAHx4QEz78WoR4U6Zk8aCkhmUAvpBRN4TKgu3V8heAQKLgA+EpeBcIbkFxINHeV0SUVThucr6XssGgGJRbPET5xRM0AfqIc8+QnwNCN1SEcYUF9Dx76+LF4nEoBTP0M/N8fcVSkT8ne12nzF5tItuMi49qpZLzom6QggEjA/dUQoA+5K2mTNqMh33y+fe50hyZbiDAjmG31bj5cpfv/X382o+k5H6stXb+DH++XjzYiXP5Uma6E7JjVU1/rh17xvolHr/2Ha9gW7IyP5czfvf7XpOx9wTQ3/Rz3iHTtBqmaXyGw80JG9Y856n7eHEUHDmcoZI6JkXwhcGAzto5MtwCtZ64lzqtTfZCFcHOSidfE5dhgU9rXTL5I2LHqYymiYLPLA0RhGaZgi9mGMBGg1JXdxV88QR2c8xuo4iGdqtPcPZXhuARhgIkEGseJ3NK8h0rXWjN0igCxnYdxYM3iBqciRE9pBKZTxMTraJMzpsaiuwTvlngc3WE1T+fQ1W3Qjbnno6pQB82rh19X/bIkr7ztWBGmLc7QSGV/IKIh1IHHaUxxDlmDhsTwuRSglw/SdMwaNmefzfvcH8Txsmdw53Vi5zW9NPJKC6Wylk/7ASBW9DnNKR5YM9Yny9HoVAAdaVl21aEDTWxxZtaZc6nvp5z4DGvUfYoga84E+zN3o0i603OAXfLTgOoclByUhyMJbvb557hNZP3S1h4Lmz91KZw8irT1xLOUyTYIibQ01pp6SZ8qam4k8SJXZACA31Q4U0LAP+OFMF4EQhoPB76NsolJPaKIB8gjFJlpugf8rvwfmeOdGM9LoKreQcDcmHxXmDWSIcCbFZ5dirycT4IDnkRCOZqQVLSzroCP7Cinjt8DP+bVURo3mgYalJaJ2ELcEde+WmfKaYowJo8YAWxiqUtLCXIVM3fs6V7/P82ow/HHrNjtDYOJlFqhu8GWS9nbWe7Tr3WYbxV3ye06PJeJ13rGE47IVRMEh2Wp0MDIY2581uYtCL3/buPWN9vhzv+OW/F4KWD/z4WwccmqamojHNceibY4+6wY85/v2N/HIQF00ly9Kh49qWUa5SORyLUPC+F76Kzt/9Edo69WkxstbWKgVsnI/ffSeHkhYdv+ZaNYEzgbHEiv1+qjmjCc4weSIUMJYKf1GwgaSEcCvp50oLB7qswmOckWEooIfQ1oljIRizwkAxTZjWfQRvuqBkuZLLilgUe90Je9XyNV9NdrkuBRoT0y+OK5Q0JlupxZGCC2ecIBw3ZWA85fxz1BpQa2eN6nNNzj2XObJQGwJ0ZgxMFMGLs8fdWb+Aa0rSycTw2IBz1eWk2Fgiyy9FKX/IeNxLeUOMOVqJQW8a9trpsLV5ju/P33nF/F38nA3+XX8y6Af8mZJyuZoZk4wmHIp9WVYzeA5ueb2jPWfELzpx/97+qGfe8cU/VPkBWYCunJvDeMhXvZtNpj95bL3rsEw+TM373+f43ZUHgkdUg7N4JEgJwmmxPGYw0uvSftbW3Rgkb1Pqc4hXpm90UiG01MJSUkXmLxFPiJUocJbqNsxnNumWoEc4HRSbY00VbKNMusjPVoOdcEomChhrSzdpcAR52pau7U1/t2YFztY8es3voncckOTi+fEQZuJjBeItGPmJbIB2K4rda6EvWD0c8grDmjInrLV7cgmsW/ffo4c0EcZ06i3TZWlw+TGUVyozMW+4fQ51045hWKUKzUgMz3meGKITy/fdjppbo8DMHm7Lli9+TTRJS+Xvy1Nk9UsMwaA01kcG2fJJAUniqcoE4rRH+ZQvXrwuj1jfb4epqWkINkkriGRj8QwqnHKEHp8qvIi+s5JYt7GEZwD60Bo2KJSZR+1L5/mRT1SzIC8uGEoYRgJeL3LoSSQSRF/parINudshkyaBYrU2xSiFYXpmIDhXrViDC0tiT4pwmEC0J0UUAL5HSq9ZqYEnxQyihTbfbqpy4Uy/XgEqrfNbCAWx7s4ZarZ8W8kuNhTFi63CU7wPvPAUBmk5HkfnlCPptCGJYTsPhXBVoh8mD31uVtofEfeklumiY1ldfTfninHjiUn0+iOP4Qntr+x7TdB5PI2ene/GHESpDnBtHSTwJMa/ICcgkisYTKfYndmYkrpvrQ6IzHLAHHBdHXr6aGOKktf2FFCircQWudCt5e+bzHX/eOPWN9vh23/Op30Ve+6zcVPWiagg9kkRfsg7xq7zGUqvoil5R3JUKUh4PRb2dp+uo856mgOj1w8tW09sS9wiMk3MUuqEMHnPO1JBxEUUXCTzY2UZbDYDUBg5Mrc+6nRKSIBWGUstNSBOAa2zt1JoQRgWM3m6hZUglzpbcb6r6bWm8qVhBF+anq9aICHUtlmDeP9jZ1t05T8+VvIa86TwaulbQ1JFLxL5muqdgSgU22YvG4qCinXkHENVEiy0UNqvP77Fx4nPZF9WK6/daPUXNhhYzL25SNO3TVVSfi0vyBbm3u4K1G5rw/TuPHJ6NO6Nq2kbB9Go5t8fWY0XhgXNhPLJsJ4p3E8csxb0QBb2RxlDhy04TM1cU31YMEbMBb9xeVzWC5zC0YM9Y/wmHGCpKMHEI0rWEd/w1NoG2oeBDL+f19PpoEtzJK+RTbBiv8fKlw/3tS4Jz9SocRrKXqTSXlbGmHEpCSNnwBX20s7PDax3D2HLS2rBshAiZiW4xXZu9V04EkoU61DRV+8ZQquGmxhaL8U5bPYziUzLT3SEFGckTfO9Uh



HgirVecUzC3/Bh12wLJm3/5vyB/8ShZecVYL+fLtBgVWBPZLYPFAf9GFRqap6DXlmq1o3q2js8hN  
29AcXNA5y6e47w5T/1BSMXAoE/f+RBdbEXx15eLW/sOXX9Xrdj4zKS31RuvBWYoscmhhTHsdy0zX  
zKTZGIYpmewgabxZJjG8U7iF6tpqbmUublcwr9LMSOcqy1yDm9S69LjEuYDPQYe4z1jfr4fnOvx/  
xMrTZIz4BhKH30tL+VjbAnv40UN+bYF/tnBvO/fOB70ZBh8k70q2h/VwlHe8XNksetx2hzy2gp9B  
CB8v9eiuf11Dn0jJVIMCigMMgIZ3EAXS6A9AughNHCsnPQoZZYOXr+HNMrJEMHnDNIg8KSgMEnVG  
F6WTTfckaZKJSLaJWHI788edfMCzd3wtVQ8cC050PMxNdk4zmtobDEpqtUpZHKqLGdAaZ2fn6RsK  
MGYco4r86jgO+52+2LQnKZTHtIlqUUPPnCeVhaao6Xl141tueXHUefxTiVeoYXgdu5BRKFWi7fULq  
IpR008Y2GiK84eyUn0hW7j6pZxNWFky6mZoNyGHH7cvci7co/bqOel709Um/+zynrE+n4/XvueP6  
a9//J8laRq32WBzvP4e4B//Ha/sBi/cBTbaY4Vy/WXxJKhE4ZC8POduoza5nk++75NTniOHFlh2  
yXAEaANSyZw2JNQc36/RjPQbhhraS+JdoqMzk2ndFSLJpO5VYAbLCXbiNQWgHet/mbFqoWjRKgiA  
dyrTSeS2VDIUYC0bNjZpMrxm6h8/EzywUlKkgijqDvqHF18tJaCNKW7pKCICIUINSwEPTCoKyD4Hr  
8znh7LcVmdEo9FIKEcXV5Zp36Gj106NacjBRKu15jUbB80APSUUn7rbWXX5CHhlqVCZz1L2qqUOv  
V0ASmpL4FXKE4ejJ/DYhq31ijobstsLCrXUEbPl5QsBiKGPWP9Mj1WP/TrhoShpk9LX/fOzws7Z  
GNBoArGLrARC0d9Rkad1/Y+1y98Je/5L+Ycs4TQUiQdeHE1lw7LLKjNnoaCkTailCpbZK6tiiFib  
CzkXE6mUrKpUpuaqMk0telUWEoKQvC0GBQwzNlsqtKj0bKMgt1jT4fvoXeTaIEqkfFIBDAfYT503  
KMxbxSlw9dT84avIp8XOhgKDeEBTmgGk8Z7pkpXlrSXFwk10Jec2ZI+sCV9XSdXpMm4z7m5I2p0j  
546S9fe8BIKLxKtbe6wFx9Qr9W2W61C/fy5i8cvXN6u3fCG7w70/v2vpSg+cawrfVLk562trYxDX  
fa4Y9p89A4JcZEvWxLquvy5MAA/5Mi5xAZbpXx+IrjjoL9JX/Gu/7RnrF9ux8aH/z9zmm6Zhf3XZ  
bml67MZhcPnOV7/Yx+Q33/gPa9RbIhp1uR/XJ0rVb7B93IvTJK00u/36zCqUnWFXI6OK80VDi+vp  
ygIBa0D9v5CoSDMCqDyNMTLjYSD19AiyKJKL19V3xOsE45AEWOpDqdRptTqALQ3FAIKHMDyb4AhE  
pWzQZ0DbPUYgheBKzY6LHBQeI45BC/ML1HzRa8lrzrHUBm3q3dDioFCpOtI0cqkHOaCoCabylySU  
nSToDox1Lgh4H2mmkvd6bQ4YligztYWDQcBtfnhr94FJP92Ln1+srC3M2FSuW+n3/393zsR3/t9  
wfVan1275eez12HPbN88tG556+MzVt+z41GnWJx3/VO9cTXkVdbzvJzB/9RMP8Hf+Ytlu24gCFav  
KBfzGHf1+a83JHMcon269KyosI0ObeMafneJvQ5/EIZ6XJyOfKg/cJhHUBOLm+CZV8ZJ8JeFJowW  
WNopTYBLYBBwnSF7HvaF82m2jSpogeVOhKYKOKAPfLESM9BIYOcVBgVwKkbBILDDQZ9moz4Kz+cY  
oWan75R6E5RyEK/1TSsKyhvMlWFBveRJoarA5syvRqKEExk9XnAbA05oOnmZbrHcPD18/32ixN5oV  
mlhgb11EBNvZOWF89Tr9YoPn738wq12701uoXLyl977ntzeqtzrJ/zSacy87X5uLezi3q7mx2H  
7u9axRKEY1Dq1731TODPffR3zNouG0EiZFhkgR9SNfLgyZwHzuca8vF6lvZWo511jfeWT3zCohFA  
erWKC/QsZe/kexiTYHuEwVe8HyMzuXpoUceY48HsUJHIHUp2PLZC6c2DEDTlwAza6sWS0bagAXDq  
9TgIg5J4U219ymyHyga4bmReFQwKoAkGxXamHNlGdJQJUXPF9XexZvfRIWF/WRxTi158Yw3LrvCY  
PXXRCmgpzE2gJHksdDugaGi56qEhQyFvuINznVlnWqNOoWpSQ/ddy8dvvoE+dW6fM6V/fto4zLZ6  
+vrjVK1/Kq11a32cHxx9NPv+ZFH/t3P/Ptwz1j3jicdiZnaWX/zSH7pxMnx6v2XHC9/2rTs11NdH  
m3f+T+SzM3x2owtGq77RnE5c5KYHdbET6OwZHvewXIS3hhPxq+IErORpYa5sXrqFaAoWd5/FTWWD  
tHSiZeTC35bNkQYXTyBuvhYPJhlj+ns2XNUsByqVEpKc2am5apY723LevIsrqXlBQyF90WRyBQK0  
UjA+1mi9FYT0V01BHSfSplJvTfU3WCsMRurxbnz0ivfTIXlw8pQVSVJGyYKSaaIP8WpWguDNzycD  
quDdynKdFEJm4dq10ixC31fCMzwdzxqVvp/gfOUr6coxfddDOH+QH12y01BXv+AtVqNb4uj2+yf  
SBOjK/v9frx3bff+sfvevv/9uj3/psfCK5+wcuyPWPdO9QNqSxbw3N31w3Le2nm1b56tPHIRWPr9  
GZmmZ1kPB7Z9f3JaOOb1dXCkY28bPJUGnG4VE7ncyn8bjqeIUR+5VKMhoeGAWHHbj8qFr6cC1L  
6PqynFyK/OC6snCUKhS5D0BfnBNXvBbVPBdJZFR8MXrZtmuELOpVegsQ8+3ZpqqVED2libezqSZC  
14iwBNJeJOM+WoiI3DITxNQusR8DZESc0ZOof/S11C+uV8KQ8bMUBXCSbSpDKViB5JUKJulwpA4H  
atTo3rwoMBJC80MMNg2JwxX16c52i+6570GCRm+5i5doa32LvuWt30gPP36Gw+AJ9dodSngDQT90M  
Apyw2H/0Hg8/udJZnj2TN/8ee//9v3ffVr1/ov+7o3p3vGuneAlc/KH3xJPDz3aT8Lei/OLoumJ  
B4G6bgPGNwk3j4H+J7L7sAJglEuisKCW2i4Wza6nH9ytupm496ON+KwGIwLh1/wNbRw9CQVa3PkN  
fdJrpYihMW85hScICrmE9FirRR9qjbmKRx22DjnFNUK/94SIWXQmijVNLOGyxULBAwkIcV6aAOYk  
MYzMS41/2FQBGhhpMbnEkE+ZZILq3NkVDv5SiosHiQ757NX9ZRXh/xkpuZe1Z6h2kMzoS8YPopUu  
h0iU7G2GgYQwSqwVGhqTyizf/wjf0OD0QcOaDa/DI9+ugj9MlP30+Hjx4S/uBL5y9JfxS5dMgWz  
e9TNCzrmMcJPHvsxuOnzvllp/V7n7zr9ls2v+fHfiHaM9bn+THkhTy+/HjVzcw0SiY5Dh3RoLNS0  
0kyEDo4E4L4PCxQGOtQPuwlJ9EFyy8to98nMhGM4VWLfouc9CyZHiHxwWmjPWRojOn412+X9G5UBQvC  
YwUrwPnEED9/PzqsRdR7dhJcgtVxZYoo3QWv1wZp/RkMz3ELiF5osNhZQ11qCKTDMnbirFCzoGCm  
u3I7zGocNutt9L6xjY/RSGnmgvztLq+IwiuCacKCvY75T7sbK1LAatSq/IGZuQm4+EBzoULk3xuc  
RyZh9fWtz/8Q9/+zU/kipXBT/3K+9I9Y32eHvuuf2Xv4UuPrmRucb9tgDLItCb9TYsc3zITXyhAY  
3soizpXnuevKreEYpnBeahTnKdcfUVYc410TA7nfoXGPlm4MMY4UYUWtFmMSAs8papIBJTRzuZFK  
uVrYpSxNmYYLJXYMuKWCjOhwuEq1TclqC2OT7opKAgFangdfVSYdaLaKKIS16oh9NhUPN/FA8epe  
v3L+LM0RL9VEEemYnxBzjs12ClvVapJ1oBLNrR6HNxuqnvAln6eMnglgwxitQsP30/3PXS06nOLg  
o6aBCEVSxVau3yBdh65mnrDCY3PrYkOK/rKIEzLgVoVbBB85MvlhVKpVBG4cpoFB4PJpMPuv3g7  
v/9m9681mw2g//3N3873TPW5/Bxx60fdzfW125u72xcvbF6+aQRDQ+P+v3FlYVmvUCjw4eufQE9e

M+dyaC7nRw9fGRSn9vX2964NMyGm3T1C15SdvxyEXxDAflkskcFltepz0mPEbmbJ5bEZlFucp5Vl  
H6kV11Qbx5PZNEV68tiqMgJoduaomqreXlBhSkTNJnqh4o6t7D0J5KLokIr8ArO+WAU4FnK7Ex6q  
jA0y7Z2q9jaQITtUAwnU8Yk54u0EtzUSysqF68+L6CHPH+1XV8po2v2fkl91ZmVWruM5ilwRZwoD  
4+815Br1YPxhlJMn07tpBEkLnwZ57uVvSpy1f6FVWrWSlSvlWXYAKgmTBqBrA5s+OgxT/h6JzE4l  
SbUmKvT6uXLlHRTe2trCzCqXLVarXW73aOmad5p2/Yto9Howf/zW79xc/+hA+G7f/aX0j1jfy4cP  
/+TP/qyc2d0v5Hzylfbjv1K27CtixcvCiLnqm0HpDKZdx2oHdGZRx+mhYUla/nAUVBzcc5p+r6XS  
5aP3sx0waSd7TVr7fJ5Cx509eIpgfJhJrJcrVAQxXThiYfpDd/yXVQuL6nBb7+kmRhiUR/32XDFo  
wlWN9H5YboLrM/UgPdU5gKyEKNBn/x8VRFsJwpVBP1WdShvKcrp4GdCzmSPgMM095WqsHi/KeWog  
hKaiWLTv2AFzm3zBWq+7HVUaixK5Zc0Hako2SA8BlDtpnq2FmmZDan6KsJvcCZJ4Up7UkAeETGax  
M3UvV9RqOPvgXu+4+O3y3VN+N7s9EfUZ8951aFl8ngT7LS3pNUEZNYkjGVjCr2EBoMt/r1Lhw4fo  
8sXL/D3AahwcqNxuMB5b4X/REcz79ha2vjrtFocJvj2I98/796y9bivv0w2mTPWJ+Fx1/++Z+6d  
9728bc//oAj38/h1bWupzRb2u0dyv0O3Wu3qFAqU8Qh3aA7oMHwIudIQ15oAdWKPl3LRlWu+OSsD  
q2Vg4es0OzTE4+do94oZW8woXjzHHV2dtgDhHTk+DX08JnzwhD4lre/m5ycCmFdv6ZIrzMVj1YX9  
onor3KyoRiZkrpShk5EpSSn82SWVUSLZWTNUSLFULCKZy0bGCdkHgxe1KaIHqs5UozMZqThhPJ6N  
VurPLLy2PDSCTYM5LxgkGDjn7/xtVRZPCB4WhGGylR7JgXNWaYGyaQCTdNqtNKSyaTyG2H6TBGra  
aGrXUNFJTMv3BYQRQCJN1cv0r1LlLzUnsSH3MeGwfjgcky+V75IM3U+iRISiAw6RW/0BdVo94anK0  
nNCxoa5WFKht83nKRocX/Pr6vycq3hzeGm73b7n4OEjd6Yurj7wg9/+tvUXvOAFwTt+4IezPWN9l  
hw/95PvPXnnJ2/9b3nLOQLDgd5LyAbW667LlP+fX+TwKpYFMg4iWr+8SqNgSI1GQ42EsWfb2dmka  
ukgHTnxYor4tR/86G3UGiF0dGjYb1G1nKeVg0fEYG6/5e/o5le9jr76jf9SGPGB8wUeVzwqvAmv8  
mJzUQwCI2QIIo2pWjiHxjBoVHVNW4HvQQkq+V2c6fz00zOpigkURgevozynMiBD690IO3+WaQyRw  
kQYur1jikSjucv6AAM2VWFr7oZXU+3wCc63vRnIQ1T1PtDKocNpQfE+oYKXK8JNYDZlLnjolcxM1A  
FsNE2hRarSSUpoga5fP+rEPf4S6bJgwPDwXchyCGzYsCYEtDu0dP0+5oiM913KlRpvb2xyx1OWe9  
4c9qmR1QXTlFeedhz96GE1s23SKfiGf4889b3vuifPnz78iHI8/4/n+radPPfqXpKag9oz1mT5+9  
r0/+v23fPhDP8uL3h+nA9q3/yBdPH9JaCrnlpY5VAlojY2zVm9Sayeg9dU1KYQsrxygkMPZSrFIr  
7rpBB3at0zNxxSAXFunj992G40Sn7Z3tqmlvk5Hrz5GhXKDLp89DbJO+o7vftC1ltgJwWhy+VkJC  
8wtFhFaMNLnqxkF0bIQhUZTsShgmBsJJ0JlTM8AfA4j1uXgGauo8s+GhMEq5LRl4coYW6KH7W+K  
jYEkbNACm1kM6A/WjowmJimUpI21U/eTM1jJ2V8DcacCkdxJsLF0ywVea6iGJ8aqqZlgAg7SlV/V  
TR5dNUY92A6iWPKMK/iKsZQAT/z4x+/RRGzaT1XiSrsWGREWlubvNnZ4mEXl+Zo8fhxevThz9DRw  
wdoY3uHOOQVqhgoxUmpIQgmF9HItCMiSVsrpz/2kaaFNvbLfbfzfn9Os5/ildsbm6+/id/8J232  
5b1b+/5hV8P9oz1GTp+9Rd//kc+ffttP9vv9ahYKgmEbjAeUBAOZMokDHPShMHIALMePctaLditw  
Q00PF+iV7zwBF178npZzH/7oY/QPfc+RKXqEnVbpyUH/YqvFZ30P7dWz9D+g0fep9W/+JmGc99BLF  
GW3XeNB198r1gTwQIKnIXBKcnyQKak4NTEDjyve0ZFQmrRedM5TnVum5eX9wMequdWMZ1KQqMoqI  
7WVQUyBv+ZUbiOe6eXatxoQS7ZMQSiJE+b3rVzzUmocfyGlbKjTPcGYKstpXRsRWE4TTSyuK8Cpl  
pZACKzH9Wbq8ShEGUoIy9Bka6IGH0/PX3e2NuJqSd0iK+iB4ev0+bnDTnQCS8XraBgEvHnHLCXN  
8lrnshbaxeomuPH6Vz5y/yz4cUYridFOoLITOY/VHAq9frsvGlouKe2Lxx2Pmc1+Rdpen5pWptX  
v/1URD+H//6n3/9add3f+h97/+Lt1jfrqPP/r9/3LD7R/76E/1Wm0xRoHKcXh3gXPJer0qi3pjY  
4sKnAvhuHzpHP+8yYukRBEbE2opLzh+kE6+6AYhJvurv/0onblwWZTYet0WHTx6XCqxn/nkLTQ/1  
6TXfv0b6YUvfQVNgq6wxMP92Y4vEy8wONsxJaeiNBCzhwwVlcloVqXFws00bYriNqQusJhaOQDYJ  
1PUj0nHr7207rn949rHKdCDGAsowqBxI4ZkkUW7UhsyHqfeTCnLqcqVeHMPZHGIIWb/25VReOSqhr  
2rF6F7PTKDK1EoAqFjFipxNVHMq9xg6qpTtjsGRrvGKI4WlBtzlfl3pRnUZx6duvZ29ImZuObuOe  
aOx9PCgpXRpZfgcI4N8Qef5b1DijTc+fZQq5QL1e21aXmzS/pUFmeXF/Gs44QB7okLx3vYWDbsdd  
S+zWEJph99rMjFsRDHbly9XecMoLi7tW+CN/CVW7L7xbW96/WX2yt/LYfInfv2//GGyZ6z/i4+Pf  
ehDP8a7sBXGCKwQc2iGBQMP2u8Pqc75qON0xRByhZzSNjUED0TVNe1veM3N9KIXv4QefPQx+oM/+  
mPyiw06fOQ450pFETt+5IF75byvfn1L6U3f/DbOkwoyqmZYvhiqWvCWVElR7VRGN1SKZn5JFpbi3  
CXp06ZT1kA9ZWYZecWvIgs/msEK4R2LpZp4VAAfLMvYlXjEaBty3MwRozGMRKrAwgSBB3JGzJDqi  
m9iJpIX+odPUmHfNeTXaqLnKgUh8AojnzXMGUY/1ZIYsr1IIYpmUcNUsEkwxXGmP9+uWLPicjIE+  
yu6qY968t+6o7bZegd90Z6xxNJCiRUR4huszeGbi02BIvzb6vXp+1WQo0KP2p5snNVvj8T4VFyX  
VeOTYtFFTFkOozHjVUqCHyuIKSddg91KBVZJm9zfknNpjhsF/h8LjSt/t/yV+73/LG15zhc367n  
ys88Vt/+P50Zt1fickP9oDvo9l6fce4SS/6kjMAU3iCvE61fusze8Zi0BDrbm3Tt1VfTmHdlzxjTN  
77ha+nYNVfTn/zJn9Dtn7qfcsUS1ZtN+eOfOn2W+p2uLx3fOd30Ute+nJenKEUVWAddha3FmSyL  
BwigljBSC2l5cqeHDhf0pVUpet6hecDgslQRBFGLhK+BYDzN7aFC6kYNjhRWNlIkadV4lGcVhp5  
xUToUAPFYBfxH+1nKMYlRa3Mvn1pcUjVLv6Rgp51UacnxfMprpPILAVq/BaCyidontWCTJ9VTWXS  
mo2V7CFphp2hxfVvE9T8AM4mxCaW6JIZ+kKtko/7r/7PsUNxZsaJ5SSe0aRQjslhv6bhH0Ygcc9  
cQapWWkXRoHaa1ttIQOxrANebZyeaPM5TxBSmXwzBEGoib8XoGEyaat8mbZjCTPTqnb6Uu4nHD+K  
xQ5iZnn98jz+lkKx6M7B4NB71++/mvuYsP9IX6c+50/+UC2Z6xPwXFx9eKBIaj8ELOYNFYpshNY  
PdMJawzJbza2lwXycEjx45RIe9Sndf6a1711bSwvEQf+Nsp0aWtMR04fh0VyhU6d/YSrV6+KAZ7Z  
KVB7/qBn6R9R46qciwbnyCBrtCQy7RGapoEkre6BZ9cNnpUWg14p0gpu6XZVLfTt2qUCfBA04iai

m1/PBoQn4BG/R2prvpYkK5WOEcRB5y8/z97XwJt2VmVuc9w77tvqCGpSqUqc0IIpDKQoAtFUEREY  
6OgtLpaBAUVFIwCYUEnC8WBOLVTu7BFRWjtdliCtiyW7QCiIizkHitzyFippMZXB7rTmfr/vr33f  
86t0L1W9zL0SoUKtap479W9557z7/n7vh2urU/kk1GQCS908bOUEX5+xymydM5F4c8zSa+D4RUrK  
7Kxthpq9vvklHMukIZAipzRCCMYzwOryveg3EiXS+ncSVk0nQAF8ab9b9nv8VcMeyUJZP+Hx7b  
rouGGSms9tKx1VoCiX1METajBj1YlqTeF+ZY4MBglmkFEAFaMDh4EJ10/k0OJahQRyFpYteeY8dY  
kWCPXb/242B6FmgCchnl3CfsxRpASz1lvDCW7KsPj2cplcF4z8SUuXPhkzga8Fwn/r9P/vL5ivG+  
v/46/D+AyD6BNV1hTqHRFfewrlLTs03YpFlbWw7p1BbpN5vkG155qZx0yk75m7/9tNz32H7C2zaGp  
ew/eAdX3W9dWpRveNlL5Ed+9HLO+nTDqeJ2HRmko5mU1DBucQsHBYAk1HxknjRjPfwMQtBYPLXJN  
emDXq9MK6KQVMOopqES9xtebzoakxg+HyIJRkCMmrUupNKUtFRZFGtWURw4PMG54HC2XfwK2XTK2  
WpIKRUXwz0ZMbrh2tdDhvFkOKgnnXYOpU98ZWT0QYnORuvE9uHUpYmfpZzrlpa90PkwW7AIhiabt  
NrD3LKe991Auv2220JNantisUoSguLBqYDXC3AEYIEtK0I2HQog9+RaSstoz9wj4rRM21jNE/t01  
bEkTLWpMUVkVTWTbcBXDKfrIVvp6cy6zFgm4GGRYuD/rkq3hDR/SzEpFzDPi+8ZpuPVN3zHt94Ts  
qV3zs3NP/iRP/94+RVj/b8x1kPLKwUecKORCXFY/CGLpXPTSa2SnOHQDMcjed6Z28kZIS3+zKf/S  
T5344OyHOqiKRslKV0o551+s1x++Y/LhRddom2d4NWxQ7WsFIIdL5XrUkfd4mQLaxUYjTD1BHKdif  
qo7ZcQWHIOWRtX7hEybdLBIORYcaAiIQb2B2UHjzaaxjNAwAepHdDRT2zLhHqJMqNmUedNTnaPw9  
AbbTiEHtRdqZWgw4X0B21NsLgx7TWe64b2P7HtMVg49KWdf9LWq6wSGDlDMkHhJmjhrdVoeZWlQq  
126bGkvFW6AG8ZyZ46KdGcr692QXWDEgp+951+ukbn5BSnXR0p2D5kBI5w1rdiMorNVQ61N/X+OZ  
ACto7001Zni7yEzXqQTVm0s5aFhd5wPnatlAmmTU7sp6bF+ILeX+GpND3RubIAVzR6yXrjOLEH9t  
oR/f3r4870o2xh+72WvXg5//52QHfz3kM2tfeip/rj5irH+H34dObR/pZmWJERcKwgd2rJkrYL6b  
jIeyuKmbQ11iNTjibzwvDPk61/+1WR83HH/XgajjWDAC/OL3D7+zS/7Knnne95rC3vVi4OUjddPO  
rUnoiijnhgDTFzr7gxRLH/VTMIpyvE6qWINuM5g1KaRul3iwVW0QpWLBOWtDglTtI76DQgOZJo2il4  
fKyHN2/19QKJY50OMKpWUmyIQULCaT/g5D+n/L1r5deuHbuoQ11eRVSV3RF1YGF64ZsTDDy8WiNM  
MHJ+oo8cOu/yrkXf40MFjbR2dVJu8DZTIBZxf92CXiTrkMHhS5NPfOo+XVc68En94WS5QDpdhrlC  
n5GUPi4vqMGBKJkDUVwh9EFOT/u5a18jQE2tLmktT9+Hq/H+hZoggu2JAJE/N9w9SW4wkYZVNKBS  
uuUtbAnUFqpUtUqdt7rJ0yp3VFSd7mut2RnsmVaTXeFUuQ3Q2b3s8MQBd70um+/MWRh73/eeS948  
H0/98FnPOo+6zSYhmvr72STj1IhmvauryhWTYub9SFUIGdLSGuX5IstC/LSF58njz26X6677UE5u  
j6WtdWjcurOk+X0k5bkZ/7jT8p7r7pKBgPdPUo9o6o2RftUSdrLWBE6ENVGPbm4SRa3nihLJ24Px  
rpElXc0e9AY6gXjBB+ULBymi6rakA/UmJA+Y70j0vjJxhqVBE1Vw3rDEA0f3nNDiCRzgmVMvVz3r  
3K0mal0KAES7HLqDtVt174q1Mub6BxwENeWn5JssCB7770xvP6qPHrnNfHQw6nhd5YvyGS4IQ/fd  
r0Mg+HWNqo3tXaZReLBt+GO/p1RNY0RDh221OCNXht6YwnO89abbg6JRsrZS2nC2t6UWgjOaJPdE  
5WhU0MFbWfU0NMU2NwGRloYiyHlRW2gBALFD8Ngq6R93cacK94PURXosliOiDpIX0yN51w0Emtao  
N1A3YMjwPUXALNULcySXyvQHct606rePhmNTx+Px69fVW2+7u49tz125U+8/fq7br01+4qx2q/3X  
f60U4vx5J2V3TzLl0SgICzx7c2RegUvDF7pmbtCLbftZLnlrkfk4KF1Cmdv3rwxX3PJOXL1r/yqv  
Owbv9mUFGpiWxsbs6TW8SWyCF1ZpIw9lQCFN0dqJ7mOL8hndVEcurTo0iKS9RcWKb3Zn1tkTQsGT  
hb+XiY9Rn4cFERpHCAN3KWsLx8kXrkPAe6ksfPQ2H3VFE0PJbeNhzp0sOsM2XT6ucodZd04ie7mh  
J2ncUnzt1P04riHRhYOOupdXE8/n2dNu/euW2T18H41RBbfeujTug2pnnKmlvQBtDIx2CDVJHRbs  
+oFI1VLdKZ6W6hXAU/0g1+A6I7UHFU/HE9ay8LcQE7cs1kWF+YYUXsc5ft0/S31RroOOVN1/ujcG  
s0wNBjXjxxsXVWZ2ujKrV6H2VSbUZcdQEeeI3gmIpClTZ8Kx+wycr9bVT6NdNUNBMH+3wQFOAzE  
43U4TNUgQ0nuw7u3feSP/zdD3/2njvvSL5irOHX2urqz42mk3mvMSrz2LjxAIUDEA61hclESdIX7  
94te/bsYbo72hjKQp7KG77nu+Rt118pm7ZsZYEvQ70j+qMKSSiJz02tqAhGvXnLZrkPKwbo1Wij  
7gqIvw8IiciKldGAN8aUlJQ3TI0goAkynocUYyDQU83lrXezXvsYroiA4z38KNfZLcXXdI5pK7WY  
CHg14yHs9BaeaPbdr80pGzKloGjwmwzD/UyJHbTtjPD30P0MmUKjfQKV+yHOnNurhcygiWexv0P7  
pEjTz1G44hjoMZmqK1BCjONpExDo9xLHXs+NIJKU9y6l1LE7c4779Qo5wguOLbMRjel6hWztX2+v  
ik4tx0nniAnbt7EiKoGl+r75ZkR3xv+07xeYfhk/MmZcaZNrjgicyfT9BTt1bR1rGZMZZshiGZpm  
pFZhE6s4VaJtRlFtZfpGz3rcFBcT34PQ73+cDhA1/3Jx/76FXPeWP9we/+zouC0b2ZnrEyLmXqN  
UfwtCFScGFvOAJ79+4NqW105MveQ4d1LaTJFzzvLPmpD/yUvPrbX6c1jBYpOjPkCkXlfdINLqzFj  
1TPQRDBa069e4986INXscva2KLiYTTUgkhn2eltDK0iR6oFmy3QBB6vHqaheiqY2FJjLKPCE06Lk  
Sw/+UUAUy8YXb+nIH9Q69JGjbG2xg8+/9zmbTJ30k4pRhu66mKq2OKFE0/ma0NmtAoRH8aTWz0NU  
AL+3qNsywIJDxkdSV/2P/KALO9/ohVhS7XzzGhVaR2axE5wK9YmndTXIXa0pY73y3y9PPPEkI2Rlh  
xkGg5QSjZ2UTedyEdaPXpWgkDYlZWsA29DyPo6A4IrL3QqSeTetvBHHDtPUpDgnk8TsFFKPsx  
FeNRds0dp85amqsJxHufWXD76jgmJTkHzsJn918VOq1xM0y+LQaCft3r59e9/9m7/wwe3P6QbTx  
urGL0GPL40KayMgdY1/hgeFcQ27nOEBLoQIVYeUanjoGhZtT7xGXnHZ65ge4/A5oF4XIGM4Hx58S  
IGSVGsidF3TxcRj/J4OHh//zeflk/99V+ze7zt139Tfux97wOcgSk1ZU8SleTEvlHU5jPT1ZXZ  
DRaYWSYw9jMK0tpL8YbfNjhoK/ufYz16GJ/SSVPjPkCBQhIrwd9B8I44kfTyziKmAvRHptbWnDFB  
k9FXC5TNjGcbjjPmC2kaRfQ4UWpsorXWXG+qdh2pKv7HriLc8ztp57dRhw07hoQARTs33TqWVeNQ  
DOp5hY7XQZ83IPcdP11wRAzRqMcXW4bfbEZRS6usYEaPff+frrqownGOqfZAUkaltR8Duhyg6UjN  
prJ2zDK12U/GaCTJre5rTo4NiBtURfhm6nOrguSBFKKBtRJQXqivldtvfDWWdRJ3ZFhFWnVMzL+G



22Yhc9YFNsfuOfeTwIA95w01n9/2be8fDwavSbn0qZMpU5s9QM9rCiSCWnSFOnuYE6RQuWqvOVt7  
5Dnv/jlCgDALnCrNcHjhLekTEpFyL2mdEjlbJTwwAMPyKf/8XPY2X/8vAzH6ChqB/K/fvx/yPYd2  
+R73vIj7BLjNRpGma1BJ1h7sbain5j6uX3+HDmpUwVyCFUJtWGEFYgQ9c5BDkh0hgiPD0cCA4B+A  
pJqaieBNDBYYIEYXyg5wPuE+1DbOSmGYZhyo6xpbB51DqUJiPhicBfSaMh2doBYLYe6Je++SYbjuU  
8+9UDOAcl+RVovzWistL8jMrexPUOfQrUWECY6qCD+G+arrRqEEaHu0tY1K2gwj80Sz04ph/6HHF  
Dize0GgxRQlojrFAUdkaUwQfcTEKIf/Kl+epXWtp8E0rKmWT0BD5ZEL3OPLMWXONJVPDb+cJUaWa  
CTuFMREvPk4n9dqBNTB66PlC6/4oR/a9Rsf+9iTzzljnQxHVxP0gJ0d61y16owV8EiZClZ6LMDK6  
KcDOf+S18vC9jPoKSuCy+eoC8R0CQ84qYlnRaSehod7YN8T8sWHHPQHH7xfntj71Nlx/3yNyNKSxG  
0NbhZjoJnCc3w99+A9184nb5NWveY3Mb9psjqPi+onh8iFtjoT61eeAUgWTCwcs7+c8SA0fdh68+  
7qsH3iSGUHu6V9Z2DY3RGBqBvIQ9VJN4QbbdnFOqygeZf+AbO/4YxXvLhgZLffmjJMqhagFw8GcZ  
hXvW5EUMcWEUsVjd98pTz32sLzoFf8uBOcea3EpXtq+2sTxyJqwJ0ixOUQQRaO142233mg/3wL/U  
9/V411hj1geHClqewMqMckawAybAmXOIkkmsc0w4DjaUmK+TTFMzULTVRhghfUdecOMBR30qRkqG  
1d48iRQWJ1KvamC8FFgt8HptSvujJI0rU5snSYcVe6Zg3Wlw9+3rG2s/jbizHPKWL/zm1552XQ8e  
UVBGPsoZQjeTprY+i/qmkgZPCJ8f9eOrfJD73g31z5JqJImNZ8mhuNT1rT79j4u9913nzz2+JOEG  
u4LqeJOnTtlvLEuC8H4oJiwztUSHXFGFblTNyqQvRGcx8/+wq/KZz7zGbn83T8hL7zgXTJcPSqj1  
SMhAx6wI8pIlyh+WEKtiMgO8TCAIIAA6oUTsrrvMXZA0QBKyG0tyHNFLVaSWpfYTBN6+gXXQy6ec  
lZ4reCQ5jfTeEayQPR2pQqltFUGGsntIGGPaoi6dHRav1WI8IWmmCgv4BDXhuvy3/1SfncF26Qt  
19xFXmj5NyaqmHiKXKqMqXs7JY6okIk3vfIg7L3qRVesY5jrmPTJpO2Rm1MMdHRR90VIHnu+OssY  
oW1ZKjZE0gMHJFZ82tqNTO1kBMH8CeM6FnW4pinoZ7OLcPJKEZet+spjbJgk4C0ERuJtYqnGj3rC  
B/tItuIvMpbAkO4ppde8aNvzX7j9z5SPWEdTgcXu0PjLxRo30hDSlQD3Hmpoihhp26TN531fvko  
hd/tWysHJDD990tt91+gzzy+H65+549IWIeUBQMvtDvQJhL1fS2bt7GmnJx6zYOYJ94/AnKZ3KBU  
6oa9ogchM0Bd1qHFDHUT5//wm3yvLP/VnbuOp1REukpRzfACaOTGV5jMjxKg4ChqgGJONZmtHJQ+  
oNeqH0HynoREyRj8LcxgS0sTtK+7Lzk62Tr2bu5QwbGv3Z0f3Au25h+FSuHZX7zyVGLycEIVHugy  
yaJTS6+Pg+yGgicIbKChx7eK7c+uI+/H310rlz+nvfK88/frfe8h7S8ijDFxhc0A93VVwLDPXvui  
eQAysDAnpqkZzS6k8l0o594tY7J/NjTgrcr3XsES0VlZR0iO6N1Y66vhKvxXGmlargHmfUPg5Zw  
2gsOWR2uFy6cU10mh3xzQ0fb7wOVq1NFaOpNIRQsJwoTGYHNEJ3Pl2xOnMuC+Fanh++de9zwli/4  
5Wv/K5xMf0qb2Aofrlpwez0cSGVNHVA3DZ0hX/1135LFucHsm1R5KUvuTSktg/L48H4II9y5jnPI  
xxxGmq8pU3CmenK0cOytDAvCwt9WR1XsveJ/eHhBiNemGNTw+guSksjTC2NhwRe/ROf/JSMKgixZ  
fJ9b3ijzC0msRZOe3PUDC7WlySZSzRVRI0rGt2AgmL6yxFFRSMj/BI80oZLeYqmE2c9nWXhSC9x  
DppvLYsg8UTJAufdf2JR0K9viqDTTtiulqXKr/i6CBqClvKmzLTCwlxGj5jMw119hgq+PLZa2+WA  
rVnKcmuu+1eue/HL5cr3nW5vOrbXqsHFK9j0jQKTvQ6H2tf4Wt33H47a9XSm0BOVESgkmbGK04kS  
FBD9mZGLKnJVZWFprYsHjBCgXGhy53aJgLDJAUAUSu4BSfbG4RY4AxUVI/k+EZcjoOZWaIMJzxHI  
iIziZpXqrLhOzUrwjTYK+kgqzxqp96ctNFQuLYtZT19f/ixNx33o5vv/rZXZ2VZXg3heUTOGvKXu  
qrBW/k8NAZVw+Hg/DX878jRdXnk0Sfl3kcOyJXX38ltayeedCL3x7B5j7WheHDBaBpvfYhYVCz0x  
YN9KhgqSM7zS/McBXmN4i16YmiNgqX494QKBx//07+Uj/3JX8r73nMFa99yqsqEGOsgfexv2hxVI  
dhgsnRpLkRhboRL6ojCUVhPrusqXJs3/JoPNfJg0xbjxVak7yHalCORrB49IJt3ncMGk2oRd5Yd2  
8GqTTeJXXBbwIw0FvNmgeLU3XOfHDyyELM5fMLDRzfk5z/46/IHv/Mh9gLECe3WVIHMC8tsiACE9  
3zgvvvZreVunY4xdqNmTCs76SVnwOHaERkz2+heSbs7lkAGS4cxkkoT5/pW/Dlk00VduQePiCVcc  
xMiLCYBoi9003RABsJh3+eJK1m0tEN9vs1MuuvnwNP2JCp6tNhpRZnxXL7sOTFnnayuv6Eop7vZ6  
TWlv8ogS6ndpMoedy0mXlBbakfPju1rudx5/yPywCP7pDfYyt2n0+16iEBr9Jkgm5993m7ZG1Lfe  
/bcG+rXg7IxmlCtgGwV25qmzZ2MH1sPXGODdpNkQXRHqj4t5Pa7H5QrfrvIKefD+u51WjkmD0z56K  
Px5iMbDfwWAQq51IBBPTMsNHaWHF613E1c7Onh905nPjyB7vk5/IaTeQz04kxVtBjnHNKrl20Jlw  
yFTYfjIS1TBH+D6jeGa3Hj7Hda40wMtXlVfffjP/24vP8975anQoaim+8wOjF5G2aVSXiN1ZA6P  
87X7WGS2gH/dyNq/FrjihMpu79joANL071bXLEpNGa9qim0Np3gYOva561NJHP4vYIRw/ABAjnze  
8w+m7SkAP2tpQJ+Hs9YBe6sSw3HYOAUYt0wktfxuftncSfhrQ51LwNK/t9+/3fT49pYv/0bvj4LH  
vEDUHooO3KaHBOQnaGP31XjOYowQevBXIiIhuPEoqP5hRNCvXpIrrn2dtlz98OS97bIj127KE06C  
jXnU3v3yJ333yvjELFRG28JuTGihQMUEgTvKx1M+aXtYWKapOvE49wRSJgHHntCLv/xd8kn/uzPQ  
4q6SIUQonttTZ/KRQ8tGwAlSbAXcWDkNuhcugb3m/7ORE0HWBJCMQXq61Ou+Abmc4pOtAk9cN7I  
vWvRiZjJKJ6S6rObwv0t/JeCy33n6fHF0ba0Or0aXNOBYAP+IflWUq19y0R97+Iz8sf/8/P6Wjk  
rQ2eKF2q+++942YZV03sGEfQiUcqJyak6Uba3BvMMTIDBYSSgMJwYAASi6sZU03hNvM3IZuoTG7Ve  
xmentLIOeZRQ6USYuIrPZrYZXZD00CcMBvQDrFMADJaTqzVsibZGpleks6ANhzrBOSB+fXVtfnju  
mYNnv5Nk6o+VxsLxF8q3KsON6bM6FkdXx5vLMHx5oJxjYInpmctLS6GFAjLkMbL63I4/AakD80HP  
zjbt59C7whNYW24VCRB4yETi4sDV2k0TRXXxr9oa99aJ+Fns8Y6lujyrk/kw3/wh9TQ/f63/CgPC  
Q6210uoW4EN1jSqXVFBMgJHFD7qqDluAkFg29kX8BAnVDtUAWY4IkZPT3fL0sYQ1TWYah/jh/fs6  
IF7S0JBSKEPHz4ot9z1QBzwR6CB1WOZRXnc0wOHV+UXP/jLsue22+XH3nWFLICaZ3LiX3zwUSPW1  
9wCkHS6vHp9FTvLbUMmPK/+HNNcQEIHg0Gs/dgswggL+2MdgN8kfHa9TCOXc511/w9ExXXOPQ100  
I+GqiJw3NNjCDBGyrqJkMqynvKZJnG7uzewDJTdtJ8/7YoQNLqSRLnK2hDEs+slPf/svZDtBAGm5

7iMrD/93vdk4UO+H1EINR9V6esibi6jAkA9iyZBvYpadH5uTh9eoSBvztyCh52ElHZ+YYFaSmBvA  
NlUWzcZad+BAwfk6NGjbDpBvhShT9MYjXSZTgSUjlUTWCi5eVo3tDIYUGUbW2sAfmSj370T+TWG  
77ALqLXRGLNpTJpjQGCVTzkPpOEs/D6bQeACj0F7Yt1Vx0/mlrql3QgieRy1pXtuGki6ssxvRXJ7  
iOykK6/6U5ZWd1oGSgdp5EYsshXYgDc0YRo/Vef+nu5810/KfsP7ONMGNf+2KMPU0CuMf2J21FO1  
nInpmOMLIkRklcV/5WVNQNu5LGJxQ11cNaoulGzkwpXshGH/3Iaa1srkuMb/oRT1tfpzTiK+KfpO  
7fRsIx1qn+N9WvdmPhbGiOoj5A8M8iyufilpNMx8N5A1dS9IwcP7jpu0+C1tbU3hQ99Lg4X0tKKH  
17TnLLsYFDjnl9t3+NYLi4u6N5Q0xLizc9Tinyth3QP0pdYe1GFaLu2MQo10pR/NlQx0BUQXWHzJ  
D1Hg0q1+aiVjaY7NiJhimlNkzppebDrwVl86i/+Qg+8SXV2my7laCmaJA9zBy9L52ANoZN3X6pLp  
7z/5ARsu64WtFASDOERp3kaKVWJAcAsQxlw394n5YY771XAPRyfKOger+Pvo3WrzW3jfc1Yn//EW  
98qt918bXhWjTz00ENESdX2GmJsFzqwpDUQdHRZP/ZyGhcifM/w0F6nQgyPErJc12F7acN//Z6m3  
07oartfyFzwWsRA94SZk9eexEdbXe0bBirrRwAgktnXCiOGkDBQTwOTSt+jbSqRRFDYImo4kqy9T  
5GxVSj+/PCBg6cel8b6tje+Idv3+ONXKqKLSAR2ALHxDB7WaxOvF2IUqPShoC+Ih8tmjyVQN7UBu  
x0rUSKATiBAbCCFw5Jr/dX9lXDPy6xxOXqlawQehavY5u+kpOFZXn/DtbJ8aH+km3mrhRG0Mmidp  
Ymtg1CYG5g/nAOfrfVqEgEFbSRwrKx69opjiBIk9WbaRhe7tMpmkZPhkFH18zfcTM1eRfCo6qED3  
l0qxdNARx4pFK/HZ/bEOVW58sqfkb/4+CfkgYcelwL3F+OWRuVTVda4remTzqwXlXScM3WYtTfgY  
mtKrZsYzt1ZN9z5avKsxPbaqkyCJ4LzQQOwnyJN7ptEqjKAA4h5Bzg+y1T1lpMFlc1Mt7dq1JbnZ  
83PHUZIPstlSm29hcI7KR3QB74fAkTvuKxZn3z8sTdNp+ULiC7BoJ6H10S46NFKDuLnkDKFnwLVK  
KU3wY0cDBboiWkA6ORBCIK3pWEmbsdKawO03EzNd11RBWlYlFgtNE40piIN7XN0oicm4GdlaIji  
qTKIsCA8DrJxOnbeTYv9aSITJWY+uJ9JuumeqizQ35GzDBTTSPzcFE7XnghQRZizkUzYVXLf4s6k  
ZeLJlZR6kYAp6/VdYvng/MbjxhVH374Ybn/4cf1+7VDF9OZVM9npIgYcIIPz7Q6X3Z1xOG4kt/5L  
x/hoU8ZWrtb6mxjI4NDcB5r6TQiIQxVMchZTH8ZtcqK8E/NpHSumlqnV39OJUiRTqccPSnhHGMwY  
J45s7b7i3+f2mtr4yqJTr4wWCKlgCpVHVHAIg/eaz9DWVoDiuoUGadtng3EURBBcmkcSqFmDZ9h5  
3FnrK//1ldlK+sbVyJ9cH2gXHcAs3Ez5ewuZW2aI7JkuUU91fJBqjMdVRHkz7THxixOTEXCMAW40  
pi68ucaNGr67IoiYlXSImPUyGs+OE2ttQsMkYqq1hrRUUkejblpPMEuOU05X7j7w1B/12SoqIq/U  
saY3o2HfNhQLYAqH+rjOk2tgaS5z66LvpbGic+hN6W0c8rGuq7swMZSodRVjN2uZ0OVJNbwO/G6r  
K4sy+euu4XKgyOpTtssAdeZK6mbJmyHVsHrdm1N5J4YeiiJiCvtI9RxxMZ9sOEn+5gd1xAX6yzRyT  
nSm2rNGm88zveRhVKtUNwlb/YDv7Y5+/DemBhBh89TXv8/nlLV6Tf4LkVHF1mvTemp8LhrfV0H5w  
5iAamah6TUcBK6bDSzrT6hzahQJlcyMqDaOO2OdjqZvCjXHC6DwMC6mHMF10VMAFdjgGZklWgdOw  
o2aJ6IpIehacZ5i+FBNi5Vx0dOWuzV0akMgqeKCGiF4naosMTFIInpKj1FHRxi6zmoaq5teMhHjPS  
QeapsSAXqIsEY2pFR3txRdewFQbndfB0maxrceMFk04EKSNOYCb2AVG9EKftbhtu2w940KjkIFvm  
7QUwaajoug1gqs+dqRW1LAR4o7L4Bw2jq7K/fflL488sV+jcWXxr/FassfX8Tb0M0d5Aeor5JevMI  
2VSKX2P0a6srByRqEAOJqOSO0w0U5Dw+ttF13kvWk53qg88wGqq7m0p6fd8r1BA8USveMTieIfW3Y  
PBiMjPqECvuNcqMmK6GI+wwK/Df72Ea+xygCHR01D5CEvsJHDFZqt7FAYt1kJGN8TlABrIfR1Nip  
9aOK2N9+1t+IAtpyHtJAgfDYrzBfSipUbHycPNKahVWhONSsE2lYyy1OH7c1jcWxk4Ivwm106rP  
Ka/SW1euCLEMx5g1DhEvowLHQWkHoUy0yNSzVtGlbRvCn5V7KrSO9tAnDSCRuIMVSwi4bld/KKvY  
oSfPatEzoQInHBWJmNGfygoZKUQBBB6srS6FCj7TjvRSbtkhi6qZqVJ0XGQLqgRwUd8EsHZROH9  
6FM2FhfZVS95qY7eGC1k5m2kp7Gi2WEypQuWHs9C8OFogauWRRoEaOc3wefZNGJ1TxICRX+oVKh+  
sZAFRHnnQ84ivE6r5VKgaKjzU6BOEqVSAeGdyIYyhjco2mhuGF1LKWdFxsxsberrNtlb3uImosLiex  
HbMYo1mjJfB3o9jH98m0BU3byl3iYrAhXvBMoZw0bb7HL731HF1rMVo+trwx+7KWv5oJq2trcr84  
hLHK9QkYiMoYZo4aaaSFQkjAOwODx2dQAD5a9vajUNN1fq0ZxxR1EsaeevaVyQ2No808njtHliXH  
uOFsqzR2hZGGzu5ipJq6ibiT7WebDgTTD0qh//OPH2n7Dj1ZBprvz9okTxQI9xQ9X/ODXptfkMEL  
kkru16Rk15wIUH/yCqxK4c8PXUCN65VO+dqqImPTNxgARLAGAyz5o2hPPDgI/LEk0e0xky0RheLQ  
pGonWWmBoiyA0gng3N6A6UuZ0YgmpFo1E/MCLQGvnhmEyyvXNDyroUav3KRFjRmD1ZwkJFSWijqQI  
aljhGvioFMV/wF8iOobVr4kNud0Q3XkNbsYyxQXcWv0HCAy6/hIzFAVteS1tNemeL3c0Wfx/6edd  
LlFSZFv16UHjytj3Riuv1db7OEmpZqKjKYT8k9Tazo0iQ7Eq0IBCpNws+cTTaFy83aVAdVz0xLOL  
YKpZGbKZ1NiY4SebQDXTd+Z0tDCAwCrp6hLjvY1nctMuSGzw6n90MrV9g2xguiPqE/lpSaxtcSNX  
HLRxfRaoXxgspw6P0+4oa5Hwjo+n2oRp5pMEHfRCz+/5dQX6Gdri9RoqCpMrUAMn0uyH+R0M1uTw  
Z8rp4QUIqp+/gvXx06rd2kzi6jsA6RpNFTpdL55TxJ1FqaTpg0z637ywsUghBZitaQo9ChqbTCe  
9rP8nZXTiNxnw8hkFFDuNKZNpZ68ZnlBoJA+SgthU6tjsy2B0jP+GptcnS6HirmXQIU1PIID4cU  
Md+L5IIXGzNO3JNpxPuB1oa26Y0vrA34Rzp5plgnJnwHsvHjbH+8H/43leEWu6lupg0iwrSRMjR+  
kjjy+VA/jWHAeyJOUY8AgoYoWZQTrmM4dqboKQ4eBnHCEMTSJfKXQ1n3jjPypkcFdZ1cNQ2a6Z7r2  
FpnWPexJmT5VAY0SBmB1Zu6hq5768q88cUX7TYthNRWWphsC65zvMYfTnOtu7PaIHOJzlaXduy0r  
eQmoJ01NkutbLFy1daqGNWA5NB0xlqVdrIdADFeXZU9d94re/cfjoewMgpbVOePkiVNDIzOsdil9  
Q28aVNBH6Md50SSDn/VOsCidDU8N+XRTQECk0Xb7WgkG1wIBNLbRWV1Na0NMJMYvMK0RLpb5Y1k  
enSWHcaDSfM1kFtPNFXytf1+sta5+hzc/NcU4LPC5yxG1gmLR461V1YZJe72iUgeJ2LUVJmUR2BB  
+uXwo+thZfGurG6diVSWALnIS2Z23gFKnZNIX2xRU9VxpRYGyAqZwIM8FLwupPp6OkvbLtAa0tz5  
BjUkxqsNoG0G2rcVRPBooYwamhSo2prCCURjR4bSzK7fMxHBniAm+YHcv4LL4iIIIP/26ZzAhDQH



EG9ihWOkJsBiTtV0a8sHM6lk05rxclqayzFx1G7OoSyMpWpH86AILSWKqlTvE6U1j9fd2OILJmly  
arWV5rMZ1d/NzVVCq9lOzzNmMrTgTTJlZRsrl9ZEhjZ+8iR/TQ6B9srXltaaVCKv4VitCM7qa39V  
Ok/PC/0GJyrmuc96x3YZvakMgogiPWVSy5WPFuoQYhk4GcC2R0Act1h8JswcZILfCmu/Sr6ZybW  
dCMNv9U3CAzzDozviQdhdecHBegiDe+9jsuCUZ6Ge4Dd5+YNAgK+saIy3iAmGc6TG1q3U4dqkOWo  
zYQvA7PMxtB9NIOra1uB9bdEYHYaydWY3q64+CI1EHeYsN4q7+yuGi qnnlgXXA4Htjzzz1DBGvz1  
P1IjbkRwfr4WSy2yueMXWMSkES/C3PetOPUZpxURz9iWYRHVO/m01A78Ls6Nk9K4qOHw3W57fa75  
NDKGutNprxZEoEfXbUGJ4V75u0pX/fQRkikIax8XBHxxKIG5jDC9bUNYAbT4HRc06N6RZxTwtAKr  
duTqiUvpB3YJe49BAFG5Vg1kNMOWbx2xYuKf0f9CzUPRHU9ZrLZNNggRvW+/Ma2R2c311f2VLds  
jjOaQXMq9l7QNJAEnmxDoaIoIssK//zx/6oPi6MdW119cr10TAYaqEHmZlqxcYRONLcWh5S4Kyfc  
ZiOCOKqEXqz1NCmTeX07HZhlX1InOQ+AOBp30AR3nmsSH9C80k9YR7naYaoNf6qmNjH6MGMIM5r  
VtkTiXtoqTdLzwwNijc4COWXOFebDxz20+Ln7I0HoRNO89QwLnpGTWV24mlorgfRHnhz0qV/V1j0  
LGfQCVQt1tdl5egR+cLNdXIW2CSldXZL/XxRC3jW4GobUWIKyIL/ta7rFv114HV3kl1cc2pNK/z/5  
VCrcvSRt19zEEbsVjuYI1WCE27fx2gtMaBDOZrIwmBe5kBusI5t1XFckCmF4x9PJqxP4ZDm5xdZL  
kHLGZ+HHeWymklvnTHjxtj1sHbT39gBtnoZRo8udW0pu0OJ7WdG/5b28v8tDX7HD77p3PXhndT8  
NkOMregmEclv7GsWcf0JsGL9jIjoIcbNw2pTa+ReVKrpnH9X2OND6R1rj0bwdvcLp6q4FajoxHp8  
j7tKuhtdYWPrggcEG4Tz+IDBWqJQAHPGLFFkzYtErn4gudHkjYiQZLMKt171C44JlHkKgAaSL96i  
/OycOKp7WIOg/1VcaMc3YnqCXNloo1sbGTkAPhyolH11lsRVVtieaKMaTbGnOWSdiRVvB4jUslqW  
PYCuPcnb6NwrSUGu8dWT4o10FJTDHjiqQO2NEzhk91sBCAYABrGG+OZ5VQZ77mw4afprhLTkyNsu  
EQM78VUa/YiWEhGsq47/MDNXKfj3bvVZPh0XeoB0zk6DN1HBMz5HZMk3UVJtLWqVZJFTHHjzcOZ  
VR4rMnfHbFGohwO311VdTaaJInnbGdzNr+CwnyuGklIexbyeT1M4bYwNS7V4yrIv9FhdKPr6lNDm  
+hDMQEsLq4qOjBB85gY1tcaWWqyd1Ld4JlqrM5qPdjr20rT6vLYw/GvIernlqxsWexzmzqGuzXXV  
/TaJg6ziJRMEXjmpFB+KuFx3EAnsnnX2UxCp3RatY2YZNbTS2OKMxVT4cYWKVUm64LDPR5tsAN87  
U23WAQx619jEp1miDyI1QIzPHKUhgRTBFI39WtiI8pJ613cLd0OZ6qZrIb0dzweyXwP4t2p6SK3h  
92VJyDH468PcDHev+qF+2LEcaNO0vCk5qjHtVsSrmvkIvFGGYBzSdjJfG17BkhXSV2HN3lRGGTf  
j26TT2ZqVOjZKq17v73eK2iTqWfANr6j15a5L30j5/1xvrW7/++LeHGvxfEjJW0ob+kdlilDhge  
/GASsLkpppaie4KrZzLWmc22qi0prSdJTiIaaIHuvJGECJrJTYTtNV/RgCw6ouRIXcRA8LV6qg/0  
tXk8bleBKY3EqMovrb7vHNCbTSgkRHSaCgrjeY6PpLEIXnSQSRpCrjplDMZLbLk1j/Yz/MT1k5e1  
y5oTYjh1EZCugakpMLChEuobgpR9cjKBp1J+EDam+ZCqTqyaIx5F5UbirkMEGOxFR9+jVWqWQizq  
z668W13NiIUTUBE2+X1Za0/8zTySx3q6WssIm7bmjclumZAKgGU38tmVAMxqyYSqbBRFVhUjLgVX  
zPvLYT6NOFYyafEagRAROV7cyRyihctcky9mmiTr5fE9L5L3/Mxjn6Wirxmr9kT4/6GbGAY/s2Dz  
3pjXT+68rbRaLQETwWPH9WH8IA9Y0k0nJ8GQyot/QnebzidRCQJ66hSU7iJTKMMZWokbYp3YbVGq  
g9L9XOTqCukRldHr9kqf6QxpYWhsFvpDJ1UubT+/6t0BijTIabr+7zo/PMZOWuT4JxJHTs1b+PKj  
JEdoyiphZPOsLpU1fygWUz8LtknjYqr1UYO9ENmKCDtnan+00aoV6+76TaFYSZ9pa/hfbiDtlAyv  
NfinYPoUqOJ9QG88eSUQfxcAdBA7amg2IKoJJYx+w8dDg6ipvGmeRbxtT0uYJ7tqMbUONUNfn54A  
kEf7YTh+igYam7XqqwcrNActQuL8LrEcj/jKh50LLhGz4BqxZUTmw0jLdIWYzSlSdvrUqVJMqtq8  
TSDFnf8orP98LpzMuC5Qb0aHM3wWW2sV737nd14PH47xjW4KZhlrY4nqv0rAAno+kVAzErWKqYd2  
9RxcxzGEQtz8/Tim905p0+JRTKrxVw93W9um8K0cDTdwKZzTAX3J7o3NZnt8npqRNU+7jhRUGKdG  
MoJ8LdwVy960SV6OI0EECVavG5KJY6JMFZQQLiBwkGU37xd03A0oIxY3RhWWWesen3czWN7e6iRX  
Onu2jPmEm9FqMNTTrbpGDR1dJto8Dfhx4NfiyjoSmI3TKsuX1+jY3ywjgoPJOFMqa11BBQiaIAlzubYw  
HdD/YH3OjxDjEs4tqkzfnZC9Jq6ZTpV/q4qJ+Sgn6UthJBkb3JJJxHthEsEoGEBDSR2pes4loPsD  
IazRWm43fAaa9hLVCsirnj9pbytrVt1ibaByZlrkkbVw+5Z6BETrn+fSwz8kgGVlq394oc+0jyrj  
fXAk0++JtzkszfXpaqc1RTciM19yC20jA801RSEclh4aXaJK6KWOLLhHk14eEWtNKlED15bupjF9  
DRV7ic8veM7PT9Kswj2TkzBUGzREeCOiPaUcUlaVTWkQZmnj9vHwvM6dcdW2XXaLlEKRkp199k9K  
WIG6suRK/1NJfMdououbT7VOqNMOzUhnYeD0Dm20cXP8XUbreGAvz16ZFn+5fqB4wiC1+JR3Oo3N  
dTc1Os7js74w10WS6qoByXSWyNPEu+Ga6ZBcP7cQJ586imTjRU2lnoOiI+c3jZST7lQWnHA6RTqk  
ylJ5r1M4YbT8dSUHyqqb4xGE4MjppSchYPHtoKpAWvEqBKcBIOk2QEDz11j4TPP9fvWSW4icV3Ra  
5kuq/JN6EnSue91ZPKwGzfZGko7aKrGiAyKxl8ER/Pz/9a282U31rX19bcV1uFERYHDB0QLDLAoR  
tCV5MZkRMcq3Wq8iQh34JVCr1LVTa67cw5p03aRExt4wgtN00yn3ZkLlkxODxTdOieOAZyYEV3  
RsJrktW8Zk2p5paVfkpzJVatG50iRM+z+7zzqP8ibiMSWDHigMM2AzKM9tPA5DiRDeuhXuzuP2Uz  
hglmwHk43MUV0ZSqG0GsVTplYxrKBfVJ5NVufn20+ToxshKhCam2z4fbXz+jD13U8UvNmh0dvNmQ  
BH4uR63A6jOLzezJRK76ixlQjRcW9sIBjW2iKRfS21VImfjTSersSaZrT0gBRCNKBIDILTN14wdNI  
4x2MDedUuw95cY5VeJQZ1aTmBCuPfiaqS1UdibRQshWkk56rUfFOMjW6FIHaXlmgYw6w0Yi/8fS6  
lpFDXj99oy02cb3WAKp+aef1cb6jjf/wNnFdHpZV+kBAHOkRkCUwENTQHYS4tnglVRqrLgBrFcLF  
QNDpEQneFqOVQYyaUhUZ8SsK6ri++oK32dacLeLIZHSVvOXwt0dcH3ZJHJMELTossHaGm3ysPmEE  
qrxua0uFs5Rr168m/NEp6aBaaSjFmXyROMjXnZOqtFqnBPiPRZPPi16G6bWJMa3OF5dpQie0STq/  
fAgmcwIFAvXltfkmhtvaQ1Vsrh7xhtKSURiVZrSWocXTRLum2lsV2pH3oa1rxl1Y1vsSIHrabd0O  
BnLwcOHolOA4fXzDrBetAsrnYjNe1MbNBF1aubcVJVqGfQGMqlLRlQYaj944wFWV2LZfFppLVREbV

ojAJHVAWrSnNWyrQuHpdGZz1obLoUtrpuH5FXaPuiVmu2R8ovPRhMy6zbK2QZqzKdYbbd1y4tFnt  
bGura6+rR5NMgedxw9flyrO1U9l0PRlGfLehgCSB7Mk+97nXHGnCTzfpI6JLFOURnmohPVZPQnGB  
veBNjqiaawRRGND6oWuMGLnNrKoM9v1WR/TpW0Mzn9+nWqLvpQlrxheeCk4ivN3X9gaZKWURBJzu  
3rJpE5UHoURtZyoEfi+zxBtF7aZxhaI8FiTWNCrV8mX4mqm5caoXlmyfmgAvZvuOk2ObS8artmk  
jjjVaV7lFkKeq8MZidxsXBHFBxkCFG2kxMIIkHc0mL0lznNnVHKHDx0xNQIFYWFFBiOqxFLjORp7  
+MkfxCRoGiAQdGg/Mnme1IMp4yWmFFDUCmjOqltTUDpnw/PuiD4YXFxy0wdqqltGmtOckRJdzWGB  
FFbxExOEUFzaYDU7TjHF4G1HYyOvjCYDNbe83NXN//W9vNlQzD92i9dnU3Wh2+eUGiqJJKGYBNRA  
q8+TDU01BNws6puWHKnjIO0iWjhwiCtzTiKsejiY5XcIF+ZdO5XR8NwKULGA0NUsY5m3HjdaPolh  
S1KJD63rAyZVcOzv59/9mmytGWzws+SdikT0lN2MquOo8rIx6Kndjff/Ak7w5fn7fXacY4bqqezn  
InaCovEBvdoDoF+hj21/3LtdRZJ2mvzLk9KdJ2kGBe/abefaURNLEHdTM6aL6qO0Yq09TppGyhl4  
GwPr6xa9lCHZugvWlIoijGe0/W1WlHJ7bpzFcharSdzcnhVqbDPztVGCHzsVGG3bqWafxyF5ZZx  
ES2Nb+0yJS3m7qyLicSTBt6pY26UmMXRayvQS+zelavKc62PVXOWseDZ5A2Ldc16+XDpflNP/1M2  
NCXLbLef889rw43eCd5lwZA963T7Z5VWVeg9G5OW+eAj5WmCxuNNTwURbtUXGAbgQktJlONN/Guz  
WhuhtpBQWwNGcwisaAKUaaxHS1VeIhJqlxVHChVWzd9HWrevJpWmxgzOaD28M4//wUGseByHRlI  
9eYVqxtpCvalVmALEproJUhuqu5sNVOobJGQuB2je20opqoworkibxqSzwSkh910/QlydHVi9aaRy  
TMtCRPOdjNLqcuompCLS+jIzD4XJeJonZpZLZ9ZzQCb1XU7/MbaOmtK3pveI1pm+N52419ihHbci  
pz3V8cwVPFIKHMNV9wuI2muCoj4xWXYUffig+ariBteXJzn992RqnRLEkkCyrBqIvEisWXPsbML6  
VsfXyFC2ny2K/B27NjNRz2sYwlnHJzJcGHT/N89q411uL7x5onJZygeXVXW2dGtratI3R5tAUE0C  
7XpXF9hzTjPNKY0vX5O+CB3CdVaP0pn+VH04uF7vSyZNDAl5QjdAePgee8ItvqwhsAJaV1h4IzED  
o8P/2dmbmkSGS0Lc4lcdNFFjKpqKBVfTXHAI1CLa5PMZFjY5NFuLKUvS05+ZfOu01UbqNF6m13Id  
rpL0l/Jzm7dwgRxLdgoEO7R4UPL8s9fuJXaSlx7aF5fkm4K2qpCJfXl0RDuyCzFWUhmVlk1j1IrQ  
dJOrYmoh+sbjYaysbFhavW6PYGJbKo4XAcW4PX6wsIOqmHBedBxypypwR8dj162Eeq3FQWlD4wuNs  
L60ZRHqUTSa+tancKplVHZItCZltEe54cqEuK+TJJIXSkAhHULYM3J5J909dmTThS7q5EKnfFr25/  
vJVV//y8FlrrL/327+1NNzYeC29F4w0aaOLjtFUYGHMMRGcY+qGNRF5L/Id8cBAH5uMJ7bqwbS6b  
PaZpG0t4ewZUuVqRTAlvh+HclKteWuTYODmslQbTFU9MZxwGmU8faFyFD02YapYy5lY4mY57cyzI  
jE889YVZDuL0tTbW9QPjJTKCEURqW6LO87STCNT8ndC8nul4AjvdtA6xT2SwwHFAxc0pMBIf5fX1  
ois6q5SrIjMMgdgxfEs7ZlsEXNctx6VBXKEzdYDvMnLSjewVD1WhlIw8DnKTGvDmWmdRN54m1L  
mtx8QHry3H0xJHeaMzOMBxRZfNgyH7Ohe8BBKGcVjXwsjBObeIj9so0nMuIX2HXtpe3kMhEm5fMo  
KxT3DVCblJ3v9Z11hGBpYSMop9nf/1M2dGXxVhvu/6m105H43ndp1p36gGkV6VG0loPPY230Fii+  
1hUbQbdtqanuRg9YVPG/ZmNM2SQImepedemKXlpb7TaSqKpIicvSVopUu7vtuj6xJVGNqUGktQuuN  
db9tgcCnHKiRovE6bzzpV+iaClqohFIEbONRwGa7OUlZrPwTGVNESAGQbbTyZhQZk1M0t6c2GGFs  
2lUZblQ2kph0ZSWQujBGRxdXpMvhBQYnwt1qX/uRIH10t7CtffycB9EA6jguovGULwbwyA6RQyde  
c09WgFMkCoXWA3VYI98kLrdD/NGdIglk47sSiuu3e23uuayLwHDD6dFGRd14bmB5+qia4iW6PKqC  
qbs40rb4wpHyiylULBz04xjNzjCHVODmfcMzMAGVnNmZFa75AlfUtVl6Tgbqbaz2Ye+c95bCff56  
melsa6urbwGzaLKdHuYkj7A2ss6iJKhJAiFZdNqcHyQeKmis7WOP+ydj+6vb5BG4cE+1I0tVVD5  
JawWiF93BJWK+A/SR3P2rgIX8SoeioMY+vrFkkDSTQzNLLK2DvY+Iax007dL1TZj6LseF89HJzfh  
s+ZuxQp9W4ratROJ6OQt7lHk3bI7uFURBSJaVG+8brei6/UgCFmEfHQZ4WI7nx5ptkZTjVdm2HT  
J64flPnunlgxQyTKWqbwrbi4XVnc3obWTJrSm2sb6jEiTXoEpNQwSspoKEi6uhLcYkjCwYghFoijt  
RpnzfqpIhSMxPqQVNrWCz1TaJDnowrOgWimeoyquLj+zk3zgli9PQg4Wky6mXIY/g8Ff/GBby9g  
xwXb9XVDILN56pp50lSyWJtcfOm5Wetsb7+m181Hw7lZUWlCHrdw6JemQyXpLB6wJEijaoaQgups  
d0FMDinwXGRb9mheSXWfcxYG2EjWUTVuk0iDq8xbqgrG6JXim6hKr/WhFExQLCWU4+DMTTXEU1la  
W3lLqYz1sh7bh7050xznx+hZ013g1plbCKk7gsLBm4Q28421U5w+D0IxqpjIGF0TNBA8ihkKCnCA  
d3pGYRwCn2lEOH+9dqBVFa0k55FppA3Q+z1WdeL3pd2x2hqCC3NjW3FlDo8CncnEYGEearSE704Z  
Z1FS6YCa7moEkSXNugqCo2hpyGrNr3RVxVw2b4uegsFMR4z8UAQB8aQGulTCem7Iq9K2oYc/x3qvw  
Vl1wL2Pitj1TyUKwXVHNN11HUlHff9YHHDROskeu0Bzu+nYtpcUvbn5T3zgV369eaZs6Rkf3YRD9  
JLxdHpiZYglLlXyFjpS2TqLqakYP7St3nU4bxDVeMNG0OmJcDc1WHSQNY8tmSSMz9yU25nVs3hgB  
35nks4M610QjVlei8xlrUJrrD8zR+/XM5KUSObPOMWHbN95Slys2+Z4tl4xHD7ULN1Glmv5YjaKg  
zLYchLHVZVDDxv157ZiGYRtYP+bFRQjQu55ebb5clDR5701+ymoB0YHdUX4YCaVsIl656Gur0n8  
aCawePwApureZEpDsJgjdeZd+pBb8B1GzKzybQ4aT/xlSTWOBqNlduKLjCj4lw/rm7kjLeeksOKO  
laYbS3I1qlbYzc4z1KbJkqNoplTbJ6ztZDCWSZNZ4dls78/LE4cXbOheygLZCa/8IzaUvPeGQNH  
+ilo+G64ih9ZiXcLKQymDyUJpAleHC8kcFqNbHRhzRxxYPOvBqrZxNCzxBRVUV+SjU8NYqyJRTSC  
j3r9xJd7VjrGkTOl1jId0GdikT+VqGbIzLFu4szFpUuQEvbCv7v0xRcQOVUW0zaNTC2NR/YAGMnk3  
FLJkAKjy102KhIW3m+weYsCWhFVbZ9qTa5vWudFGVWp2ntLLHXBFPiz//Q5JaJ3DCuVvpaErB8HI  
nQU/TG7LKYq29rdkp4kx+yFSbQOXOcjsZx1ylwvWT/WMC9yyp9bq5an6RR38n35tBh+za3NDEoo  
hoF+KXTsogQTaCZMmvy+fLkoqhlvj/PmSpq16fhrm0c2BImnD1TKa3QzhLFwhsHnHQICqjp62bGy  
XQZWsC6HzYiuSxf/pXf/v3VZ9KWnvHIOlgcvBwEclfar0wriAZpagK1EZJpxEiVUati5UXTij+jE  
zqXqYRDItroce0gAL8HIQViQ8EN0VZGRHrXjMe0jrCPGkJoitGBI33coIioqXTjaNJ9YJ1VjYhem

```
5bmZfeFF7cdY2t8JV4oVvoZUJv65jlQ8HDQi/A1Yl8XNqmoeNNKu+iSZXjunmYdBoTALFr7XEowv
+vOPfL4U4dmUjTppuKouTrspC4wwbZtXLFU4zW9dLqiPq8Ml+UzT/2stfQa2xGUKr6acAmTA4V0q
HNiXWBN0tBF1vFW0tQ2d58W47aGBEAmbYXhcJ94pgYLsrBpiVzXbg06A2JIZnsQ/1lYYxsZqKraT
n53juqqD12BuFj/J8c4NXzwNC/y3txPPdO29IxH1oWFpUsG82uytjamxAg4mdT59UXBidZvmvL5H
FL5qI4ASm38AQ8LVk5tC5Y4KA8PbMkWCs+kKKl2kbM0t46uRFnM2udsMURWLTGZkF0dCSUdhoh3M
rstfLF9Ouc/7yzZvuPkjgq+1UL+poancGHphDVXqFWHazIBnJcY7pZt5+i272DASYlVldoQmawfD
hF7nhrKXNvWZCalbDRhdeM//MO12kVptcvbYp+bp/MvxRylobo407VGk3ZKJapUJUk6M6oA+yUyf
xpfwKSZEI4Sxk2MgqI0MbRbU2OySIQs6lLjJpLnCwqx615bLEIuolxskmmvgM4yg5EuxVTXWVXdD
W9eH9e+Xd3mz00iMxEy/lmpllTemct2Db+LVOoafPdrWJsyl/cO9frZJ59pW3rGI2sxHJ+6NL+gW
617yqqZDSdx1QMXGosq0pX17LDZxZ81vVmldBdvrnQejI/N5gRMosDb8fbJqoL5J62tu6y73D1j
eMu5xXfhd/Z8Shsp+GihSzqGM0En+NJLLw3GNODD99EA436tIm+sTdm8bUcCADZg7Wix1RoVeGDUq
9PJmuRz87K+fFD2PXC90cXZeXAYzoPrV0mVLuk6ALfe89dcs/DD+nnsUlnGjG149x0D5kZqkeM+
FlrbQ410gqlld19OqnOwOMKxk5nlwZiCCexDrcDDdqmXdvMkQ7/s7ZGWWO8XPxbOATfKseF2uhIh
3MTnD43B/prA4c8sVqfdMuqisqCXSkW/z3br8gMO14/zQhnz189Kz/abRy2GQoRXXf+pw9/dPqsN
tY3ftfr+sPxSLVz8gGXTUGVHg+fLfJSWSVV1RyTwhhYn9uym2gc2JIdCctZxnoFy5J9hQFurjat9
O95kn6JZcLyJeuOmcZW22OJf854ZG+thAh5wuK8nHfB+cZnbNUCGzukn1ZqaLbInWmTpbGxBA7lw
rYd4aCuS96blw7o3ILMbT6BQP7x2jKWALWHOxE7pBP5zD9+Xue3EQndNkrIlqlq05dqItjfjTb19
R+pGiScpu/q8ZlnZpKdvM70/tp4X7P2nsJYdE9MHjV4u4bAn8nTCDaITZ3URcoKjmx6HcqZ7rHJ4
1b4wlg2cc2HY3JdID7T63C9JOecdq8hmUFh5Xwuvmy5mzk5Eq6rGdxtLplDWAlE5K1fjhHo/xJgA
HavyI3ijlo8AAAAAE1FTkSuQmCC', height: 150, width: 180, top: 290, left: 50
}];
```

DATASOURCE.TSX

```
/**
 * Default data source
 */
export let defaultData: Object[] = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
    Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
  { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
    Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
  { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
  { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
'12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
70 },
  { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
    Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
  { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
    Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
  { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
    Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
  { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
    Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
  { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
    Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
/**
 * Image Source
 */
export let imageData1: Object[] = [{ src:
'data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAAOsAAADSCAYAAACrfzewAAAAGXRFW
HRTb2Z0d2FyZQBZBG9iZSBjbWFnZVJlYWRS5cc1lPAAAAAD10RVh0QUxUVGFuAFBvcnRyYWl0IG9mI
```

GEgeW91bmcgYnVzaW5lc3NtYW4gc3RhbmRpbmcgY29tZm9ydGFibHkuIDt0SWEAAAAVdEVYdENvc  
HlyawdodABVbnNwZWNPZml1ZlIrxMzUAAABGAVRYdERlc2NyaXB0aW9uAAAAAABQb3J0cmFpdCBvZ  
iBhIHlvdW5nIGJlc2luZXNzbWFuIHNOYW5kaW5nIGNvbWZvcnRhYmx5LiBae8yiAAAAGWLUWHRDb  
3B5cmlnaHQAFAAAAFVuc3BlY2lmaWVkozTnUQAAs/pJREFUeNrsvQmQpOlZHvj89593ZlVlVXVXX  
9MzrRlpDh1IYgUWEgYLIQKWBlsrwPbBBActoNDDlgbFrBgl3vZ5XRgmzA2axtY1hAYWFhsLiGB0  
aBbmqt+nqyuM+//PvZ93i97wBuxEev1tDQj/19HRWVXZf6ZWfk93/Oez2vVdY1mNatZL/5lN3+CZ  
jWrAWuzmtWsBqzNalYDlmy1q1kv0uU2f4JP7fX0b36/53Q3bdfdtD7L57XGdTOu6yq+GfqvKsqi8/  
4u+p2z+Si+NZTXR4E+ddeN3f9TJ4tlZ2/L/xalnPr4brWZekRa+I/ZTZ3Mz8NvtYOP8FdRpliYn1  
1N+9k4QRqzaXsrD/60XhD9x5Yv+0Wnz12zA2qx7sI7f+1P90/3r/2AVR29bLeajfLXstwZb/eF4D  
yfxb2L/1jVmp3N0hwN0B114nQ78wMH44kX4pY3KqtDpbaBMLmkazeZFli9c13/c84OvuPy2b180f  
+EGRM36L1zP/Mq3+50Dm78gQHxDLRxj1SrGbJEgSQqURY3xpbN45HVvxOTGTaR5ghvPPAvHLQhKA  
e4WnP4QvW4H5/ZGsgkq2K6HOM2RphlqeXyymh31h72n272tL7zyBd920vzFG7A26z9zPfl/fJs1P  
d7/1jt3Dr/q4OadywhbsCoLVZzjvswpPjNk9wie28alhy7gjW/+bDz34cfr2drD/PZN5AJIOxC7u  
Cwnny+xtbOB+y6fQ7c/QmlZyOMYlewJ129hemcfYa911B/tfjRP48975B3fkzafQAPWZv1/WL/6r  
s9/3XKR/Yskdy/Np7OgSEukAjrba6PIMyzTWpgxRlEUiLISLdvG5UfP43M++y344Ac+gjpPkUZTd  
MMQSZYhXixRi0PbG/i4eH6M7TO7iISdHa+FoqzQG23ptWoL8MPOTRTJPxGW/a7mk2jA2qz/1/Wz7  
3yzs4qc33Zd+9VpVvVn0yXKvEKRE2Jre4y6KiFYFQCuxJQtcXwyw+3DY7hejcUyxuWHz+PtX/BWv  
P/xj+HGU1fx0COX4ZQebty4gc1egGWWYOPMCA9dPouWMKoyxfCCEGVZoi0+bibfa6sWEh+kTl09L  
gB+84Nf+D9kzSfTgLVZf27983e+bSNKkt+x0XpUPzDHghAdOq1Q/uMijieaLa8EffkWIctrrOYzZ  
EUptwtcP5hhKWbyX/j0y3j0wdfjTz/6BK5efRpvfu1DuH59hm7HRTQ5hNXy8fDDZ7Ax2hWTeoVKG  
HWZxHJtHzvnL2B5eiSmsSes66PX613tdDfedukt3/JE8wk1YG2WrN/47ne8qXTb/9J1vXO2fEyOE  
8AKPAQCmCw6QiQ+al7FSFYF8rzEbDEV0OZYJSUqYd5FnOHwZI7YcmBXfb70i98goOzjPX/yFF5+6  
QzmixzT4yOxbifCyEC35+CNb3wdotNbqMUMzsXEXzoAbuBjICA+Pb6Bnd09pGIX2/nyaLy99y2Pf  
tn3/UzzSTVg/a96/f6Pfc13L+eLdzphN+i0Q4zGbThuIIXCpEKuwoo48WRgB0oUaKQj3E+XWCVZ  
iiiCPNZiuduHWjwKa5cJBUQRxN8+X/3NnzoY0fI6gyj4TaefvI5rGYnaMkhMESiPHLlLB577DHsP  
/GnqFxxYLuN2XyCs1ceRLpYCdF6AoRJs3EPrlfLx97p2v+/L/9aebT6wB63916/Gf/Wbr9Oj2b  
1ZZ+saqvXGMDs4IyuYCVB/wWji69SyS5QRB0AI/uzQuUMSx+JSu3I6ETRMs10v5f4V2u4f9wykmy  
wROKOarsOp4s8L5vZfhieuHeOchl+FP/uMz+PDHPojz4y2sqgKjfoDP/YuvQ1BUuPHkR+C0AuReg  
N3ds7AFvHFSi3+8gN8aYnrrKoabg/lgy/Odb/6af9IA9h4v5zu/8zubv8KLZP3JP/1G6/pzz747m  
q7elJalm2dCm0WK5WKO2dEhbj37NCIXsXSDCKzeu38bt/TlOZ2L6+i0NAFXwIPawMJ6Pk5MINw+PB  
NAONns9TKIaTWrThp8sMjDl5AsIhQtB+PRNj760edwspyJqSzXqCv4dYH86LmR0hShhHsnE4W  
SDojNafbeLw6lUMtvm6zo+CBzHftP+k++5cd9rPv9DzafYgPVTfn3sF7/Dmq1W7+4PN9/AAgXbE  
mNXfEMCllHZsNVGuzWALX5rQaYU47fTG8D3Brhx+xDz0yUKR0wluU8rFN9W7pGmBSJxRrO6RMdz5  
MvH8STGYrXEW5d2MTm4jb37ruD0dIUb+7fhCXNmRY5ep4Xx1kB8YxeTo4UCeB5VcOQQAmlzzmen8  
IMQd/YPUcjh0061AzjBG+e33/8rey9/U1NacY9W03XzIlhXf/0HrWUUVxtl8YYsjQQkFnbPbKLTB  
soqQSyAu318KubsXMSDu+MORru76IUu7HKO+85uYHT7GxkrmOZLXNs/gtvvYbClgXa3j8PjBU6nc  
2R5jNgGjY88fVsAO8VqKX5vvMTFvT1sDjfUte7tAPvHcl1y/cwxXQO9128KgM3QGffGJhckP72C4c  
w6LKcuJfUymJSyrhenJdGzVxa997Dd/uNV8ovdmNV03L4JV2PZ/GJ49+4ZBXgvYTpELKFenM2G10  
Y4EpFkOHMxmwmBbKCOBXOAJyASYrO0VJnSE+cJ2C47jIYlW4lsCp2K2dgdDVGL2joQNT6czJGJSb  
2xsCVtXeN/Hb+Khc7s4uX4V/eEluUaJtLiXGPZR2h5u3JniwkZfrulimZbIj4R5+2MBEcaV10XFY  
95t4XgpprIcMEXuYLGCXW53B78tv/mM51NtwPoptz76y9/zA3m8ehMZMRXgMapbxiIEwGhFDqbw  
w6Wkxn8fgdpKSZslGD/QFiPJqv4mHMBZy4PKsTw7Xe7GPXbYrYwCJjSWUwR9jpYLBZa3FDGHm7eO  
cIwsIV9Z7i8tYl0dYozulewt3ceTzx7Dflxgn07Y8xOTwSkY/i+J8/VldxunWeo5MuSbWOVlvzOV  
TN7NV2ZqPE0wuaomE37/vfv/qpp+2+/rQk4NWD91FlP/eb//Bbb9f9ukdEknQozRQIAH92Nbfmew  
fFWqKsputsdJGKWXqsVYmFF2w8wEXCvkhi2bSv7rZYpDg4WiIsCAwHsroC822vBrSwMh0M8deMQm  
50QyyTA0XwuJq+H/ZMTnN25gOXJbQV9laXwwhb2j5Z45ct2xFcOFmxJUSLodnB85wYuj8RvRkt+J  
gcHzePEEGIQIFsl4sseI6/uD5K8+MF/92Pf7BeO81Nf/LXf3/TLNmB9aa8//flvf3h1fPVfdnauB  
EHbx8WHHhVWEzRpZi9YuIKLDEa9bG1uyM/S4SxNcGMEzVvZ2Ia91q+mMhTAW8qzFpgazBCu5Vjv  
lwiK2sczlJMFymGalzmYUetLk5WC7Q8F6elgizitelgCd+6hQuDECd39vWgCD0B8+EpPO+CmMUZV  
lktLFphNV9hvmJHjYpM+amY3V2w0yfpC5zI74b9DaHeQF4j00hFfzWf/NCzdybf+sQ3fPEzD1y58  
IVf8vU/2rTb/ReuJhr8SVi/+1Nf+Vo7q3/r8GQynh3egtseIIummc8mwlqp+JoD+I6DZHaCZHKIa  
LYP2xG/NLQx6ngYDgJ44p+2hQXFWRQz1dNC/rq2kGZiEPsOytws01iWmMpTYdJQQGrJNRdxJKCS3  
1cFFvMMZ8Ybwt65XK+FWbJCKWVotTrY3RbGzOR6qTD2/iFy8Y9dJ0Cv18aJHBbiImO2WOLa9ROM5  
PW6bil+rC/XdcSELzCbz93r12/1w1b70mjQ+8qnP/BbX3D74+/5N/c99t1fswMasL4kl1u//2FffX  
1XVbyzixZl98RGjqMbR0QJpAiznOW7dmmD/1nWxeSyty12sx8Uxrpzbr+3j09wuD/BVMzXduiiJ  
2Ztf9hG4ISohD9dgadd1/DF9E3zFB1h0zovxM9kF04OX+4TCRtmwsTMxdq2g9AuBYQVqqKcttFUN  
vZ2t4TVu7j2zDPaqXN6MkEl4N7Z2YJlQ3zUCI7cEFzjudv7GPQ8efliC1eV+Nhy8KQRjk6m4J0Hg



z6mp6ed08Pppdr2v/raR9490Hr2P/6HC6/4C81m+M9cTQXTJ3C97+e/c5CvZo+Xln355PZNDMZjp  
DOCKUYRz5QZmVNdzieYzGbY3D2H2ekcz159BqONTYibiTqFmL0pyqzUIJMtAEwFSLblCmE6ci3IA  
ZCIaRohl58lCVveXLgCTJuQthws4xh+K0SS5mg50cbDFlrdEVj/kAoCd89dwP6xsOlygu4Y4AYBK  
ex56b5zKASQ85MVNrf7ODhNcfXmdVy+uIvldIKHHnoZ2p1AWfhATOmWvObjoyM5CFI8cN8ZvPxVn  
47Do3nabvtPnj1/319649u/9qDZFY3P+qJbT/z691nx9PD3b9189nJ/cwe9jV1c+/AH8KBs4HxmY  
UnAJhXKtMRCfNG2HyLav40zF3ew2X0M+zdvCZhqCBxRlZb4iAMNCNm2i1Zdivmai/9awPPbaA8Dn  
B0PsVhk2myeMoLrOvJpO5gu5+gFHQGUHtpLxFzOBcRtOHmpihH3338Fd25ew/LoUIDaEb90Jc8hp  
FlE4rvWqOIV2nJoHO2f4Fj84/vPbsIu5YRWGC6Wg0cYdrHKxaxf4HSxUvB2O0N0e1uYTg6QrZbBw  
Y3lo888+fEPzWenX/EFf+sf/mqzOxpmfVGt9//Lb/6Fk+PFOyYnRwjE9/TaLraGFxDJBu5tnUUmD  
LgSgJQCrFl8xCpfIRQzd7ZIMReW7ThMzzji16Yqu6Jtci57TwVkj1zPDeD4ngZ8LPFT7fLPRB0y2  
qt2C5H4pKxgXEaxAL9CIvdl4f/OlgBbu2cUzIe3b2E2maIUBk7kMLAsSx6fYndrLOzbUskYgl8cY  
1Se+MVViePTEwx3dmAlMfqDDpYZD4eOppPGoz4u3ncB8XKFqMiIeq0xptmOypfpuD+/+mLv/aHv  
7fZIQ1YXxTrqX/3vW+ZTQ5+ZXY4CeaysdvdDbQW0xR8enGe0jiHFU6U6mVdHogJq4AsnIxOThCv  
EpKW/vwB2NhNTetHUuYtQQLC23PR9jqwGu11YdkUUQtLFemS/lgC60rLitbQFeothLb7LJKuFn8U  
gKplM+elb9yDqgZ04t1bbXjSuU5srLs1JB1lep7iq0tZnWq+V09BOS+sZwabhdIAVJqZnQx18LKJ  
18sg9HWCP3+UNUrevlQAJuImW3hdHqCbcBCi5bbr2xnMz++ee+dbv/L7mrxsA9ZP7rr+h/+MluuTe  
bS8TBMWUxQJvRRJypeRiZMVvwxV+192lp4QNMTAqoiIgu1Ndob2gIcq9D2uDAMEbYD9Vld1zdN4  
YxHCTOyha7U5nHhrqJQH5j443XJLLYwYikALARwhbA5Ga5MxGwVBuUhwZyqGNMCxFyDU7ka1PUCN  
YV5/VzAmqtWk7xuAl0YfSGWAEsl/HZHbsfY2hjJ6+vADhx9PvrJq2iCC+f20GrLoWK38eyl4ijH  
Ifinz/46MPz17/hL37da9/6t3+u2TGNz/pJW7c+/O6fWS6iyyhjBH4H1TBdt9Nj4FTLCeMoUpEyI  
S8BBzQgxN+Refu9noCwFCZuiVkbaw7UcwkeH05gC601FUGsN2Qu1RL4ZOJb2gKO2q01guvw/qWLW  
gDuC4hrDTPx921huVjYN0MRpnpofL48L01mvyXAFp+2FKa2HTGJbQ1QVQXN2wC1PGcSjPrPrS0Pg  
2FPCyOsThcbNK2zSCPMY91KbicYboyxMx5jKe/t9sExzo83cP+lHTmgOhgf3MLHP/ihvhv4P3Hzu  
Y8c/OWv+cH/q9k1DbN+4ln117+7fe36jeeWR3fG6eREfb0zZx9ANDvE8Z0DbI66AlTx4VFZ7M5o  
mWs5u3m5hgdMZhd2kKvI2auANelMew5FOUWJnU0YOSIz2q5tpq3tQCzyJbyVQpI1SPUfteayBcmt  
ARURelqKkY4kz8SgNUU/Jb/CZOmKwEs00Qucna002/2PO360VM9EB80TfSapfE3kyvZTWE1MLIsQ  
IwzMchLIKGTfZdLiPM5TDqy3sYbglxZm8Xi2ksrL3ASMzqjhwO4XCE/dtTPpMTXQGrZv3PfTKh  
/7aN/0vq2b3NGD9hK4P/Nw7f2dysP8mT9AUJTU6smkFAQI8oda6w+RwH7bYqKuTifirgh4BQ19My  
F5godPvC0g9BAIEulzKbd+AJhTzUkDiOMKO/F6L2VskmmXNk7kWQVCWlGas0K4wo6WFExCwEsQl8  
66VYeJS/FsW5cMKFISZmOcZwS3sWzKPY1WocgEeJdQjErAxspdfC2dZiHphNW14bsZj3Jf3egjKnX  
ZRiky/kEGIqaraKsJBDIJAD58zuNrqnivqv4uXsjrhZjNAMQ7GOHq9UO8//EPveef/d71phGgA  
esnbl39tXf1ju7cfiadz8dVNhNwhIiJH/HkGO1Q/M2whXwWwRKztczFRBbTtSWs2ml54td5mgf1q  
bvkhwAYyPaMmj59RHVQawd3PzurFACkuznHwqxlJj6qQfAXzilgrJe6wor0wrgSjFTa/qthQFuR  
dwJSzMQxIBULWYv87j8R5CXVS6vdyA+bixgjbR2wg3bch+BsbjeQvzjWA4ayxfTWg6mWA6MsNPG6  
dGR5n6PFgnmwrwsddwc9LBzflNeq/i/0VRNfAbGHDHbxbztbuH0wSg9PJ5//TT/+W/+h2UWNz/oJW  
fOT/Z+38mR89NzTApIUy8kSqmM8XgbZx97JZaHd9AR5ownU+xevh9uOkdbfMEgdOB1u8qcHlnOZ  
fBoAKc1MKDyffVvLdujbarsKDtf7i8gc9gNU8JhRFi+PDGNS0EWEZSAEBpFaSdyuxaTuBJT2NGgE  
rWWFJAapBKflswppOnaofrYTiEHg7A3U7V14CroLddVU9sucn181+8jErBbYilUWYFITOkN8VOjx  
RJdVxhcLjidzCRbdhFFN3Dm3I48R4DjwylaHQ9nzuxpwG17qxecPbv9k3Lph5pd1DDrvWfV3/gBJ  
4/nzz37+B+cW050kYgpOT2YYDzs4uL9F9Huj8AQ7er60xjuPQA/LDWPykJ6v90T01L8UktAJ0x1O  
T5smqlUjhCTlf6iTRkX+qQCZitfqknJvlbWBzNdQ5a1ymEhDMpVrc9kmJjEJy0LI9XCQgdGhOWZN  
GpcMnpMVhVmpn/Kw6BYp2lo+rJsUV6UXEH8W8vV4gzuH/qoFGmr2m0dwbGKhYH18Bhf2MN8wmaDG  
ofHU+SfHSiP2FGPwLdx9twenEoYt9fHaiXMLMx79uJ5jIab6eaZvTc9/NZv+KNmNzXMek9XNDT6y  
2p2PPbEv/SCWMO8586IvxaUCJ0cgVMhPTRA5u4WOpsj+GWivaF+KD4mTWSCVUBBhrPJoAiwi2kdY  
Vn6pBodEqa0S6rl9xhJkvsJMxKADA4J00F8WccLFmJUP2RKx+M1GVxizEh8zj1DFCB2wbQQ1uY11  
R/kYBDWL3hfpzmzkGjSbeV/mXCthZDK4K4dJHC01+uyIOVsvp3BrT3O++TLF6eEBRhvbIE+PMJb3e  
XByir4TaleEGK6+W85VgvcT6hW6PcqcjeQ+++OsDliwy7/pYs5v+bDWyLvdgZdHiXbZtB57nqILDe  
HuEjS0Pg15HfFHxPdNCNuYQw/FZbXVrt30xf9vw2n0BbR920FaT1hZwsYJIA0U0g+Q2VQ5rDTZVa  
n5CGddRs5i/s/xQ/NuOmKm+YFZui8/p2CYgZckhYlM+mzhaRWQKKVytgKpdgtHX3/P5XAGV5nBp7  
trEMOuPK30NLJTg19ZUBb7Kk3pMKzmsomJreqVm93IRYSVsPN7agGeL9RC4WnG1MeoJ37K4I8XB0  
cT4u5api66LFVJhe7/d3f3Ib/zEqNlNdbPes/Xsv/9pxy7TnXgxEVYrVXTMEdoKepvoCRMxIuzI5  
nRtH63RBGLWEHmbsuEFaPQR5b41Uy0V86YCUbsnP69UfV9uqG1qkQbJusKOFX9cG01gjdIyIMV0i  
t8RQmcRv6WC38zFsoSwZrRYQMm2OHEZ4eamY40pGoqwab5X71t5jqodMsZEYCqb83SX56aAmv6qY  
AGkPGERLWbzSg4R+V+aaRTbFVN7Oo2xcE9Rb7TV7N3c7CE/ZvljIn8H+Vu0Heyd29WDP78xhO/Jg  
dXpacnlycGd8WQ2fdDwN9pd1UD1nuy4tkzF+J41WdhPX3Hdn8sTNDdtX0KiwqzVhHY/+KK2Sj41  
N9t6GApLbQnGMX3r01Ka2stu6v6vxarmwhYYUgGhQibyqq1JJG3C5sSMPI71gwz+BO04AhQKwJMS  
wYzAVJLwFdoiiodhCgKVdjCtZov+sbBtLf/nczHYdNccJiOT8UC9RLIfTWcq/NdkT0t9XHYNYC38f

B8uKyTA/tsWTK/nKpMaBLX21+6MO+LT+vpYX6yOd18AGsiLkefsjzffFKhFmTS0tDNnZ2f7cZkc1Z  
vA9W8Vq91201+37QReV00VSeuKLDgSsPTF3ZWOKye10QtmgBTjtDmRHKyNVjr8+Phn17cumb5kca  
e0oE1peS6uHjMMJbY0jMMmoHv1SoToNFhJINTvlfdWNazV/qazvmWsLuGxeQ/xgW8xkzq4hoyvQ5  
KsSELHmWMLmmt0Eeqe3qbTK+s76v+r6WqjEJrkv1YT1hd1ZU8XriAyAU05qjPhZRJP5pJAC3MZsuk  
OcpEmoSxymOnruNo2efRRJliJcn8LtdhD3xXeXAOTk+7PzTf/hXnGZXNcx6b8LrcF5Visk6W1ISx  
cXuxh780kZ70FbTEGLmscDB0ZoCAQZV0dz2uvEbWoigPqTtqy+qaRSatRoBEojQIQ4Lkz4RgBAAp  
Z2uMVyp1rD+XMND9GULFFpW0sW6MF9t50y7e076oHXto3S0Onlt8lZqxtPcJn271Y2cT87nsNkfK  
0zJHlq5D306Fa8hh46ViWluV4aRmUMua402V/I+CwFpSw6pilVXDFSJCR7HMQZnLooZPMZikeL04  
ONaVly5A7Q6nX6vJ3844E6zsqwvuDr4HDaycs2jhcVdnY3ZUP04fv7yoJax2tTgLtSc9e2Q1YXK  
ADtNWNy3KJdGXOSJqajBf68f2WYk+kVAasGnpRRY7WfSbNVwGPl8vtKywGpcghjNBuWpK8qPnJts  
yoKcJiuqY1PWmnlkmFma82ayqJ3o8laUWzr9Up9kPyPKR45RDTAxfoAuod+iMApdfxkp9PCahZhF  
rF6sCNgl8c4K2FuV306rcEIVn+Ak9lCGPdUTXDMks92N9DvsZd2EYin/Tny4KbAvwHrC7t+4wf+  
pPk8NPkFJuJbbRaQ6TzCcJOH36rRyNZWC3XTWmlXGVPCvDSH61L06PKnCqZy5i0tqntJd8JK7P+1  
+CkWr8ifF02qIzOYv6y0hY5vd8a+mreKuBLYyorqgpToM/UD69NNmb6BbWaupx8rq+BeVzHpGkUk  
Lwi72tXpqeWbXrMs9KCv8vyws5OUWlnULRciPnf0TxxzMAT2FMbwfNN51BeThG0+3KI2EiFnalGw  
drmyeQUeRYzuh1YRfWlDVgbsL7wwaUk77Q63SCPlgiENE8P9rG7t4uwK74pm8HddUTVdZ73Aa2aB  
Qqe5lBpddJLRNw8tLSMUEFZrs1jPkZzq5YBER8rq2o4WJjYMkxM01qY22JahiBms3euFb8G+OWap  
Wlea2Co5uVNhVR9d2NYyPj4dceNtkBnBFmB/+fmfprRatSCQSjXvCf1150UnrzPQg6BtvjohYCU1  
wje9z44mQmIK4w3PRzvH8AXk5pCa0cnSy1JpO+73Ojg704Y7Za31+ysJsD0wv8xHZxfLiaBK+yyF  
P8riiL0ugNNgdhrs5FJS7OhTX5Tc6g0NS2TdhGHVoFk07w1SzL3yCiusmZhghk76ZLXW/LJh/e7Hq  
PzJIGYn0OvVAoBSdcyqtS8qP5NDoyqNWLe5LtSn5aX+rJqtUnPc+LOuif7S12WBxvNvVolzE31e+  
718BfafZ3SyNNv0PBNV5vXZ0cPUzu3TFY5nS218ZzeS59uYnkxQ5rb49xu4c3CC2zfuaCCqWQ1YX  
/CVxNmF46M74pi2cXR0gr5sOqvOtdhAfUzZuJZd6+anOUqQ8rtubJJalekUOJbZeaTq165NX1Spm  
tC2sJOyNDtp6kQZVwNBPATWwNH/q3dZqI9LM5blhxVLCHK5KlGwNLGkqmEiJm2q4OXiYzW9w/PDd  
tbXLjRXzCIKoxylf9m89QGqD57a8nCDosu2lOWqRaCPI6NBaxVJoOn8hp6/RYG/RAH0wU8ObSy1  
JQtnjt3DlmeYDmfYVfciP07xzg6nfjNzmrA+oKv+elszKojNm7HwniBW5lyPfqZtqkQIQMSCIZtb  
TUzK6yzE/RLCV4GfKDYDgoiuy40r2mpnZyrr8qmcQPoDBZ7UqvcBJkqcxgY4JLAekPJ/Dnrhym/w  
gHLGoAQDKgLBnaoRCr/1wNE+2ANyKHBRPo/2TJkdGtdJPE80/NwcB31jTWAJH5AVcR6/yJLhNQDV  
VmKFjFLIvuczSN/m+Sshxhv9BFHKfIoRtBp6yyeQn7XH44xn2dZs7MasL7gq6iq+XhUms6X6GgdL  
RQg6jfw1frPba+LHEwhrmqZvbCPpolKhbA0XR1Lz0w617UyL5+kH+i1TzWDbz4OeZqZ1NxnZmXwrq  
rVpahrUmYz31NNCvHL1Paen3kry36+TU5NYdyNDGMNWE3w6GHBi2mk2VpHp0sTyGIQiqDma/Y45  
DLzahmjz9tsMKDbTb9WQMr3TOXDPEkUsAR2FC/FN22JRyDMXBrNi8ViqamdoN0MpWvAeg+WbMQZS  
/BS2Xy65cQ3y6KlyiD9Q5qYhRb1CfgcZTj107nhafqS0Zjaod9KELH21w3MdZGtKwKYhRG2EejWV  
jlhrHoNYGU75kUZWbY82CxNpL/phNoIoIUQTNUoLKmn3201X19fx11GVfeTZrpxTRW8bKkr1oGpq  
qiNOqEcBlrPyy/9uZjUaarBKVfYlJlYtgavasOulaOPCcm2stKY7qH46RPx7VlHzZJHpn4mpzNc3  
z/FMk5QpA2xNtHgFWx741n840tTKdTrGadNENhRHphzJnCWPimiKEWjtZbMtam6u+Aq1em8a2K  
qC1FLR2Fa6LIBjgSU0+NaVZXrqfkiDSgn9X/U2NImuAKYUTdlHGKXlOdt0QVJRqYcG+RpUzTb84H  
pUkHBXxrtTnJUqhLI+cdcXrPK4WZJj64bK+2wS0NoVpwjNwxuCTU2pQqSVmrTVJkDknFdZrcx3aX  
EAwZnmNQDBMFmb7nJlFmuaZrMQ/LxPs7D7YbXZWA9YX3kwp8VxaIs3zNGCdd8m8I3OUjqvstLZGh  
YGoUWiZnlQ+0PGf75xR8DKaK0xKlQYW9GvFD/1a9VVjUxRBM5i6vgw++fJ7YWayaeUE8VAQwFU8I  
CoT0a2Dgfi2Ky2GYIcNkVgyGMQGcwGIVXkajLLVbi5M5JlPSb9WLYJSI81Wve7S0dk6p1BCSw/hP  
D8WkoeVrdfyTHKK0V/LRKY0qqpi0E1eayZnSkvM3/lypjKmPLAQpqcIggB+nOfkkg1wo871D7/Hu  
/DIG/LGDG7WC7YsL7xNBlsIY5A5okWENMjJdW6KDZHCYWECEZVg0mks27KHiAs3DWm7DH8IK+oAnt  
70WyrBv/FYmbtmf6rXhBF35WVfVI2q5DW0EaGmxvsXbNKXlWioDSilhBnXYtbqow578r0D12lpA  
wH0MX011Wu237FVzg31MIBGq01Aqa5sDVSZwJcJOqmKIq2BtSmv2k61Ke7gZAAFO12z0JHQJkKA  
EcDeX9yClBbqpD7ruTvM6fekzyqFXAYdAnfC7QRn4x9cnLSn995atzsrgasL+hq9YZTL/DTQPxBF  
u1HSYH9/X0tatCILLtWCFABhio1WKZ8jxHYgr6nMq9rNH8d46/asnEpDapsK76e7GS5LYD0uqqbp  
L2rBLGHy9aOMXhEUFcsA2R/Kx1Qb/18ch0W79dMrXQob0ZpfTCft+TcHCfQQcmmespjttSkmdy1P  
0ybl4dMYor7qaJrPVBdDd1RD/XrdiWV2tFkqVWvoe0FNOWiVXf1TY5pnqWSSpsb6lg+Hy1NK13Y  
pptS2ogJvRyFgVHT69eanZXYwa/oKvdHSXR6WgqKQzxOa5pBU7VBD010x1jMmrABBY2uYlu/5K  
jTak01VBeL/ceF12Z+CiPchKGhK07xWhmYetFJJF/qfJpZbaSdPrSauq2Y3/dWa/a6MBivrwUR2e  
WbzNTGqy5RQsbY42XYnZrXNEZHiH9vymIqlyRrBni9Sre676R4+N0FbZdrzqqY4AWoz/1rr6A2+X  
t8KtNWuqii96qNsC6OK+TyZxFr1xJpkzqj1WraYyT0c33mmqWJqmPWFXX/1W39YCMpdMcdXfoki  
5Em4utVJkp7t2pJI71iouZppvUO7EopqJSvOVf5K12T9yQA1hVMFRXyEZhraA1/ZdQHfZYMVhkgV  
sKcPBCokaQdpPQbyZLaeSM/810Fa81Zqowui3laesYEJvNCTGUVYmP8uLQ0VcMvvm52AFxRLcOih  
1Ib4kMDeEtfsI720EOfOePS5HzL3IyTpD+eyTnAie25uAKuMHacVGpZTGaJMG+lxM2JBLG8NdYRM  
y874/jIZjXM+kIv2d9LyqssV5GqA3JjctZMO/A0faKtZILQbDZR4bPCq5ThsjxCcXjT6Ckp+9YaT

aXMC1MsjgDD6Y7UF1T/WBvASwVtncZwvJbpXaXaYGM6emqX/T2FppHyxUxV+jMxL5nvzMXUTgUME  
VabiLCYxnkiZmiSaA2xoy1tGVJ5HC0FT9NALprkWom1zsfWMM3xrFwq6kKB7ZbQSiT10VlwIddhA  
Drk4KzSjPQotY3PUiZl4Gn/dImNXkeLPhI5xOaLBL3toQ5zDhzrVrOzGrC+8H9Q2/43wq6PpFkVZ  
Mul2MaB1r6WTtdULMmmnd58WnyyWoM/QoeYnk6wiiNQDCJOBVCUWmEUVcxECooNN4fYOHDfChIH7  
e0LApy2ydlqGsVsv1RZNqEJG6npyRRJVvH401ZRs3g5U7kUDp7KxFy+feOqHCicIperEr8tjNsK5  
HBZztUE3djsY9Traamj4wtHL1bqs/YGfTHT28bKxZ8pH6qKPxvrs8oo9pP0eXCoNrExlwOPFoS8T  
Dko5quVStxQ54k+albwdeVotzrCqLGyLv3ZsmyneVmdNjurAesLv/zWP4WX/H3B0phiYrFsFDIrg  
y40dVdH11XVj3NtCvHd5tMj7XNt8Za+RTFc5lgzjEafleYJVmpYj4DMpUvPu7JbYSjXfjKpH+us  
J7+ITt4yIDMt1pmHCPByRGSy7jQ6qWy9BctSss5MCLLF90KSTJDTqrl+QVAcjjcd+khnZR+49YhF  
skcw56HXuBpoCk5OUW3m8mB0VoPxjKdPZzVaaqm9MT0zep1wYztqpZqUxvshj3MxY+P5VDJK0+HP  
qsvJodNnifP+7xU+GcLXSLvubacVN7DfrOxGrC+4Ct0wsmyxkI28ZjscCpMpWMr5HeLySGOnnkCv  
d2Xo70xh97WrjFp2QXD1IrY00lqidnhHcwPbynj+Ny0nHMjgLNdMQ9nc5UntYYbykrMoVZ2qfpJK  
rhtJVo2yMcybZSk6TCZE7oCmN6qswW8zVHLWCLjx7hd2dswhabYzPnFXVRZffF94spZqdn9LCh8  
He8OMViOZXnPxWwJ2jJ8w+6woyBKbSo67W2E31ciqrJz1k66NqZF1TkBOFypXW/mRxePqPaBY3pV  
JjZFeujXOs7iWUhr7cdCNN7Pfl5kgqY183OasD6gq+/8T/+s/qHvvrzDwQK192ixqA/guWx0dtFd  
LSPzb2XYXj5FVoo4LgsqrelZJBVS2QWP2yhNbqI3Ys7SJYLM91NzOksilFGU/19V1vcNPhEv5Ump  
oC01GYBqKYv1fMJIA6bUnEJMSSpE5pWC238Ho83EXS68MXs7Q4vot8bod1tmz5bWraUZgl6GLQ98  
VLXynDV8LYAfld1k2Ynx+LzRlitYjV/KQZXR6uYLJWBow6TvK8404FauZjlacUG9FiljAlKlhhHn  
IHj+8LiprmAPIylm8SV0MfXJmi1+qof/Hdls7MasN6T1e8M/17ghr99uH+t3+n0NPdJP3K0e1EYa  
UtM3MIEh8RcZN5TRRY0n5QpENMWlwsghbaeEK8Bz2mxB68iVe2paOr4pbmf7ndYQsTopj7Vggaka+  
pnaKM70iPikpfiovpHbR6vlaI1xd7PWua9UGGTfKev1B0dFZUxXDRUsQg+B1Rar3hQ8hCwNdAv02  
jtYRQTMJ8da96t1xgwAr3PJlgoxulrCqJIxAthETN+VfPUZsRZAp3IgJGLiO+2hmMXJ8yMlQ6+vY  
0RcFV+z0PJbTXFWa9Z7tzqDrT9tT2/Oq43Nvh8GWqzAQVRVRbHt0ohl26YUjyMu2DJGny6lnEmSG  
zYVPzW+c0vM4p1Kg/Y3B+hvbIrZ2UXN8j0BsSWmoq15U1dLEAl+9VWTCBmVGTjJPDN+YVWaua9Ka  
/QTBZgc9pgsQ2HZjhYwOL4DThHQFjoeApp+LdUvzeNIgz4dRrAJ/HakwPd1viwDVzTJTWueAN5nW  
WGtLXE8XJhfzVI5gNajOFbim1MBkQOt2LyeC3B5X0a/3drGuVaASE6asCfs3u82keAGrPdu/fXv/  
NHyd777K95rxYt3ZIGw199WFUNOKFc1Q8voG1XCeNnpIRYHh8JiY1lib54Egw/GtG3jm2rM4vLUPh  
72fgp3ek/tyCDi4+Oij6GohhK0leXVoChq0cohqgvMj5LNjZLmrQErEFF0tZwIMMqCjoDt4dh+hs  
GN/74r4ziMxT12NImdHEyC+CzuHRbwQ0AXoDQMdhux0B/KcpmiBBRYWSyoJLgF4LeytGsIqq2haA  
FHm60ixGYLFx5Xynr115FiLNKp6XTNTyUuz0B0/OU8T3Lc9MLIy/d5cyPzvNzuqAes9XZc2B19hZ  
+M33YzMuDo2BxJ4VqmNJDleeK/pbefxOEzt9G98hosV1M8+0e/jj/6wNP4WDRE99Kr8Ht/+D6Mt  
vq4b2sXrwtiPCIG4fU//mOcf+1rhAF74sPONcBTCwvb0SmqxQnKkzsoxa9M40rMyxTz2UquzZGRF  
pbTE9y+Psfe2UtoDzq4cXwNUZThg8/cQNEb48Yz78PZURcPjhW8sjuEn/s4eWKCKnkCFx59EMotL  
e080FpgP9TUIjK9T7tQmV23imlpNDHWzp1ZY1nOMIJvvMohUmWl39F+zGC3HU7Zm/tWxCGRLGwOYm  
w1DF5fP78z3o/yJZjc1YL2na3dn/E2R0+nPDm6paUm1B21hE/PTUpWWBU6evYrMbePd//ZnMUKdX  
HrT2/HoF34hkoMFWhsPIpnd9+Kpa9fwps/7Uoycqficc5RPvc+My2DXjTAzlhPTCBATUS0XKOX/+  
ZKC2bH41SWmSzE/5YBYUt+o8LC50RHfcopcrv/6t34ZThfCbrcjPP3sBL/zS7+Mhx57BIvWIp74k  
8fxuu0Ozo1HsAfn8MxHn8bOmWpsnD+vHTYlFwqqQoh5b6W1qirSBC7K4vkItapW1EYnzhWWThZHA  
sptVfY3LYK18XPfz091Oyx+wBk5BEKfg7kEsMkqe9X99xfNbmraes9W/Yf/2gqRf/mZM/cF+9M7p  
oMEsWk2Fz+tXC2wuvpRLE4j3L79NAZ+F3ubI1w6fxbhA6/CH/zzX1K1v0cfeQjnL34uqniF85f7G  
Nohjm/sYPax5zC+chk2e8zWA5MhZis46Dir1dQtFgvxXYVdBbC5tULOpvjPkuT/0X7tG/CKz3kre  
ruckbrCwfs/KuYnh2Vt4+yVV2Jw7iGczXaxOLmK937gTzDqOdjptDE9mcITP7k7HGnkmC5ona+M/  
KnWE5suHR23sVaiYM2zJ+zJlgFPm+rF0m+1VD2CE+FZK8w2917oYSToYVPAT62nthui2D/tDax6Q  
y5z0uyqBqz3Z1VVR+y8HuxUWEXYxRd/lWlg6VzAFSA/uQW/NcL1N96P+xiUSUskwqz1ZIBsA7+Nv  
7FX4kZ0B488tCEMFICe+KGDp2pE02NsJPZw5uFPE58xkksRHB0d/VhqY7mZxVpbRta0EnOYxbiZI  
JRDkY+eu4MxpWesVlhc/7gGfzqdHQzyCNMqwtD9y7fi9p1jJNcex5U9MdfTmUqkTk6XSIWpz9Vd+  
ELrtTx3Z2sEry9+rLB1qRVatja9c54s576atrpCvzi73eNsG46XpLiatZnTWr47aNmVgNfHdqeFv  
W4LroC3x7Y++ZtU01X/aHH4d64A39Vsqqas92RZjttel+1vgpGgHoUZAfULUOKKiys55B6mdtuUL  
LAXbgmFJRkdLecR9vWztQJiihAMSnl17q7S3gPPITelcfEHBaWWH6aJnVHHA4FvAttmavrUxXgZ  
o8GGSqkmLcANCLKPPKZr9fKJ7/OsHzfH60+/qze542Vi0e8GLOrH8ejVYXWZg5b/F+nb2MYbAsYB  
0Y2xnfgkA3ltVariH0BRuBN3oKXZsjEnrDqdSE/c75sqJfDyrfMNL1xv68MSx83cBz0A08Y1EFP/  
077BbADOQH63VDnxZYpyyftQEYML0QD1gas92wVmW0VVeCENs6c24NNKROXozBCo7jPQchagkeY5  
qYtzeu4R/Hf+iHaoSVmrGx+29Ts2htb8kFtQlxgOOUKXjHXyHLtr8XTtGF0XZRwtX+VQRsvRztNB  
WAOFnCuzYejHjrDDQEx87VyoDBAJozbFnbd7sVGLoWQytmqNpQzpmMmq8v9syZV92JVqSpXOCn90  
8Kkn1hVte4UstnOp5Fft8zmEC1vgaEwr6s1Eo4Gkgj8jXaAUF7XhXaIoVeJGzjyesQjFrchWcnhJ  
N9Dz21kXRqw3ksZ2MosH2mwsYlhbywbPINFGRQ2m1umWqeuY73NSK6C1HpMqqxfaleLMnAQGrkUF  
jjYLLpuIaw407yn3e0JILt6LYv9oeXCPDV9R3vdrdPKhfE89ScDYa9c7pNNbiORawfDLR15UdRyH



5edcT3UdlldTMERdhZ5KxOhgKblfkkSwc04DKLXDx+EoyXXOVqfkFCbHSzO3LO/+GSrN1YZi8sZyf  
nCc3DLT3nNhVF8jzxd3N9EqU4xoAut1jcB5Ke9B5V2ymddsKLOaftZ7EWDqbyb15hiV10coYAVb0  
GCUAnV2DTdzYR31yhICEiNSn7xvDQpy/JYSOH12mgPehj2LRWKcCjvwtGNLORflxxWqm/mmpRK7  
awlQistVAgf1Ghoax6z7dnYsDL46TEQHavsqRY+6PS6zEye055UaNluWa2b0hnNdRzV+aW/qa17j  
mWmBFBreA3O2qrWsqrlluFcn1/Yv+u10Rf2bMkb4MHRYhWW3Gtva4jVnZsqQWqH4Xouj7xuYVSPJ  
jflKuWa6a/+pNXsqoZZ7w1Yu4PUSkfisPbUv3NL0/9pfm0g9Vslc0I3ey+So6xCF61VLhhhW2pr  
sCRjSij9YhIo85ghS2Vf9FRkMjMVDot9y9N2aKCzNOcrsf6HIY+BV1ewtlc/qsYTaFE8Wqy6QzW  
i0YUTZWQrEdl10yLFqQnwXaNysvQwBU8EaeG61glw2QxUL5fWKOfH8SCzL1r53pmzcVqCRa1oOv  
l2i3Wlr3+ula0eojo9RrirkWzF6/S46HqVsxE3o+KhuzeH35fFnrzSbqgHrPQowtQdVvQgye61gq  
CM0WAhhrrwdS6fBh0yCu3zk5nDkVFuaTffN0Le9ptj7TJHWZGCLSVXpwjL5vbTR562ypEViL+RCVJ  
HXMfHRCtCxfR/JNs7eYrSxS4H04R8crXTiLU1UxpAgbm+ZtMT0VospqUC1glvZyTo+tc1oFtDsp5  
fWWYr46qlZYa9634DR2birKyfC1UZzcrx2WqjkeejTjkIPg3YXWVpjtljCF30cJnayXMGX1xW25  
E4C7tZohOVqAe/MEM5eA9YGrPcKrHuvqK2DD+UoVmpSejqkyTaCESGbwmlD2rZ1PGpafCmlKhOc  
qtVzVAVIHScRq36vwRyxfSIqt/LtfJSFSeoGUXd4loAzhk2qiCjahSpajdVrCryxcSkJm+5nla+1  
ksCr+GLWUtyXM1NdZLcl8qMLHioal07az+valip/jFHZtjMHWtt87qs8G7Em/50YJsxHcLYvJLnr  
RAGnuoBt+VnoZgJ3XZbyzCz6QTFItVIOVVtcaZukli7VOCld3u5di5WDe7qgHrvYkv3X7SEmbTC  
gCXwSUV2A5UhZ9kSb0jzmVFlawlgwtTkcQmMc9Rv5M0rlq+9HfTI/9p66/Wc3DE57XySmVOHQl57  
shGKikSzL4nQDKT4jjFjRPlwKYCmtvskmEUuvZUTIlyMGqkKyt7erCYca6Vmb70+10VKHeM6hN9V  
duMxOADXctFvm421yZ4ywjGqaMp74cjMDgcmeoWfstBR4AoZiQOuAmOWABJvJLXWiBzU/WrzeDoC  
p4wePu+B5uumwas9xKtJQtiff5562qlCvVkI1UldGqjhe+/TNFoK0Ppz2rL+K62u07FGLPWUj8yV  
ZDquAuLvmJm9JrkvkUcoyzkKd223CdWRqOCTpE+rXQQsxuyf7ZnxLfZzE69RYKUDMmCfM/Rulwdt  
qxtbnK/wgxPpiYUu4KIPmt9Pb40PRA4FitRqlqPGgWxCr6sB0LTD/ZaHvp5ICbuQPXRH115eaeFi  
4yHkJEoaFs+P9XHVME61qTM7MI7/+DvNRuqAeu9M4OzyNLJa6WtKhAsf1B2Za2sHWiVkuVUT2HM0C  
KQTzBmQUcFSGF/v7rxU9XHFHxVLsU5XQmqseEqIZ6pgJF/RxyaZ301h8Yd6kUj1PSRpQaJvpF6ozh  
Kla2MxMbyNjm5mrtoqw8f6aC7bN0CyrSLV5nKMzFHReV4CfmRpgdg4xWMX0DQXEYQS/qU5Bi8EOW  
ipVylY9erE6sKrTEZ+0Da/bhy/XDpMC6WSuqaLuqIs8i4wU6nooVhFFCHutI+/s5e9odlQDlnsXD  
S5K26LagjVdTzh3dU6MynlyYBTNTHafUFq0Xk8fV9PXMekQYD2MuFqPrjBqEWRYDq3SqiH VATbsy  
YI11TyiKqGYkDRzHQGNaxlgJ3miQ5fTYiFg6KK3MRQT09EiBU3dWMY0pknOtAwfo9FfRodtoxVc8  
WeZEffWQBhb4taT47QvlyYzudU183oqmuL6fkgNcOt1VUFCDi45FOYntxHNJnJQEPiuNie4muItT  
Z6Vush9zqL9GW9v6oIbsN5DKziZV9p+zWqgOjWsynrdu0OcdECra+allqY7hWWHhCALCZY1VjDql  
WkW52ApzLpcCJlvV9WHm2IShADmWH1vawG2tVR5YQlirGLcZgcE2NkaiWcEUzxba6035Y7jtttYUm  
cCQPH+rszZ1LZ1Nq1+6QKJAtel/B20dMUKLnwcFI9f8XqsZvbYkmdJaP29pmQILBqvYu2q3Qo1CH  
x0nWE1Zb7wy8284KY9DgrIU/U5f/27V7BS91539V81uasB6bleRV3UWA1sXUTGXyT+zZabDqc/HW  
17LsI5hXjE/aQpTEpTtyXvkZKwRYd34jKkSpIGjZX+aRvEEOGebViI+sTCmK2ZkkVLiRUBH97KwV  
L4FHLUYBhr5pYJwJqCJaX7q6A5hyvYAYWBM2+g4SUZi81wa2hYLINYm8NrONbdqq2w4hb4ZJfbU3  
2agS4cmq7iEq2M7NIqsw7EYdXa0HNIiy2uTQYn5wSkqYfte6KLXCbGkwHe0QLcXaBEHLWE5fm4OX  
vWZ/6jZTA1Y7+0f9TWfX9W/9kMro4JPVs30T20rg63FslXdyR1IKVP1R2sCbzET/3J1Jq11emKud  
pWB+cU5rVSOKMUndDojVbzH2sS2MwF4tBR2bAmjZardyy4YT56TqZrKNXnPqhRGo+L9nKxW6AHhD  
LbhsTVNCy3kkWRZF1PQDHbNvDudoUwTHpmpdaZZTPJn+SG1fz12gwEym+LdvlEUZpCpMvNptceVZ  
YwC+u2xg5EcNq1Wg054A7ndxnL/CIG4AOzMKQ/voDfQpd7/6/+giQQ3YP0E+K2tUUb20/LZ2mgTw  
b77567VfLU0QCSgFBMWq4myZDE/Rh4L83SF8eyBMYc9YaewtX6kY/xEnfgGY4qKD1gIeAiaomyhp  
txLGRsf1DYawhaDT57Rd6qXefIoEmALQ0e8m4Aw8IxfyfQOWbMiK6cmkERWlwOFvuvdemAzSb3Wg  
6Is1kOW+ZwVB2eF5iDCOjilnr5DQ79YyiOzHJ3NLoLWAFZ3KM8bwJuLRbEQ6+DgUADrXB0/9qp3N  
LuoAesnBqxb53/PquJHGQ0uBUisFqolFwKNqDIXqkOLWXifiskrpm8+OUI6vYmq3UHQ34EdCqvmp  
vCBYy0YPWYqxGYDQDTRA7CKqIqXGpwqWp5GX4tsgXytWQtffIfp8pHHe21XzMwT/T81R0vflqefq  
R/LgcYaCCPA89hMXrWcdVpmPWjKwk+N4/zWNRgZvGLJoSZhqclEg1sVEo3/69QUdMtRimuQLacop  
2JBjKH42zb6u27XV3+7PD2A59ZpZzz+kc1v/Mm82UUNW8hyy7K/76upl8qt8ZaqM8aXu2QWZcIE  
nTMnbIbR/y1YnqI5OAZFJ0Wwt2L8Hp9IeKWKQdlzXCcIT9ZKWt6g76qJLJgnxVE1HMQbA+FFSKFp  
ohXXK2GtTo90aomipr5/aEKm62OD1S5wm+ZaK3DSgoBwRI61WBTZ/uippMcYnSXymZvsnoUKFEa  
qF+aLWOpJU6V4emdGkmaZlgFS2JIjVlizraMlUrIpmttA9W5zXzfSfy90g46uMUBRZNXHkQrXT5+  
JUf/uUfbXZQA9ZP2LJe/dYF3v+rvytO4Ds0/WLyMWaoU22qeyoOfhEftTg5RLR/FVkvVo33+1Qh7m  
xqV5SRyaz2e0aXf63ng/JxMy/FCBWuZJsgpBi5gyZIFksUCq+kE0WSpA2Y4prEWXzeZ3oLJzghAx  
Q9lr6mqE9q2KYZg4CmawZ3dQntwDqZ9vXy+7a3SAgtaA6let6jNYcF50lowwcZagfTdf7PmnXMa  
2eNiAbWqIgoB4NjW1qGaKXy+OUMCDvCrh46l+9/Ynuj/+Zm9zRg/cSbwjv3/VX79uI52dDnTOGvm  
XrOIFPNulzWB6cRsukNRLMDtF/xBgRhX4Gqpqqj/BZrMTJoesUdCANxYltp6gdyAYBWEZHRhH0du  
R2KX2m3QxQc20Ef1a4FnK4mYxnpJfOxgMKkeGoNERGJHC0f0eIYQSCvITCC4qW22w1HOpXpEKJvy  
shSYUoZy8JMjjOlS8XzZrGWMmpzOuJrM+1UMg0F06Prsl5Ynlcn4xX6PwzPXbi5+zlvf7N9+dVNU



KkB6yfBFD7zcFnMb06wuH5O84xM+HOza6BmPQpxeozVwT7cnctoiQnqaBucKYTXyK+2v1Ri4lqqh  
6QupBtq3tOpfDNA2fc0SOUJQHLHvr3fenGCilrJ7VLzq/QN2WZxeZzeA6q9b2rzK4psK7gyDURNT  
m5iNL6oEV+Hmr/CljSJWcGkQ5qz1Y7rUF/aNk3RbM8rtRLLTbtgugl+W8DZkdckZnch/rc1gLU8R  
sgRkXIgVGWY6G1h1mvNnYdf+7lnL7/6TrNrGrB+01bl+v+nbOZHuaHZskYz0tJ+0wB2TF/1SPzMC  
DuXHxM266jL56zVilY6gGLC0pzUfCvNSgZ7KIDG4U9ppe3rlhbUV5pjpQpKLuY0MmHX3NZifHbRw  
DHF/axwuqtEyABXqWAPxLSW68ap5lzzZKKyMb3N86b0kAGlfKX3NTNZmb6xVGWClVe19ubWKjF6t  
5+VaSO/1RVzvQ+3PUApvvPRcx8U9zfGwTLFz/3hb817qmCJSb473nrq57722xt94Aasn9zl+L0fE  
sx9tWXVfZbnGYkWX/3RIksQH95E96FP7dxdt2YnpKwtORPSw4JLgKGYymK2EioyB2cWIAyOUQ20  
0KhW5etdYF/CWclh8cIsyfmbpIqfZasLGJBvXuqD6t5WNYO8lyglj67a+xAQFVq6mZ+cgft3obWG  
lOJnw3wHLrMActkeaZgjX6LY3RdmOJhAI30zQg2fWq2BbqtdSdOgVx84lJM8eu3D3Ayj8XclgOh3  
8H4zAW/2SkNWD/5gSY/PCjK8th26j4tWkdnqpr2TEKQE8/HV14p9wsUDKzyoe+nTels3mkFGkV22  
PsqjOgmppoInLfQd4GehTLNtAywFNMzIFOKH+gGG6gCAWWQKAuXdqXBKAWE2gtL3Fm5RRzETBxq  
h0nzngr+46pPFRob60/c86kgKxyPaC5U109XI9S5mvGegKeFkXRZEeuYyMtLY2s9F+2SHSzMWA1E  
Qb3fVeHKfdaHQw3Bg1YG7C+CPzWnUdq+/ofrbTvUxjGWqc32OFC0LpdMRUD4c3kSCuAIL4czVZ1+  
xgtF1BW8oV4ohVIlnKh6Rg7rrUogv4v6ZHlep4qA7Z0lnP0nESLR8sCaY2Cwd9VYpgg7unmiu09  
sA62kkTGPCqOlIKx20jXUzg7Ih/7NTrmmA28BWmaorMz9Y/Y9xrt07JCiZ2AF1GjYLN9lofzVLH+  
bGavJwOV8hBxGZ0Ff8OfHmJjeh+A9YXC2Dr6tfqsnyU5Xq1YxQObbutPa2tUV9F0OKDJ7CazDC49  
BrNXRYHApTUN8CmacruFVf83YxRWduoEGqRP9XwbWR3YuSOsCdZpJYAIq8FFI6KZXNwchlVOsWts  
l1XLD8P2MvKlrqWflpUppNWDgALXneEQF5nfHCEUiVSHA20KjVfXGtJMCuZSqZ9GG1mdFsVHqo1k  
G019bX9T8m21vSSbXraMRcLoS0Wg+/5+rNWf7uJADdgfZGYwrX9A8I0XyFkM6Zagsq6UBisNYTf6  
2mNLpvCs6M7OFh+EPWqgCfbtZs6Q09VUZJlsoJRIHNVo0kL+WmSCtNl8Ryl/DgVE9trd1GFXRSJS  
ZWUXTbvLDW9wrJAFjfk8wiFb0YuVrZcKxADuiVmtl8jaIfae8rGAuZjdeodI9REGSPa60HHxd0mB  
Oo2sWuIgafKdA3ZLIrW26p2oe2B4osXeayHE18CUZauV2nwTCx97F081zSYN2B9cay6tXlqpceLE  
uXYUVUFSyVOOFZDxdAKYc7BDvoXPRx95Cr++Jf+EMvaw3G6wETA+tfe9nnYee1jCFem6N8dnkMtG  
LNYvne8QLmaa4G81x5R8AlFukSaLNj8o89R5gnSNFWTLB07TOVE8QpJnuF0PkGWZRhQ8rTrYoFG  
QSttpjZcxVbU7mZWhMz6+aD9cBk9qZaplywVtaljLoF3VvxVa2wQykJY46zk4idOL0OqpUcDnIwD  
PltrYRKk9nRycnJtza7pAHriyMi/NhfrvP3/uMDB85lqAIKJVky2bwjOMKCZXQHTncXvjDS1kM+X  
v+XHHZ4PR9H1Bb40+eewrVOgMuvfwOCOEU1OYSzfQ7e1WuoOqGA6VXIrrj6B6onnUAsj5r74f4GN9  
CDCcj5FIOyclSvkk5UWKgT3X0Idh5i//204eTrDnzx7HdfjJT7rvvxxsp0hXv7pW8/7tZ3tcyZtI  
z4yga8i5X5o0kf0UdeiT9RiqjV95Imf7KuiPvOrWsko7zNPI6MDRYZu9TEYDnEkvbL2oI+ds4/97  
je96yeXzS5pwPriYdcSf6+2qt+W3d83KRoZC8YZXTSFDbLJ3VAYdmuIvTe+Cq2NH16/CvGWZ2/ja  
L5CuRCzdnMbmZjM7a0zSFZH8v+LcNt92HtbCDBGYvKK77h3RqVc/I99CP2rz8F5+BXCPjHif/8Hc  
C6dh//gK7B84sM4fe/jKPpd3IiWgr+O5mHvf+wSws2esj5zr35/Q86UCHlipqdrjznTNqXkWk8xr  
wjidYGVBA2YYV9FSG3tBDErr6Ya68sI94QtmYU2EaA7e3u7z76mke+tNkdDVhfBkbwnyI9ngsK+  
ty0zLcy2GP1NwVMifqB2gDeEp9QNVbGIw8h/+BTGLzyZbgsjGr95m+j2N5AuS2m7s6DSMXkbQ/GR  
s6lNUJ+7gwsAbGrQ8cT5NvbOHjmw7hw+VHYJ7dwbMfY3T2HZHKKw499GBMxQVeLKT7z7B5eeWkPF  
15+FuMrOypoxqHIZEItN5wvdVwk28kp4k26rEoj5EafWaPETAixSELHGNCvDk1JYmXeEwNqXrcFd  
AZ6jd3zezxQfuUbffXGzMXqvwuhW8Jp3lOm7f/wZx6rOsXtFixy40dvCQIVrCud1k1c6pyagZ  
tr5lZi9S7h+B1YqQHjmnqWPpnszlSjtvnm0awMtbChEGC13JYOaa4L5kFbmJfyLMcnSD78AbRvC  
EPu/yay1QL9lQoPCAu6/QH6D17C5uUx/G6oqRSaAASp5bW1GydP5siU+R1VntBRG3z1fM5aM0MoM  
nkMix8o0P2pMi40zI+LvWjVuJ5t3roDncwizNceuDKv/iir/mOL292RQPWFy+7wvmbYkJ+sPasf  
q06RaUKfzPJVEObA6VKS8vwWKZvXbwgDHSdEvioerZqLnni53oHU7QXS5QfehZZR1jw3H1VIyWHT  
8ISsxbLGZznbuLSM9eQfvAnxReuMBA/MxdTmjnRmqr4Ax+t7b6Y1G0V5aa00/PRA5ZrCJvmsayFF  
Cz2t7XHNdcKqVonnxemfnndZKC6SewUEoa1OVyZOV1hl+Tkthw+Lux2G3VviMXhR99nhd2/2eyGB  
qvw6hV+5tdcy/7gx6/Jjn7U8n0TiaUig9dRitIISWX8WUtnZGG382dh3zw2RfwbsuHJSHLfKhC2n  
c+ExFy4UaXpnfrgOip24UQz1IncZzBAQcEHjmdkVA7EtFDRE+Z2+iH8LfEdKT9Kk5eMycooYWQ2j  
6MSJmW0WfzVSpXe7LWWkouc0jLstsnXBRI0hClFyoZ3NpsLu9I8pmJhIq/Jbnlm8p3vX905f/9nf  
NaX/d1GYb8B64t/VZb1E3aZ/UiVVoG15qKjZmLCfXbZ2OvADYpEfteFvTnSAn4cnJpRGa5R12frm  
QLSE3OYmknDkQZ+qISIO5mRM/VrLfYvOJeGLB46ci648He7yFiCyEBSZRhV5WPW+sYshGQ5YiGPz  
Zny0WFXRqFQ/VQxe7Xpnf9niaLfUsa2nbYZ1dhQy20P2eIEabaE43R4IKR+b/A3PusL/lZTAPH/c  
zUjHz/Rq7P9j4vSeq6oUtMPyolstTGBHRX/No3aOkFctFvtGN8SIN5/Xr53tAXNctedMx6L92t1Z  
EZzrTzWbhdGmSynMr3ubIGT5/HEj06fHyI8t2nEtIVq2f4ml4DHIBCF3Sj3T1EzBSRZVb7qdfFDb  
Q6RukxifhVysnBkRsE6Y6s2SorUceRISvlFthJfd3GkQmzsuAmGO+99+We9/T3NBmjA+tIxhV/1D  
nEB3XeVeZWWRaJCZDqMSQlubQYzR6ItcSzmN2kTdMUK3hrCum8L9aaYmWPxVUCypGPui3gKlbad  
8qJco7gpQ7N4/0zLXRefh7BhTOwaVb76wlv8sUZsB7HeAhrmhBXZUTGswz5aqVdQTR/c2XVSospq  
E+c0z/mqB7mWlXQey0K7tHU7QmYYy2HLBanSodzeVxy1N+99Lbm02/M4Jfc6r/p635u/ns/9vVuU  
b2BDd1sfaPmEaVTCBysmUz1jzT14WuxPmuJ0WaN8AD1sCVg6olDGpneUmEvqxyo4j7SBB5/nvdhe

ZZqIdGXVTYW05aFRn6rZcCq0iuxaXmT3+uwKWF9FjLkearmLlUtaDKbemPzRYUllhIWJoytTfLt/  
lhTNVXOxoKZnEMr5IslNh5+9fvu/6y3R80n34D1JbnEtHxzXlUf86rqsIWggJfq5mfAhsEmBphM8  
UGuMqI6/KkyI8bZj1qpFrCnpX0c11iHXTVjjeIglfoHKqRNMDNCzOJ9h6xLm5uMK0C12mLNkY0+f  
dRCxchtqleUFFBbIk1zZPJ/BoqMGgKmlXSFTouU5wq4fSxj/rYGWr2U0WdmS9zJLUSHRwgc5yi8c  
OGvNp94Ywa/ZffvJv+f1Zb1rUWf1D6dKjCU5uy06vLPQM3bZaq+qPqz/NCEOTmf1cpqI0jGbh5tb  
XPVJOW11HzOBajxStvaHLnNpnKmQb2QNb71GvCcPWVSLdSGYoQ3ExYmMHMGkuQFZmIap+L35jo9X  
QDMLM1aL7jUCeJC/Bytkb00UNg0WSCenKKUL28wuPPqL/nGWfOJN2B9af/xg94v1rCfY0FQzTEXb  
BC3tR7IBI0IVqoFMtBUZFr1pKWKjLgKM2owqTASoJSCMXCVL52vmum4C8vKdXYO3eDKl/vY8kVzm  
vNrQH5fZcbk5s8AFQHP41Snzilg08xEkzmOQw4CttkVlRnGXHEciN1WEXCKRRCKLMqID25iceMGw  
sEo7Vx44H9rPukGrC99dv1vvqKubPfbZcenOn6CudY0NeqAZ16FKeUrVgLmFaokEjDIBQqH63yZX  
L98t9KgsV2TTeV+2UL90Dqbm5GTiMT1zeR+1rrpXUNJWtDAFE29VsyvGOUlK9J3zQjaTgt/S+3Uy  
xRhMx3g7JiGdsekmoqXghI83iJeDbF0VNPo0uRtO7GvHf54Z9uPunGZ/2UWK3+uV9cLfe/16+ty  
zTTdjJTpjbN3VBB8Hit+xkJSH3SsWCb4yxijdpyfIYWNLiGnet4TooTMz1SMJrp6YX2meoBICydM  
4da+2bWK1nSMbNgqfpfaMom1S8GmNRVhplAlwtY89pDRp0nKhPmleozuRT0Ztuc/P7o6SfRk9t+u  
43wwqWfV7l32nzKDVg/JZb3qi+p8Qc/k3MMhg5lokqh7WtwSEdFgtHaREv9tAc8Z5wpWo9YFBCSi  
dnMHqWGdcnAZOgyU1Y2DJqZQJC0liRTWpQW4ZQ6V/xNNodrKaGYy3lqel+ZoiF7UhyNvm2eE+C2i  
onrNHXkZjCWP30TFPTYnrjDoK4Qm97DGe0gdaFK/B7/eZDbszgT53lu15WWJ6mTTTQxIIIXwSLW  
CPMyema0kljWOIX2qmYt0VkB12zjG9dkKAaTUFLtZyoMKEFEamAsLCNKUvFBmFU9qYWZbJuKjCaz  
uTosQLVpG9iYfNUvhdq+jK4ZKkZrL2rRam+bJPe+j1Pcty+eg10EmNja1NzucG5y3C6o71bt259W  
fMJN8z6KbOYT7WdjoAhUgCFNfOWuQ6Foi6v5XNdHvEmTBauixqqUkHsVutKJ5jIcF3QzzWTxzVCb  
JnAEf3STBg2zUotxs8t1kvJASBApDxoIT9LolwHGscCOoJUhl0RmMq6tgKVVUtlacS8KQwu0MZs/  
wRd8Xc3drZVUcLZ3JSvXazyNLDC9nfK0//r5lNuwPqpsWo78+zSFB1kBQRPyPeZdrFw6Li2qbLTx  
cq15pYlSgSsQyG00pi6zMVC856ULPW1uB4MBrHAniCjaJkAOSuKtYgoDfTaZlZNlprobxIJu7IIg  
uDntXPzHPRXCyMyrNKmHCXJ+68Op+jJix2NBur/Vv0+yvEZlPI990JcOHP+0o2bT/+V8+ce+IXmg  
27A+pJfTK2UrgdfTN9IWCyRr7YOSS5VeLswlnXDIRy/NOWAlrWenM7SwJgFusrClcWg0VIHWHBMu  
XO3hrfWEY1xlqsUSw4zyoMMzLwqgUjRxFWSGV+V8qZ2rdFhgpURYz5OGZ2HBa+dpJjeOck408Wo1  
9PUDwv6i/4AwfZ5JG4be2f006wP4mj5LnmbDVgbn/Wlv8QMzRgk4qCogLlWAVYkDEt5Fg0SeS2dh  
M5a3Mpb15CVJiBUriYdDkfXKcQWKDbkOmKw1rZLT75R/SbVosWA90KqokCSJHg6RgHhBabXEFEMw2  
suCiEx+lwlbZqzL0Ppfy4B3FWOxf4AzwwFGG0Ntf4stIBqNMHj1m+D0d7C9dxGe620+XMC13Eu39  
p/+ouaTbpj1pe+zZulMFfmFHSmMzfa4LgbvaVv80lzn0YIqq6X8rLKNlArb4eibJpGWJNjwLcU0r  
dzcjFMVwGuKhtVHw0TmNSYcUMVywsooHFZkXbkff1foOI9M00ZkXlRGSdgyYwQAHh5yvWg6hZ/kO  
L85RtjrcOPWmHMw1v2PYucVr0UlJvrOuQvaIHAYnWA42sRyuWQ74PfLC/6V5tNuwPqSXsKl7xM4f  
LZlewF9z1D+Q9VfyoXaOvqNBQtT7XVNOAiK6oLEk84+TU1UN/2/2XsTKNv06jxwn3PuUPNc9WZNS  
MgSqJkkJIEGJDBgY7Ad2+nEjtOdWuk67nYn3e603Z1e9ortFa/EdjdZnSwbD01jQIGYCWFLQkggR  
gmBxBAQAiSB9MaaXs13PkP/37f//Z9zn9zdiWPj9+S6WqWqd+vWvWf497/3/va3v911Bud8qjOgg  
cs54Q37ntvbhVZwv+e8qU6U6w8UNGJtFd00WZ/PZ6KDPSiABgG3qM7hU+hrzTs96WzvysZymCwdP  
cCump57j92RUR15ze2y5Iy16z7jwLGLpFebka2NdZmeX5C2M+yRbiYERJecPPnM044evXzfYPeN9  
QL2rLXoq6QOJqnU4FmdgTSjPWmnkWxtrsg4BLMR7sLgOGpRSzMwMBp7X9XyAQShPxaG20XTuft3z  
4WtPRAaXNg7gMvNYvdcQY+IEhHApYxT7fwMHkydqWpdgrD4Hj3sYK8tdfc+Lz16SBojTfaz7rnPW  
B9fkK03vE1mD13CSXiHL7nIHx9dVtdXZGFhgZRF5FljYxOQRW2OTjT3veu+sV7gAFMyfgpKDGABs  
ctGEqk3R6SRTqXtDG53b8sZbxLqsCatpH7IFOiBGRUchHivDG/dP7pAceERC62PKpqbmJQFQGIgM  
5t4ojqjXM7ZSTOKeKdQh8C00BeKp84zLkxNyOzCPP8GzKYzeU3Sy2+Uq268ReojU9Lp7MrRyy5nW  
H52bU1mIJnKdh2RWZfH7u5uClQxtjbb1/zLX/3f/sdf/Oe/8pv7d33fWC/Mx8jkRtFe6z1/2QS/L  
46dAUWTUnciecztyV6rJVsg+oOBRJZRSjBoYKUbWHmU+NmpqlCIJ6nkgHzUWWVfB3FQnBvtDpyqH  
nslw8LlvuASo2ka4xo7LnTe2ZLp0XFZuPiYy52bnLu67sLqztz3ycyVr5JLLn+p89B9absN4fClV  
9Drb7h8dn5h0eXbPXKn4F13dnZJm6y7Y//iY48177rrw7/2R3901/YTX3ny9/Zv/L6xXnCPuDayh  
dIIZFkgjo2eVlALayMjtCeEyNvOWJETUofJhbxod4s5KI6hSzN4JFSmaVgGxtkW2KX3DZyTI2LC  
T4BIk4jwlfSoc7e9NCv2nGeu0bd39nYFsfXPhRxxTI6Me28ciotZ/DtuUslPnyVXhrRpTK7uOTC2  
h0ypo5dcowh80+eticZqlblj7HO48tWgKUWC+N6vSmf+8zn5F2/9/t4bbMeJ//nK1/xstNf+eqT9  
+3f/X1jvaAexchoB1qiFBHE/ydLENXJ20X/6oj0pDPcSNrbLdnb2pReq8tBxiA+JL45PXZ/h7Inx  
80AgIpEEeJcVR5qIEikXfclKjQz6LpvwYME6SBH+LpBPaeLjh2Wqel5ajrt9TLZmlyS8ZfdyHGR8  
wePyMhYUzbPbsvEzIxMT09Ly3lOzMxZmD8gbWecSaMuUzPTsr21Ta+eOIP+xCcekve85z2y5Twvl  
KayetF0h/qR66595bVffPwrX9lfaF+R2IYJYe0//vIeg6c/O5ae+fpzSVRbjDalDmqBtRpb5nKIO  
EGWFHlpyvM9hL+7W9Le3ZHO3q70Kb8CYKmnYBGKq84SGB4DHUZgDFUH56ohjoaRk7FXfRi0erJ8e  
pn/PnrskDOyOYknpmSQNuRsx4Xgr7hVJo5dIWtnN2Tp4EEyqs6unZXDR464QxyR1dV1lmimxidkc  
2dbRkdHZWxiVNbX16glVXOG+sjnPio/+7u/K9vbuwzBGS1gqFVSx+vXFpbmrV7UJx9Z318F+8Z6Q  
TzS579UH3z3se8kEh31TFNoCkNUM5hPVxJ2xghb0NDbSlTYeckUjeH4GV0y3ZaCS/Cg6HcF4Z6Dm

1N6XOgvISRft830+oasn1qGfoRcdPFhmXFesjE+yybyVecR+wdfIQeuvVF2yW4qZMblnl3nNdvdthw4cpi1142NLZkEc6lWl01nnBg21XTHvbJyhhEBvO3jj31e7rzzT1lb3eZYS21ScNGA24hqids86omMjIw8euDQgTc890Cn9iVK9431AvCsJ7+eFM989r1I8qPssuHsDJ1KDM+ZNEYpqMa+VEqLghLY53gLtnhDY3inw17VnGiwB55AwnfGmbW7zhNvsjF888yKM/ZUDlx6iUzOOQMDIypusnyzVxxI4xW3ydzSpXJ6ZdkZVV0mXUi7ub1Nr7iweIA9rlubmzI7t0Dj21g/K4sH1lgGOnnihPOWTZe/tuWxzz8id999t5w9u6klJeeV6y42j2MlzYH1hC945vHx0Y9c/tLLf+yDd969PwJ931jP70e+9mzU/8aDz8SSX8Z5NxgORT3gSBvIm6MSj0wwdCQiDCX83IW90B7u9dhEz1ILSjWDVD0r1B76bem78LSLSHl9Vbrdvw545ucX5Jms85wVIq69KQpq5PfJ7NXvVZGJ6fkzKnTMjElztB5e2NTxqammZ/u7e1Jp9OS6dk56XW6BJa0ntqXkyePy5gLg9ec1/7CY5+Tu+66S3Z3W/oZ4pVb4VXpUZuS+HAYxopjmZgYf/9bf/AHfvqXf+nX8v0VsW+s528YfPqbUfqth56Ja43LEFZChBtEfRgdbg9GZ9Qm5t1zTZZBUCBFLlOmupRvAWsp7+45I24R1AEpApPSe6096bu8lvNqID86MqaDkqnNFJE73Jq9QvpLr5DxQxcxTN5yOenU/Cx/f3Z1jchvs910Ye8GDQt56e72Do97enaGBry+tiL1xoisLj+Wxx57jB6111dWFHjMRLyh2kjPKvTIdci+NGUav8bOYECSF1ZPvOvRzz/xD/dXx4afP7ehMPfVwy++SB4RaK9FRDxzRVUUrDERYGIIItqYW+N+D201Yd8rJpP3qKTP3RciZvBmNp280ZT60LxyzE47GAae740f1M3Jl8rY0St1ZmJK1jY3ZG93V+YOHGBYvb01K4eOXcT3PHPMDDlrw+WkZ8+elXFn8KPjYzTglt60jI6NyYkTp5xHfUQ+9MEPSRvMq rzghlPDdACpa7N7riEwIgSgWqQ8uvOFNhQkYvb2Wv/glptv3P70Zx79n/dXxb5nPS8fj/9f/2LkqiMTx2ujo4uYjwpeLhhNQIMzEulHODc1Hp1glwz1Ek6Jc5Usxi7Xo7kjW2aQkS0EyfqqzaLqqB1FrwLgzNVbuct/0otfJdv2ATM8vStIckfX1dY7yGBkd1larxfddnF8g6QLeEmR8vM+my1Xn0VjuNoz15WUaPb50HH90Pu9y1LvU+jBpjcaisgnpeA2MlOOWCx3PEXlvi6FYsX8fbAbIeWfmZn7xU5985Nf3V8a+Zz3vHjtPf/XObPGVi7VGXafF2S+gdO+8HEZjAKKJ3WKHtHANrXIIzslUGmi4jLGMzkzh6nEzOCarC9/PDrCJn7IO5K6S78HIZOONfOnCYRrO6vEIVCbBn8+yGM54apVm6Lifd2dmSOWfQvV5HWrttWVpaoiGe8EASqI4nT56WRx/9rHzorntcTqzgFoy0gfNI1CEFWiPKSszg0hMFJULPyFKRN3fOo71J1giMLacz/8vbbbhr5xMof/eX91bFvrOfN4+17/2D+2e+evPElLz0gS+OTUi9UhZBeC2R7yJH2B+xEzaIuJ4sX0A6GFKifkUoPzJGRLuwFCQJ27kLNOmLIjZpkjXEpLn+T7GTOQzfHZG52lnRFkBRm5+eI1gLVBY83adZ1c22TjQFTUzPOYHcIIAHxhdddPX1GRicnXMjadv/uyJcef0I+8pF7GaqqRjXlXpkCEqh5AmUosrEa7rhdHEADh8HD46JzhyGxKE+5n/e5aeDcV1bWf+n617764tdc95q/++/7e/uh3/7xnoegEubZx5o99LF9TMrsnDwYuf1GLKHgdabKteC0Tf1Gr2oMn8HJOeLywOh04RyDbWGGQRT4iVi7ROGQNRh3EsluvS1srPblcmZKWmOjsiuy01hrAhp8X3XedCDhw8RAFo+uSqNkURGgXOybtbFN8j4QXxj26uqqTIIyNM28FovUfx56QO+58r3Q6HW4UFnKPJE0CWgzTUBaBAiqMN9IG+ziJw++hBwXkG1PXYah4D2wO/LtsR8F8/+r1HZ2+6+fOf/+xnHvsrX9bZV4r4S3ycfuz+1288/8wVhw4dkme+/YxkUC4s0jBekfpLQFDxcw6VhrYzzC5JDvRGqMPmAYVGZMoRJRrBur0s2dZfKv71D61fdLp1+JNOLi0SCAQrhMTc3J51W1+WcxaXDbGdbObMsUzOjLnccoUGCWgjA4x0Z2sbBAZZWVs1IvypT39a7vjAe8iYCuqKbBzISfC3kBYIMr7MY+IY0eyQ3yH0rfnaK9r58DeZL9xod9HAefa9Hz55/ORXrr325XP7xrr/+Et7tNdOvj/Ku1PI3x7/5mlpb60SiSlyP3qx0DCYnThwsQCdB10aLaiHRactmcsni4GfmgiXMyiFJfclPWR3yLx7FEKoc0vLvBvz5w6KePj4zQ2kBWQ087NzbhwdtcZ8SoNuCgSWTu7Sq+Lks1zzz2nYJE7jLW1NZmYGP773/AGer7WGDfux68YJIUzHeRqzJfZWKopQFiI8B5wTgVRBrla9j47sdJm1bG5iSQnNG/QwShrKz0Zf1edvrg177qv9k31v3H9/yx/syXf2J39dRLm+Nj8h0Xen57dVe2V1ck39smoaHgVPKM3pXDjP3YC/S7srba2pRBe8d52g7RX04nX7pc5Mp3SO3Q1dL3t3Zyeoph6sb6JoEjhKswVBjm2NgYobv4/cyM87Sdnvv3thxcWqSRfPfz79C48Xt42omJCfnYA5+Ue++9V3NRT2yAt0zzwutAZdQhVsmCoVe13BSbd91WCHOLnN4WxkuvS6+c8gvn3Kw1CU6hjtsYBQDWaHZ6vX933bWv/OTNN91w1X7Ouv/4nj3OHv/Or4zU4ua2Cxu/8dwZObE5kGXn+WYPHZWxxqS4mJm1HdaexhF7TgH+MM+DQj/AP7xPIbVofF6SS26QZOQFyerdOq+1C2FEaAXJN5KUTU5M0OoSwi4vzNAiUYvCaJBnjzyMjY/wdQmXmp+5vzm5uUMh7fHxS7r3vPnngox9176ceVQn5LpTF/FhsKB4JpmoiwtnMgCSVolEwSeNcm9zgNeIaX6dk/5hyp5BFReINTQUGDq1k92m39uP+1296/fUPj42N/sTHHvzk3r6x7j/+wh7bqyff4QzqkjHnmrdTl67sStb/VxOnNqQiy5dl9rkgox0GpJORFLHOMccmktQF3RhcnYQCCWZqM/wOfp6iWSHrpW4PkEFCISmE5MTDKNPnTrFhY6wc3tzi4Zs5Zft7bMyPT3JEPX0aRf2Liww7EUI3Nrdk6kppR1CxiVyXx/8wB/Jo4982r2+06TfxOHLkXrSei3hd/HRABQYaZjIp1FRSvshH0ckEIt14SQ0dGjAEYBqJG6DUBNvpfmZoJARu00BkjTdqbs5Ukve6jaMU6+78bp30+//Tx586JMvesR4Pwz+S3gm+vm/GpubbULyc7tfyIbLPTGU/LmVldlcPpyuD3W0KnEHqgTNrfG8qWPwQZTSO8NOa87yWv18HR10t9bIj9r1jUCFXhSu+d0aldMULM0gMMCXZ6jsaML5RpAA7Boy4tLRCNXT5zh1rBMHCQHmCoe00f/P675VOF+STD5MR3y5DTW2+o10QLn6jHUpANVMNBLaJKLMan90xjEqzoeT9GmrGudZfIegGoj9CYkYEAgzWnr+4MKemo2+dWh8mUqt3n+ZSLFH5uZ2fnqdtvu/nWfc+6//hzfexsb77ZGcAl03MHZ03ZWHZ6BZFeMAxPbLrlfWVNzheWnTcbYddLojouOlQqUvS31pB8/qUiM0fJbBpxxrLnjHvEeU+EmTA+1ENBD+y6kLflcr7Z2Xm1CrqQFmQGeM2d3V0Kes/Pz9IgAR7BuGfOq6v4eVyWT6/IH/7hHfLNbz/FPJozcUB8gIZxTvYX59RMfAgrGhYjdzUjo7YTyfwdHW6V5hVCF51UxIh8Z/dc5Of2uL8HpREZQIahW17cLXXvUSeqXHcRtpJCMFbWeeYrnZd94I233/x1d54/fP9HP35q31j3H//5Fzxp/Ea9XmuiINMrEqK8oOLVkhWdnNZWd+RJReyxlM7EjtDrYNaCJAUPa4LV7uw9ypJI0Vd8UDNDhJymoaxsrJCI4Ax7u4AeOrLwYNL9KDL7



ncgQ8BotzZ3nXHXCR4BODq7vsqe1nZbO2kWfxfJUrr/gY/JU9/+1hTOy6McA8+IEgs4yWjXKyKNP  
EnYj9R4YajIOeE58Ry+s+wUqZfFcGd63rRg2QleFAwm1G1rdUWRMeQZ70PD9wqOaMBv4D0LfIrbF  
OqWL0dsnnef0uyn6WvcLvKNt7z5ti+7XPct93zk/t5+GLz/+DM9Wq3d6wdp76UjzkhcYiatnW0XS  
qpXGnEGu9PJZGVjj/klumXS1rZ209RnRF76NkmOXsv9gg3bzBvhQeENEX6CWQSACOWYra0d55Uj1  
l8Quq6trcgCOmncogalchJK89jNjQ165YmJKfasold1wYWeQIu//OWvyhNPPMHRkz0wpdxSAbDEK  
escvt6k4cOb29hImmukHUOs+yLMhdi4aIkGBt/r6zxZhN3YcBAWjzRGeU51NhvoaMq689ZqqL4xg  
RPwPLML1gmRN3CMsZO51+ILYFlzdGyq2Ry51YX9z/zk3/zxX/37f+9vR/vGuv/4T364tfVv00Hu1  
tKYtNodGZmaYY42MT7KULDlnNap7V05u7Ejujy7P7OzuSb/XknTMec6RaXokGBm0jnDr4BnhCZGTj  
k9PMW/dPLtF4GhqaoIGCAQYlEJ4zc2dXdZcsTkA7QXgg1wW6g4jrLnOMRw+efKkPPTgw7Lb2mHoC  
08Jw+R8V+gIg/jQTYu0wbJ+2qyPa07pJA/vjQ0Ez+M7+3ElpwHTECNffwF8FX6cJIEkoMEQc4sUg  
KJnpZGqqBt0i5XhlEnHeX3k6FDN2Ntp8fhxbiunl4+6cP7nz57deOq/+lv/5YtidMd+GPw9ejhDm  
3X52EVxTdX/evCKS8dkY01D0NW1swwbztzFrG62ZNYZ5Eg8K/Vjb5B89pjUcq1LIicFiITFi+Zue  
FiIaLdaAI72XNh7gF4QCxYGC8+z47w1qiYLBxZoJJsB2867TjCEPnn8BI0ZIEjx48fpJT/x8U/K6  
eWTVJqgdCkogL2UzegAebSh3CO+CG2LIhgaa8LMZxN6d0OMLTwxCIbNflbnnUkxd08GD8/nnbcFK  
QLeN9HmVw6V5vt6ehPeg7XCHMOehZK07byrITQ6izASBALj3V6zlepudGoffAHf+BNX3fX7W988  
N9/+Jl9Y91//H8+nJG8262lRYAo8AJYCHlZ3BlBj54QISEU81tuQU5dc4Mc/rGfLXxsQVrOm+xs7  
Epe22PISIQXNVIXziJZq1E8e4eTyJewDnKe6rrLQYH2YvHC02AzALC05zwrhk3NzE6RbHD65CkSJ  
WC0CKkPHDggjz7yRfnCFz9PxQkYLowNIzjQWctWni/Fgu4a/F7pkIn2xIuCson3tjBEM1Z4cfOGY  
E/r9YhGCNULMqZm51TNEaqNMDqUc2D1+I53doYeE4CKPWNKc2J6e0xwR24fa/1WCnXHaa7oZig1J  
U13/V/T6/Wf+Gs/+vYvj4yN/OD73/fv2xfaGtrvZ/0ePJwXiLrtzjNuYV5WbzZkc23Zudo9WX/6a  
7J58jvyhSe+Iu+/+08w+lh+8//+gBy65OXsmhmfGGUeeerESRLwL7vsMi7Gne1NiqhRA2lji+Ewv  
Ci6YND0BoNGSWRvtylZ81PBuKFICOKvDnLAQBvO+M+sLNOIDxw6KCefOy2/8Zu/5cLwVRoMdBvSM  
FkWsem9YaMbZ0NaaMich5NsdDAJZaoMKmutyedbp+bCZ5vNMZcyO2Mta1ln5OnT8jM1LSM1sYki  
1OPCoQq6DnHsXptAlZ+rVqPLl1tEjmuPU2ZzSeIZXxQ/z7XZPvabiPf4+Oja24zu2NnavJ/enfvv  
yfbz1n3H+Hx8Yfuv8Yt7AUS7QuImHXcgnNuYXxaDlxyhVx11VX0rocvvkyuee2tsuRC2aRWyDNPP  
0WE9+hFF8tFF13CnlLkqPbsyF/Xz26yxAK+LlBdLGWEyAyPnaEsLs2TWLC8fFrGJydkdEQJ+sBTg  
Rgvr6zRUJFTLp9ak/ff+QHZ2j7LXJHUP6ghZpEaZaI8XXWGPCO9X80YRzWvUfDNY+Yg+J3bIJPc  
xIeyMBKmlpjbejx49ymJ6eY56aR1onx6GUKLOW5NrHXkiQKMPBZFI1LVRQuylRxAog3w3Lk1AMLY  
8SRHpcUcUCWke/2utni+vrGzz13/NS3f+on//rf3jfW/Qcf17321RdPjE/f67b5qXTQ4dyXbn8gK  
+tbko3NSDRzgOMpXnLpMZleXGAoCBkV5I9o/D5y5Jh6OZczguAAr7mlu+eMapekAXg2eMdRlHlqM  
T0tCRAZU9La6xDxNSVCGAdlR52HhUfrd9o0NgBRj3/pCXnqyW9wgXPcBvi7VHTI6TFhtPSabtGPj  
SiVED9RzoGHNqXCBkPZgp4bXzAQgE0oL2EjwbHhuMBBxr/BH+bchQOWRD0gzH/w6PmWRxKOchVA  
aQVXoiNGwg9robbMFTlK2tIjGYCC51htAid0fWDY9ja3r7s+edPvOutb7rtmy48vm4/Z/0r+nj5y  
668zq2y33fGdfHVV189hUX0za88KU888Tl57JGHnYdbI8jz9//bfyYvf+Nfly9++Wuysr0nuy6XB  
KHBwstedyATk2NchNtbu2yHQ354YGLBife7W84w51UIdKvFGilnozoPCmOjav5um4v10OHDnFh+4  
sRx9Z6Sq+DZ+obc8+G7XKib8z0R/sIgVEDJvVXsc1CEkT1sHC607LjtQ9qamygavGwjYn6q3m4QS  
i8GVCGHxkYanWEjRgB8CqCRKIkcftwDpTcstNUu9SAWjp2G6gzY6s04P0Tf6JXVLh4XHntv30t7B  
LugTwWes4XEeH232261OFKtWE+dOstNz09MTH6w/fe9+CpfWN9kT+uv/66qNVq/fQg7f4vSR4dy  
qNi6hXXvYb54/bmWXnyqa/Id595ynnJnCUV1A0/ds8HZfQnJ+VHfvpn5SMfeo+Mj40x1MXk8alpp  
RH2vVIgctN6I+HEcXhePDczOyM7uxuS9TI5fPgoPQnC3unpeU4wh3FMTc7I/MKs7O60XD66xvwWd  
ERwgLGu3/ve93IBg+TP1jQsfBiNW/C5W9EwDkVwm9rqBoEzF45iQJZpJ+Hv67VmmHsDY4Pi4Qja5  
pwHBqd4fXWLs3MOHlkhJw7oitgYrKZKGWRjYtNbHvv9INOG+4Z73/5AGwfoaXMJ+k5leKzDtjBRL  
yb9MSpl0IBKw7G6a8OmCE5wd965X4QcN897Uwpc9b7xuhuv//Lk5NgPPPCxhzv7YfCL7Pha61+dv  
OK/ePk/2d7aeqrXbb+ryIorB3k2NeHywptedxNf86UvPsJ6YL05RrAFdUcSG7od+eKnPy7z114mf  
+tnf4GeDvnk5PQEEVrIfgJsYq5Xj0lwGPeDl0NIjA4ZoLFLhw678LpDLwLDxeLFWNa+VGgAw9PCK  
FBLxd/0Wh2GxB+++yP8HHg3bAo5SzNF6J6Bp7Pm8Z4L42G4+JmIbF3RXmws8Ias7NagChpww/8dQ  
n+E9r1BVw4tLfK5OG7QMxK1BYoMRJegmbalppjk4/YL5rP5MAaE12rDu3rUqjjaxkTLAcdaIwL1XX  
+VkfHh1n+X5Mkx2mwAGfoGbDfp1OshdupLFurm+9fyNN7z2nd//ptuTfWN9MRJpda+qvfpV1/z63  
k7rObfgfz1Nele6S9rEbr2weIIE+KuvermcOmncmgWe0eqF+BnGMT4yzoWzVhJaZhcPshkic6nb1  
tBwmb6BwtpQuQfX9+jRixja4r3PnD5JTzMT700lpFuSIOEfCdYc+kGopnNfC64iCtozK2soqFzgAp  
49990PyrW9+U+feIPSttKxlvkxCpQeXC2fIPV2+iVwSr4dhaS4IOdGMnF0VAe8wNEXtF0a/vLpG0  
gb05eDBgzW0lSitlXkqFSWaoU5roTeeG/jGeiLahYQ+2KzIg3EmvuZrx6TnoSqKrA97nnIHQ6bdh  
qahtAvj3ebRT/Pw+4HPEuHaqDVGFgdp/nOddu/bt958y3+3HwZfqKDRta9s9NPB77nd+za3cBedF  
2ticdTjOnmtE5PTXDSTziBe8pIr5PFHHuFCSpo5yxizc0uytb1B4AUKho0m+kk3GEaeWT7lQuFJm  
Z6dInl/d29Tei5cm3He1lBgvaFKNBhngcW/urYpkxMT9DZnnTFOOWOGR93ZrtmkJUSLSwRVEBJjc  
BTEzO6/70H5+MMP0dsh30V9116OnF9npG4TgHeiV4VxW6gLEgR+pkcDcQGAldNYbDBCpDjhcWy6s  
B/GC9YRar6IAuAlNS9VJUMz1BrE1Dz5nyExOMO+NEWP635PUAkhcb0RCBg1L8LGzNjXZFRaK7rj

i9G7NsgCwpEicwbPY4ZXnu0Oeo2nxDZLQiEgHPE3wNBOHwGnecl7mvf/3KV77yZ5uN2j987AuPf  
2rfWC8MIx1znuM0t+Be7W74UajOU6SMO3wszdFxF3JO09sg57v66qvp5RCKAQVtb+3IqFvMzYUFa  
bV3GQq3ujsu5J2huuDmlprLQWdJKgDLCB5k0RkkHiA+bIFL7N4HKOqxI4dly2tTnn/+eYJKMKrTp  
0/L/OIcvTYMs+Y8E8TO9jpd6bT3fGP5lnzta0/KZx/5nA99u14mVMNekPNRmxxp1kO5w8CuQEYwj  
ahEAtEeHhUbFkLtVqdnFWCoVEBgTUS9RUB0VSQtCuErqytpQV6wAkU1fT1mzLoUwGbPqjfWgJDi4  
chHE5h+g94djCs8n6e+Yd+F2ohiwEnGBjKG2beYe9toaqsduojabR9iRrohFDiuHifwRb6byG0MT  
WewV/Z6tQeuuuqqr4+ONv/Gl770lWf2jfX8DHen3c26x3mPq91iWgRwgvAqL7TdCyHg5NSM9wZpY  
NdccsklmjvBmBHSuecmnTGuLq/J/MwBWxEGgxaxllvot1x1RJji14WzA5ZhYIDI/1wYFrWapKd7  
q4ziI5bXNpzSrTXedntrRYNM3eL8/Sp59kWh+MEPREGDK+7vrYhT3/r2/LFLzwhqytnAnAVRZr70  
VzR0VOvc3EjnGQ3jDOAtO+MlueohqbC0DVHFDKSUB2BsaKeiw2jwd7VUR47Dd8LjTP8lSgIqsHLK  
nCKWk9gSoHNZVkaCkg1jBUpsiC0hs2AIbH7L8sHAT0eDFQWJsbwaMi0RsIIBueD59Dq107bOi8oj  
kLpB+eoIXeJXBdpLxxfonXmpvv817jvT7zsZVd9udlovO1LX/5qa99YzwsjfeXBbDC4qzfIL3cLd  
xEghZUaIGqG/1IQcmBcCA1NksQW86jLR3cwb8YZEAxmcnRSlvNYpmYmpb2HFjZxhgpAadaFum5Rj  
U5IMxrhOAqUW+A5Wc8cneD7biGs7fTkwmFZlXOOMAdEKxsW56EDB11/3d3ZkHn3czYoZNV9fODAI  
hc/yPlYmMdPnJLHvvCozlMVYTXvqKiv1kvJGqpZCFnQ+yLpQTtdoPgPA4ABYSFDGhUGUDDjzmWgv  
wNbiSE0qJR4v6QMf7Ex1D0x33ScOHZAe3MYTcOdO/JHk4/p9z2TCSJrmDeLz+cs2j4nu2e5gUxK/  
K81G+zk6fYyz7pKSJ2088WxF54ppeCaoltRkemc20odN6ewm/jn4il33Lf2+v13r3n51R91G+bf/  
eznHv0L10rdB5j+lMeNN1x75FWvfPkjnU73W+1u90bnQRbZp5nUedMt15pxng5CYyzss6aXhoWak  
YwwtCkyjCakhgUEXk+C7pYlabvF0U8zKutj7mmnDxKAKth397pEKMFOQR8n81v8e2FxVhYXDsiuM  
loQ//daProXvObZzXVpd3Ziym9t73BD0YReVhjo8eMcjQHQ6e577uHvSi9XoqjwonqOg6GwU1vUa  
n4CQEFxcTCUzIgKJelSoXBkbMwboII2NPxCecX6OZEXW4sJVunGFsm5tFfyj0WfQx5LIw8ibYSUV  
McJOSbwdXralO122EDxWnhPSK0SSPIgmtZ/1UNj84VRmrZxof5en6dl5oHPjHPM85jUS5gNXtPvd  
Ra7vcFPn93Y+tZ11776F37h5/9ptG+s36PHzTfdCNTlPy/s7e19w91UZ6TpVM+5POzu2HEpJzJQ/  
aBZMI/qTd5sHRmRBQKAtYAhb8TcUws34X2npyakM9DdGojp2sopGR2fYo6K/BahL4EdD3LASFHKi  
ZnDpRoSM7fL5ejRwwx3Uc6BkU3ifc5uEaianZuUDfczyjRTc7POUHfld377d+Xs+oqWZ7go+9wcc  
G4Q56ZhpWV+aGwgXczqLUEVhFHSYL3BFOJDW7eakB8C1DLPZSUVGBxz3khC7jpgi50PwYvMe08VA  
y98J0/d85HtmODwYm/Xqkih3hHpBNhU2BARTaDrBqAarp95TZSY0kEeQnk2Gehu5I1Wv/IwICwmG  
0v88eQ8VzEBcn7HZ/S7vct299q//NGHPv7Vf/Azf+/AvrH+RRrpLTceueH66x7Z3W092W53b3Q3b  
Mp2YeYwLo/DLk3CgLuRE2OTRD0NbFFurBfepvqgLiDUNudmp0n3Ix3PhWWTMG3a63CiOiZ3zOl16  
vbWaiPOo2JicsTPAQkffOGxUQUWU4GFJTOA80xGS4QCTPIQSQA08Ow4SXHZ9oMi8GeWFiZlqWT56SD  
991t3zr6W+ShKDKginlQ3HslAuNmlyY2CQSQ0CUPSNDbypZEajSAYc9bXVjjo77kNAqExNqhRt  
wlxWHJcDwaHDYyIsuq/T5p7KiG4ZC71x3WCJ7IzLJJRrzLgL77cgy2qGis9VlYmYhli4kdDf14zuMf  
ph7Q0tzzXM1Px9f9wu9IRcw4nR0RgZWOCp8xZ/ZTXhqb6bcwWM2WvH7Tba7XfP6xL37qR3/kh+r7x  
vrn/HjTm26au/6GV3+y0+59o91u39jt9aaiXhDNC5fYxpV6QjrZMANlPW3Eg3gPpZ6gCIsbHvXbz  
jgQcqI8Mj417Xb5lOAQyjdTKyq2vbvbKZPHv+u86A7J9zAetLNNT03TU8BIC7/QsKCAupIr3FciO  
9T8dzd3ZbftDz72KH+3urou086bwkhOfPc5Of7s8/Lxjz8kaa8LzJMLGYvdBbEyMqKbgIJAYkaix  
0sk5N4KwJSelnlu7jtuEvGEib6cXj7Dv0epxsJcy+GDARSKNsMYaD9IvMdPJGXgamzZKVSL/UgE  
2N/n/hSkG2/wvVC+AtkHohwZ9D3gFPmQaw0vD7z4T3vaX+gqheewcTni9i/zgAwbySo71ZYU/Z+d  
kx6a7Sp3t2fK5999ruPvv2HfvDPPST+Kwkve1tb5zY2t79k263d7XbcRcHXW+YUDHIFeq3Dg+UN  
ah2HyW+j1PvIPK/pYXFQbKA5IktTFxWeOTjz52gMU+MTMj2zqY06yADoCulybawLVmrMj7akKy95  
wwJqhCdIPNJSU+fM3H3d8+326jTD1zYnMmmy1F7/QYpgthAjrhcFoa8tdciS6jd7dHLwaO8+73vZ  
akIQmMpAJaBTnBrNrQ8k/v8FTVMhuv+PE2YulEf4aYjftobyzn1ctod/g2Ko0md4m9YU/XGxfesq  
eEhIkHeH3uWEcj1wSSacR9c4KbWUw20s5+tm6es9apxsfdVLGTXkZL9tOPHCXhFSHQ6FfozBABwK  
xMxbnPO8hvBNEQL9RqPk3/mN2yytDwgp1Pw1IHCgyNaxpAwJBguc3rNiRMnPUh+9RP7nvXP+Piht  
7+lefMtN9yztrH5nVarfatbO4sD5pMqIKOymikRRrtwmm9od+bCPYy0KDQfBKlrfWAtilzaYKZiYb  
4PhPfvss86YFDhCt8vExLQUUDFwoTXyz5dcdinDzrHRCff6jgqSCWlvLmRVhQWAJ5BYQVmk32tL2  
4XN4AOvrq/JkSOHKCW67gwTxnpgfk62Nlscchozyfyv+8P2y7PjiLLhuvxUAMOon+XCUBP2aL6EYL  
VDri964BvS0ZdjPlRrC1DMrp/m7pcWDNHCEpamvPr6a5RBu07MVQuPyuZR8N5kJKv5aFq3nD14F  
9/dY5taf55UPZt25GSLAWV6PzTSuc+Idj0Ca4XKp0a+1FRozYgGaU30FuoWvtQU27EQbk5CvozXI  
VrBd/weoFa703n7DTe89tZ9Y/1PfPzYj78jeuPtT/yq8zsTPuAX/j157sKiIbep35cJ51EI9TlqEB  
ZJ6owPDhYbIEHWUuzDKENoClmcnh93w8uYXnNWC7hqEhFBWmOpGzV4MJC4AxrTloiNHSR1MatBUm  
vZSJhErZJ4zZoAlvb0u024gH9rrd9z3lnqcQpFLTB3HMaOeCHJF7nKvpvPYD9z/Mao+4BxQE2UoL  
3EYWWEE3IzEIgd8twc9dyIeOS7zVaK6zhHw11f4PvPzi6yvmifS6mgyFMJaKG2tcPCQaEzgeEiQ7  
ws1qFoyEkomBi5ppCGhE8kYVBnr2AqS2XWv/h3+n8QqW5pHeZkvR3qvjJXBfxO8Ug+rYS7/WiOmu  
BQv52aDzcAjyKzBfbUwmxbn5jaK5u5u6w/e8fa3JfvG+h/5ePObb3mHW8BPTfe2fz4bpEdTD5DQC  
/X7oYaW5joxXBIJoWfpReIAvOBmwWDHnNdCyMecSfpszjbPgZtnYMRXv/YVbgiQBsUNhrfd541Oj  
TKvuefCYJe7jo+yYwRGjxwSTBp4SRg2c1lsCnudbedRW/R0YCPBMI3WuLgwwxIOAa250f786U99V

u784Ad9HVUFzvKiVHag0dVjT/XLA53QckIsRII8US0gp2YsXNTuc5fPrjA6ABlDNzEJmwDKWZZj5  
mkWcsPQOQMhhdJNrrIwocRDr5cGQMtyQzM+3De90F6fGBKnud8QMh0irSykRESxnodsZ/F+enliU  
77g+3v0t4iRGsTBY2JXLTYyJftn18FQ7vK7b4WHHJXlflD04LLTp0//2r6x/v88fviHv//i229/3  
eOtdveDu629K10o2RwgR4p0MaYeJSRimKJ2GAUBME4u8yRzM0CrK1LusqnN1+FGoQbpQyYbzCQeS  
Tx98gznnS4uzQpUDcemJqXn3h/Nz6PjE5RrabqFLYV2nkTaY0nmERq0EQZv722w0RtoKwnyhd46G  
AlAIYTLMOD50ZerOoMhNfATn/gEARdsSGiVQyJXROjrPR01TjD8KSo9KuuduXbZYGFic9KmbW9oo  
100CBdxbOD+QvepXhsNwmcgVjCFYJnDh76e7ECAShI/X1Y1hW38hoW6KEHBWxrDKXhmv3GUaKw3F  
KK2qagckzfyJAobLX7mJpXwqNyp4J5m1Fm1SpXryMxz9+EzJ7HHXi5VktbVn4q0nhz7GjJrx7ECY  
5EpiwFS2eaCddTu9H7q9tture8b6//L461vfcM7Nzd2/003M0B/IqYSk/0S86ZnbJnirYbKez7wz  
JWIwITldPA4Q2yk5sgQR5aTWiohr+2yJRBSSezx1FPf0vcstIsFYBa8339wHndhbp7h49jYVGiqx  
s2Gh4RH3dxYZVmm3+2ozOeoAj1jE+PScBsGhkWR0bS4JISHjrBXdw11WU6fWpbnn/s082qEv3CTj  
WZT64U+pEQt15rKufBE827zrJrnDXw04YkKDFVMnCq3uS5H3WdSdnRUwR5sBCTqlyJP0UvO2fRKe  
GzAxnlvzzwalMJM29qYP3rjNxD1V5xXAibtUfVPGOpzRR74bTYN71H/ndJomWm2G/KFh7zc2L1ly  
omyLsLGDSHYw+Hcb5ISyjisDRdq9FIMwsYeFeX5unM9un5287f3jfWcxzve8ea522+7+fHdndbP9  
fuDqQEL98NIIItk5keoCAenFDbdd30AleolIaXXwZOPU9Y2G4HrcXNUDKqeIRVE1PdGbC9QYRrOwt  
OinWuuOe+2WHDx01JTE8YkpGg28N4wRpRfUVkcnJonEKNG94QwHdcseRbknnVfWcYoJPTTa8PCZ8  
Ngfvucuabtc18LZWqU8gU2o6ZFU5m/GZPJyKn1RsuZC7dg9Z94XXh86w/OzC9Lw18Vea8ZS/bdtd  
ghfDWiCoQFt1WMoF7blnXyP3I/iQJRiaoUEfoTemSLfRalbJHsaFnVcCxEUhtliCGFXNUUGOE9T  
eTNFCjsb/lheRRCaNxeOy97jbK7LayOyVlmJiRZSIXchv1m513/s3PXC65084Y3XBu5hf7G717rX  
4y5FX3w4NFx1MHQxO0uymSvny/Ck+YqwgPoKuQyU2XXGEqYAgzwtDcarZHI3SUFz7Q22mTeVX3YD  
Q/cWb8w7d8hZ4098MBwOJbPf/7zcsP1tyhBwBlIa2tb5haW5PChi9lvunjwIMKkLat1HjVNMepwx  
hnkeFCIGicEk2lvT/NehooM8xo0cDSSo/H83e9+t2xsrAeApl6zJu2BIrWNEtG1sD7wc/3iqoJJM  
ZBiX+vFA80DY86jo+WNC4i/U+DGWFow6CqQq1JjHnNazJHhVRDUkRbjnepn+PUC1xJBaboYJhdHM4  
JH7RjbdPfYzghB2x1KqQlRGTabD95usecQ4qtIbY268WtaBV4zoiQeZhsohX48Sj2RXG91tMyvoT  
Xn9sHb8ZiMeVU7TfHFza+dn3BO//VfCs95yy7XRdde97FdX188+s7F+9k+mZuZuXFg4eM2g27vM5  
U6XuQXrvgaLOXRkCw1RsBPnReR3t74ycoqcZRTkqFwYXtNWb2JBr4080hqZq4Vw+179OXgSqZcLR  
iJf21A+7beffsrlq3Ua5uTsJatRn5NrXvUqOfaSl8js4gEZmVatYISmOuhYc9FxaC85I0WbmQ4kT  
nQsM3tdolnj+RjwBdT24Ycflme/83QAZsBKsgWK8wHryYynGppWG75Lj5H44y+BNtAaVYxtJoSui  
UfKzTA05PfoayZDZYZh2qAvDw0qnTiQwB8PbYwG9vHnQetLEWk7xsblHeAvvictatwxFcBYxiT8O4  
y5rPlyGwxYc69+Uz4HdZEwoOcdTa5T12U3c5HK/2cUwETxd+vzHfyXC4JtuevWBvb32V3dbez+/t  
7N92cGDh5vsBW21pNUBQygjyJjVyi6W3zD0Ah/U1+5YiPehDksZqQ/LaqrqgNw0y0p9n+pNOXeBm  
7Eqpc2rAdquberw71499vnH+drJ8RnOtmkNOnL55dfI0sGjMuXyVoS6oBaynzTvs5QD8EcR1IR91  
ZwG3mwSDTbhMmNM4XWHXUj9hS98gV4N64MC3D40rPm5MkbXO5dVd07GY6WSEA66xQdDxWehkaAa7  
prhV91FcSKha8UMylhL9LKIQvLII7P1oU2R3OMiDqFvIP/HWiLBRlijf/X1VPHIciVNsfswtUoG  
hrUbPKpoSyEex/Fobab+/CXDQMobfFLN29YZjXk12vk3i8ajsKAUTBw9wCku3YHX/2qV4y/qI315  
puvu7LT6T2+1969xnnQ5uzsIj0Gam0wQiXNZ6oMUGnBAnUP+RkU731DRAn2Vkc0pbxUyhtMRbxaI  
yzoquxIqK/5f5eopC9nRloa8neF5TFCbw05UOSaaAyHxRw7dgnRYRgchMzwPgshdJkCjT2pAiTw2  
Tz6ng98TqEnlicVkbC+Z4+fVLe9a53yZkzJyphZhyYOQh/oUNs4maWlxpvdYhCqBiG1VuwAYBiC  
ama0Enj/7bba5cbWVHmqQGB+VxVUD6Svmikf3J+ayXpwfJYvq4WaYSAnDHKK0OY3QaTFENGHELVI  
g51FNTw9PniBfcxqpDzBRNHiYvhea7IPDxuNuh7MoW+T02UTFH12ASgojSUhhTwGgTPip/deUhb6  
R+/aI31DW947Vi/072v0+0eRc0RHmjMt2CpsWiIZ2wZgkV+50a+2ScXNOeF7g7SACaR9ZJpfyJRR  
W98I80xT08rL00VSLBFXQp0ZaGQH3k0WXD7RZdjMXJFIffffz/pa+AIH3E5Kjtp304+MzvPRT06P  
sJIYengIRotPfNarT6HVqFkg9fjmAB41V4v59889tgX/fuXIWWK7OKM1Ch6JCvSK/VLy0crXoJM  
1Ao/rOVzhk/dImnZ+aoRhiMJo+GjJ1lFQ/4WL4cV/SALQJgrhlFQWXQrqkZHTpj8D52jxURLoE80  
AsZLkclzVd9dGAePOAGIkP14RJTKIKR4r4U3oCDMHISpBRWekKWJLQJGzCbHzIdzsWWOd8aqABUJ  
WzWTLiCdfB4mm6d/vSL11jb7e5D7V73MjRoY5edcQs41PAi6/IX3mA8n/rcSNkkuacLeug/KvOnz  
LeokYVipOxCMt3GiT03LzGjDQuIjJo4UOOGmDO596xxEVDiD33oQ0Rz5+YOydLRYwgIoYEceSRKM  
Kr4V2P4C4/abKixkG0EL1tvBp1dG/iEQ9nc3JI73n8nmU4kaGSDwKRJ6koKYD5X03CVYWjqSfKhC  
bsYyvfmMgO9QSVee9vU1AyNIYS8tSgYAY0+10jHNRiKyh9KyDHJAa6UbWisaZkz270N6G8R+RqtD  
lmjmr8Hw/fBVAYn5VQdr6GRjhdEGwyG2uMKgoxJyKktH7W6tIKPXvtYU19z1gHRVuYjeURsYynz/  
SLKFU22ETp53GP2huuvS150xnrj6659h7s4ryZJvtMhKUASC5Mk3ExV2tPWLdABsfPZQLQpEUUMG  
4mqAVRBAWru5JrzIKcyRktZa0yGSjPVckDZeeHzSo8ActQv8houSFON71Pl71d+5VcIDB06ctgZ6  
TiNEIR1GLGGjVFQo7ednaBSXVvjSfAIS45ut9OXz372M3Li5HfDqAnzZNqipoaKiW2sH8ZKzTojq  
VL4qnNk2FPqFhpUCXudLumSFDIb5F4su0wJLEeNDWA4byy5xTW9vt54w2CpuM5UBeg0w9MCQqSa+  
4tgsDZZXSJvDDh23t08EspK8PTmqav3zDZYgl+e5C+e76sEmEHoV418uS3GZsExARZF9MsSkEjo3  
02szp5Hw17buMZ87SCwqdzvpnr99OCLzljzNpTXLi9tIgQhEWB8mjVT8XqySmzIEkFw/atc0tyK+  
8ZOqXvZjiINO2Z17AJ6STmuIfKUM18PjeMyjKveeJu3wsVRKPJHipkoAj1cLtB6LX7/wAMPyJv/z



b/Wdp3Egxc1bVFDvgRPZvleaL524TwaAOh5vSqDhcArJ1fkW3ffHQgag8q8VOS9rLHWtauF51SpK  
cNY7N9WI7QFz03OWRaID1BQbLqNpVoSqfJ9EdZmeZ/1G6gJVj2X1jS98cb10FJoZZ66Jz7wc3H8f  
U1NkjBIWUKRMPKSb12plyYxja2MaDxqDEaae0+8X5nW6HLPgUL7lCUTz1zyYFVcK3tmC2WiRawL1  
/xaioY6fyzKkCIL3t4UQ6IiHyZQ6PFPua9ffFEZ6+tvuv4fdFv9S7hgKS2pXsUYJODaYjfUQnst1  
CqsJxIXvyorQvqgD3VpxOg/jSTA/VSPbzbCuIhq3mPT0bRNzgriFSPOxIfDJew/tOnkeuNTP0jppq  
ae+Eco/qKsOWF8dpabt1PQsm85R5mm5HB3K+SgtjbtwufD83ciHZnf0+/6I8xroReCprBlmIAIz  
KUR25iqi0vZOdrMzmE/v9cVmMPfPv/8d2V8dIw5snoVj5pXWEkcUuXC9ixVgAXpBzYaPEcv6gM+C  
9v1uyK4WablJFVuKcqpRGrkpRBCW14tkdVNff2WgnWp5qqRYheMeCIfCntJiSKTAGpxaLpRSOt+A  
/KblKH3Csp5+iCkSr0njnxEZplYVeeQ5fnwPY/85sD7PwhKIu71t7xojPX733TrvFtc/6u7WE0k+  
kAiVRg6DjxTekWfs+V+EreGm6myWxBKJX6SmRQVKF/Y4cH6q/cqLPiO0G+pYxn0huWeYiZDobCGm  
ekQqBG8rUmc2Feo8ensUmteP3bsIuZ/MDiUn6zli6MJ3euBrsIg0dsKsj5KNcbRRR4b+c/79reel  
Y9+9F7NsdxrEWDY+Duk+OANq5onktqH0k8+GOqEwcaIPH55ZYULH+EvJ8D53NvCRer4oq0wIM5aK  
lIpz2Ko/GOBxOHZOpSRsLnWG7HKp6CsU6vUe0OombG8GVIO7/0YHRX6d2QKZRKohmWdOA/AISMbN  
SunqcGixs65rsxBtaHeelUpI+spigYiJb6mHPJxH2X085KqKP6e21fk3yuKCR+urCFCx18Uxvobv  
/GrkfMwD7uTokoCuhe5Aq+VVLBYvSl0dvqDsteRuYbvWQY9pdT7iaGP4zsmIhq1TWIwPR8suphzR  
htB+b3KeDGjNQPVxvJiyIBNnQ830Yw2Ng9duXF462PHjnlpzMQrMyjfGHNjIO9CKdNYwzOT0IyU5  
+Y/q+AG9r+/8//QNjogtrlyVGEI5OY6T4m/pVHU60PsHTMipBHwyPCKJUNBE/R3t2U0dMLGSCA/I  
BQMORo8nNSHcnfdEOpkPIWmcYmC1zavFXmONFQH+beehQQJUd5LjyBz0xWfHxsQZVFFhexCIMcMh  
ZFY4RvE+yWDYjjAhbYfgr0W0hWf9tjdNS9aVNKYoiG7b0LJDhGX/6sgHicGaKU+SLOBPAJUREATN  
256/Y21C95YH3zoE7/nrsM1DBkHKjoGAXsZGfd9g9EQ8lptTA4XPPck7agWclJQ7qrME441dB6Wp  
HEsHE9AYF5Wq/khwXHIT6tlyCrSaARyM4BqCHau5zXk9IrLr+Tnow1LF3NCIAdh3e2tn0nTUKew  
AkGiYJK/j0RCXz0/o/JU08+SUK9Cpk5L9oYpfeG19McOKUnDuSHClNjNxcNa6PY56CivGSQH8anZ  
2TC13Uth7V81cLg3HvyLLCAsNBzJV8YH9ir7AfAjvIo+VctWu9fRMV8eE/SFIuUobFFS+UGUQKL1  
ssaVe69dT5ZylvkaY0hVKXaWhRKfrqBWC00DrV3y3PDRmwsKkiwSjHkye3cxc+rDU0RkYTKAJvUo  
/CemMI+fUEb6y/90j+7ttPq/BSJDiivVICBkHtl6iGzgYSZMZyqcFfNNXfxfoRh2YjPnzR3UGTud  
nTzpFaqSTCsRPNisZkpVQ2hsFkUmrszwPVyIKETx4y0KPziy5mLQuuXuVSkqvHwfpSB69owMDlBA  
j9MCXmsDhIujxOPs+ub8r73vS9oQIwm7rTngaUkPFFVUEoqSPbAq+xbfZTlF/c5Z5ZXCWhtN0wOM  
ZiMimkbox3XwCv1J7w3fSViiAJNMDzb2HQwc0115Cac5aG0UxQlh5geKNOYJGEfmxZOJDEDJFEmU  
+jaqTCdJC+nxPlrEyflYXSW965cXCWdUfPpvCgq2sV63ULdG0djQhOahm6ec0klADRDGiBZAJ0Qu  
bjf0c2B+5QVrrL/z0++sP/rZz93pLmkToR3VAUXVBLW3skY6mAqY1fyks4i6u7bLhZqh606d5VqY7  
gHc8WEySzzuUVSOFESytPPjMajUHqvyosqKlYdYS4UPxXKvOGsASDUM5uL02rRoVofOMNHXCFPxW  
pAaAPQATJqZmaIXbeJ3knKEBTedrFwIDz30kKyuLTMehndE2afwYyRU+GwYtQ2GW+Hk1i1PjeoKN  
Ll/L6+tslY7PzcnI/VGMB6r7Ya5qHkZ1TDUKzQ8jTwImGeVOTUeoSUqy3psXZlmHhwsqY4KMgU0n  
fdQSQqpR9pt81EgL6lQISP/uRIYYwp++dY4lmzUGBNPKdTj1o1WCi065JVwOB8aERKIMD5XtmYDq  
ahGhOhFXb904Av3oeQow706FO5HL1hjvetD93yqM8guM31X242xoJoNBWEQFquAmWeaJLqQIyn7H  
G3XrmrtEOkrTae3V3rrKA6gh8mYUFmvWhoIBpcHgKdQYRlKGj0vyl+ABFeRZYTWR44cIQML52mKg  
ghlwVqCx0foGzWAhkXlFw1v8+cIftedd97JcBWzWDAbZuCBKzNIRgyVsLy6kZVIsErUqL1FbCKHd  
1+cX/KMJ11oVvIJ5Qq3cvtk/cShRY1ADze+F3YnmYbyIO1op1Def0FqWGOgprXByeecywV9hh7Z  
XEaRRyE3DSXzk09GE3y2IDsnpsMC4P8SOvf5BYD1GsgkVvMhnHPPLPPwgZEr+y55I150KISbVjpx  
m8MWsoqgTxTg9RyWy0Ate51r78gjFutb77tf+r1uq/GBSaa664Nxx1kmeYzaGwOIU8cBQie5RfPc  
NHfQpWzmvQsfx5uYbO+TYXbKX/plQWMg1qlwRwmJlDEQ0aoi0BDTuoAZTKk+yPew2uLlG5CBw4cC  
KWQyEuEaCP4GMsmIEmwzJP1BDAKhhdrzU7lTD70R3fLqdmn1PuD8FB4FQPk1NY7WouCqkHpftTKM  
LDLaslad07JLYZiPwyKRhqXWkNWlzXPxRpmZGFnQqAo9/eGsqGeZUSerrunIG5E0tD3YR3V18fFp  
HOGO38ism+9YJuXdVXFlyKGSaYSasRSUkxLQTVciyz8vXXxmCQqjdor01jdOqIkHIGBaQ55b4qPI  
/WeJefYk11kuDTkFpDX6ypCV08o8URFlata6tK44Iz1p37qx6LBIP1HWVE0bRciLzMvieZYyNpHm  
FXci6aSrKwLQRIemkZt4OemSlWrx0HX1cgDVue0NjDbgbVnsQhk/qEOm6Jef0NXj+fCMgyKy06bO  
Cy8YdHvK664QtUm41I5r1Ef8/mkIoZ6/JEZ6jlo5ild0j2PuTfvec97ArvINGMFLxrb8K0TxhhKx  
haifKgHhxieuXQAnmh1bY3HsTA/z4VIj5+UQtyJ12giz5dyKEV4jh4/0mNO6lqnNn4uzpHKjLiHy  
60+BpqXm51vlE+ScgnqZ2aezVXT9oswoqMIM4DVRGo15N62fKHRoxkxH9GVnKTJQpAmgQ/KAH1j  
XNT4ddeVza8czi0L9vVy/w9qkYFvg4MJJ6ls6gIUYt1HVUBCyospwvPWF34hTvWKFvrlHkieUlnq  
zdHgiK79kpmFUPX2esYi3I1la+ZE+UjCckHVRkvh5RICuL1lJoBwo/ERj7koUYq54R35nHsPcyrc  
BGJF0s7R+TL1CleesX3adhVCe+M4WNoY2gA6Lcwm9AtmCYN9I73vk82NtdL4TAPcFCUu+8FyXxYh  
w3PrpdtTudygbFJ7e5tsUwDBBrXwRQl4sqwKssdk0I7U7I88ilwWVj4CBdr3iAiXw4zDSL11DGbE  
YoAs0aBgGKobxLSii8DMXPVrCTKQ9tcVRKVTe/1RMgtP+U9QtmJ9M9apXNHm/HpDbOS3KI6TgWJE  
/TEELXKN1hDkS8pSdionHQuSGelBwdjTqIh9lIUJedOaL/wjPWeex5I3cls0j6gs9vPg7xIn4X6J  
AhnVbmrWmyOwoIsYfg8hHl4P+UM971rluOSN2qfQZQ1aMTmoecJLSkGeYq11GpvlYsOHGUZltRe  
assl8ZNBbf58OHD7PgZJbjk56p4aiRzSN5c5ylHpiRpuq94lIZ96uSy3PORjwS1f+Sp11fJxQ9sz

VIVPgvrDOfzdOZRiIAXGDQEObWhoIJ321TG0K+uSkwdHVRCKkGvpTj8zqlMaURkmzTskgK/RaQ7pmlMQM/7c+F9aFG/P+auiah5y/mv8bgwm9pEE+VPwmJPEQH5fGFXvRtGrHi/hNoQ9281TI/OoTQ0koDuGqD9Zxzf2GkviNw+b7BAVEkw7ytdsoKfuAq61TXh12dWECTLH895hea7o5ZMcgTGY1gh1q7wyvsBAPCEASJKgxq71FvGLQskReSi5RAokidLT614ErRpqGWhgtcChdrkirpRvsrBAJQ9chMkoUmnTsimMBgWiDnALukNUwa/Vw0ITSUMjdeyJ53jPP/mTP6aiPkkenlppA5vs56qRGYUjixdhcE3zbqQE9vyJE89TywnGqp9fV1JHhQfNvNHNl1GubJ6opmURIuUOsPGz+bkut8YsH2hPwUjU+Hw7otY6+KUPSi10SdK8HEXWRXSSB9aS7YuGHheWesRZUD0k/zgtynwW9y9TqqdNcKfB+u82PgMITpV+SnDMR08VNpNJsAa0Px/uyirJJ21olysjrLzSTldcmMY60qx/0p3QKV6EKOYNNiOpaslaUb2ksW03f5sxi meZSGihS3whX0LoSxqi9xwI/0JIW/Hg54a/WkwvhoS5Avul0oXCdijb8UVCbqwTySdLAEY0NBybbFaG+RbsymGZg+WITNZWz8of//Ef89h6qbZ4NYKhZmGMhAE4VkeFt6PH8BPSrPaIzWltfZ1GMT87x0iCqoqV0LacSVppso4bOkIEJTRRWc33rDIYB8PsQe4jHaY4Mj0545H5iJxfZSRpaYMMqHNRdyx+9hgnobtnKNXyzkCbpta4OdE6DBGnh6YvmWVPfh2o19iwb1BtsEQ13UHnlCk21Co/++qZzL9OSSV11C09hTQEENZOhur91ZnmGbU0Y56ZXF4yxuovj1krRwCapdTdnKcxYrQhOz5HEYWZnSXWLwRWKUZxP+yoSSrF+9C61XpwsZKhXObcx7DI9HA4bvWzaluYlWmqryPN0C2fSy65uOK91TFEFXYX66DuWhUcs3odEeAPfUhaezuhc8N4vlbwD721RRzYUyq1GfmZNPGQG1+7vSdbZ9epMUWE2usAczZmpZHB5HDsuVWx8C09B8jziIpmqvJobWLIzcFtRjmq3jS514IsJ4rWJV6TF+FwnHPYVhEMNfPEf0+GkEoDeRYP9dJat0/kkVrxZAJlCXuU26cnZRmlpzn66hBqBuF9p3Qe2z1JK4eb9ws22GSAZxHXoJk3Ihi5ZcXRbby15JgY8oRGecMVYQM6heksbY7nX/qjG6R8ipxLZRvqqGpkQ74eyygfLi7JXA/vffLPVhQ9ciB5G71gAjex0LbWlCeN25vNR8KbKVKV825xPhqTlKVsjSu6NEjF6vRJFaG0PAMC0UbuSs0Rh+KY5bN3Xffrc0KHi2HMfcHgyEUmMLLJOWXfarYkMA7rioX4nhPnDopU86j6siORkXhcKAgEl6XKVB1XTOIRBRA095fGrdnYVnzub3jgAO09ihVqu0zCgRFRYwftJGzUnY0YpqShOaKnLXiNJRqzDufq9ZhPb85S1juehZqkIXPeckFz/Ny0lvkJ8j5GTg1Ke9ZNceMLOY2HjX5595AcX08/1HorVRh8JoPs6MS5EIva856f6OysfP4BxeUsd76htdFr73+VW900/yfg9VhpQDL14aUBTHTxQTMPApsCC5rrrm+pmwWzgO7xDRyonojeFg8gF9CrCvIVZrOTlyEsCurNDHnXgLtmo8t/z13toqFPWWXjjY5X3rppQxT4UmNOMfdHXouacVrxPS2mKkKr9rzOkAGAMXeaMPQJ3iNQuCskuEVmrz9qMa+ns+pUyfIeJqemtI6ZKgpKhoK41ZxORlSyY8yLY9hM9CmhyS0ChqtECLqKNWakYV/g6GLZaKaYtSJcrAtHzwXuUUeWvaL9kuliKwMk403bDvaI8iYR4MHptSK9/KRIk5MCSKWZXY7G0k1SclAKkqViaE8s1LOK3zrm1QI+8ZqKnXkwtC3kCoP+Bydrigg1BeMsd52+80Pdnrd424xfMTtuFNxVklQ1ZDNAABc8EasOVfkCdOaG6VeDaEfPJ+WHPLQ1qWdMnzCGajHxm2DRx34sYLaxZ8NXVDzcvEQ8psbkZVeuQhesxhSzjsX08j96IkjR46xWwazbCwvDz23GBs50NGKnOTmfr+ysir33XefL/so9U9FvdIALlVJDxRXkyKEpZyJ5NXicX4AfGBMaHuDsHi9VzkqYGHo3o5I8HWri1s0RuamCioUnPmDhc/CxkQhITogz/AW4ZOJueeTXuyc/WA3YZHwY3rAGnSv3QLIhgYjynpQcU3jOmimgMVC0QRBVBCp9Xk11lav2Jz0OjqdICwwNviZbeikiCXIttGCX67DGLRNeAAYmq2ZyECosWFP08nExe0MgxXHGI1Lwxj/Tt/52/elut1b8aQKJfPNEFuyCQaah6veb2g1Oeh0p3N9IVUAkTDEJ3uZaWbEDYXUZijwrICB+dqQzplRHyz9EM1VKFb6EaCoPzqAQTfa4EKZihzpsK3fHc8Fk8CLR4YMkr36sSinR+ScEDjbKmiCS2ZeURFxx+vNfa0WjAi4+hnqg7c16CSn5MBQgaPP+ssmkxTNVa6/L6iswvLrCHVo2iLNPMQ9Cw7h+lq95Wy7IPlYJPazGmMfnZYOMQ6KRr+I9gTIZegjdK56vG8lQp40ahtZtLSeVMhD10ioSyBNMGxZUE1fb/nAsRpgpdF6QAYG2KVWjFst3q22Qeh8HYdp51TSh7W9JUIWwI4y9dE9UkWKVEBpHlVJgNGRuf1bv+j031pPPH3+nM7omZVDIti98HStW4KfwLVCEs5kHnY1KjczfiDgyT1cvb4hR2ST2YxRqAZQCM4nQvh+HAI+bxFJpq8o8saYMo+MKcKcqFBLqmNwDN1ATfd3WygR4r+mZSZmcmB2qCdqEM5teluVxoKohbP3whz9MT1j1maKjRlFfDYmZn6IMFUGdic+Hh9k6x08+L+PNEZkYm1QedFxVZhCS67Mux1LzQ4211xRG0CBRYhAWvY2YKAXJAOANXTQIYhDckWtNkS/zKVsdTcNTVTMN5pk2ZCRhWtYlmzK7qGiMmNIPBcax2R0P0QtWZo04RlUshCdXsBIBRtwosaoEZgpElqtW0Ljg5H7yzEZ8XCdNi5C+mYpkK7POJhXdVK6P5f+eW+sP/bXfuTV/bS4WJN9L+UILxkNC5KBqokQKEeCnvnphB1VH6nwzyuQwvAnV+ZMt5MGNUNcbFvwBrbQU8TlOMHqTS9R3/J4zhX6thJDNZR5AVFC95bQJTI/P09Di3w4GfshSGzxK8rGAwtB7733X11ZPTPkPUlehnXUenlcia97hs6YubiU+sdKGrQ68oilP9dHhmUHga5RzTz0JNKBQV4PG349BuZoe/DfGk9rp776uqkOnQBmb5xbsqH1nnjq4wVvWXqG5FZlKjYWhEPgXwmsEYjTyQwnQx4DAaQWDeWhufWf0yyCfAMr2SZFrkfQiahUcBKNlbaGwpRI3MMAaVuXg7yChuMiasFj1yqY1alX83DmubvW+sm9sb74bYMYdFeQkV9q4GsbfPiY2WoLAu01qyzRQ2YiFC8D68U4u3gei2WPodieGkWWbYU7k28xbzBa94yYotR8DdIleS1NaqG2hseFJ9XHOaHAVMwLpVwDma2w0ZrXOTLAzh+hMA1L4saxiqKlqzcpQV33PEBffz2dT9jK5lsiqLa+ZBWSc2+YTO65wa39rZwzeXgoWMEd8LmY3ROUSG3JDSL93UkqT92at9G1Xx8uOsIqS76aDnw2eWqeLCWHHI+CXlxVJ2v47tf6mxqL0W6DUxkKuS9b+JLIBkJEExKiK2sSJ70RvOM8L11FUTmIywgrifiIw9fBI8+ttrA20IG41EN16uOVLuzW1hrbKVIQsAzrvqqFCQMLUaIoc12/Nur15vntWX/wB95ykQvvLlYtooic3apIlcHy2JVIXuDEsSLQ4DKPgpyizJorZGUnA3NTvC9yRF3o+jfKxCKV6XlZi1+SyId5vCqQVf00vXroYdRcuHxJwXdd6GVFBfAYEv/S3y8dOERwaXxkVAXeqC4Y+5YpkU67RwOE6NlnHv20nDp9vPTQFZ1LTlaL6y8QG4emsNUeTSYTrz9z5gy9+vJYmObFRUkeSTNVeUjzPGBnuW8z44AKT35AWGggjklrs1wVGwKGNPc8q4oocHNMa6t+7mrqgbQ41E10AFRjZpchaq0eiP7ihdsthAw19cxrK6WVWrcn68Nzx1E2V



KpCKmX6XCqUV4bbRZi5mlTa7WKqGRYhzC+G0OHcR0CWs2pKNVybjkPZUUNrw1/w3YDIPy0S06+Mt  
dXZ+4C7gFMFu0gS38EvoTex8PzKIEEZ18PNrIahyueNwtgDvB5hGIniPuwk19S3OGmeop0cGOrUt  
0K5DY6KhpE7vRMe7bWZm3nJSWVolkVhfmrQRNjr8VkWkBmNjifqrqftaCMWKKDpdIA3v+OO+6Uf  
i/17Jw+vWmjloR+UtvUkEPmYsoZaTmGwjdxr6ysMCSdnZ7RRSL6utgrG2Jh6wjGmE004Vwk8zS/g  
hum5fs+rq+EuFnQyLISctQaKRXraYJ935xu7KPY59a8dnEUhizHceHrropZWPg4LFSnFENGy1kp0  
q6YRkTuN9MEv2E00UclC+MvrNUxDXlKmbSx5p6GGPmm9aLyczAZa+wwPFOMm4Y2SgRpl7g2xGQq5  
WlprE+ft8b6A29906ILaS/Pcwn5YDlGz7yBhG4VBY+igNaGkx6SGSml04CSUuTaQ/9W2pCo1IRt2  
EyXTOF/CGdz3oqUw3jDrU2jIRqjeqzSWCQepiUWck7+6huzLWw/dOhIgPfDgvDdIKWxx/Lww5+Up  
59+mh7JaqjNWp0SKDbFG+ca21jESidKWZNOSfXr9NrMU+ve63I+eIVIDtIBjtwleLWpbOJGiBrCR  
yW513wIrOFpSVIAMo/36XXb7C0mljA6wWiGERI6YsIkrKhUELumLxgqDtgmb8HnKsGO51tVTAS  
jizGa7EYYq9z0JlXfl+W1YDBmWPadDFsnp6pf5ZnVRXnEtbtJT6ShQb0PIiCZ0NlPuNV55Xpd/Z7G  
5/pP6/nzuf+89ZY2930Pe4CLCK0SSsnoaFHLchHVj0cRcsa9SEV/cJ3Qxh5YeCleMxRxdqOGhQOo  
oIgCHdvXxPDDW42at4DRuxlLHzdjVzXvGz1stDT6mbokRWP9GmEUNIareRTzX910WXUXTIgA58Jt  
DeowBfaa4vj/9jHHmCEUCLJmGToKDhd4gFuZasjEQQsbj3W1lblkMHDpc8a+RhnEnrCSRpGGkgd  
amIpnmDyrlGLkA+NnLXyvEWK6gWDACUqxdJGx5kiZv3thspIN7FJBRX0Duve+uilFYbRfdr2Ex  
o28klfGJUX0XF2sIdlnOtwstHdPPeMoOofzHVQRc4/GV4cyGzmnQrzgRlCPy+FaHtcoFuryAWu8  
oQtSuing16c5I+cl8b61re8cdQtzstNsiXzHTXGkTX0tfs6WUA+xZA9IL15FNrjkNoQJmNtGkbyY  
okn++dRaELH+2g3yMB7WN+i5ZzP5OU7cpt4nZe9shb+BQVSzj9JAthk4bjdVB3JEA/t0ECqQZwHE  
YEC2ZEyk8bGR0KopRrGEP9+Sp588knN8zyohMVhvakG2qBORJzfeqjtGdUPRnDi9C1ZmJtnCIy50  
eblygWvnS+UZpHIS7RgTpC1sSG39bItUgQPfGTR8V5+grzem35lyHSTPajm+bt2raJnkSfw49+sD  
deioaFYqiLh9Y0y8Z08lWPOFUUvfDOD5MM9xkR+Sa6Iw7BmQ4S5xnx4W+UXE5vIPEXURwyRDbvye  
TVxERmOonBfiXsUeaUUV6Y31s9bFNk5HVzcmHujoyPL56WxOq/60/CquCZEO/3gJBgd29QqDJWQe  
Pu8ohaVQ6ViLwFp4RkMlTlk3+9ogzT0blp/J8kRqCt6aL3hpTmVKsXaqxipNzrXixnVjeULtmIlw  
yPqw/j6imZTJTy2G4fpcJBMMQkRYyCB7YNj29vtSFyvyT3orClUd4qhrXV9FCVlMOzycTyUTkCGF  
Ke4uXWW4f7s9FzZKZQbY6ePWCBOvTBvy9UrWIOEje5IvUETYYZPSTM/MjJWD42Ixv0009jNM1UHW  
4UZqFIOWjYhOimGm82ViVaEYoZhFWH6XNbXULJkNVmPZNfioZa1ktmkLCJGD7nxxHWGjaU61alzm  
VeeNITZojpjSXEjztQ7s4TGNKkfrot4R6HXC+BDdXlBWE3ihVIREzZmM53zzljf/JbbE3cit5lig  
XJzc6+o74Ek8fQuu0B5FIrhdsdcpoq0Uyh224U8GSA2KzBOoKwJlWXUOSeFbowoaP4wdRgvlQA11l  
bhgRfzYS2HaRTZ9JVOE4G6ap5VWOZ8LSRZKIkVetvlnTky7P+5zqjk3CJ8fWa8qjLa125LPfeYzX  
PwWibB+nGndlIvEj65g+FtRL9TQqkfPsb29SdUHY8i0+wMaQZaX4Jwee0YVfIJsCa6L1lULAnK4v  
ql6X48BKDFA2xXt+M8NgY3IruLlHlyJcH5dH/ai77ah7XWJbgRZmHqeqUHy3hXDUUFRswrlnHenG7  
fEMU6uIPHnFRzgs1PjGcs7iyZRoUc0/y1A1H6qVF8UgzNo18j527CLPA5nGvG5uFYnQJytDTQE2s  
MwIE8BUGrV6/+JjF+fnnbH2er2fcVd3kb2bmXF0y/AqowxL37M7ywlqRh20uqf11NiOm3lpT3qGL  
K8gyL5zpbLDM0x0fwpCAOVhogbDwyG2Ua6yKxypIEUg9ds80WC4WRbGBA4JZtsksahsCrCFNDc/4  
469wfKKDdm14b02W5bN5a2WotZWziq0Bc7YPLU9pbxQcCZo5rr3OX78OZlFwGLpxMS180EWQsbcz  
+QxUArnWPM3Aj12/hzhrWIenn4f5rUasyr1g8Kqhlp6xJJ+aTlo4vWcGolmuN6ss2daxonjsrcYu  
SA2EhNUCYUYAl9+FEih9EJcYzv+OiKmSreMxtdB4F/bZm0loKqCptVls9wAxtPQM3nqlSIIElW6f  
rStMA8oc9AtRqQFWFJ6X9aFhZo1rsPo6PjgN3/r3xTnnbFy0nOWN4m49XtKbFNAkIEL2i0y0HDYX  
5Bzux/iKHsBxG6oo4UxNOJwg6tqCwXlOEJeldTC0GUJFzsOeSqL6IWG4mmk4RXD4KIy8TrOQ7uTN  
boTzSxeWD+DZ028qr7mTsNqE8hh77vvfpIL8HHoBjLvpeSHwdDMGG4eJkfiz2ltbY2h9uTEmC5if  
05QyI8gotbYEDPfQ6qRjl8IJ+IV4FIyiYK26RM55eq+dRjKkLZpniBbKsNps4q0+zqDJ+tOd3Au  
uFhWWW5ru+nwhmbiR6RLXmZNXzdWIzpxBDeTwc0YoqOjswrOaOE8RV51lcASq8czKRdqdUNF8mH0  
pnqNMFAcIgbXvU/1k3Eh3WhGaAiSIBgB2uv2WzunXddN294wy0Xux07eFCU3Q+hiwU7makRIJTN1  
R9sOkRD6uaRqvsZ8dxCDLSFqbJAWc+0KXG4fn206it3NrvIlPukmHQ21F0TiBG5D3URAprmji+dm  
AZtaDh/gTxpddK2LrrJ6alQgCeBo8gDMR8G8vAnHpKNjbNhhku/KPtWjYxv+auVpGpeigZG0+m0Z  
Le1Iwc8S4rhZVoE4I1jGE3flzKdyLtU5zeJMt8lFJXX2o/jKIc8VaIKdx1Y+8XokYpYXESNow6tY  
zElW2LfSsdCPR8SDNPxIfE5Df15FDSJFRgqQipktnopcXKDBHrXXEXE654jkGk/IsL9RYE2PIegIJ  
4mXbK2pPrBvVRSCGUtVefwZOXcVh+1WbpmzCTq3CVxuIYm5ar6wzV6VWwLrD23TT39Z7Wp2l+Us  
boLdoc74SmyfAqrWUog1GOREfTh0il3LyqfBONSmJ6TusnwEV8yGugqVZXDihDxjaAQNJfsvIRJn  
XoAhlZvmRiIt+6cCvUMFEPFSUVzmgcLeFnJEMlhcJ6Bw4wSYI8BVR12jxxhdxlH1RfT77ruXdd1Tn  
iDripzcGkPCeq3ux1s0Qh3aPCcMcXl5WQ4fPBB2cxAgPGuSebFqQ1lpoySXU/AMn1Gz9/N13xQ5f  
Z30PhiUjc4wI8PiLztnJKgm6KJNFIOoJZUOQdjk6ipJg95X9dAwNpuuZx051FbCdcd5pv0AphbeI  
HEatVipgiiHN6yklWpkhPVTnTNbCnhr2E9GkTe6qJ6qWogfhe0esM9XiWoX/TJq8LX4KNBJAbbp3  
1MBw9f3LT+1EDhx6xdGcmfjvOq023R/67wy1jfd/kYUto7lRVmvtEVGkSy/ywLMYC+ldcSAYeMnY  
Zf8Tm2MzpNKXgBjySSIZzMD9kN3oRtrU9IisKOyInCESZMy3GQRXmUwCFHpqrGBQibsbGp2WVoJh  
yqIoge07EqGoIOPgRE9Kwvm7rPbnV2GqilntJNTY/Lvr3yNxmbnaxMDcC2I/ia6U505U4tsYKmWj  
txxLZ9ZlsnJSZCHTVY2Lj+9nOEK9ec94prQ8AdQlIj8uEV6WIirqda596JEhmbBxBVKJ7GCTB9aF

```
Yc7F7TlkKF7nvnUwpQSLD1Fih4AJFETx4Jn/Z6PFnwph17e11R47TFxzkUcvYE2jsld1JNtddkk
OpGk3iwb288Yg73yggsyDBpJ4hxU3JHVcjGVUwSnyPrp8+yCokD1PkQSJ3Xk1VD03q5biSghUMC
Wr71QjLhngzvUZrZq0JY+65VPDr55Wx9tPe2wdFvlhkurvYBVPjaLoF09PdDeERRKt9iEtQJSstg
ABHLAGYUINPqfOgwFMF3PFhUy0pgpejHGmiuyJC7Toc3AAk7HiYEG6epyjV+qKK0ZIGgIXlgSMTB
Deggp1VXtGYzVK9+e43kE+BggFOCI+8Zq105H7wwYfIGdbP0WNBd1ApfdLgaEvoGVE90DoeEIm0d
1sEeg4ePOw3oYFolaEfhhmzrOLV5hmqIdfG5av5Tpe4pANaCF01DGiEI77UlnMzURplhZkppR6RD
dqiTAzvTx7IF+UG40fnRv6vc/WGpq4fRaVEDgNr686BoBvEBuokvVvWl2aw2EJNDgN0G3gmXo7e
1E/8gJCdNKzvllKK4aSSoldMgj/R3WHTzokPqAAUaFT04Oomu+z5gUAsUfHulgaVieggMAgWjEbT
SpT7o6ONrvnlbG60POfuzNpApHTRdCh3g/xmaLkXhrjROti/XABrNivzKN6yF0093ULd9D3gEEqR
WC8GPURCQhnCEkRdhZ+fqsvqmrxm8GY766Ih8TPAm3Qz2EZeJZULXMaQvehiedaC0z9pDa8fmxyI
lDYWNZx/23t7MqnP/2pgHoavVAL6F1QxXfFBxbSk1hAz2WV8/IkaMXKTvHkygI1ORam4y9CLXkf
nFGEaR187PxPE0y8vlapJPoQq+ons/GxoYP8WohnLaiv0hlViHJflbGXSI0xSaUBsQ49rVtExIf8
AgyyULdWHxVgDk8ucSMNLcQtiMq6BdKMFHQ0eROPcWwaLiw0yiYWRi1UeQVTW0sHV/zj8MozUFFJ
UKCF2XK4sdhAGwLM29qcWjER0qV1JueSgklEDuuiOUqPR/e59ZDD32i+LPalf8jwAAZs7UsKZQag
wAAAABJR5ErkJggg==', height: 150, width: 180, top: 62, left: 50 }];
export let imageData2: Object[] = [{ src:
'data:image/png;base64,iVBORw0KGgoAAAANSUHeUgAAAOsAAADSCAYAAACrfzewAAAAGXRFW
HRTb2Z0d2FyZQBZG9iZSBjBWFbnZVJlYWR5ccllPAAAADF0RVh0QUxUVGFuAEJlcn2luXNznIGNvb
mNlcHlQdGd210aCB5b3VuZyBidXNpbmVzcyBnaXJsIOhbn/AAAA6aVRYdERlcn2NyaXB0aW9uAAAA
ABCdXNpbmVzcyBjb25jZXB0IHdpdGggeW91bmcgYnVzaW5lc3MgZ2lybCCprqv/AADyDUleQVR42
uy9CbBseV4W+Dv7kvtY97e/qQ7qqquqprt6pWwFAQNXyVHECUHEGZUhcEGdcTBicNwGBQ2JAMNw0
HFGHRxHR3AmJrBRERVZG16ofZXb7nv3ps398yznzPf9zt5X5VNd9MNJVR35anIuvfdzDx5zsnsz/
X/fb/t+R1VVstle/+07/vjvkcD3xPc88VxHbMsSyzTFMEQfn3LDd8LvpSpLKctCijyXoiJ097Iq6
+ek1Jca+M3ADg3DFIufYTti6ucYeK3I1/wP362v+2ff+WeNsD0w2sN9Y+fousmfre7AcP3QsB33U
xxKvf/P9LlPsBV45Hzb9u745W3GFqyv//YX/puvFBcA9V8DVse28bDEVGAGZnxqwrXNYv/rPfvf2y
/gc2sztJXh9t2/52i+THMDK8kLSNMMj1TTLJce/8wLguwDir8Ia+b1/+Ru2X8jn0GZvL8Hrs33zV
3+BmJat1s3YWM8LK0r79+pmKSU2N3bx02eR220Llu32K97++Fc+IwaoJzFpbJyzX+xfwNyF/zUG7
A//P/+HsVoupMwyUGdTmvx0bBeWH0dvVvKbf/c3bv2jrc/6OQTSr3gvPcYacOonmrCuDCTBwsI/t
WBpbfiqruOod+t5LvXPPBvx91Y2E/kw/4yfdav/QT//xcd4z/+tj9m2I2ukZeladuWuZ5H1nRyZ
mN/FnxbyzBKC6sLXF2zSuOodH0vdxwPTxdg7FXh+mFuu045PLhcXnv86eqxpz+wwWG2YP3s2v7EV
75PLq7d6wxYezsuP02wmwBreQFWgM8A6nz8tL3OpgGliQ2/OXAdr5FmSS9P0n5epG340YFhOW5VZ
J5R1R5e74C+VzjOHB8S5XmS4fNSYHmFz5o2291zz/Ompu8vzDyPWnsH6dW3va84vP5YGYTN7Q20p
cFvYKB+1Xs1/8L/NH0hleYjCC5y2/KCB+e/iAR/whv716DEfKrc/B7g4W/2N9w85+AYRthlgIN5i
+16PTdoNePlwjTzwmwEoz2ny6u+EzZL29hLkmq31eyYludYeZ26VpE7hulbplFgsakKw7RL27QK2
7UKxwnWRZbOKqM8sQz7xAmap14YnhiVee/sF37y+Pyld58NLz202rvxWBq0BhVXju3dsQXrG2L7Y
7/96Y313Dinnz5gf8mbGIC114s3G6Wh068ugOrgcYDHU3g8aTtuC4/uejW75fnBfhKtrJieKY6hC
WPZcBx315AyDH3PhMmFpZTA9hoNvKZVVV4Xdng3jlaNcgHDCV4sZW6k6bla7otgWJalBX+zHLNwb
a+A1Y3wWAG0sVUZM8dzT3q7hy/4zc7z8+MXP7Y4u/1y7+ihs53Lb4nwtwL72YJ2S4N/7bb/9sver
v1LzWE+AOWm0lvJZ0KJDVDi6lNTYl0BngQbftJyvHdjXwP8uw0aXGA1sLM8nwAPPgA2w/E0yiJfY
99NfHhi007btMwGcG7hmPDTiPG+LqzoQbReDZk+siyHSwrOxTaSaGEkSSxJkugjx1IBUOI4XQldT
yzXEtEyCs/zxfE9MasqXTGml1nOzUrO8fm3mr3hh9s7uz/T3b304db+zbtH1x9bma5XyLYAYgvWX
+3tj37pk6+mY14fwh4iHxY4s55wXPd3+u3eM1WeH5VFFhpVkrVwH7HPUVhkZ/BV3aIoI4A1gC33j
LL0+beyrHx8qg0Azg3TcG3LflsSpxJna2EpK+mEEjSajCXJYnwqebYCEBv6nB634ch0eiKrZSR5G
kuaArjwlXmsXhiIa5vSCJo4uUxZuYUDA4XdwvL8pRf4U9OoXvYbZz/t9Pd+bOfKwz+z/+i77vb3L
q9xjcrtHbQF66/K9ke+9K1X4fdfBrBdAPaZVrvzRelW53HLMgamaTTWi+lb4tVcBnuXfrRI4z72d
6cqMpjEvAEQtRipTZN4ty6wyPssvjAAxHW0hHW2xbBMKfJUHNFHZzuyWpzJer4Q12tI006J6zfEx
LGQsCZ4j+W6m4iNGHmxkPl8rsdY5LGea7PVAWA9raAlmFhkDFjoClw/Xk4kS9IiaIRRs9MdtDrd5
xUNB093d/Z++Nj77/rQ4VufHj1umG2p8Ras/8W2b/wtnyflxur8Z/B7fQBrALA+wPAkAvrhzs7T
wWB/5Yiid7mN1st7KAlJq9oM/eAEFmuJ3cGqZlaWRLt51rtFWZhplu2WWS5JFvMTSGvKVRkSRwt8n
itxvIR1XEOB6+o20hLiYduOpFkk0XKmVtNxPamb3bpMEQsIGLAC17ADUGELIiwkjrOCrYezfQZFr
myghfcEniM2WEEA0JdYFKbnZzKfnRZ+0F1eUxB9uNVq/bzjOP/m4KFH/93D7/3NLzYH+2tTtoDdg
vV13v7k179LCnh1haZOXlFAGgCsBcC+BYD9QKPZeQTFbb0VH/JEnmV7ZVnargeglLk0m6H02r3nR
YosL0tgcHKzyqt2imNaLKcKNm4uLGcCYfbqg7r4KFOydCXyEXgV5aLCajtFF802CisbSPswqo60
DAXPirAmKwUtPxcLBTi2D6Ad1+iZC2018TC4sp6DQCDJqdxLHcrhgAa2Ab0sF70h1YaljmHKA9v
38Pi0SctvuD2aVLV55rhM1/6/ruv3zsA7/1Zw8eeXqGs9/S4i1Yf+Xbt3zN5wv8wRqoFz9fH8AaG
8A+DMA+47jezX63exX0dw/s8EpVmc3Van7IKiITr97ZPZThcPCybVunwOq714spALNWMFQ16C2AR
7BmWaqfzd81Co2PybMEVrQ1zKOYVX2cJoPKwEiBlyfrmR5Tgf/gjTLyPK/ViLZpa2CJ52J7nsxgL
```

fOi0oosPk/rzY+ZTkcyX8zEDzrSDFoCViCddijddhN+cUum4zOZnB8X8J+jw6OrrwyH/R8Jgsa/O  
HzosR+7+Z7fNGp0d4vt3bYF63+x7Rt+89t/OYC9uL4Bfv1toJpPgB7uNnz/IcBjP0riS1EUtSpaw  
PLYOr1d8axKcIPP+r1OJytSmY9HoLQJqGahJRDT+RIWMQNQEz2SLE/EBYV1/RA+Z67AZfGElKkGj  
8J2H5QVIMR7aYFb7aHSXu4viWcwth4AGWPfLvYZaeMBn4efKT72yX/HaSRZYer5a0oKq0CWVXJyc  
oLncqXX9G97va60m77s7++LBQs/A2hn5/ej3nDn5Nr1h3+s2Wx+f6e/80MPvfeLT/duPpVv76otW  
H+1AWtsAft9HGCNDWAfwnT+G17aanc6R6HnPYp/90F5W/PF4nJh2BKtFqCtvgRhS8Lak0GvLU3fA  
4AXa1Fp8WgJ79w9kcl8Ju1Wb9MOZ+AnQFRkAIupAMwBcIKVH+8BRPB9cRgAIADWHV4FumOcQ6HPw  
adUS9rqDSVaz6XKM1nFK0mjVNM4Ftv8Ydk9gHa5XMPssYZ1JZuu2/BSfn7p+Vwi+MXM0/LawB3Hi  
YZy48plabUa2O9S5ufHaavdHR0eHv5Et7fzT8NW+K9uPv0lJ5cfff+/Wwn4G27Yo4jPYTFgnTVAqY  
DdleJvVrixrV8zcvBAwBZ+U9wO8v8F3fRiVxk3LMHp4j1kUuR9FqyEsagJoW5Vh2Y1mh1FVmwDrW  
KieLNaPjGdzseFPVqDkL7z4vEyma/HDhgxhaWm9CURGU32ysJ6ZqDjHlUNRAUNX4NatMa001Rgz  
kbHWJ1z9UUTI5EGKKuJfa2WEwmbTezHhf85kMSPQJcJIf2O4T9n8Fob3YF4oMbn5+eSV6k4Bii0W  
UoDvmscrTS45fiBWvaz0UTms5XsDVTysHeiReKyOx3dHb4QJe/a3V3J0cFh8fJP/9C/xvJydvT4+  
7eA/TQ361u/9Vu3V+HT3H7ge//OxxOTj/9lU9ckLfz/t4Aavr8RNgadRvgILFELz/1VXvRLy15bd  
rD08qKRJ41PH1TyutvFoWpDZcr52bFSUBP/Pj07lQR0M83remH6juv1SvtkUli0FazXZDLWnlkFd  
1Xq4sH64bLlNdCUwGKa+NloAYzribKDEgBbRQv4wKdZFreiDPuHlcYZhM1AugBoDj83y5aSgwoHA  
OtwZ1/WyzmOZw3wNsX1TLABX0Dp9RgYkOLfC6xhg9VKMlh5tbbdHavIM286nfsqKm/6rjtn56MTX  
LG4s3dle3NtafDrv/2x3/bOTxZ0slgAgMcRgPlb8Nhttdu7zdC/ahi2j5eDzxp7WZYP1+vFYZGXL  
sBoGY6v/meOF7QaTQlct0G6nJ1Ld/dQ0iSXk/uvaCoFUNbIrB80AL61Bn94DHUPraW+ZwEQ2aCvg  
efLoNNWi2rC2jLwJEYurWZPWNfIHG9ZbkANWmwAlMy7AkT6nka3LS7pMmgwfebZYgYKXahvGsC3H  
o/HAnZQr1LwezMsMHeOT+XkbCQVfOjAo8/rwvrm0u/0pRV6ekx5uk5NsU6Gg/YHr924/g+7vcGPX  
3nymdXR2z5/eyNuafDrTEXMC3GN8uMpMe/6ffzldwEAYaPRGPqusxOnxcjz7MtFmb91tY40AcRGk  
eU1bQZFJbja3R21YHiP+oJMj4TtjqZDlVnZBaPrteR8fKrUlBSTVcT8qdFbjQIzCJRKp7cnaTyXG  
L7uOQ6mA7/R9+h7NtXKLgC6MAxVbobbpmgyfXxSM/gY4lrpqKYVfm8CSko4b+N2xKtnr70iUxqDgk  
UQ4phZ8ax7vdHJeR6Gx751hTwIsJHfuHuN4SpmszuoySsMVVkrS2lpV5Zb5eu98Ur7ffP7ZcXn9+  
tj62I9/BGBNt3fXlga/rtsP/pPvkU/MhY09/O+r+Zvv+7uOY/sAcel57qNJFL01jvM+ANC4yLk24  
RsqpduUC7y3klFd/B2+Y7RaamSWOc8YFrSCVbQ9fxN5tvGw1F9uNLsAV6T1xB7AZsJHnc9GAGILv  
m2LKVeluFwImKKhNeXBLgC2uijC0bQLBdOyeIH9F+L6vu6/yBP8LdWOBM9tqL/K57hYpfESfjFUK  
LDr2njUfrUJ/zYIfZx/ILMpjqPRldlkhsVnhYUhlAWghfPDQmXFAYK1zA3i9WplFMnteHxvuf/IO  
779V9PFTLaX4DO3rIzA/s3v/0n9nWADiPoAwW+nV+F63iGoboK/Aa/WU/PF7PHlejUAZiWcJcXwr  
BqajO7JCLsXgPEVILZaVUZaYeJAhwHU1VgMgMwAmGwgjxRutCenhLWzBRZKmrBuJf9IvAxuWckp  
5RXLSYseAZfMgZYF6xagvH3YJkH+9ckwd/Ox2daWsjzIMBYmRSC9nY7O9Lt7eixcbE4OX1Jivi8O  
X53rVz2Di5LujwRE5/fagSaY93f35VG6Qqx9LsNuXzpCDdXlGdHV4XiE6enI8ngc5+N6+opnE8wG  
k8ejtLqN5yfjT/v3gs/3/zwB/+3rcjNFqyvI1gtU2/uP/1V76mBC5QBsF9cJ3AkSJN00et2b7q29  
b7VYvYW+KuF5YXRajXz+YrAb6lYGi3rHm7kdrsrjmtJFi2k2epJksxhcWMT5dPCDACPvhFGGv/Ot  
DAizzNN0di0lNwnLFuRrtQX7Q96sFzsJINFLbM6pQKAM0DFet/5ZILXptLu7en5nJ+fAYzHmi+lz  
8p64tX8RIHugdI220NYy45Mxwx44bj8Jka7kaPLbwGdhr+9nkq5nomVL+XK9Ruy0+tisYkBYk86s  
NpSrGR3fwcgTmS5juHrTmWygF/tNyy46c3T0emTk8XyC0ej8Y3pyS1ne4dtwfq6W9bNTwM/H8GjB  
atY4uHt7e48ZVTp+wGOHdNtL9brrLc4u9egj0jL1+52xGbUF8j2HBsUM5A5LJwJi7pcTTWgVBq2W  
tiCZJdRwOdtBCD5ZeVFBCDC/4OPKNpHa4np1FaZhfx0c334kzn8T9Jo+qrsoqGFzmdRmM5Zwm9lp  
W+I58JWV/OkO9EI69zuibelyWrOgDFdBDMtt8aqC8bw0LbOGZ+TqsZysH+ZSwapqaL0sVYBv2uX  
IVVbcJPHmLhaDZ8jTC3sThpmkmPo5LpdIr9hG6eF8PFYv2e2Wz2ntnopPcz3/dd23tyG2B6vSyr8  
dolzpNasSEBcm7u7u68t8jza6kVpiwFXC5OqXMkw+Gu9Pq7WlTrghKGfkmKAIwVQvPFuaR5KXGyl  
nJcaMEDG1TyjZRLFK01aBMBCKX6uDyAymzqWtvYGHwimJgj1aAX9kVaDV8Q1rSmzozoMuhU4vcIr  
3MdS5/nfl1lbrJHiypOT0Zkso6XsDAziOQFAXYWhU183B6BD+KCMHGdxLGtjBgD2lJiFXLohy+mkj  
mLjfiBCgLYt3jLHu5kKskHFM5xjifoJcMxN/FvEleon+K1Jcm26NJ4Zn00+4gfHrIeMt3falrL+i  
rf/8X//YQXsX/re/0DL2gdYdnaGw19/sL/zO6qyupFk+e7o+N7NOI4t3wvl6pXrWuubrOawVmtxg  
FZSRM9hOYRoEiFtbKzzJRgvShj5WlqhNikfiKeZWr/IdEpdNE8raRs1Fa6KBI+CuThx4Ue30t1hV  
fBysdDXeR67YypN/ZCGF0Wlhfw2uTuOqYBkiWEUL3I+mw3SOqKWvTJ9pb8RfOgsjeJed10vImspidl  
nHx6Fw6nZb0NRqM9Quv8WEGWqElV4+O5GC3I038odPpaFXTejmv2XQs08WcdN1ylw1Yb1v1x00on7  
9+51f+Jf/Rt2/tyC9Zf2fZXv/4LHwD2z/++D/gA68MH+7u/CZbqPVlW7M/ns4dnk3GPlrEdenLly  
hVtJcvjpVo4lhKyRpe+JYM9k/GJRm/XALJQYQVOXJwstKuFlJ0legboZRLF2qNKn7Iq6+J92xQtF  
uDrWVThAIxr0Gh+TgngMmdL2kkgx/EKzwdi+a4CEJYMTBrWF1Yuy0u1yGyA195VQHwF33KGz9a20  
3w+a47T9VwaraFY2P96ea5VU2zR47GbTlM7hZgi2t0Fixj04Yu3pRU44uJ4+5227OFvgWuoK8A8M  
a1wkqRa67xYzd20KI7OxpN3p2n58Pj0xN3ebVuw/rK37/yGL5U/8z3/Wn//K1/3Ba7nem/b39v/c  
gDsnWleHM1m0G0XuCmt+TatWtycHAAwDQevJ+plX5/qBlZ3uCL6X34f+cShKCLoMVZtFY/tMzqj  
p9sY1ELbYWrGwW0YZ2+Kah0o9XZFEPgNbTCoMeknHVvKkx2nkjQ9LWcSi221gN7ulAkOE7Saz/sa



IqFBRWay7Vcpb4k+kmcawUSg1UMUFXwjdkr69qhOH5HUzukygyWmDSa43cbdBrMwa4A2H2xyhjn5  
klDiyEsCUNbdvs9acHC7sOX7cPKRlhcuEAtVpFlW15ntpw9MltM37GK570f+0f/8zYyvAXrZ779n  
T/51fJN3/3/6e9//Q9/id/p9J4cDAZfVWbF+7I0vz6dnPdYJrizeyAPPfyolt95jqtAaneH8FFDW  
KYzaTRbkqdLreltNdrS7+3BR22o37ee161rjPISCARfLT9a1pVJmxJCg5VHYZT6LRR8cewA+14qC  
MOgA0DNdT8FLJrv2uJYbIWrfPYu7Wi+NMBnP7DioLTLZW2RHb9+n1HnkoGodQS/t1LLyhTOOkqUR  
jM6fdFtxMoq3kbrfAT/t133L5SRHF66CV/XUMreCh3ptppYLEzpt1uytwNLO2zLpaNrGmzLsHhF0  
QpExT+YLJZvn57Prs7Pj7eR4Y+Pl2yLIj719g/+30+Rr/u2/1N//1vf9Fv9Xn/nba5lFAVcvvcla  
f7IOlr3tX8UgDo8uqW0MsbN32x1hWWI09070gYNvHTpGgyuV4tyVxmAY0kXl1jYHYFGUYL2aqURg  
M1qJDQ0EcAcK6X/dgGqIltpdw6rlkqdp5OKerKM8rJIOoXjoLK+G2iLHMHu+dRNqi2rBT+ZllkpL  
BYE+q5lmuvrqzLR6LRNJQtQbpZAstiB72U/rBe24K8utbMn057ZDsBaSgg/VyuxCB4sklA1DCuQP  
BpJu7+nPrQWbWDhiJbzTQlJKb3uAD8zbaUrWfWRLUxcMytZryosdq84jvXC8vazydGTn7+9CbFR4  
E9v+71/8R/qz//1v/9dTnewew333XviuHvzfHz/3XEW+a3mjtbiHu4fKHjGs3Peu6yPgK+51nrc7  
rAvnTDUDIrZYAvCNaWdBQC632vLE489oftgbnOxmMtqvpD7o/satWWETgTwqDhIXzRwWpsgUT01j  
lvlsdIp1qAS87FpwwRKqCJnGfOnKnJagsiyaw9gNgFOy1WLyU6ZKF4Ak7B+YAE1/FZaV2NFeh2JH  
e7AAi/1PBqNASjvTASqxuN70u0MxTbWdf+r3lGaHJod+LkAftiVdDXSUKq228GnV2Deuv1KzQCS1  
RZ0kJFMF7eVNURJ4vquu49zfrwRuv8hnp+TJmy7crZg/fs3f/zn/4DV6Q13AYtnlovlV5Uij4S+7  
/d2Ds0cN3prMICFSmSEGzgm++pDugDQ5OxlnDfARcvxhwditWrqulqOJVvPtBZXaWMSsgogsmPFx  
79bu33ZGbS0k2W2vCm3jo9lMVtK6nZruuv76lOm9FeLVP1S5l9Ju72gKYvxvQVEFDNA5aivmsSJR  
pwpKcp+V1YnZWnGXJQUMSPBPZme34NfvS9pOteyx/kik8VkJL0dnHq+VDre7OxKvBiBqhvah0u/u  
glqa+Fno0iBzp6JX3XBENoK4Hw91pY8duIwp8zzvX9yJgVWNB9+93w2keFwKNPZAoxkYQWDYXu1v  
P/oeNK+vtb3ML1j7Z34JYGf1rb93/7NxpNVjKIvD9Pkq+Ab/XuqsgHYautKvad4b4kq2md5wQwK  
lDWVqevhQxsTXOD+ib1bEN9QUZVka+SUV8oXamiYKfpm0rrfdUPBHAZifWxz8A1Zdh1rW9d6GBiP  
74Ha5izdthTqWwzqzQ5g6UOQHsplMb9NkBdsXOYplJb56j4EAT4WxmryiKlW1jPywDXfHYKP3pAt  
WHxLBvEtXQX709BsaM1ALV7BFDP626aZhes4RznFWoHkkmKzRm0LFsMezjPWGm3H8A3NipNHZGm8  
zpYDpsScC2w4qVgAzYL/UG94zjVCHhVFWaz0SmTJHplt9d5dvL8T64vv+MLtzf1rL+0luz1XYBp  
rcUafzFuIHe6YeNnu52uzs4B01b10Xt7F6B/2ema9nZu6wpluPbL8r4/EzpbduAR91+sYB2Gz1B  
W3QSQHQz1XxwThr4obCpvpqiJXWBsWDjxrCoh/2G7LTeljuHZ/I6fLYPha6aP9rHYcpcrYNlBqwa  
Tb68HenYpkt7bAhoFnc0AAop+M7sGRHwpCyRRUJALOApQ1YoIFFx7AH0sTikIH6epahVrnAcZ2fn  
cnuYCgr7DcoAmk0Byop4zuNmsIusGCVGawlzrVYa/UVa5s9HAsDarTmebGSNhY3A9fH8Vry0ksvY  
VHAauRZsM4N7ZHFAGdWbQ+wMD0aRfFeGi3ZPLSVgNlGgz/19u/+7rdaZVHtwgK8z3SDx1vdgeu6t  
j3YBSBxI1NBQYXGDEtmJ3fVutDfm5/e0RyppmJwI7MA4PjkWEYAL4sRytLWwcr0M91zXpmu5OVG1  
KyytftQrFwxAKScur4AOqNBTVi665f25cbVa7Lb7wL4AuuL250BHMq7ACxZgDitAXbtum2OFksWcb  
F5bfMq/0MfMizrdo2AvGDAAaIM5858RKLlybNU6skifvjVVFVcpxAX+PalzNED4PKyTi05bhNws  
iGz+7cAxB6etbTQosyWak3r6LEly9EtabR3sAC5cnh4pMfDa1ZqvfIuP8vKyqLhuY1rUZJdTvNkG  
xXe0uBPvf3Q3/kW3KtZiBX+KdsPnwGtFR/u6v2wPbBoUSHysF70cJ+CZsK39BsN6Q53tJ2NdbVs+  
CYoaHVxx6qvWiYLaXd3AWpPlrOxRmtJadmpUjFC7NbC2xX8UEaDKx15YyowDAIsY+CmI2HT1b3dX  
em32xT41s4f2aRSCEwtPwTQM3xeo9nXxvJSVffhV3oNrURqBD3sPK07ePC3OJqpBlROQOonZlq2y  
KCWadf+bVfpg5+kRarpKYuCTGZVS6Di+NlNxIBVks718x2/qefDXj02BTBn6+Jva1j3xuBIF49oN  
lKzOZuvsNiwmmoiDfi+eR4nzSB8oddp/8L81k/HB098YEuDt7D8JJTDYgKiGgBwbzOM6hHcuIUTN  
M31eCRT3GBZtJLOYE+6/V1JFmPNpzLAUwKonpWAZu1rGSH7PFfrmq4r5qMymM55UTtg2I7G4cX0W  
V1UQEvgG9+m5bJq0qPWiZHcylI2mOeg3tEU/uRA+j34ss23ygvPPytRZsq9u7fUr4WZU+uaJ208B  
Z9JuqxSpGvQzT2JpvckCxxvSZs1wtTI+2pzNCWALFRYF9p1SBkZ9TNBZ5kYtLB7z1Qj78GTyGWChw  
vuA0XZ7X7J4IjEWjNOTl2Q42NfPzrKF7s8N+oKLBdBb6hsTsB4WkPNbH5b+1cfk+lufoC/9hHZG  
Xbl5VduA8BDysZYrWajm+f5tbxIe2VeTGQ7N2cL1k+0/TCtqp3W3pouP51Et2qTI+mZ7dNlsgl6  
7UqNrBX04Jl6zaPJKesJlMpoSeu01dKqGDF7R6uAq1aokaSaYe48QtYM//B5zlhW60hb24tF6Q/C  
YpKkNJKMc3CIgfWA1M1v9m7XEuJAmgCAN68d13u3H1ZjIMDGvFkO1trjS4pMHOjKm7hhFhIGNFNp  
NHb04WB0V2WCfJ1pMPT6ak0OvuyMnwR12uqomG/11PRH6nKys2xQPI7IFld411+Spnyha69WKEh  
QiaPbsne/tXVeupAAVeT08kaO11YVmpnkj7wahNLHize49K639h+TaI4+J9dyLSrHv3mEAWFemR  
hyvr4HR7OE9rygN2Ij1u3381heZCYrXBkhv4g4cFF10LY3XNkdIMLHfbIXyFPv12BRjJuxgE9ZA  
LTmCKqWkRtq2Z5ZeUppHdDAFqwJO2lYoVR5rTaXZV1WcEq86bXqqGcYg6Wx01C1QurtXYugFOX+  
jGIZKYGwPKyDhC0AFxDQUXN30uXLkk/LsU1czk95eAp9pQ0Jc7n4rsdjRBbsGyS134sI858HxuG4  
niqqZYGf0319LZ0B5d1MT0WXAVZwyp2QjYBrGQw7MkEno6tfBaOqVUm0m33tOKq1d2T1fwc51DJ+  
dmJjtzY2bkMgK4kB+22cF28/r72yxp4Pd0BLkyrk+eksXdTDq4dyTF8/cMr11R+dXV+PzC91qHje  
JfKLP/ZLVi3Pusv2n7wu75ZG1dwt12pqvKpMo2/IE3iR1bLuZknicqlXHroCWGnXASg0tIwlykl/  
Tdb5tO5TEGVx7OVjEcjiaJakoXdK8yrmlQfZNSZKGae5QB6IrPpGXzYJQAYlfunr8hysdQbtqxYl  
G+oAv5kPJP55L7+fXI+kcnZbQAw097akgsC63RxUL7Ljh5P2/FqZYLMc5xskQv8jhgsQ/RDTa8If  
Wb4mNrNA3DRz8w5fiMBRW/Wdcy06A4WDPqoPiNawmkAlQbQmo2OLgra/cNAEUCthSAs1SS1Zz1yk

7XPLaXi68mxakmxg4eMgRSZc3hWk3viNQbS6e3IrWc/pMdH5RrPduJuq/Fss9f96PT1D0U7j7xna  
1m326sbVn4OKA0Ny36oLLL3FEVJaQfTdnwp2ZgNkJ2fncJSxGpR6VMycprBxystR8awKobtaPQ2i  
emXdgHgcZ3W8HHDh03VLYKNpTi312hLGxaOBQZUC2w6PXwrjsqEcq6N9prCH9Q4DUDP3lKn2dS8K  
kGasFQRVtL2PVi2iQStPbX8fN2Kwmt+VxeHjA5mmWnTOKVBQwB3MbsjVnioWk2MRPUUEGVpKmi0N  
p1TlQ2WksfFjiFOk1PNKFD72WwhZ6NjUN5L2M9YQc0hVQP8ezk/U/98PJ2o0uHuwRHO3wCYezIf3  
ZIAVDvsHUGyP9V0U2f3uizP7wD8Q3n8qc+T2y/fxoKQWK1uv4HLcGTyfgsL5/mb3W/dgvXjwVqUt  
mtWl/OseBvujsGoq4e/mVQ2oBq+dr9YBJ0hzXBHp6+lcSHz+VrzmxEC+pqzO0WkBb9NG8YNW8JuT  
8v3mOapGJ3FDR/Nx1pi6APQHlUkmvDrgJwTWNfZcgrAhWqJCQKTAmiGoX4tAcNCfsvwxQKwFWWAS  
+/K4+rXchJcGYbSWIE+M5daMbhj1ur++MrzfIlzoAhaqEUcVDlU/zhj5RW01epqlZLlf6MOqcvqdJ  
+MJFiGjlAasXq/LAg9DRjJn8ehcBoOBVGkpqZnK4vxYusMjTe/ce+Vn655YHMP0/uWa0mOfMUCar  
mYS7lwVA59Z4Zj91hA8z9P6arKQE7uyinTRyAv7sErWg8JxXlGzvgXrduP2r7/rj5u+bbSL0njIk  
PJGlucd+JVXqDDPcYeUXGGwB7e9tAZXZbaYSpIVWhbImzJPM62tDRxQQjvYlKzXhQtWXij15bCoP  
GYDd6k+IgsraPGMwqpF0LCjHkAew2dlCxoDQWqkUIzL2NWdbqECmmUUzGqlC02GpAq4rHKuPCzS  
YHtg11VaeBQZZYejkHPS7LSQnv2oM4nnlYNcTFhrpQ1wD78bQK/hJXlv+nnUnyNqhKk9+ZwV61zr  
0uAD7V0mOcfDnqS4/Wu3ZE5KHpn94pcfvgcdn73Rbl37442IFDln40FJiwodY9HL/8cruNlcUCFs  
TpItlyICd/56vWHJcDxj6fHbrvT3zMca2BuawK2YH3tBlA5pWEegpo9nqXpNfiqe2dnZybFtOmns  
UghBe2NcFONP/LzSk9JCXWSG242Am13CBoautIElXUpuULLyKQ/Bc4MqhI6msZwGG/mLbm7vQkuZ  
Wo5GahqciPbp6PC3/T7SP6yMlarmAGYni2t5ACqpVFWy9iUKapYN8scAXqKm7mk4yvptFoy7OzJ/  
nAg89VclqsIQL2n0qPhZuQFG9k52ZyiaHRNK1D2YrmqVSvyRJzcUd+X4uOdb1/CkJ9ryO27d+tp6  
bCQezt7ei5WEMh6fiINGcru4cNy/97L8uILH5a9gysS4pg6g0Pdrw3/dXF+W2ywieZ/H4bVKGQ50  
o6g3u7ex6Rc3mi2uy0snDt4uG/2INNwKX+z/fjf+xazMox+GsdfWJbV71rMJ+85vnf7YM0Epu3ih  
kxkBCrYaO9qlwuT/fPJKSjgZfXXeAP3GUxxARptd/NUCJsbp7yxTYxBKdJYx7JUgqVky5vT0NkxF  
9+DUVf1SwJLPlutAdhCW9gMUMTveqyUkoUMnueKVj7SjgKsthPUKhJqI0vtM2XNAief+1g4WF1V9  
8hmDyqlxqD1L734bD2Eym+pHx2BnjbCoYy1WL+2tgH8bAaSfOyryzJpnm+nJ45qQ03kflKUNRvTY  
fnb7b60mj78zy7oeaYW2XQ7Mp2d4W0R6PB1fHZcp6ncujmf/bhZPJPM7g0VdtPCYQl1mVSD5vJo7Q  
etv2477vUWenWRJUrz79/9Pb8qbdhsN3mznL33IqYryGqzYF65n58/E6+V+nGYejJv2m06mZ+qHc  
XIardgKFNh2murjkXI2fReWyRejTMT36Wey79TSQBBnx5A6WqWmJn9HDCqfddkrf1OBnxsWGw2i  
9mb6XAui/h9T8WxTYDAeVtwW918TMA+Brwb0Vlmar+z6Yh3KEfWo/MUsUJVZvgwgBrxfpcWnqCj  
H4uO3R8jsNo1ALhLMRPASjO1/HYR4v3mVqoUfe5spW0g5ctG0yBfbNS4L0dHYsBe6+VTmmeaT8sF  
xDMkkn9S4AzdEJN50zPXhE76KjbYLCBgSr97L/F+WbRDMeaJWUdhlaZrOmZSc41JVVFLM0T+dFG  
pd3fvoHy8vv+OJqalnfhNuH/tnfMhBD9qL5+DcsZ2e/L43i9y6j9e500jZXUaFqQTH8TANWMEqSj  
bVraMCFDeMBREAXlpWWI2Rxxg8tZpX06sXVOEzclQUupFWoE806tSoCiNLXqiQ1WULj9QjRgRFmWX  
ApOiqTADVMFBJCocj/2YWD/1UZfiQEpf4tkg3ry7IivoZR5BT7zZIJFo+udgVRt4nuqQ8r7FLT1  
6x0wYkWc5nNZjLHApTCB9eRGCx4gN9MX5fRaVYwFZtxkt58bapQtKiI4QXwOUGP53OZzhd15Brnw  
Zpin6DH9Wm0O+KFBU3x8DpyFq1K1GDR8axQktWZVjY5Ya2eyEg6SxVtN7xjOt5HcRWWG0+H42j1o  
0WevoTdn+BMxp/s+/x1f/Rvfk7e1FuflWTLDuxocb4Tr2ePx6v19ThJw+XsPgxoKGk8kTrh3etAh  
bMr3LhLUL/F7J4Wy+8OHgcIculh9VnID3+Nf3dwk+qQC1Y6sE3MLLS9jRS4pP9JORS7qoM7paHFB  
OZG4V9LDGFB4SjzKSWMWqOba2SYmhDUAWbjeFmPJfDknRVzSVXhIVMF/9Iotb7YJBgYoY6WWnroc  
lZONVcli0a3p4BvUALxY/3Hk5EWbHBgc+HgbwYWGzawsWOn2VHqz04jx2ppI8LFYKsu3sP3c5YOq  
XMc2VphzACuy5pmCqo1ezrrlWWH1G3KQcUpLdMcXgcVnsh89IpUeaUpMHby+U0TV7EC7zYWhlm9M  
4kWfx041Co9/2Tf57/5zj/84Pcv+Ka/vaXBNyvbT/yDP29kq0UjW0+fSJPoC9I4emy5mA6ADKOiz  
CcsjKtN1XP4ZbAK8K/oDLqWToPhnlTxQikr1fw4gY39oc0G86ClyqDSQXC6OtVCSDK0m4tVidYDs  
3PUMOUhyApW7f8Uje7Sz7lQKbaV5laaKiJAL97Hv+mrKAau6vulpohYfEBw62JQlbUoOJUKlQ6nG  
rVmeIzWlGikWp09tYBSlor/jm2qnAsVEhn8MksylbpmWqJTKKENaEUsOrarAC/u4UFIKSGE64Br  
bCpDQV1940uYE49WUAL/kGRHRZLULA8r48t7B4qA0gYGeZEd8czwRhO8P4IO61wXOsiy45xjjM8R  
pUylE/9ePE/fp/ceP9v/5y4V9/04fD1cmXG60Uny4rrWZocmZYH4+nDVQplAbpGxcLl7FSpMGyEr  
GFVOUpChcdg8dqc9gaKSwyytJB6TKwJpqVl8T1FypSulpkCVWtjqaPke3UEFzexwR5XtsUxCAUwk  
GJfKbDSb4k3eT2NTTTaSxUKBowY0PF1Do7og9Pemm2VC1Cj4MdRPRLAcxms62F+jxHVLRRPma92ugMYwHwYWGPlt2Qg  
6Mr6iIwcFQlucAT0OsJPq/FHkxpMWLN862tc6VdQg78607OVfWNp+f3gnR+/jSeCvHG0HbcPbzOx  
GOqWsqf5uOD3/51Wxr82b79y7/xjQZ8INDxnYMQi54qsnw/jtZ9FtKzbI5Bm5R5xGgBK9jUOTSd4  
QEOJLWIllpf6wRdVQ9ktLTT6dd6Ryz1gz/GtA5vSP7NAehMo1TLSTN15KV2uJAi09c0OFC5tFvRS  
QXONBDlqy+nVJ2WV4NQjqabNGrsbXSdYKedUF4jtxTMuZFqiQdICu1VNVj0QB0lWlIeB4BAkspAE  
v3vaL3S0kONNGPhaVKRsTWU3bxAFh6rtlahcKrdCmDwQF83DONLDN63Mb1IH2dT0+kAYvYAN3t9  
Q5gZSeSdTr1c+Pb0u4f4BquL6bDqwoFc9005Frjya6czTwf9sp64QAXXlKrqnogDrersoxqxtPVfY  
2jeVEUSb2qwJvHKsGyHsgiflybpYl1m8vCymC+YFywU/zfNMjapa/mXcWLgzTKZsAr1762Nqha4fP  
aVApaKC2wy1/Y0N345Tz0wlUNhwbTtGTTcBFKZXDEaNY+lzqnZo1nUO5MZGVft5deCvzuFyU8ule

RlDRcB1vAYpLUSnQddzSVSQjJpK7PSpyrq9jn5jQR/XtjYpHVuS9Uzn4BAkDB5VGWi5TaWJQheEf  
D7CMbnS37uswbAcYM5yAwsX6DUobKliZ/Br11MuU9oIYDFS7Xdq/5gCb2UNWo8qi0z19B7TJoYcn  
12XY5pgNWPtkWW02rabqjDBgVyWSTnUqEgW5y07bFauF7wE+jsGzafso7PxW18dlPtpbD/47V8vX  
/SnvmcL1s/WrchSJwyCmlm0fGe8Hj/qBf00TqJOvDTBxHKj35vCV6XgdQJrRTE5jS7rZYm+GHKN  
JdYaqQ003CrjZveIlpoW2n1zPqeolH14Kc6n6qDhdXSsrWMqwJvUgWppkuMTYTY0hub3TG09DrOQ  
v3amlJzFCSBa7puPXiqdOt92AWYZq2ISKte5oZKrbDrJezs6LlZLiO+ZzI7H0tnANpqDETGpwDkE  
ACe46PX0tk/kssA28ntO1Iu40dmCSj1QJbL87qtD/tm2x2vS1kFtcmjFhUVLqKxh0098TkSCKfYC  
FpquTl4i+5FA6CNlyuJCtZFT3RB8hs90Gz62+4CF7obr6bMWDzdbPUCnPOP4jvjbKGm1AUSq8/Eu  
v6r7/hD8hu/+X/Z+qyfbds//wu/1wiw9Fdl8WQSzd4FKtoyDMvCDWNR/S6L3kHFVJU+KxVkeV5bP  
KZphv2BmkNPb/jZgxY3s6wpnrmZNmdrDL1Uf7RSuIgOU6a4A2lwpb6qs6lAqhRctWSOVVtR+oxmo  
T4cKSzHV7DDRfWcGNDCgkH/19rkZ2ldNW/Lfxu2ljAymOVrbtdUIbf52St16gd+ox8OpD24rjnZ1  
eguaLElq9lIpUaX05Ek0zNpDo5k9/CSzm9lb2uxnmvdMpvTudhQxJwC3+yNpWSpxgIm9wBiQ3K4E  
KTt9KPDzqf2vbHwMBjHyDjnxbb7R+IGHY2Qa5M+pxOkScPz23Ne1/PTV2R0/6XHSXC+13YD+K3y5  
OZh6Nr2ix/mJ/o3gfqD3/6Htpb1sy4MDjPlut7VeDn+dY7d2DccB+5p0mIUlnNoguZAzk60tfaVR  
fekauwnZYtcWTDQCigerARVGzxcMvxvFwnCp7NdW4shaCxZ9seWtNqS2nUBgAabchUwYzS3JJCBC  
i4GLRYIsKqnelCMoNZ1ow8sam0Zoa10oTAo0MRUKGlq1FjB7pQqP2oYqRb06yKCz+MefLfOr0YsO  
ZydaosbQc+8rQ5xNm1ViKDV98NdoVaZ6YA5mJGEvYH02LbHHC4WHSorUoaGpmYaL7QKi4BlgmviI  
tbc1+vEghLLOdBjyFZTYSTOWutYIk1Xo0rGGhOgbjFzqzTBDECDH+MEM4ddS77f18XiDKC993h/9  
yBO49WP44V35ZN34ZSvMUZ88KKuPvjX/mCdR9ta1s+e7Qf++jcYzU6flvQhyzRv2r5PQZbQtOyi0  
RlqITxzmDQa4saulAxWiK7mevMHni0xJU740/yy+d19yZN6jq1lbCwp0x4bn9GUGmwElKo9sAhCE  
yFGBshWbekIcpUBrSlu3WXjPJhfQ5rsmM6rU9I5F9W1tLmdAHPdWgCNLqFGnO3635xhQ4vrcn4rp  
58Hofa9EphqyZJav5izd3T4s1CmtCt2gHMB35XT278g2XIqnZ0D6VNDGO9JVpMNe7A0GMUGehchu  
ghsJxStnKok6F8HO+DQq0wF2SR3tP6XqZ0yW2uqhsyCfm5B/SdWXZ1GkqaxOxufBCsKhGtPbktPM  
mjzTVY7x3Gcl/AIDBF99WF+308eH1+Dx1fjCRPvr3jtf+Db/sAWrJ81dMJ1TVCwrlGV13EjNfElN  
slWbT+ccpJakqy0h5OWipFa3nQEb0AB73wtvW5b2r1DDcYYla2Wig3TLkDH+lmXtJZwJNB4hQnGB  
7W/1lby1Ms/fdaN1bxQAGRwRjIYznoWq4LV9h+AtpYz9ZSGcoAxUz4KSNBKliBr7TEWGwaMLKzy7  
LocUuNU8KmtjSRpww8kbLZU9ZAVVoxmh+0DHYHBAozRKz8h8fw+LOSuTpJj5RGVCju7e9qsQEpoQ  
VFbc7ksO2xpsClZnEjKWbO8NjjOJSj3cjmnlRyL4GexhsEod57VQTMVfLM1wKZFFFw2sxSXxZ1yA  
WNUfjE5r0dUws/Ny6Jw/OCteZ55eHyBToF/9VFufpp4hHh8Gfz5g6LIf74GcG7qQ1l9dgaR35Q0O  
IRpLPMS1I1MG3fTFbHAWjvzDctov6yQdnKQikqKW0S1aWALNA/uHyoliCBZQ2GR3Ln9kfxGk/9t  
0olU9qqNEgPlakdSq+oY6XFrkZd6MCHab4a9aU1pW01ocMaSeYNbNWv571FC8S/aypHh1Ix7rTZB  
wDvOqWV1krI+qoCnw+wJpR+pQwKOVfGQoRVWZu1CLg17d6qmgYA2TMoxI4HtgC5UKZ94wWx9osX  
sIvzzMXrMKEj9nHfnNZp6kCXXO+cAG40GmTAI4rWS1UBcJt7WL/C51CkFA+pjFUoBawzhwineia  
GmjgpSJDtDCQmCAwsdhe3C2nI93GFRbLsbamJBgy7btdHfwHbELp/gkBiJE491MPeOi/Qgu2xjX7  
jnTMEtDKYuUP/BtX6uL5pf8mb+3Besb218lDEYjaQZtZ3LnuT3fb0e2G+RVkTtUfGAZHSt3ArcFK  
lxp4JEWzw99abMIXsO6lt5oVGfo9A9UYZ8dKYyOlqNuUr28dYVRVdf7lsZGtbCmykpz+ZsKpFkac  
LqIGNOKMn+qVYNGrhaPFoGWuVTwFg8iv/Xg4xr4Fp/0PSw4te9tcsAyrHJGmsnihsWszioVWCHkh  
RrQcpvtuqgDN28Uz6WaJSKtjgZ/fP+SjmbE8qWpG2NZSNjZkyYFzs90tbTQLgNtDmBl1t/tCfNcJ  
fZNUzqzk5dVD7i7c0noFCznM20EKBSwjCLneN0xLCarpkKxqFaRJK5ns2ExvAerveMEYbleUVYnF  
WcnfLnM0w4M7GVcyJ9+zddqbB7g6XKDri+uNkAqx1g9btVDqrmOGKwmzDQIaG0t6xt5e+Hf/mPDD  
T3T9Dv+7Oz1XbJQ+HfwZbKQFGkRRbV1o2o8gaEq3EybdZLdLoXxG/2ZHp6QvSv/xWobv3ijZxU42+H  
7JYor+ZdWqoBq7CkvNNye5oNS37AVB1U0jI3KnmR3kDUXJUKSuBaqm3WzkANKg2sZeM3jJQxQWAg  
SaxNrNXWQ5obL5K0HCmZzi0mXXftKz0J+3S0b/nDOWuQVo37sJm/XM6lUazHmrlU0YUQNzywTzU6  
QAdgI0WMMVtPJZ6B5sLahr2ujoHMPRfa18qqJ+Rli7k04fdz/AcHOGdYxCj2pt02bq1IQSUL5nlZ/  
cRqJvq4vr+jvbS8lizzhHs5sDvepNHonGd52sJ34jIHi/OcZ9HZE47nD9MkvvMaV44jTq7SquLKL  
ihzhZXvF2A9T8FOSgtmFSyphEtgWpZd11F2S37kb32TvP+PfucWrG/EjaLaie9Zhuu0cEN2nKBp5  
FXmmEBFFkdmdvDhXKusZjloiu3LVH/T9Pe3hJFCHw0tCpUNaYFojh+Mn2GBOSFEQjVVDhqmNagQWg  
01i1D5pbQLHqUsI6iFRplJdAntT22Gp9CVRAHMTp0Kxp0qftXZCft/SQxOWkuWKZM2WBDyopQmAF  
bpScEAyjjgH49eckv6uznvF1lQEGrjJVUCTd+ue6Kh6hSdHa3/TUGNrfa+io3Hi7sS9i/BcvbV2  
rMmmAuT22jIdHKiM2tIqzP4q+kaAAToudxY2iqIc7EDrZjSciZSXlJfDtkKd2GIFwpUhenh1NeR1  
q8s4j0vbLl0RYubuVW5zf7BiCXEYCFGmmUjsA/OGDnC43E8OGNzgcccrlKGS3UHj+hBCg0AtR313  
7EWEqx2HS/A8z/xPf+dvOvrV20L1jfa5mi+r/hjh14btiqqpWbRUu3cppVtF6YQditbxahMHagE  
WDqENHy0EX0OPk7aMq9W8/JcrVWwHmBs9H9tTSXqpTrItKrcp+laiUxwHNxIxobCkYfUdMwVS3kz  
QwKg0nqbpW5UmF11ldZB11Djn516qm15VTas2y0nsmell+rqv6TLIpwq/yQn1CHhuj0VZu1YUU+  
PotgET7Xynnwjwso7TwLamKT+XDZDXDSVDjOJd4/JIERQnchIZKsXRbA4Cxqek15qAFVrrMchUO1  
0TqumICBftmWWOynmiuxWQ/LFaTOF3pkCymw7oNU9M8Eto6DpKBPVjiZHHcQqjFxnps+wYkoMp9  
86sQg6TaP3luNZt0H+KgN+nb4qfMa7ArbpoRFLNMs6mm4qa8Ej5zU3zNqyig7NMreW9Q3pr8IEu



ABpmkaHsE4D0EEXq76xWk4MtSy01BTHzrKN5bM0qT/cvyInt35ehpcekXg5gzVhZwjcs3SGG+1Q0  
y0X81KZgqFWU10kUFtWVesn4KSuXjLMSj9LK51Ie/k7eKnjt+vUjwLa19dpIwDBT60mrWKQ03NqS  
u3UZYQsWjfnV2vwSjXNWpmi9AAYBs5otlMXBhc+uAe/kvTYZDyqqJUrOK+VKRJaY8qUBvBbcTxJS  
skaV6ltQTOL81dKr3pUmbQHewDaWiL8nQtZtcpUFI4VUgRnvDyRdTSBjzysB2GBeazgf6ZRrIuGA  
ct8dEhh8Lb21urMOZ5rVbYc2zGcVv8+fNjDoiwd26ya+Iw78D0fij3fBBsa4dw58+N5POa4LiMcY  
VJ3G9I/NbR5X6+xZW2KTOzNYmnK5kLKT/39Pyfv+Nq/uAXrG2kr1lOnSfB7yXp5JY/XLcPk92+o9  
dLIJqd3m5aW0DEAU4zvyWDnkuQ6TsKXMSnk/Pxc1l1IMhxOduHkaUdTGLUF9TZlNemmqnUG7ba1  
B/WzMuswbxby/yqUqTYX0vIr8XQCVT1kJ++HnsVqnn4ZSbm83agLfczL2pwwfgdKJBqKKWUWX6A  
pa3curfKzuHDytKjRmX0hwrP99sKaVmAYQ2gmPRyMoIlpGR24W0v2AjPP1X51DVesOvZwqJWkyMA  
uuYj3gGF7olYaOL51oqas4AkuOz5nkt0zRTMTZmpY/v35Z+vyeNsA9XpTVJi6RX1YkDcFVYGHCYj  
aWVRy3Hb05B8xdpkr1YZukHLdu6h1Pg4B5H+YwhMw2mG5RarpUgzU0w6cKaaupM02ebRfCi1GlrW  
d84Wza+axpeowHf7cZydOsagNXI4admpIm0T9rhYj1QbchlJGKpkp0UCguaOxqkiaJIilgGVjTlZ  
lNTNxrkpBVrvZJ6tdYl1gIXqvB/K6kGhw4X1u2g2tzbBibXmZp2uqSv763QP7ydzE5RSoBKcrG4i5  
d5ULzHwVVOXZ+j1Yp1+SKjviLWpk2v1KFY3JOVsm8Vn+/bsI01LG6qwbRCLXyuc1cdVmgV6aZez  
1RanTM9s5ro53KSAKpn5A6kv9QIztCLDXhRkTEIAGojr8/dqBeZ9eyuqv+bM0KBPrvVuYrFb6T9u  
stlhMu9kP7B1Z5Xpj8nRXaI98HJdOZk04Zj5oblLnAuge8704b47y+ydIzzfYlTzEzjJOa7dkzjVn  
2PUT6bCRj0fdhMB1tVSL5C5wedFK10pP/13/4y88w/+1W1RxBtiqwpW+/Wj05cfhq91UOSzb2hDd  
KVUL2kEai4DIIPWUCOVqrQAC6Fgx2sm4zPVG+IcmChNNGmvyvdS44sAJZ18YNj4d0aCK3kQYKIVV  
7+J7XFibJrKlFwJNdJr1K/KvBh1OkdDTvRlqf5AjSPberXnVbt3ygeLgaaHhsjEeBrQ4Ykz6qoVQ  
xx+xc4ivJ9ibByKTKFyGz4q04bUx2P6FwAk0TXLSosv2OVjOaE2t7PRvdjkiFkEwSNMQYMTLGzME  
TNyzQ4kS8sSNz25Vt20v5yNpEgLFrvPV2fi2aWWE87mIxmDqcxGt9lFgdP1fgFIZ+/qK7CM5ziup  
WFZ5/j38/F6+VyZx+cAc4XHNZz6Y+1u9ztw6l8E//QZ+KoBfpqmeUF9rRqoG6v6AKhlDVRleL/78T  
9/zzVuwvhG2dDV3osndA7s1vA6z2auKjAV+tTA2CwIoew9fi/k+jm2kxWB3CJvBg2Zb86SMAFPdk  
CkRzjItZVPkQMTf+l1tanQZrKFqA/tKgeBchwnXF1GpK2EK0KglSzbpGOPYyXGy8W0ZpdUeV2tTL  
CEXtb/GpoDiouDfUYpcWxNbBbPVj97s01Cf1tIIMoPHnCzgUoFQyWND+LCBWM2h2CHLDNvwPUNN5  
XABYvJKN1xpcYAWq7qYgokjFeom06BvSr0oIBLAndfXAm9UVQqyE6Nu9aNB4Xd62rWzWtxXmh/gG  
NqNpuwMD3BFHCLTjge5L9PR8RNFF1/GtWBKBuTdAPL9OyaVudPoDnb+Ij7l3HbsHsC6xt89LCDHQ  
avzVLPX/auwor+DBRAXIP3PqO8nAeobXfD/TUODy2hmxPdf9g1xLsPSXU7W5w03aJuctulghTejR  
ANCap1A0aL5QmJ229D3JKCzWOIkAS0sZJ1lNfjzrBYpq8wHVpD+IASqsoxVNwkt/NoCp/NZqelrs  
NbYEduzazExvyFhYoicGQaQtJqnqjWK1T9VS7kZTgywY8XStrz6pOpmAAKQ0WRzo0pBQJubSqJiI  
/UiF/XJtJBcFjRZGLS26lraeGrshuA9YupZggfk7XIGcd3lEwJZeJZAahuKQkspt891Kgv38tIM  
BvrKTC+xPXM4rn68Brtmy7gix49hPMNJV+cSx7NsfDtAnd1/XWju4u/rcQFvXasfbXSnI1DohHNJ  
lcavR249UZymu4txsKYyi3S+N/iNP8TTmeG6/pf4Zz/AwfYZdHyGdvzn4FRTQHTGzGRdyDN1ZJ+m  
kdL1fvJv/unQIe/fQvWX7PAUpYaWR65levtpNGq5wVdl+CDDyTp/FxvcMqlUD+IFoFdN7ef/Tnp9  
br1TR/neoPfvX9PC9zDZkPScaoBFQKKlk1FyipTq4cWi7nMpyuZw9dlaQIjqnrBqd/kuhtliELlY  
fb2j6Tfa9czavQbsdTX0xxhpQ5vTYMYxLJeHQReB6RMtayvzeOSHRDs1DZSP7wCtVdlwnLDImJlB  
ckqFi5WLJpgVRNHeVDLUk5HdnENQIulbsOr5V1YwuyrCBwLODy26LF2GrR/NS90X73hDbgGMwDtv  
gSdIxnfyu1np3dxHYc630blW+KpNtBn6zks6G3ZufxWpeDtsNYuVqHwepFKce06YCVtq6ralu9/K  
M/S1/HUR3AoZ0Y9Q/pFLFgTfAcT/Px/qyK/BwB/FQA6Mhg6/wyBKhdvq5/6e39a3vF1f20L118Tv  
m8rTbTz6f2G44d+kWxmGhRXUxFaLNHQsj/HaGkqY7qGhchSrUrKtGbWlFW0wO+gtUYqccQxh32NB  
mugid3ioMkrajZNZ6rgzwJ36glHcQbgOHUkeNNR42yS/znuoLPxufrA7VZT2m08eqDdZV0UUVmbt  
rhNa11NKWvazJgK/Vc9BQaCrHoRqGhtseDw5qTVWyyNmj4/0YL6LM42Cvy1vhMDQZYbethHMTqCqD  
GitvYJjn0mlyLS5ndVbKsBN0s+Rlgaj1lndZwsLuYTPORvf0hY4v7Wr7XI+3AfO/5mMzlRfoj88l  
NXyVITkdKQGh3ndf1na3T3seq202wvaGlflLorEbeHuUhQSfzppFbj7uggYV8eqLcrl/BMc8xaOlK  
XXTbJicf22Yr8DaPw+gzkg7NkBVL+BTAfXpr/80tajGg1XwjceJ3zRgZaBFTzjsYeUaeSSlNDEj  
euwXQzWhpTPCwgiU86Pj6XTHUiAGzdaTjRtMgaoKqvOwFL1vAmHgz211ARqDEsRR5EE7R6otaen6  
ObmOydaGIZibBkNBwF41DdZUykwKxQY63VUS4nSysPSMv1TixJaWv1j6Y0GENp1xFmj2alxQ2VW  
asYMKBs1hT74XR1zmo9076jn88SSgJdhcYtFibEEjY6YrVaSlTZCxyo8qBX54CpNJEmao0zlhbcj  
2eYqcB54DDE6ww1IMX0B9+rM1/LfVmOT8Vg4QRF0JtFPbAl17acEjj3pYdr1jLM2UzFoMJC6cocI  
G92j3SqHlwHTDNPvSySqhvnp845ZLX5e5odus/snsWSA2v6z2u/1BzVQLU8/B5ZtncJdC+HLF  
SlwbVEvgArQ8iIWGvW9CCYZF/fIRfR9A9Q3mnV986RuGHENArDWaeS3d+LVclGyn9JrDnEzTvH9r  
eGjuSqqzWFRtJSNkGr4ddohA0WMVpHe1FNQ24sSwPV8ihv/+gOtJA45JqVjqoM9mnXbWy10RhBUm  
6iwSrPgdu0Y06DYsc47LUFNCZIM1nAVxeJpI3ctlKZ9POwPh/WkJbU3PnJ9Z+U1Fa7qaiWNQQL9  
EH54HN+GOB8OrCajoRhUwM71FgR09ykGTdFFyzsS1M9N80TA4yUDbUBnrDKNUKe4dhYJ2xSsrR0t  
TTSTNiQ34eVdDvQz05v47UL+Kzt1QsuVguZTk8kqWJd3MJGQ7r9Piy7I/FqhuvI6fCnG9+3rRVTB  
VwHvzG4C0qfGMwQW845jOW8SJmwnvbwK7Dg3QDbmIH6rgDUMY79Pq7Ju6QWP/8o9ZoqlDFWofYXQ  
H1ti9wFQH8KVvUi5bYNMP3aY1VDq1WRjcxGOE3Wk9SkEK5KiSRq6fz2roKGIt4BLGynPdSxjExrT

E/va081hbpJ+2gt2Cqnc2c26Q4H1sTSKWYftqKZm44aWiZOoWMuMmGENKutKntWax0jQxcBgZ4pc5m4p+Kkko3NFMdmKWKk2r+MZ+nN6JgbvaZMo8qkvkZhaqCqHqAFCLn1kKm9nR1p9XfFDdtKo3O2yVFZMJrreTBlxdwxj/Hc+nr4qT26rqvjMQzX1FpfrQf2qI7YrvWEGVyyChWSq1ZTZSo+FoJqz4Yfe6w2ioUWdSugr504nlkrWLR0DOa811oGG+FwqhWodBIv1cKbQciWv6umFGe4NhOsJg2jMuEqB7ej1fLzglb7JaA9xtfyfzhzYXUPD6mZV1NV9nOhBXRtSucDobQD0pXd+3V+JXntPfBz1/YT3zRvJur5pwEopTbFVT6LR7XtZvDg2bW/tuEaYJCSzBVgt3GwJ5UXEUm0jjkW0qaK/6QwpK53bpgEdU+q0jMWCi aCxmQguGytVaDqDqzetMQM5HI14PpvJ87fu4Yr7OieHNJE6u7v9rjQ9WzWSONApMFks4GihfJaCm rIIn7pFVPtXtfBcXAZ5VEvNAuM1tSCiLqSoazLKqi6waLZDCVqPKcXmyEgOSo7iqUa1x+OpTFcZH vAn4X+ft1b4fQGfeajizd1GT9qhI/vDlhwOutLrgSJvNJeUxhulLnDqLzIgxY9r4mFaSbl4lwVE z2vpz1ryt+sqW0GV2Q1WsiqAYs8t5SOU2fYtjK4EDNJvXNcS7wW7sT43qn0hpcFFwMLQZuiTD3Dw upjWsvKstZes/sxnPKTeV7832AyvSLNQHFmPNJibGPFUMDGoo7wXUY0/vETbK+N+hKYb2TAvnkCT I1h5fWPuuvWs/eyZPmyaTmzslx3QRNddnwwwGQAmCvOLQM5QdjS+lQq4DMwo4OIKd4NKvjavCrVI Uhx9dMgbcUROGaQXLfKycj+dkXbsv985kss7r3lGqDaTLSTuTcn61z6oNt7vb7VpOJX02h1pe Jb00Gk0HfV1COimMtBkMuq81uJ6y6S2k13rL5Eos+MnzdQia5EOzsmExcT5qnWPAMrZfCnnZyOZJ iWIZUtuT8/1ldFU7ty/q1bWdkbS6gzk/nImPhaC507Gcrk/kpv7Pb15+Uha7bAu0ZNqowlV9+Wad bpXVfvDoCdxSqkbWE23peqhVLHI1k5m76obQSoewQ/mqI8sGkmjc6gtdfStvRyWPTjQNr205Mzb5 KrfaCam46/xWbZRFZCME6uWac+9Rvd9+KJuFUYRmo53VOT5K9T1xvFRZ7jcBJNMpcCf5dubanwGy wKj6X0jXZwP0tXiJozT7mJyauMGMDivSYuRamQcdcHxF6SnOWjpleOblmtZrmOt/KG1JKAJ5H47k MduXAP162g+ldFd1gGw1/P4fCQfe+mu3F1VOiWcSon0h0uAPoW1FfigZ6DXZ5NzSWD5+8MjVfnn0 GLWHdN/TtOs7rKhZiUkdNe/q9CZjt2wVYSM6GQ6RT3hqu5zpTW18gL/TT94Dd94PAKIDE+S0pIf+ +iH5WY51NfKBJB54nd2dTrd+ehExmenssTxrtJCZnEuy3ilY0Dqzh/RCiadMMBJeWyWp1Yyelpl0 jQO045qTWAd02HAVw71fAIKHAxrFQuAyPFdHGciYetQZmfPw91oa3eM424KMLhJpQO9wCb8Nj6pc Ecb9+HKdOGf+thHHy7HQWkUTwDI74Jb87DtNfbBSu4Cm9OK079qH1ULgO/99L+UYzwO3/GbPuH9c fB5XyLHP/PBT/gc/87nt2B9nbcP/5O/LGcf+Xe/6DF46wekjFY16JYTL0aXsjThCt3ygtDkgGR2j KTrqQJAI7xm4UwnAB7809OX+XymAR5aKlbjcC5Mv92Qt1y9WqvJV3W9YQS/dDjdyHN3T+R+ZMgxL NdaA0auvPld75VLV6+pn8wpaqMptY/6YIiW3L77CqypL11YWRyUZFoJZeClpYRuOGLhRmXURV7UI 9bmAaPOv2rS0d6kJURre9M83USINymc+Rx/t2SN/X3kliuyd+NtMgUlH+4N5Nbd4BpJe9819Ogw g25ev0h6fYHcVv1F+FnLgVLmMxXaywut8XK19IMax1t6Zj1PBubfrvOvMk0QJckKx06RfKm/rsTy v37r2AxGKivW3D+Da5hGHZlvTyHi3IAKrzQjidaWFJxsxgZsirlxCKe6Y/GK6/ide5YuVJaZLEaNJ F176XLuJMtpuJqOutF8dN1vDL4U7mu7ypNFpRUhFz+kvPBN7wOwnwx4b2TAvgnKDY3NjWzIx77vr zE8kThh+5bX7v4sNX46vWHVDAktJmKaRUXJcBNSwoSDqFgiuFonWmL4aj1UoSkdvr7hufArnbqMk K0eRSrL+UI+9vwL8FGP5bkXXpHfDCYt3IS/82t+n7z9bW+XPJqBZjvvyNveJbdHEzG9lvzG3/olo IigjM2hfPiFlznZDuC3YAFFVlEuk9VS0sLQelz6xDRbZX4hVlppPW7tthiWAlQPZlLFxAbutgfgm 1qn040zuTcay1ve8QF5y+NP4TxMufHo22W9ShTsT3/+b5RL167K715fVuzr/2v58q/43dLpBLLCQ kVX4NLBVdm/ff2aoMq0pjoOcjOPh3NnqUXFwg+KuvFactExcNxyUYAycpvrrirHRYWIwSJ7Dec7EbX ZU9LUDxz8YnmnM26ZdXdfmkG7Q0yGds2mlgwmFM8yBo7WRh2Bc37GgqSjNZHHW9udcyDPPLAFQGM E61Ltl+UPDwjddgddKbG6xq5UpNR3BOqsD/EaobgB9anjtp7j30UViEE8v1clbqtlpNzVFSTYDC1 AxHLOFzThdU4hetBaYwWFGkWjgA71X1Pjng2Nxx0vrBAgeV3vHn3L1/R6qcWLF3MFTa2mp19IZrt 3q6r3t3npXdjg/fmJeTu7fw945cfehRGR7ekOeff6meugYOy/LAM1YbFZVKouasRGJdcpVtVkv3M 1WirC2b1BTacpsK1oIR4Ggps2Umz334ozLcvy7Dgz1YtUCGg3258+LH5KmnHpYbV/rysZ/7UTk/u S+H1x+S45ef1+d+6ofkUq8j73rrTfn8tz8qj968KoPesAYc6KoWRKhYuVFBV0bFnVovmXNgLSoqc pIenm+EpMlp/fyFjhWLNNDIG8jy9duzhHZ3dkzGYSVYyql2oH8vJ7FyQqiI5xPdQG14whlt1NQMsLF rzW/QStNitgIcWClgZV/iJcJL3XALXTNJ8kkHSxfapg0i/13i1YP9ON+rPwAQLOYW0rwapld6Xc+ o//VxX0LieguDPXb65hbfIClsjzA1X3Y8mdYbhqRXJYI/qpSwpVa9N3rUnL3ylgxvhwu+GrdakFu DZi2aDRTfx7Hzf51cOBDJq+5hI/+P3/FFTvwuwcXgPNs9UK/bpnnpHL1/bl5Y9+SK5ee6t+Rq+3o z4mBxxfjMpgSmcJS0TB8fxCIA1+7X8W5OTAKq1V/jUHixvY4AjGeCmG38bTplw5eFhTSqvZVINg j7ztvbowcSE5PDyU8emxUtdbH/k5+fff94/kJqzsFz7zbn6sevy5KOPyP7eobTaA/j0jlpW1YzaS KVe5H3pKfSugUdgC63RlKujxTK91q91T5ghmWQ7HET87pjKYIly+u/cMZh331ZDEBy/kZfOfYz3Oz mS1XNFsNfSTMINGEQH8PubeWmYwbRSHZnCd8VqrenpHfjct/uFEfwRg5IvBp0HMT4Ti/pGKzX8H I8GG6+Wpnzcnj9+VpL12Q6++mG2HntO2NPxh5xJyim9VHRICAcvNKiU1IGSS1I93hL0WL/5qm4ba /rBg3GNOQceEyNmBcd1lOreuHJJRqtMbu/uyC8896z8+x/4F9Ld3RW/2ZT1ZCyZ8bGY8KauHF2R+ PxY7tBvbYVYBALN7bbhyrrSeudJ3WZIMHKiLOW/m3GGmrd+wa4Gk5hA6gTaK0v9ZamsNRj+Js1r PDi/ESmd16UE3w+LVmERerkpI4GczRiv9sWZ3hJ3vOe90jbt+Xo6EA6WJQ4tJ21zIxak2YbnOPIs JdtbWb4bLSOTVh3Ti1oYNERR5a0ikWdj1Y5VSobWpWEzb42n2cp/HaDBRKuSqAyF8t0EJZFyeAnU yo1Wq2kDR8abgsAnitNpiAA9ePEHyRVOFFWs1MsQLUvnFeuLkfnveH+we/P1zxR42M4wOmnC8JfS wv65kzdXAIUbXzWi79Fs/vNdH76VG16Dy9Obzut/beAap3oWArLriuIE1VYybXmsCfRrptUrarpm JKu6tw6VR5c36mjm/qQUsMm7AoLH6wzFCMKJHdFqjZXgOW9GHQu6n2clpVLIPQ17f//+y9B7Rk5



1Umuk+uUznd3LlBbLbWktmVbtixsbJID9ozNYAbDM7PG5sEzPDBhHm/AngcGFmEeMCzSEJ6HgWEYY  
AwM4EV2toJlyZKVU6tz9823cjnTnr72/9fdVvGHmyjUbDukWrdvvdWnTp17r//nb79fYsnqVopU  
b0xJ6Exeo+Y8EkOLkofViJ6iE4BcG+qdpEMgstmlMrYmyldiVgVmBCmA3QgSlck/877K7SPQ+LKg  
avEIBE9KDDEROHux902RdceE2Nz8yVBF/kcslarVRGJdvl9XczPGgrbDPpScA/LNBCq4pJukAj9I  
G9E2Op4AFPweXiFtRGKs3fPg1ycP0uA+wg5zCQSNv9hb0tek3LemuMUAZ67zblRfMUmhgAQcav6  
e5sC/yx1lzi63SEsWLU2+QNKPBQFEShbtBrU6e9KUEjZnXJKby5mjd+E3EHG6DJxpp+oZ712WawX  
5bGatj2kxysWvHTVBZjb0m12DjU3L7weAFV0vFwh50SVKUcgbv1+y1hgTBd1eR3eJFBhxQzngbv2  
JT2Zbd3+H3y7L2kuAMWQoypsZmk8MIIX4GG9x3RS12eq9Bio8TGURKgoOpAYKAAH3GhUBYoIpgBM  
QAQg12BF/sIkDvgcBOFdFKFLJWPo7iEjoQo3mGIVhOwqUmcRDYOhNTCJMjXCoMrWjF/jTkPBLIpI  
pvf0yh71M6Xyfy592aPZ/k5AWA4gu21tGI7zeCUsklFlqZysWUIQEL1NBTCNCypNOyLEXqgfDESv  
kcFGkwkZpEK8/b2JuUKJQq629RjI6409onBsnXRqLsj+OB6hQ2215GpHPwen2XYWeXoYiRwzkYuL  
/fEL1UBcDmThqPr02hSQXGrXKLtFzSUKaB+d8duNg5+VTzuPcQXMHpOtx6/7IvB6RU2m8JzulYax  
XNB2GqMhx0b7HwYKAcA38mZIp6MiBIV4LEA7DXtB5sh5iy7rU2Vowq9ykiKTHJ+gCVS2SSgDxWKU  
1E0dyVsFZ1UNKy7VFDMEokv3keaoJkleRzYFYSVgX8fSq6dyMQPvH3Oq0g7iaYqddn0z5fNktSqi  
KKkMuB9wY6YTgfuOT+EopyJiV5JgS1NMp6pQpRUc/MK30BxaA+qGduSyq7KQ7XSHXt7TB2BhiXmU  
FciCgn/HYEEsr8X87AwctCPEm9Gvk99kIdneWGBKHwRAwGV+f20s3FWVPgwUDAE9aTIN0ELpzovq  
vI9ft6ouyVgilrzsAiBjXvb1Oe/AfLbK0guFZorHcv1zpfqc9C2ubrb2uKNELO3rka/ObzmlId2v  
lCv+oWEyM9ETvv16Vv/z58ERaAoGNrsFQ+yHdw06PYKHodcAeeriocHzBAjIfqesNeSCrAojaeCy  
0WKCI8KBXAJVagdaSCirxJYkhJZ/FNVN4RiMV+MbajwGsPjQpdiguZUFaKE/zdW4mYyMwoqBIDvI  
ZYcjQ9BHRVPsc5o2PthvN4LakBckD+xPNlirgbhoxwOhECmkQ8NUbcTNHQYQPNFzSpmxqNQ1EMx  
gXjFGZG+WCuHulTX0UAGjmxoaUjcQmpKeeTvogewAcwRKqzqBLzPQKJucteEPIeE9GHXHXOvkiXz  
z0qc7LALepDJklc3gxVRR5q7jBQEK/1OYc00bZiD4zNckbocWgsdlv4pQjV+xOckgSWWzrr8PXni  
0XpTY8RKQiiZnhvXJGHPhuLR3t9Vu1Tp/BABSbwI6fcmQulh4IOdzFqIRhc5T9IyiBW+CtjQRii  
LYELGIAYADUD4vU1syDYEjDPCcWpW0o9jyw4CNMTWUqRk3rwIv5rsdGyiEozmFqMjNcDaZbQE+qx  
ZZtWfSJGB54nCxe3kVemD76wI7C0oqXnPV11YcVjyduKpoHV8++ijgWe0KbPbcnAH9LyLgBZIBnR  
OvFY6+K/itaH5gvRZ4HCKDMtpqeyosxzoFcUo3o98TnwP8UsIJJdcDTPNAaF/AiAiNWNr/wPGEa  
nHEHhTgE6Ca5vcdp3Q0UCJbQnDOoT88Nf8tMOUDpTuT70HeKyrRLTcnYBVcm3jLKT7byctXCMMmg  
/aJaLzJYXMzm1852EW+DaPmDRgk4MVpgAWj/ULy0WebUZtfrgY6zU9FFzSaCOM72OAn4TidjEdgw  
7vFL8+FyL2K9UUqVGpisOEK4hBYcSshQrVB/5mlws4PAwIYQpgM2csAojdt30tYUCJQiaOtKRGaI  
UYLgWNLS2Wg6YOfkaVG4GAEEB8h4kZ7CERj/F5Fzt1A+wK2f1SghYhMyA8UZUd16vPFxq4+E7ifD  
E2mBi+OCEA8v5GqsJw9NHJpjP6pIRVTWAXCI2OaStl05F/ZAPIeyb8hwYGSVYopm0RTp3oCNZNDI  
1EhPB4IPywbCnc6vMHG4HvCyh9MoCTv06izrarbiSJec4tF2QSRX4P+BQYLInCE9kgdMCMYBnrNAF  
F2FH0b1XpgeDfTRMTWVFVPLiQul/UhXLsVRUHFLFUWE2cP2ep3/QieResMA8TC+iJX0uR57xvoUH  
ZBsDERgaiTsBPCMsUgMJmJc89e8cuz5pQ/kStVvW3QF4q1EihKp+mKZIRSTCjAa8o5MbIpuSw6D5  
QDuQxoRxxgJqThNeV4eThikGidwTYaKtKeIzIHgc9B5h0ML8oMm+NRNimgXilfhNxZCR18GboMjj2  
sAQ5xRiiFTICeU4sprWqWQsakq3qamfZlGE2qeNtKe0hFqToDtbfFUuib7s5RedjwVDBD5L24Kb  
wJS+YWXtTMZOyqqzvBsM0ZGS8uAiOqeQ6VAHKVLa+q2juY6nkQjMtw7aydpWrzEA26APwHap+Uc  
6UKVAKdW3hjCfCZolqAqhly+HDUESK7pN8TmhlDnpfQaOeMDV7n7uaF/d2dLeqsnpylnkWB7RRrr  
+CTeNM18qIv1Gsaxud+7BnrUwVeyv7BgpUZUF2O2Xj449nBV791s1Rt3GG4pUm/3aZ0a4d63TYbg  
S2D2GAqGA06KtQT5vlEVT7Z8LEg0ee0Dcy1RkJflulqqRKWUgaL3BRE0JE8zpxw9pvsDYRmeHqdq  
KwCvYPwGV7RUZM88MaCj7VV+Hfln0w+I2apNbuH/Fs2nFTmW5UBKa8+LSQBeK/C81jaTUKRSrGA8  
9U5VEXbA0kCaHhBDJemeljekPcSmhShN7U1+bgr4TJYErOoK2gqU79GKQ8ojihFysYm7teo017jD  
5+n1s4Fqi0clTlhDB04fa71dM3SiLFAvyiGC+9aLM9J3xi9WGCKIXDxS0spkQcYGzHwn0Gao78ju  
GLX8Sjs72CQ8BCH13P3/d4Pf1GW9qK3/5wmCn/yY89Ynypj/RyPf/Ac24tsI/mE63ldFDGHO7HWf  
4F+q6sgcbxj5/ySTK6QpbxhjH5hviJFGSzlPtBSyIXRnbVCDxVBDdNleed4mwUnK/bKIWseRda  
gMQYEMUi/E4Iipq6T6uYuxXxj8N7RPx/Bj+T0PfNNWfTUTlmMgtM/PJScEVGxi8PPihgMBS1zXZV  
QSQ7C9VfV4bkYKmnCvvhS+EnBZkbCbNKFBRsU5FIsQXOKG8P9pKpAbkRbEOUQvasRjN42CitXOZ8  
+cqe9hTlK80ZhVtr/JbtVkB9gkfdOMfyMLNDRQPEp1Z4J8tSsc9EdOy8yXKcRojg/ehIqerLB4U1  
fbg/FKQDFvVSXE74rn+F21pImL1WY89Y306DXRuy4w0fsh1nMex0MJA0WqimlgukmVwW8JEW0R+H  
fGuprAfKhkMnwJp82SCGJoRjgEPq0CoZE49oKWY4E3NGqhgeakiNUOYKfqsJMz8KgpQeSM8g4R4U  
9LCRALbZUmiq9eJglHKQo+nu5B48ytBicJ8qBFGwkU87cnairUQXhdGBuMVzwQeJ9OYFdISznWNa  
VFLRvIUOEL+bds691WE3+JphYw8FYOGt1fQSEPaRb32Onu7Al+KR6uXTrODbdCAfwYjRa6uRg8dC  
Y0xdijskXw9frHO7xhRqb4kXtT3KyrVBRSpqSQ7+Zk88M9LlUqlPFGGo934v1DJz7sVZp/VF46c  
ikOR1/0XOvJt/20kuG84rFnre/j8djf/AZHn8PznkOfLFbqbKSOMhZxLYH6FX2L5molqhZyVCmVK  
ZdzBeSfJqoPC2QQpkN6Q632DfZEw9EppbObL6JqKr3URBaieijvg16oGKSR6YhAeSQBV0BQsjiLU  
zFy5HvKUFW7Bbmtai9DwmhpxD2w3i2gaQzQEgyA0rMWjAShifyemH710LXFcyLNHhr14kspso8  
IeEy5Lou2LECI+B/pLPnCopEtXzVZ4V0pJhZggyLIpVoQ+TRsMgorW1Vta2vND1KGX3SCQ+sLEkb

Gy5UpMUjJGE2A4zxvlyTb5iGseC4DXoZKSvm0uFtiZoU3312ijsbce15aMD7AK2a+9MgkHv+n/9c  
1/SElPs/tt97Bnr01owNrIkjtscCn9wfq55V2N+mcqVKpVKJSOUPZpvVGh5sUILJSttzc+xt/Vlv  
ZoO2A0VYBw52NrWlnhkLMipJzKmHL7wKsrc1M3OVD5pai4jAVZo4aTpgkxnglaZCjE1GCFJwd000  
sihTDWT9WsvG4Sa+ElBM5ooQzOzJy8y8bCkNF/k/U1RUJ01ptV7eqoVY9g61NZs9lMRrGlIrb2qN  
GEz9oi211lEU04/1+0yjqAzNnAEIJOZaoX4AT4zwFj3mS5fPUTgOBJmE9oyEzVL5LVawbvHPCzLFM  
y2SoRVUqC9Lvs0hD0X9bfn8pm0HFmCQpSow3I5fKsbF5tL9o9aFu8Lh4IxfqCRf6nK59q0/vte6e  
YYtlu0t/YxN4X8olWvbeTbIUj5HtXKR1lf20/JclRarOVpcqFOdwyrbtQQg0e93BdwP5od2L5BKs  
3gzHYoKSEdU3ZSZyiIjZcjCeJhqDVCyJEheqLww0z1WoqloFRuVo4wChOemrzOVEQLOF6B+QP3iS  
BWHklQD+xORs5AHUFZCqgb0Uq1pb1Z7UgMYWoTEbn6X2V9LOqr+jwJjiKfWue+sZSW6s5YOy7U8i  
MTsKqdFu0vu1XgibSTbLkqLaoRh4S+ehmKmt+wL60bS/eZefZLyMzpgleoU9DfEFAFBjDRRffK8  
3x5vJmh2ARK1wwF43U21gUadS7F9ZUTn3Zs9w/zc0f+Np1EW4f/2Q/+k6hdhBxRPx76w/fuGevTX  
Ica8wLbZk/51waFv5jLV0Psyj5aJiaIyUyhWSnyQ81kqkmWIS80AxSak5i2Oi02XtlyEdyEYVw+CK  
6kKES+nw01BG7mqv4ogMut15TrVvUulGSqhMryTYP/0cLneCCDHI7y/tuY/ShWvk/JgkgnrDUPzC  
gO7zDkcr3rFmZQqtbcpxWxhmGrOQV8AWAB1I3uypPBXXS3pTGfeVSh2erzV75LqkVaMkKEWKMLM5L  
zYvKYjx9bV5Y2uNxtTijS5MbWqPehQmSsEAPdXxGFjelAJ+jINACDZqKLMjj8Xz+GuxeZAI5HXIY  
yXtQN84L6AOaOay3wbXqtwX1/ED/v2n0iS+jy/g4nXf9lPxP3XBXPetPyF94eljz1j/1/RyZqNkw  
qBwxUNFXNYgCUD/mqBgO46phrjBcACepEQEPwLpd6YyHG2IAQ16W9L7hN7NRmtD1XjESKwZHajS6  
1U9UVN4kYBKimdVWRnYfSv0R0i8p3qtMALxLLww+SJE2HhaHQZLKWZCDeE8UsatZlj5YVvkNFULY  
vG4RhIqXiS0UwAPnOXTlgqxAW5A2wWFLFO3aBABWEq3ViliqdaOwQYk3lZyc1NFCoa9qxErsh2mP  
EdVgdVGAmz02c0WPb7Roc3BkLZftgMfLZKCFcITzMeGw57QwA6hgM7vgeEC5PTYgGSD4Xw339wvG  
rpZOFZcUIa659DIQaiNiH361abrx2kyMeMk2owBIn6KjpPf9lMiBSpyoHvG+hTa6NQosXCmxvq52  
zo503b741H/d1c3t8PU8lORrEB1kw0BJGfDIJDCKQAPgJ8VbiClAndxY00NuOvwlUzVqM9SzQII2  
k6ZXLHfIEw9qC4ACaEjUcwJUmAhhRZCCBiNozK1An5fMWT0NyG9IZ5VDX5LZRBfMUefB+FxZkjuS  
thsMHqGKjKfI+MQEuG0KjDxNcEibSWepfq/tsr9DM3kIOhFRd1Cup8rFWgtfiUFKVMx90/DfvRhk  
TsrfR61GYb8/ZnVNeqHIRU5chkOQhrzJYDdcDjqCvABEYsCmhDlex2RIULJK+TZrqjwye6XBJQr1  
cnycpRqhFUEKQ+nRJMB57eFxmACQAs/fzIelnmDdaEm8FQeCfLvaM9Yn+oi0ufts75sBLHYe102  
p3ffeTxRx85d+HSCMS7PhutsB0CKQS8MHYXKLIWtIYx0gb/Npmq0MTOKPiYJYvCv7XNsXbiQFo7  
VTtyXUhSg0DiNr4FX1OzJtiwQN9BXfiYX3BZSyjawqPbEnImpM8k3Ke0KhK7jmtEOOzx5mE0/DoK  
Z8riyJpbcisK+kQWlOKqpZROqsiS3g8XR6CZjJm4f3uqo1VlAKldxgmVluttGPj2aYTU3c8pvYAA  
+lsyIYie0NPOeXnYeIpkUpxrGQvM3VNqAds2/IQIWpI5EQkzuGGpLIFcoy+4rCHtpVMeelguE2r  
B0Mpg8Gvdy5R++9Jo7jw2maOE/lrrxO35Rptenfuv79oz1qbNVY7cV8T97mmklMKhKpZS61vcJD  
svG3f4wG8Ng2bPAYKd4YZR2wCpoCeeQI2t1qzegbr8nJpBg8aDKC/UlLEyyrjDSTIw8WjRZMLzoD  
zqKPDuz9GKNffs+SeHJ5NzZiY9X4a9AC9kyh0q6RyqiysAvg1kQhuLYWrdVRQYwCuivIxxGeIjBd  
lMKYvreJPGTWRLi9fWYm8ptDa0Wrnq5s7aFeFhrN81Q6AGFKdYcZKLnw79b3eqKrAbuHobe0R7DZ  
paCRxkjfXx9ne62tHZQdHJsJesB0vWdzYtskjb8zHXLBAoXOa+hBKSBRQZ6SuhhnQLmXZuG54/Hs  
WkPY2PpzOMPfRXf60btv/49Tyns6KZ3/oo8ns7jeULybfXpVkfSRf08bPbVatXS0SMHh8PR0BsHo  
YEFaaOJ4V3BXAggup2T0BrwQc/IC3k2hrAvb65Ssz4nFC/sqJWqG9Fs2kdEsLTnMAYt1G6p3FTgd  
TC8SEa3xYiNKCEHhS3xpnmyOXYkYVe4nxSYwZGhgQR8wlKRVXkdmASlBQMvpwtAGCXLQAAeDqRPm  
6LNgp/ZClssVWwoypkKA22YyuAMreRuGEoIS77B0LtIW8a62GSq0i/CYFSpMTKHYThDRRC4CxvtN  
uekhkzP9DnsLeRrcjqkkoBlglDwme+Jsyr+qU5DhZKHDApQdftsbe+QQf2H6VaY47fljcyjmik+  
OUU2dDZa198SLiHhS7GzYwCImz3ts8cD6003mpdvInJzFft27f8Z/f9/o8GQXddNjKbz+9XGpIe4  
W907Te9+lm/iu3ng6HK0suMJ3mQ2fwEyrqGWeI/GIR3g+Zc44HJ2iTsdjrU4YdjW/q170E49Br10  
6L0zWEzh3TxDAxwfnWDr8qohhtmkizlMMoU71w4WLAXD3rqWqDJWMGcsd0DbTV7MiW2Vn0FN0cq  
rEcVPIiN3SIKyAJGIdImE89XSRTMmjvmHoPSvX7GFqNXQYLN5EY3tm1FUafHgreMAYRXKYkJqWSn  
an6LgpK0wpvGitgfuooQWc58WTX00rBR1WmEX6rsNag0xdWpeeazxdEpNq3xwLNFExxPBKuKcEim  
6plB08PwH+1foD65x+ikDeuxx67j05c/yKqNxb4I7B3HWN2dyy5e6151PqdNd7cKlApyGWWawwG2  
5wr53Lsly9evHzha9xc8bFaGNxabcNy0tHhEcmr8jU78ix96zqzkL3tj3aVesmiXRM24on+TwYVe5  
p+tvob7juU6uXqk8cf7M6footuwk7MsfNeeLe5Iqf14QcVBM7x8icI0oBx7nWntDV5UmquZU4YhS  
ix5YqY2DJqhmjrtqKkkMQSVZFtiXBB9EieI4I9QtHVNADmkWKsFmTAULIaRm+Sv7VhR6ObxFOUiG2  
FHkkcZQqr2rIvY9tF5fqEoLBKRvaWlJCgtGhqKXGIuEs47ymKQ95gzGmKkqsKY6nZG0SQacacBFJ  
HzG8lqA9znM5bSZNre3pXKOW+X7iNncYW/InyegWrnKIEyIfL+gQnPkqPxyYlKHwx2h0sh9wrjb+  
dNn5Hke3+8kGQlxOSIEeGQYKsbpUt5tjCSx5+Y00r0P3m3n/LnGJDRefvdddz/8optuWkv9dN0b7  
mQA+id8bz75a9+pSOhiZbz15kF64bf92LNyLT9/+qyGqS3X+Ox4GH8zNBC7V33d2/sWW0+xmLsjT  
iYblmm1xmOM2QXSWnDFi3GQhwIQh8TA9lqSRzq8YfJa21xTPVUB6cczXiQshkTv5JnuaUrPVV8LZ  
knlrE7CUIWMhQcOEGFDF3X7/CMUDBESJkoSQ96fQ1rOATPounJOHbDX2j71EO08cj+NtzeKvSFAC  
higDJhzbppz1dwsqTE4AWJo+KBibDsgDgD5sytbX4kyVHjNWH+ldLfwPMnchNTNzGY5eo+vbTAAS  
2OVwmFbV7xNUQwARDLg+2QXazLoiJQCD2HMMAXhh0AKgAELjNRhU+m1W1Ipdgs1uX+RoKEsgYEKW

YBqz+UaCytrc/MH2OA79mA83Me/fc19d9/zkm5o+9jYEGp/HoLwZ+0Sfh6BIRLPFyRrWpaUnvjgb  
8N8hq5t7xw/evQPHTOxhF0vVGRfWap7nuzdczrzjx7xQfA6NZbFz+La+vTnrN6qyaqKhgYYYY5YpB  
TbV0lDwQtXGUTOhUtjrUEUB2HN+HPa2KRix5yg0aNBalUKKPIKhUHii twjvNeFNJRmM2RM12YgdC  
rfWhVfYwCB4vixTKShGoTBlerlZ1VWKRMAdy5ieJkWXGxPPNppZZX1qtKZCUYlhEmlARDoTDDf01  
BCM6/zaKhVKRY4+fIowJB7zNWNwn6/DyxXY0Y9oPBjyZmYp6tTZKFGiSM75miCvgXx6wtFEp98Tn  
uHBzppuxVmQ7wnjJwSAYWLwnP260WjM01xzGclCMZ4k1w3Hyeseve/eq7bHQpUh+OvPRRC+FwY//  
RWl7MmGmuq96UmeNROQg+rxZBJCsltkQ6kfPnIk12bG3asb269ib8spnkuNalWptWGWlavs5vqqF  
DuoUGTf5dNmpyv9UwKppirPtYwpAEIYdlWOaShpSFKTsKrNI4j5VF4/NRBHeJsKusBlklDzlt5iN  
OVrYu9qZ2oiRvqvjaaihG1UyWJvZbKBCTAentBQly0hLobkASPUbY+pf5y2aQwz0UsDobG1veoVt  
1M2IUXRqnpPWmdSNWs1gMEWBfn1rU1anm9QOO5SjCiADTSeRMJgOOB8v1ht0JCNEd65Pwxoro6KL  
nvUXEk2AdzFWnWRAG7xhxw57KyeoWNXH+dz7mej7UtrDM8CybdV8tmwu14uX0wMJzdeaNYGjmMXL  
d59L1xcm49SesV2f3j69KOP7CSHLtYajWaC6rJseNiDeSOJg/Gest7jtIU5lqOLv5/tZUXLQWWwa  
dJh5B3ff8FR44d7fg5f2t1fXUhTTNT2gquKbOujpMXjG+fvRm8HOQXe1CL63dpsdbgO5vqPhLKw  
m2q/JOUIPC0Boy35qAlIwidVWocmJcTfeILT07maZqva2eADkr2PV5iQoiysxyZBcK0gYSwu0sP  
wPdYwTPMeIMUzVrK2GqSHMoahJB6eO+8Pum043GFPVphTHW0cFui8aQzyNGK4+JgjeSYttA3omKM  
QLwIRsDqsCYtMnn8yoS4R0M/VXglfHVtHwJywdspIt2kz0vPxc0r+yNAXUyeKOMRgNpi1k2RzqBS  
T30OYscyoIZESAIjCaGgSKzG/VaVtnyaqaVtYuuUTx65Ei70+qWur3RMD6AX7++tXN2cXH+o2ykf  
co1RL71SoLwvTD4GY95p9VgU0+R6MWXqUoqPIXydtmFJIn/gSPKWzjkXV9cXrq32Zw7zXlQgkJP  
VighblFORB/ma675ghVSuxTPUeqnwmHbBfXLus+oAKxT2ddSft25dmm3yeCgEKhruFNIwHiG5qex  
dDVY9sxFbMfWARsRA4vYoyUCcLYBFiC81CwF1qObqHo/JEi7QVTlWcKdFBhkw1wRDMq+grI4LSO  
N2LE6ObjYKlGj+sIpR0OqgwY+LQI3YAVWi2Cni7Ab9nf8CbGBtCDuTh700RaSAvDaAZy9cWjPszw  
AgA+Jxfyp8FtDyl+hy5qE7HQ855axIOG55L584+xp56wt+3NXGaIYrpqfSmzdgrVrPMBomqWXQ8Z  
9s108HJ649zqGv5ju+eHA66r+912ke2d7btKGWziFJAEa6whiII3/Osz4LImBfHqEO33XG3cfPLT  
mZorWCnN9JAOW1ZfOM0ST8SBpM72UBv4K28US4VJfBLe7haPxK2/UcXyZbUtq3vCBh6VonpI3V8  
4JPvbixTSevAJO/GqJGziUhsCUkPapNk6nI2xSYQKyqu0AZae5fUw+KUwAdJYn0hI3tKQ4l0Ji6+  
aJSII9CyT1h0ArdY+sdgT9LFKuIAgaaxbMWi+qt0m74K4MG02IYPKOL13RDSambCkYo9DMCEdLks  
0iPWFfeVW6uIolefVK6tHqJdrbWyEUeCO9OkKFMxYmpZsgcRWFCUWYL1erO9irZHJX4jqdg15Mxz  
c0v0XDU47vEaUgdGkD8c8+nLkcwrigVDGkMoDEAERLKRgZIA6Jg5KactPPfOOaw3ij4zupNN5xY7  
LS7zVEwusmwzbtzfvlcqdLspfX3SyPOiDVB+J5nFRYcIvAr4r6Uu+POBwyZCilHogiOKqymYsn6/  
X4UhFE7T01PstF+yLacRw7s3/+LpVLpz3kfnliSY5I04ReaVar6plCGwsC2d1ZpdXtTVNxxGsZUo  
2WiEpp1MwoW8V2KDFBVRsH+QNZMvhGhqjklRLMUy4RA9oWH11eCwwBB0H+Ldk4QKpZBy9QV21A9T  
MU+KFM8pGdNs0wXsqyZ15xtaTICNmWfiCSzQydpqxahlWKAkkUzXZbZPDIji5C2TKu8Pj5c1IQA  
rtkoTYvSuY4xu02MP7Ds7b6Ia2ubdI25/u2V5b80ZDwOJR7mMv71Fjcr7YZ0vLCCuV9DKZPOOQdC  
x46EW5gQ4pjQNoP+j1LKN15R+H74rCH9tiAy5blliuVemdpeXF89KprcnOLS4fjOJpvd/sW0gd48  
XyhyB852POsz3SzFUyFm0741KcfwsrMvfylJzEer61WW/Im6LpEUSxTMMhP4Qj5jxzyvXuZgsC6S  
/NzD4eTSYk391dHk0lulMVsGxbVqz1+TY1cDLG7HPJdvHyeDi6t6K1QVYETkHqZu3OfmV700++Vc  
ScKlyucwZaO5i2Vq1KmoYCKNxBfMfcgYyMxItaNiB3OcX1pL0dikuGRjKpwYSp91QEa7vtmWRGv  
Cb1XfBRAemEn0+nwQCA0KkDDBhGYcgo3G7qL3V19sDBJKIzZ86pMTa3QJ2tC0J0FoyGVKutiIBAB  
2Tq4E70oC2EiHbMPDCbLBo9fTbq1SpVYXzuMSG1DdzNBh2yags8mcEmCIUEvZ4kKp0Ijc8A9IsN  
qQ84bhZNh6acZhBqsQ3bddqr10sXbiwbnOU8oL15QMr+cRYG3e3xiv7DsZTgvA9z/pMJrBQAcDI1  
codgUGYTb2BjJ3xooTRYi4VVC0w1igMs3ASNiaTsMI79sJkEsxxrjXmhfX742H3/xn2Ox2M0YW  
RMry8Z1Cx4JGfK9GFTYvUC0ZCGDadppkWiaZfU835KyFjSnqIPJL+KYkCQKbxu1N2B1e3izlvzdSMK  
0jWUs7dkiHneRwKpgLcT3UPVFeYMY8KD6q5kIS3MFFGfhsf5Kupyc0wApjFMxoapbGY0swazSmz4  
pPH/KazrZKno4jGz1/bWKFnrR2q1hYlPwWXMNICFIRwBJybIsqplktULBWE7BteFe2biQwcpJSvL  
tI4GMim0Vg5RpVKgVZWVvhcQfAtscXJe0GJHr3YQBTpc4KAEi4nTCGnmXCSR5MQf/dcbeFAki8U7  
Hy+OGeb5jz/+d1xGJnJoGlogvA9z/pMZqrBaKBrSLSr8Wy30sWNKU0oG4EZx5BvzYocnpb4+6p60  
Db/9g5e1acchxNXMz3B2/m3hVHgJQZDA4um3qgKq3+vH9OFC+eoeuIkOyp9fgk9UezRGFPdKXvDL  
jAG18FqvRQ4obRxYCI2hrMNva1VPMUZ53i8wUD5fAqsYm91ujmZDSXMiNqKLtQwdHtIHHRFncA4  
qO7grFqVatUVXYz6fIqAxd+KOCWU8EKYvVkmOXYEJglM66ymVCiYcOWOGB8DrzD3ffdl1BAGCB4j  
0F61t5aJdNCAAC00M1s6rYIKwiRwgTgr122ylyGz4bs0n1WkVC24Bz1jx7vWLjABvuFm+oHbp86  
QJ1W77Q8CDMJnamWdhjCMLPVxVKy8xUqJxEcgc5ZDby+Tx/5MC0rVze852CmSYub3DWmdXt5OD+/  
YkLeco9Y30mol9DdvApWPuOux7AHy1681okqJRNjSROEgikF9kwFnh51jNYF9EZf1zm5+zAKL/5p  
/7H6C9+4pu/2/L8ZWeSVCKJogoWth1P2GBrMkn6aWONrj16NRUKJV0dTVWZSNQZwesVS86LXBmMi  
QILFLtQWGRp8qTQfFUMigY8UoAxtwk5hXUOKRsiszwjSiYx/Hq5uUDI5KL8Xgm0QvNUbos1q5jwF6  
jcUtDDNxn0iRlWbI4SqooSXCv+xqUPlJ/HjyqaSkcL4G7tgiUyx+EstKlGsEJ3hSiwV14YBBY/fH  
xVg/G7U2yCTjR7Keb7vcrjbFLig4ZSptXGRQ+EJlfiLHI4WqN/bJL9Yo+E4Ir9iUzTYIc+v0JFjJ  
1W4HisaHHjZOAvltaZQ00zkvmLjFHOacEBNhvw8n//SmeF5NuVcw8q5bs71PGs84vw2aBlpWDe6/



QvZnrE+A4bKoY9SZ9Pom5tuvD771KcfTFUoKugigxeQdE6kisnxL9sqG2p2gL9vGNL7oMv8WEU3Q  
Ru2rOA3v/f98d//4jvfbNvRbY7jvCxG/qhnMhvNBrV2WrTV2qIc51qmgN/VEHyiih/K27qeYvyX8  
DGTiLMVE1iikeOxwGZiB+OvWXCxmrFqmAV9l00XG/LqBmGucEmgTxRs fHDk7Mxdnlz2NmSAXUlP  
2Eo8IVrU+Kwn8M0DxsLFjome2SYHkCLTBuipXAi8JxCb3qFoU7vr7ZVyROlK6tbVbfc8QkKY/bMk  
x5Z1ZrM5eL+otpaZyMtCuiL90I118gCqtTr1GtdpoVmhrBml6hczLORNgAVpDEbvm15Ir2R9/NS1  
cZGMr94gNobF+TzeRjCdZOpTKN9I1EUtHM59AYM0cLUEiKPGD3dspzHxRgQaudGnM7XS8awH80Iw  
veM9e1mXEXASS2oRAimjVkh54aTV2WqjyceK4WxarZ4/Ae0+YJpGEdVzEqP8zkuWqYZqXOk9A0//  
v7Zzvaf/Nb8Qf/wzu/wvXy9zgZvQDvF20QnI2qUqnQ5c11Wl05QJ7pSENdmKzDgHnC6YQdHyRsG  
RTNcX5eaKN213YubtDahbNUK9apXKiK2jg4IKAGgLYJQB4XMgtB/aqKKslwAvH0I0pEWqxXqXOn  
q2stG84vOaETnRuYinqBDQOodzkTSW/uI+vr8jPmw7Ik+4Rq01FcQ7rPHhagDLsGZULvk/RtwVkk  
Y22zXn/gw88wBtXoIbRER2gIMaGkC/4nNPNKefZki97HKaif5pEQ7rqyGGan18W0eTJJOFOUoq3mh  
2NsPNACSi13dEiKXBkYHCFahSEA4LahMwueKXH/icwZB2zovl8mIleQdh0IxfE5cI9570o8t5Akk  
3FsFaqpW6yknNfwRm3w+Tp7xvp0Ht1OS8k1xPFu9TWbAnCm7HxXeGES0oTXjkFzvCavgsHygr2fv  
54H7S2mPFLdXvns4zX/1281f/3Tb39xEazvdDzvxXmnIrlXkRfpluYWtdrdbtDi/T6F+NBBDATOmN  
KF8bezp7FEoMo8oQxerTVrktZ6vzEnYF29e4vAlIYsNw7UyYUmwDTWHakMdYMJhn1fsgLjAz/cVp  
y4bqzu3rHijABaAZgyp9o1Ra1BSKFDYvkzJpK+YKcxolrflL8DgKU+YUb6VYH0gzXKhZV30fLYWCQ  
hsG9+e2j/81hYHKU0v1ftQa7rABJFJtNwVOOSCX71FzYZk9LW8q7NurzcNURc/RcNCjcRsshyFlu  
x3J2UvlulScwdTR2lmlenM/TdCXlhlCzlGzkSxiKM71/KIQhSecCiD3nIiItCnQRZS085UF3gww8  
6ZRQy084Hs9tBw3cXKSx2Zxb5BZtrtnrE/nMRwM6R8OnGdXdnKepC0kYEDTrGWZeYR/wx6STrOxP  
paw+0KoBu+LvBVZ47/+mX9Fb3jPf33Smd/w7343+fOfEouNwbj3p26ce7Pr5UxoiKJV8/iZJ6heb  
/Ki9AjJsJHTqH8F/ICZWAd8Too0DFGB5+SpcfAoh6dzNN7aYENiTwwtGDY29BkB28uGfXL9ApkFD  
u843Iu3L7GHKZBdrJDNxt1sLpJdW6D4ifugPkCcQHE4WCCjVKC0kFd6rdYiG2tBGASnE0GJsDzE4  
sGlqAVQRqqkM+Ta4VFRqBKvmgpiMUVURnb6zCP0xBNPCughzxy3BLuJ3g95NLNWoXm2EiLeU9sP  
s/hcLm5T3qb7a1N2tle5/chGV1HZ1sdYYtgYyzVqN3uSP5pc2oh/EtCajdr6u+OJeARIMwCL59x1  
GJMghE/ryhDA5DtsH0lbdLu80eMTJt3iTAIdyyrtBWzd7XtRpaBnYLTBxCE7xnr056zfk6Fm90vs  
71WfJcsXUzJPDXbSnfx4uvncnnlVQTgrw3zswxlennDe/979qvffvNb5haqv8HO+Dtsjm/z7L2gp  
H5p9QIdOXhMwQM1cTZG3UgDhSyINoP825rIVI/jIegdsTfSkFercxjXffxsHPRULZcNL1ebF4+Z9  
DbJqJfFs4oGS7/HMZ5DRr9L0fYar78BmWzAKYD7RTbsakExKYp0JG8UNoR+EKKrdo4tcCsF5ECPV  
QbdrxDBUqp3maoAq3smLaf+aEB3c64Ko2+wUSZ8HuCCi3xtjVqTFpYwqZRH9zihYrEoEpsQl+7tr  
NM259ZQns+AEHGDAC+VoLLQzuGIAIyHGEPs98YCxjYvYqQx2vFPxESwygdRLGhf8N/fM8rGJM05  
LTEgapgZoch+HL47+zGH0wO/ULxLEcgGxSHEUC6KdpDU4LwPWN9Rgz1ymLIrprck77XnjZNs03+o  
yH0nfDi7DiOmxpaeEYd6YyA+/Md7/rPn8x++R03f8/S8vyaOez9WNGumVhwp86coqWFA5yv2WpTk  
NIpQVAdmZLUURNF7SIUppYwLhhZj/PZntunSelgJNXjmo0eJeOFzqGlbURkNw/QBL0eoyF7WaiZ0  
hxRwoZiLy6zYZYp4felSnn039h7QCJRCIZDATAAAG/4saCgpDdLut0jfEu4TlvDqLUE5HTXSLONR  
kyFO+nBuz5BA/b28Jg5zhNtK6X9izWq1RdkOF9U1vmXeZ9zcI4Meu1VNiSbtKCKNjGEhA65OCKZQ  
qlEolFfvKXNoe3mliqHsgUa7/Q4rG0qKUNMu05Q1BurMUf2tn6pzuFxpnbvZuabvoD8R6GTDfodz  
h8chOfsSc0tst1TruNvm2Y0icddAKEzIQgv79szlqe1wGQ5T1I732WImDrXJ3vdKbEDP389y7LP0  
apVrZ3X/tvf+Uff+/t/55PJb3znK3/adtwemfYvYIPQhB7OPffeRjff9DVSzRTdc6EzgidjAxAwf  
aJyRkcBHsxMqXtD6dGucTjq84IfsnkFoeCJ47CjWFO2z8vFO+yBMaFjAo4Hi+HcNTMDyspFqYaar  
i3AfeUVDen3GloAy/R8AUXEwi0caQ4nxQmcaSoXVTDXFWA8B51T3jxQAV9dPUutrXUqCoNGRziW9  
++/WmCRCHJsEcebySo8IIRMmxv0JgzgU57nSbg7w5j0REqeKCx4Qe/vtCYo9G4L/WsMhuVbXo0g  
UhWash8MQAcaMlgsCEIjXK+4/Z5ji/EMi6KAbad8TWljuuY41EAbBXvufHYM01ThVLtft6L+qadS  
4NBN/NKjWxKEL5nrE9rNfizx+Cyf4yPVEu4fNZryPiSem7f/b5b4//0PV/9Hx996P6ak8v9SJPYz  
trWNp099ygdPnxMeJuUjkwqVsfDqtGZnmLJZtoyJsbmIFRlW6JdahQ4fE0KZMmstxpbS1Gk0uz0A  
rh3ldJ4xsaKcNkUPL5TIY2EwV+Rk0nBC+0ziSMxtYNcNSQFrdRarNobJLIY5qwgJ2BGtJnYUDutD  
WqdfZg9ZlHGyxb37SeH/z0ebFDY3SivV6VSc14QWONBX4Ac/cFEvDDCZIO9HwptyHs9NyeFI/R80  
YPNFRZQPBxQsVynHuex6XBCIUjPefOAYDYG2NNI6eKADQKbgzD88+snjmVYqZH5xZKAs9TEchoZ0  
WTT8bw7eUM7xTlWEl87KQSwzFx5RhD+rF3XX54Z6xdvY7sh8T9krEQr4HU//LtflPm+4z9+dk/s  
QQzvI7+YPPSw5Fh5ujcuXPUGwo0WGL4IEF25vNSNmcmco1iXFrdsTQZzUTYUy58Q041T16jxh63Q  
naRF2genpQ9ZyFHFhu06S15jV2uYlNhgdXtK0m2kUWeyd2KNVuGi83Zvqjmv3B1BX1qfBVR EjIU  
b0drj4hQwVoL+WqdYkcoI+KcbjmvuuoPLcP2ScNgURiQ9vaOE/tlgU2Qs7DDYfPo/RqQYcUrLKLc  
JfzcKCRiujJFssiGRKzQYF+ZczhPs4DJYFwoobGE7SPMGXKiMIQAw4IhYT6BRsvSZqao24nC4bjn  
uW693nl+duyJN4yE/bN8TgzNVKmCcIlbN4z1ueCiV/hWac4Xix45Fi3/uYPfNHne/uvfDCYm5v/w  
UNHr729u3U6GfGCfeTBu2k0SYT4SeFzSWvVqGpsNpWNnF6LCESZuwLNltJwnc612mwoefafh76M  
DnN10+KISivGKvQfjBDzUBIQkyWzeps8GzTudup5k02ZYRM0E/dVVtP9OQQBr1jzjsNr0hesUo5f  
gxa29Tt7ojHxchafzimdm9IF1c36NyZs3Szo4txyFuQU1N6NuyJS8U639+a6N1g0Ait3UJtCUgyg  
RGOo4TP2RbxKxgseqqZSFdkwt0cTgK+11DyBtNwPYAMAX41gFMG/Y7VjyJzGp9MSiUyk9YpcUPC

+TySHX/NUGURnzj7ewKgnACQfiesT7HjNbnPMkrFNI5bUxZXTLb3z/F3+D06BTrTbeubDv2M7W+  
nkKOOS8685PsMFGSr/0CoCB4IZFqNmZfBOMqRZqpom79XNBx41lw1gIKyMlJYcmBsQqxk6NG5TSAQ  
dgTxTLVYLhsRsaTk3bl1RXo30in41act7JHB9AjBmRPG2oEkeL+tryP4HExAQMxav6vvHCYw/CKM  
Or3OE9sdXu0021Tqzdgj5sJVzEZEaFSbkPYOQtF0ycYD4SdcAJML98FqB9sba1Rt901DNFgaAKtr  
SIYMBV84+eiAgwjQ67d6WxJDp3zy2yoeXh3Y9jaMPH7eqMZNfYOXHJLcx/N4uD22M534vbF9HMQh  
PPm4+8Z63PFSIH99fIFnTd+FnriSzje9ksfyZKwfbZWrf9Rc24xbm+e5hzSpwc+czuNACAxGdZkJ  
pUoROCaVfVepkyt5Wpoik9DK83BEwtPcCSFJSErhZq7o2Z01aSOfrJV1SxFcWMyi40dDEFxrmra  
DfNC5S2bKbhh4meFEowABD0JQw1tASIYZd4E2IDrC1QfeUE9VqXKACOMWNUGM7yvvqPU4Lx1zK8L2  
TD8AueX7ImrjTpVKzUJoXGZBQ6BPc8RoH6vPxKh6gHnrJDYxJW5/Lt8zqX5uQXK511R+XN0bxE6  
fCoAq8kRSCH+9KYW4pKtfnVQqX2Cf6gH+qPovMb5x+PFUG4S+2NU6qfLLIihlI32DPWZ7+hIjd1B  
AH0uetuX/ndv/wlnbvW3M+p2XCzXJ3LsLgs15RRvFOPPkATUp4wUYIu4lmV7OOVVUmtuQqNVVRnd  
RFFsUukqncLI9f0p8pYU6XjCg5zGZHb/ZMnujQu50yzma6NDMNrzqh41qGJhZ9YQPrRSKnaSftHL  
WwYbSDiUgnN7b9GcMYLh17I3nKb8808G6BFBc6zF+bn6cC+FZrjXDsKe3yenz/5iyDCp5BKytLV  
ODn5zjPTPhix+OAPR5U01NBWLkyaJ6IEg+n6Pmwj5+fUKVEufuSBNE9Q4RvEvD7hZxPpoWCvmwU  
Cysufnix51C/c/4nA+yoY6RDYTD7RlBOFQAQCWbacTbnrE+OyzyCvjhDH8oORua4qgmYqTqc9Wf6  
B9Cor6g4473/SB1uh2q1eY8jolNz4H6WUZV+UPU3Vmny+eeoDBT09zGFWHqVFTrypAVA9YZKaHhK  
TWOcVoYyEPCoKdq5YKMcjR7g2JS1N6k5KxTy1DNnNc+jb0q8CrhaGSBIDfmKRY9YVw6jOVsFRg6p4  
H2mGxsGA/ZoA1o+8VLKlascjpucP+dp/5EbqFyWqVLik5sEZCdD8j001dmw6rUV8dz97VMCOVxZX  
qGy77DHLXEYHAI MED15JNqskeTjwDgj7B1wnowUIWdhE8hTpcI5s8v/Zm/tuJ6Q1zm2m/E1BL1c7  
oKbK/ydnSu+f9jeuvfcQ5/qFQqVZGm5nkGqM0Lo/SSC8GdGd/V53bp5zf/9n5/58rI+Tj/2qG1b1  
tIkToqVQRHTCoaNjBeh7c6rIhZvDm5tiaLuDi/cCWnokGJD1BQNivxMkZTHbJgiXcyLGARfBMJuD  
kGtxJpJcohtC6TR01QvjRSHppqrQD4Judi08izeNhUDn0LOCjdaIgz+hozDGRIBWBHj6b2qIfaYF  
3nz+MvIbR6kkMNxG4DBUpUyTiUmwy4NO5dkHE4KUJzHjgdtGuycYsPNU6M8T8VKTTYabHAIzXw5j  
AJpDnngOmYPHuvidQJQBV+OX5mXgXTPL1B14YAQPtU5x071h8I6mUZBWihWB36hyPlG91HDzf8Nh  
94PbF1+olMsN5NGfpytb3TIYcP8un/7vufUurZp7/gnHb/67TcDuTT7/vd+4HXC1geeJtMwXMD2X  
sTmcFM8GW8U8qVxqdQUWYiw35JV6JUq1GcvmzQPUEfxT5IHPhnQugjUz1KjbzJrqhgvUEBStElq8  
FzC4ikAHx4QEz78WoR4U6Zk8aCkhmUAvpBRN4TKgu3V8heAQKLgA+EpeBcIbkFxINHeV0SUVThuc  
r6XssGgGJRbPET5xRM0AfqIc8+QnwNCN1SEcYUF9Dx76+LF4nEoBTP0M/N8fcVSkT8ne12nzF5tI  
tuMi49qpZLzom6QggEjA/dUQoA+5K2mTNQmH33y+fe50hyZbiDAjmG31bj5cpfv/X382o+k5H6st  
Xb+DH+XjzYiXP5Uma6E7JjVU1/rh17xvolHr/2Ha9gW7IyP5czfvf7XpOx9wTQ3/Rz3iHTtBqma  
XyGw80JG9Y856n7eHEUhdMcoZI6JkXwhGGAzto5MtwCtZ641zqtTfZCFChOSidfE5dhgU9rXTL5I  
2LHqYymYlYLP1A0RhGAZgi9mGMBGg1JXdxV88Qr2c8xuo4iGdqtPcPZXhuARhgIkEGseJ3NK8h0rX  
WjN0igCxnYdxYM3iBqciRE9pBKZTxMTraJMzpsaiuwTVlnc3WElt+fQ1W3Qjbnno6pQB82rh19X  
/bIkr7ztWBGmLc7QSgV/IKIhI1HHAUxxDlmdhsTwuRSglw/SdMwaNmefzfvcH8Txsmdw53Vi5zW9  
NPJKC6Wylk/7AsBW9DnNKR5YM9Yny9HoVAAdaVl21aeDTWxxZtaZc6nvp5z4DGvUfYoga84E+zN  
3o0i603OAXfLTgOoclByUhyMJBvb557hNZP3S1h4Lmz91KZw8irT1xLOUyTYIibQ01pp6SZ8qam4  
k8SJXZACA31Q4U0LAP+OFMF4EQhoPB76NsolJPaKIB8gjFJ1pugf8rvwfmeOdGM9LoKreQcDcmHx  
XmDWSIcCbFZ5dirycT4IDnkRCOZqQVLSzroCP7Cinjt8DP+bVURo3mgYalJaJ2ELcEde+WmfKaYo  
wJo8YAWxiqUtLCXIVM3fs6V7/P82ow/HHrNjtDYOJlFqhu8GWS9nbWe7Tr3WYbxV3ye06PJeJ13r  
GE47IVRMEh2Wp0MDIY2581uYtCL3/buPWN9vhzv+OW/F4KWD/z4WwccmqamojHNceibY4+6wY85/  
v2N/HiQF001y9Kh49qWua5SORyLUPC+F76Kzt/9Edo69WkxstbWKgVsnI/ffSeHkhYdv+ZaNYEzg  
bHEiv1+qjmqCc4weSIUMJYKf1GwgaSEcCvp50oLB7qswmOckWEooIfQl0ljIRizwkAxTZjWfQRvu  
qBkuZLLilgUe90Je9XyNV9NdrkuBRoT0y+OK5Q0JlupxZGCC2ecIBw3ZWA85fxz1BpQa2eN6nNNz  
j2XObJQGwJ0ZgxMFMGLs8fdWb+AA0rSycTw2IBz1eWk2Fgiyy9FKX/IeNxLeUOMOVqJQW8a9trps  
LV5ju/P33nF/F38nA3+XX8y6Af8mZJyuZoZk4wmHIp9WVyzA5ueB2jPwfeLzpx/97+qGfe8cU/  
VPkBWYCunJvDeMhXvZtNpj95bL3rsEw+TM373+f43ZUHgkdUg7N4JEgJwmnxPGYw0uvSftbW3Rgk  
b1Pqc4hXpm90UiG01MJSUkXmLxFPijUocJbqNxsxNumWoEc4HRSbY00VbKNMusjPVoOdcEomChhr  
SzdpcAR52pau7U1/t2YFztY8es3vonccOti+fEQZuJjBeItGPmJbIB2K4rda6EvWDOc8grDmjIn  
rLV7cgmsW/ffo4c0EcZ06i3TZWlw+TGUVyozMW+4fQ51045hWKUKzUgMz3meKEITY/fdjpb08DM  
Hm7Lli9+TTRJS+XvylNk9UsMwaA01kcG2fJJAuNIqcoE4rRH+ZQvXrwuj1jfb4epqWkINKkriGRj  
8QwqnHKEHp8qvIi+s5JYt7GEZwD60Bo2KJSZR+1L5/mRT1SzIC8uGEoYRgJeL3LoSSQSRF/paRIN  
udshkyaBYrU2xSiFyXpmIDhXrViDC0tiT4pwmEC0J0UUAL5HSq9ZqYEnxQyihTbfqpy4Uy/XgEqr  
FnbCAWx7s4Zar8W8kuNhTFi63CU7wPvPAUBmk5HkfnnlCPptCGJYTsPhXBVoh8mD31uVofofEk1  
umiY1dfTfninHjiUn0+iOP4Qntr+x7Td5PI2ene/GhESpDnBtHSTwJMa/ICcgkisYTKtYndmYkr  
pvrQ6IzHLAHHBdHXR6aGokktf2FFCIRQWudCtU5+bzHX/eOPWN9vh23/Op30Ve+6zcVpWiagg9kk  
Rfsg7xq7zGUqvoil5R3JUKUh4PRb2dp+uo856mgOj1w8tW09sS9wiMk3MUuqEMHnPO1JBxEUUXCT  
zY2UZbDYDUBg5Mrc+6nRKSIBWGUstNSBOAa2zt1JoQRgWM3m6hZUglzpbcb6R6bWm8qVhBF+anq9

aICHUt1mDeP9jZ1t05T8+VvIa86Twau1bQ1JFLxL5muqdgSgU22YvG4qCinXkHENVEiy0UNqvP77  
Fx4nPzf9WK6/daPUXNhhYzL25SNO3TVVSfi0vyBbm3u4K1G5rw/TuPHJ6NO6Nq2kbB9Go5t8fWY0  
XhgXNhPLJsJ4p3E8csxb0QBb2RxlDhy04TM1cU31YMEbMBb9xeVzWC5zC0YM9Y/wmHGCpKMHEI0  
rWEd/w1NoG2oeBDL+f19PpoEtzJK+RTbBiv8fKlw/3tS4Jz9SocRrKXqTSXlbGmHEpCSNnwBX20s  
7PDax3D2H1S2rBshAiZiW4xXZu9V04EkoU61DRV+8ZQquGmxhaL8U5bPYziUzLT3SEFGckTf09Uh  
HgirVecUzC3/Bh12wLJm3/5vyB/8ShZecVYL+fLtbGvWBPZLYPFAf9GFRqap6DXlmq1o3q2js8hN  
29AcXNA5y6e47w5T/1BSMXAoE/f+RBdbEXx15eLW/sOXX9Xrdj4zKS31RuvBWYoscmhhTHsdy0zX  
zKTZGIYpmewgabxZJjG8U7iF6tpqbmUublCwr9LMSOcQy1yDm9S69LjEuYDPQYe4z1jfr4fnOvx/  
xMrTZIz4BhkH30tL+VjbAnv40UN+bYf/tnBvO/fOB70ZBh8k70q2h/VwlHe8XNksetx2hzy2gp9B  
CB8v9eiuf11Dn0jJVIMCigMMgIZ3EAXS6A9AughNHCsnPQoZZYOXR+HNMrJEMHnDNIg8KSgMENVG  
F6WTtFckaZKJSLaJWHI788edfMCzd3wtVQ8cC050PMxNdk4zmtobDEpqtUpZHKqLGdAaZ2fn6RsK  
MGYco4r86jgO+52+2LQnKZTHtIlqUUPPnCeVhaao6Xl41tueXHUefxTiVeoyXgdu5BRKFWi7fULq  
IpR008Y2GiK84eyUn0hW7j6pZxNWFky6mZoNyGHH7cvci7co/bqOel709Um/+zynrE+n4/XvueP6  
a9//J81aRq32WBzvP4e4B//Ha/sBi/cBTbaY4Vy/WXxJKhE4ZC8POduoza5nk++75NTnioHF1eh2  
yXAEaEaNSyZw2JNQc36/RjPQbhhras+JdoqMzk2ndFSLJpO5VYAbLCXbinQWgHet/mbFqoWjRKgiA  
dyrTSeS2VDIUYC0bNjZpMrxm6h8/EzyWUlKkgijqDvqHFi8tJaCNKW7pKCICIUswEPTCoKyD4Hr  
8znH7LcVmdEo9FIKEcXV5Zp36Gj106NacjBRKu15jUbB80APSUUn7rbWXX5CHhlqVCZz1L2qgUOv  
V0ASmpL4FXKEN4ejJ/DYhq31ijobstsLCrXUEbPl5QsBiKGPWP9Mj1WP/TrhoSHpk9LX/fOzws7Z  
GNBoArGLrARC0d9Rkadl/Y+1y98Je/5L+Ycs4TQUiQdeHE1lw7LLKjNnoaCkTailCpbZK6tiiFib  
CzkXE6mUrKpUpuaqMk0telUWEoKQvC0GBQwzNlsqtKj0bKMgt1jT4fvoXeTaIEqkfFIBDAfYT503  
KMxbxSlw9dT84avIp8XOhgKDeEBTmgGk8Z7pkpXlrSXFwk10Jec2ZI+sCV9XSdXpMm4z7m5I2p0j  
546S9fe8BIKLxKtbe6wFx9Qr9W2W61C/fy5i8cvXN6u3fCG7w70/v2vpSg+cawrfVLk562trYxDX  
fa4Y9p89A4JcZEvWxLquvy5MAA/5Mi5xAZbpXx+IrjjoL9JX/Gu/7RnrF9ux8aH/z9zmm6Zhf3XZ  
bml67MZhcPnOV7/Yx+Q33/gPa9RbIhp1uR/XJ0rVb7B93IvTJK00u/36zCqUnWFXI60K80VDi+vp  
ygIBa0D9v5CoSDMCqDyNMTLjYSD19AiyKJKL19V3xOsE45AEWOpDqdRptTqALQ3FAIKHMDyb4AhE  
pWzQZ0DbPUYgheBKzY6LHBQeI45BC/ML1HzRa8lrzrHUbm3q3dDioFCpOtI0cqkHOaCoCabylySU  
nSToDox1Lgh4H2mmkvD6bQ4YligztYWDQcBtfvnhR94FJP92Ln1+srC3M2FSuW+n3/393zsR3/t9  
wfVan1275eez12HPbN88tG556+MzVt+z41GnWJx3/VO9cTXkVdbzvJzB/9RMP8Hf+YtlU24gCFav  
KBfzGHf1+a83JHMcoN269KyosI0obeMafnEjVQ5/EIz6XJyOfKg/cJhHUBOLm+CZV8ZJ8JeFJowW  
WNOpTYBLYBBwnSF7HvaF82m2jSpogeVOhKYKOKAPf1ESM9BIYOcVBgVwKkbBILDDQZ9moz4Kz+cY  
oWaN75R6E5RyEK/1TSsKyhvMLWFBveRJoarA5syvRqKExk9XnAbA05oOnmZbrHcPD18/32ixN5oV  
mlhgb11EBNvZOWF89Tr9YoPn738wq12701uoXLY1977ntzeqtzzrJ/zSacy87X5uLeZi3q7mx2H  
7u9axRKEY1Dql731TODPffr3zNouG0EizFhkgR9SNfLgyZwHzuca8vF6lvZWo51ljfeWT3zCohFA  
erWKC/QsZe/kexiTYHuEwVe8HyMzuXpoUceY48HsUJHIHUp2PLZC6c2DEDTlwAza6sWS0bagAXDq  
9TgIg5J4U2l9ymyHyga4bmReFQwKoAkGxXamHnlgDJQJUXPF9XexZvfRIWF/WRxTi158Yw3LrvCY  
PXXRCmgpzE2gJHksdDugaGi56qEhQyFvuINznVlnWqNoOwPwSQ/ddy8dvvoE+dW6fM6V/fto4zLZ6  
+vrjVK1/Kq11a32cHxx9NPv+ZFH/t3P/Ptwz1j3jicdiZnaWX/zSH7pxMnx6v2XHC9/2rTs11NdH  
m3f+T+SzM3x2owtGq77RnE5c5KYHdbET6OwZHvewXIS3hhPxq+IErORpYa5sXrqFaAoWd5/FTWWD  
tHSiZeTC35bNkQYXTyBuvhYPJhlj+ns2XNUsByqVEpKc2am5apY723LevIsrqXlBQyF90WRyBQK0  
UjA+1mi9FYT0V01BHSfSplJvTfU3WCsMRurxbnz0ivfTIXlw8pQVSVJGyYKSaaIP8WpWguDNzycD  
quDdynKdFEJm4dq10ixC31fCMzwddzxqVvp/gfOOur6coxfddDOH+QH12y01BXv+AtVqNb4uj2+yf  
SBOjK/v9frx3bfff+sfevv/9uj3/psfCK5+wcuyPWPd09QNqSxbw3N3lw3Le2nm1b56tPHIRWPr9  
GZmmZ1kPB7Z9f3JaOOub1dXCkY28bPJUGnG4VE7ncyn8bjqeIUR+5VKMhoeGAWHHbj8qFr6cC1L  
6PqynFyK/OC6snCUKhS5D0BfnBNXvBbVPBdJZFR8MXrZtmuELOpVegsQ8+3ZpqqVED2libezqSZC  
l4iwBNJeJom+WoiI3DITxNQusR8DZEsc0ZOof/S11C+uV8KQ8bMUBXCSbSpDKViB5JukJulwpA4H  
atTo3rwoMBJC80MNg2JwxX16c52i+6570GCrM+5i5doa32LvuWt30gPP36Gw+AJ9dodSngDQT90M  
Apyw2H/0Hg8/udJZnj2TN/8ee//9v3ffVr1/ov+7o3p3vGuneAlc/KH3xJPDz3aT8Lei/OLoumJ  
B4G6bgPGNwk3j4H+J7L7sAJg1EuisKCW2i4WZa6nH9ytupm496ON+KwGIwLh1/uNbRw9CQVa3PkN  
fdJrpYihMW85hScICrmE9FirRR9qjbmKR22DjnFNUK/94SIWXQmijVN1OGyxULBAwkIcV6aAOYk  
MYzMS41/2FQBGhhpMbnEkE+ZZILq3NkVDv5SiosHiQ757NX9ZRXh/xkpuZe1Z6h2kMzoS8YPopUu  
h0iU7G2GgYQwSqwVGhqTyizf/wjf0OD0QcOaDa/DI9+ugj9MLP30+Hjx4S/uBL5y9JfxS5dMgWz  
e9TNCzrmMcJPHvsxuOnzv1lp/V7n7zr9ls2v+fHfiHaM9bn+THkhTy+/HjVzcw0SiY5Dh3RoLNS0  
0kyEDO4eQ4PcxQGQ0tQPuwlJ9EFyy8to98nWqRWeZhm999A+cYh8gsosuTjr8wp7C5oPiNDqDclT  
BQCQRXu4EL1LTL1ti9RYY4Nlg07r4nMhGM4VWLfouc9CyZhihXeWwmjPWroJON412+X9G5UBQvC  
YwUrwPnEED9/PzqsRdR7dhJcgtVxZyoo3QWv1wZp/RkMz3ELiF5osNhZQ1lqCKTDMnbirFCzoGCm  
u3I7zGocNutt9L6xjY/RSGnmgvztLq+IwiuCaCkCyv75T7sbK1LAatSq/IGZuQm4+EBzoULk3xuc  
RyZh9fWtz/8Q9/+zU/kipXBT/3K+9I9Y32eHvuuf2Xv4UuPrmRucb9tgDLitCb9TYsc3zITXyHAY



3soizpXnuevKreEYpnBeahTnKdcfUVYC410TA7nfoXGPlm4MMY4UYUWtFmMSAs8papIBJTRzuZFK  
uVrYpSxNmYYlJXYMuKWCjOhwuEq1Tc1qC2OT7opKAgFangdfVSYdaLaKKISl6oh9NhUPN/FA8epe  
v3L+LM0RL9VEEemYnxBzjs12ClvVapJlOBLNrR6HNxuqnvAln6eMnglwgxitQsP30/3PXS06nOLg  
o6aBCEVSxVau3yBDh65mnrDCY3PrYkOK/rKIEzLgVoVbBB85MvlhVKpVBgG4cpoFB4PJPMPuv3g7  
v/9m9681mw2g//3N3873TPW5/Bxx60fdzfW125u72xcvbF6+aQRDQ+P+v3FlYVmvUCjw4eufQE9e  
M+dyaC7nRw9fGRSn9vX2964NMyGm3T1C15SdvxyEXxDAflkskcFltepz0mPEbmbJ5bEZlFucp5Vl  
H6kVl1Qbx5PZNEV68tiqMgJoduaomqreXlBhSkTNJnqh4o6t7D0J5KLocIr8ArO+WAU4FnK7Ex6q  
jA0y7Z2q9jaQITtUAwnU8Yk54u0EtzUSysqF68+L6CHPH+1XV8po2v2fkl9lZmVWruM5ilwRZwoD  
4+815Br1YPxhlJMn07tpBEkLnwZ57uVvSpy1f6FVWrWSlSvlWXYAKgmTBqBrA5s+OgxT/h6JzE4l  
SbUmKvT6uXLHRTe2trCzCqXLVarXW73aOmad5p2/Yto9Howf/zW79xc/+hA+G7f/aX0j1jfY4cP  
/+TP/qyc2d0v5HzylfbjvlK27CtixcvCiLnqmOHPDKZdx2oHdGZRx+mhYUla/nAUVBzcc5p+r6XS  
5aP3sxOwaSd7TVr7fJ5Cx509eIpgfJhJrJcrVAQxXThiYfpDd/yXVQuL6nBb7+kmRhiUR/32XDFo  
wlWN9H5YboLrM/UgPdU5gKyEKNBn/x8VRFsJwpVBP1WdShvKcrp4GdCzmspgMM095WqsHi/KeWog  
hKaiWLTv2AFzm3zBWq+7HVUaixK5Zc0Hako2SA8BldTpnq2FmmZDan6KsJvcCZJ4Up7UkAeETGAX  
M3UvV9RqQPvgXu+4+O3y3VN+N7s9EfUZ8951aFl8ngT7LS3pNUEZNYkjGVjCr2EBoMt/r1Lhw4fo  
8sXL/D3AahwcqNxuMB5b4X/REcz79ha2vjrtFocJvj2I98/796y9bivv0w2mTPWJ+Fx1/+Z+6d  
9728bc//vAj38/h1bWupzRb2u0dyv0O3Wu3qFAqU8Qh3aA7oMHwIudIQ15oAdWKPl3LRlwu+OSsD  
q2Vg4es0OzTE4+do94oZW8woXjzHHV2dtgDhHTk+DX08JnzwhD4lre/m5ycCmFdv6ZIrzMVj1YX9  
onor3KyoRiZkrpShk5EpSSn82SWVUSLZWTNUSLFULCKZy0bGCdkHgxe1KaIHqs5UozMZqThhPJ6N  
VurPLLy2PDSCTYM5LxgkGDjn7/xtVRZPCB4WhGGylR7JgXNWaYGyaQCTdNqtNKSyaTyG2H6TBGra  
aGrXUNFJTMv3BYQRQCJN1cv0rlLlZUnsSH3MeGwfjgcky+V75IM3U+iRISiAw6RW/0BdVo94anK0  
nNCxoa5WFKht83nKRocX/Pr6vycq3hzeGm73b7n4OEjd6yurj7wg9/+tvUXvOAFwTt+4IezPWN9l  
hw/95PvPXnnJ2/9b3nLOQlDgd5LyAbW667LlP+fX+TwKpYFMg4iWr+8SqNgSI1GQ42EsWfb2dmka  
ukgHTnxYor4tR/86G3UGiF0dGjYb1GlnKeVg0fEYg6/5e/o5le9jr76jf9SGPGB8wUeVzqwqAmv8  
mJzUQwCI2QIIo2pWjiHxjBoVHVNW4HvQQkq+V2c6fz00zOpigkURgevozynMiBD690IO3+WaQyRw  
kQYur1jikSjucv6AAM2VWFr7oZXU+3wCc63vRnIQ1PTdKocNpQfE+oYKXK8JNYDZlLnjolcxM1A  
FsNE2hRarSSUpoga5fP+rEPf4S6bJgwpDwXchyCGzYsCYEtDu0dP0+5oiM913KlRpbv2xyxLOWe9  
4c9qmRlQXTlfeEdhz96GE1s23SKfiGf4889b3vuifPnz78iHI8/4/n+radPPfqXpKag9oz1mT5+9  
r0/+v23fPhDP8uL3h+nA9q3/yBdPH9JaCnrlpY5VALojY2zVm9Sayeg9dU1KYQsrxygkMPZSrFIR  
7rpBB3at0zNzSXAxFunJ992G40Sn7Z3tqm1vk5Hrz5GhXKDLp89DbJO+o7vTc1ltgJwWhY+VkiC  
8wtFhFANMLnqxkF0bIQhUZTsShgmBsJJ0JlTM8AfA4j1uXgGauo8s+GhMEq5LR14coYw6KHY7W+K  
jYEkbNACmlkM6A/WjowmJimUpI21U/eTM1jJ2V8DcacCkdxJsLF0ywVea6iGJ8aaggZlgAg7S1V/V  
TR5dNUY92A6iWPKMK/iKsZQAT/z4x+/RRGzaTlXiSrsWGREWlubvNnZ4mEXl+Zo8fhxevThz9DRw  
wdoY3uH0OQVqhgoxUMpIQgmF9HItCMiSVsrpz/2kaaFNvbLfbfzfN9Os5/ildsbm6+/id/8J232  
5b1B+/5hV8P9oz1GTp+9Rd//kc+ffttP9vv9ahYKgmEbjAeUBAOZMokDHPshMHIALMePctaLditw  
Q+0PF+iv7zwBF178npZZH/7oY/QPfc+RKXqEnVbpyUH/YqvFZ30P7dWz9D+g0fp9W/+JmGc99BLF  
GW3XeNBI98rlgTwQIknIXBKcnyQKak4NTEDjyve0ZFQmrReDM5TnVum5eX9wMequdWMZlKQqMoqI  
7WVQUyBv+ZUbiOe6eXatxoQS7ZMQSiJE+b3rVzzUmocfyGlbKjTPcGYKstpXRsrWE4TTSyuK8Cpl  
pZACKzH9Wbq8ShEGUoIy9Bka6IGh0/PX3e2NujUqSd0iK+iB4ev0+bnDTnQCS8XraBgEvHnHlCXN  
8lrrnshbaxeomuPH6Vz5y/yz4cUYridFOoLITOY/VHAq9frsvGlouKe2Lxx2Pmc1+Rdpen5pWPtX  
v/1URD+H//6n3/9add3f+h97/+LtT1jfRqPP/r9/3LD7R/76E/1Wm0xRoHKCxB3gXPJer0qi3pjY  
4sKnAvhuHzpHP+8yYukRBEbE2opLzh+kE6+6AYhJvurv/0onblwWZTYet0WHTx6XCqxn/nkLTQ/1  
6TXfv0b6YUvfvQVNgq6wxMP92Y4vEy8wONsxJaeiNBCZhwVlclOvQXFws00bYriNQqUsjhaOQDYJ  
1PUj0nHr7207rn949rHKdCDGAsowqBxI4ZkkUW7UhsyHqfeTCnLqcqVeHMPZHGHIWb/25VReOSqhr  
2rF6F7PTKDKlEoAQfJFipxNvHMq9xg6qpTtjsGRrvGKI4WlBtZlfl3pRnUZx6duvZ29ImZuObuOe  
aOx9PCgpXRpZfgcI4N8Qef5b1DijTc+fZqo5QL1e21aXmzS/pUFmeXF/Gs44QB7okLx3vYWDbsdd  
S+zWEJph99rMjFsRDHbly9XecMoLi7tW+CN/CVW7L7xbW96/WX2yt/LYfInfv2//GGyZ6z/i4+Pf  
ehDP8a7sBXGCKwQc2iGBQMP2u8Pqc75qONoxRByhZzSNjUEd0TVnE1veM3N9KIXv4QefPQx+oM/+  
mPyiu06fOQ45pFETt+5IF75byvfNlL6U3f/DbOkwoyqmZYvhiqWvCWVElR7VRGN1SKZN5JFpbi3  
CXp06ZT1kA9ZWYZecWvIgs/msEK4R2LpZp4VAAfLMvYlXjEaBty3MwRozGMRKzAwgSBB3JGzJDqi  
m9iJpIX+odPumHfNeTXaqLnKgUh8AojnzXMGUY/1ZiYsrliIYpmUcNUSekwxXGmP9+uWLPicjIE+  
yu6qZY968t+6o7bZegd90Z6xxNJCiRUR4huszeGbi02BIvbb6vXp+1WQo0KP2p5snNVvj8T4VFyX  
VeOTYtFFTFkOozHjVUqCHyuIKSddg91KBV9ZJm9zfkvNpjhsF/h8LjSt/t/yV+73/LG15zhc367n  
ys88Vt/+P50z1ifckP9oDvo9l6fce4SS/6kjMAU3iCvE61fusz8Zi0BDrbm3Tt1VfTmHdlzxjTN  
77ha+nYNVfTn/zJn9Dtn7qfcsUS1ZtN+eOfOn2W+p2uLLx3fOd30Ute+nJenKEUVWADDha3FmSyL  
BWigljBSC215cqeHDhf0pVUpet6hecDgslQRBFGLgHk+BYDzN7aFC6kYNjhRWnLlkadV4lGcVhp5  
xUToUAPFYBfxH+1nKMYlRa3Mvn1pcUjVLv6Rgp51UacnxfMprpPi1AVq/BaCyidontWCTJ9VTWXS

mo2V7CFphp2hxfVvE9T8AM4mxCaW6JIZ+kKtko/7r/7PsUNxZsaJ5SSe0aRQjslhv6bhbH0Ygcc9  
cQapWWkXRoHAa1ttIQOxrANebZyeaPM5TxBsmXwzBEGoib8XoGEyaat8mbZjCTPTqnb6Uu4nHD+K  
xQ5iZnn98jz+lkKx6M7B4NB71++/mvuYsP9IX6c+50/+UC2Z6xPwXFx9eKBIAj8ELOyQNFYpshNY  
PdMJawzJbza2lwXycEjx45RIe9Sndf6a1711bSwvEQf+NsP0aWtMR04fh0VyhU6d/YsRv6+KAZ7Z  
KVB7/qBn6R9R46qciwbnyCBrtCQy7RGapoEkre6BZ9cNnpUWg14p0gpu6XZVLftt2qUCfBA04iai  
ml/PBoQn4BG/R2prvpYkK5W0EcRB5y8/z97XwJt2VmVuc9w77tvqCGpSqUqc0IIPDKQoAtFUEREY  
6OgtLpaBAUVFIwCYUEnC8WBOLVTu7BFRWjtdliCtiyW7QCiIiZkHitzyFippMZXB7rTmfr/vr33f  
86t0L1W9zL0SoUKtap479W9557z7/n7vh2urU/kk1GQCS908bOUEX5+xymydM5F4c8zSa+D4RUrK  
7Kxthpq9vvklHmukIZAipzRCCMYzwOryveg3EiXS+ncSVk0nQAF8ab9b9nvV8VcMeyUJZP+Hx7b  
rouGGSms9tKx1VoCiX1METajBj1YlqTeF+ZY4MBglmkFEAFaMDh4EJ10/k0OJahQRyFpYteeY8dY  
kWCPxb/242B6FmgCchnl3CfsxRpASz1lvDCW7KsPj2cp1cF4z8SUuXPhkzgA8Fwn/r9P/vL5ivG+  
v/46/D+Ayd6BNV1hTqHRFewrlLTs03YpFlbWw7p1BbpN5vkG155qZx0yk75m7/9tNz32H7C2zaGp  
ew/eAdX3W9dWpRveNlL5Ed+9HLO+nTDqeJ2HRmko5mU1DBucQsHBYAk1HxknjRjPfwMQtbYPLXJN  
emDXq9MK6KQVMOopqES9xtbzoakxg+HyIJRkCMmrUupNKUtFRZFGtWURw4PMG54HC2XfwK2XTK2  
WpIKRUXwz0ZMbrh2tdDhvFkOKgnnXYOpU98ZWT0QYnORuvE9uHUPymfpZzrlpa90PkwW7AIhiabt  
NrD3LKe991Auv2220JNantisUoSguLBqYDXC3AEYIEtK0I2HQog9+RaSstoz9wj4rRM21jNE/t01  
bEkTLWpMUVkVTWTbcbXDKfrIVvp6cy6zFgm4GGryuD/rkq3hDR/SzEpfzDPi+8ZpuPVN3zHt94Ts  
qV3zs3NP/iRP/94+RVj/b8x1kPLKwUecKOrCXFY/CGLpXPTSa2SnOHQDMcjed6Z28kZIS3+zKf/S  
T5344OyHOqiKRs1KVOo551+slx++Y/LhRddom2d4NWxQ7WsFIIdL5XrUkfd4mQLaxUYjTD1BHKdif  
qo7ZcQWHIOWRtX7hEybdLBIORYcaAiIQb2B2UHjzaaxjNawAepHdDRT2zLhHqJmQNmUedNTnaPw9  
AbbTiEHtRdqZWgw4X0B21NsLgx7TWe64b2P7HtMvg49KWdf9LWq6wSGD1DMkHhJmjhrdVoeZWlqq  
126bGkvFW6AG8ZyZ46KdGcr692QXWDEgp+951+ukbn5BSnXR0p2D5kBI5w1rdiMorNVQ61N/X+OZ  
ACto7001Zni7yEzXqQTVm0s5aFhd5wPnat1AmmTU7sp6bF+ILeX+GpND3RubIAVzR6yXrjOLEH9t  
oR/f3r4870o2xh+72WvXg5//52QHfz3kM2tfeip/rj5irH+H34dObR/pZmWjERcKwgd2rJkrYL6b  
jIeyuKmBQllintjibzwdPpk61/+1WR83HH/XgajjWDAC/OL3D7+zS/7Knnne95rC3vVi4OUjddPO  
rUnoiijnhgDTFzr7gxRLH/VTMIpyvE6qWINuM5g1KaRul3iWVW0QpWLBOWtDGLtI76DQgOZJo2il4  
fKyHN2/19QKJY50OMKpWUmyIQULCaT/g5D+n/Llr5deuHbuoQl1eRVSV3RF1YGF64ZsTDDy8WiNM  
MHJ+oo8cOu/yrkXf40MFjBR2dVJu8DZTIBZxf92CXiTrkMHhS5NPfOo+XVc68En94WS5QDpdhrlC  
n5GUPi4vqMGbKJkDUvw9EFOT/u5a18jQE2tLmktT9+Hq/H+hzOggu2JAJE/N9w9SW4wkYZVNKBS  
uuUtbANUFqpUtUqdt7rJ0yp3VFSd7mut2RnsmVaTXeFUiUQ3Q2b3s8MQBd70um+/MWRh73/eeS948  
H0/98FnPOo+6zSYhmvr72STj1IhmvauryhWTYub9SFUIGdLSGuX5IstC/LSF58njz26X6677UE5u  
j6WtdWjcorok+X0k5bkZ/7jT8p7r7pKBgPdPUo9o6o2RftUSdr1WBE6ENVGpBm4SRA3nihLJ24Px  
rpElXc0e9AY6gXjBB+ULBymi6rakA/UmJA+Y70j0vjJxhqvBELVw3rDEA0f3nNDiCrZgmVMvVz3r  
3K0mal0KAES7HLqDtVt174q1Mub6BxwENEwN5JssCB7770xvP6qPHrnNfHQw6nhd5YvyGS4IQ/fd  
r0Mg+HWnqo3tXaZReLbt+GO/p1RNY0RDh221OCNXht6Ywn089abbg6JRszRS2nC2t6UWgj0aJPdE  
5WhU0MFbWfu0NMU2NwGrLoYiyHlRW2gBALFD8Ngq6R93cacK94PURXosliOiDpIX0yN51w0Emtao  
NlA3YmjwPUXALNULcySXyvQHCT606rePhmNTx+Px69fWV2+7u49tz125U+8/fq7br01+4qx2q/3X  
f6OU4vx5J2V3TzLl0SgICzx7c2RegUvDF7pmbtCLbftZLn1rkfk4Kf1Cmdv3rwkX3PJOXL1r/yqv  
Owbv9mUFGpiWxsbs6TW8SWyCF1ZpIw91QCFN0dqJ7mOL8hnDVEcurTo0iKS9RcWKb3Zn1tkTQsGT  
hb+XiY9Rn4cFERpHCAN3KWsLx8kXrkPAe6ksfpQ2H3VFE0PJbeNhzp0sOsM2XT6ucodZd04ie7mh  
J2ncUnzt1PO4riHRhYOUpdXE8/n2dNu/euW2T18H41RBbfeujTug2pnnKm1vQBtDIx2CDVJHRbs  
+oFI1VLdKZ6W6hXAU/0g1+A6I7UHfU/HE9ay8LcQE7cslkWF+YYUXsc5ft0/S3lRroOOVN1/ujcG  
s0wNBjxJxxsXVWVZ2ujKrV6H2VSbUZcdQEeeI3gmIpClTz8Kx+wycr9bVT6NdNUNBMH+3wQFOAzE  
43U4TNUgQ0nuw7u3feSP/zdD3/2njvVSL5irOHX2urqz42mk3mvMSrz2LjxAIUDEA61hclESdIX7  
94te/bsYbo72hjKQp7KG77nu+Rtl18pm7ZsZYeVqQ70j+qMKSSiJz02tqAhGvXnLZrkPKwbo1Wij  
7gqIvw8IiciKldGAN8aUlJQ3TI0goAkynocUYyDQU83lrXezXvsYroiA4z38KNfZLcXXdI5pK7WY  
CHg14yHs9BaeaPbdr80pGzKloGjwmwzD/UyjHbTtjPD30P0MmUKjfQKV+yHOnNurhcygiWexv0P7  
pEjTz1G44hjoMZmqKlBCjONpExDo9xLHXs+NIJKU9y611LE7c4779Qo5wguOLbMRjel6hWztX2+v  
ik4tx0nniAnbt7EiKoGl+r75Zkr3xv+07xeYfhk/MmZcaZNrjgicyfT9BTt1bR1rGZMZzshiGZpm  
pFZhE6s4VaJtRlFtZfpuGz3rcFBCt34PQ73+cDhA1/3Jx/76FXPeWP9we/+zouC0b2ZnrEyLmXqN  
UfwtfCFScGFvOAJ79+4NqWl05MveQ4dlLaTJfZzvLPmpD/yUvPrbX6c1jBYpOjPkCkXlfdINLqzFj  
1TPQRDBa069e4986INXscva2KLiYTTUgkhn2eltDK0iR6oFmy3QBB6vHqaheiqY2FJJLKPCe06Lk  
Sw/+UUAUy8YXb+nIH9Q69JGjbG2xg8+/9zmbTJ30k4pRhu66mKq2OKFE0/ma0NmtAoRH8aTWzoNU  
AL+3qNsywIJDxkdSV/2P/KALO9/ohVhS7XzzGhVar2axE5wK9YmndTXIxa0pY7s3y9PPPEkI2R1h  
xkGg5QSjZ2UTeyEdaPXpWgkDYLzWsA29DyPo6A4IrL3QqSeTETvBHHDtPUPDGnkBndT8sFFKPsx  
FeNRds0dp85amqsJxHufWXD76jgmJTkHzsJn918VOq1xFm0y+LQaCft3r59e9/9m7/wwe3P6QbTx  
urGL0GP140KayMgdY1/hgeFcQ27nOEBLoQIVYeUanjoGhZtT7xGXnHZ65ge4/A5oF4XIGM4Hx58S



IGSVGsidF3TxcRJ/J4OHh//zeflk/99V+ze7zt139Tfux97wOcgSk1ZU8SleTEv1HUt5jPT1ZXZ  
DRaYWSYw9jMKOtPl8YbfNjhoK/ufYz16GJ/SSVPjPkCBQhIrwD9B8I44kfTyziKmAvRHptbWNdFB  
k9FXC5TNjgCbjjPmC2kaRFq4UWpsorXWXG+qdh2pKv7HriLc8ztp57dRhw07hoQARTs33TqWVeNQ  
DOp5hY7XQ8Z3IPcdP11wRAzRqMcXW4bfbEZRS6usYEaPff+frrqownGOqfZAUkaltR8Duhyg6UjN  
prJ2zDK12U/GaCTJre5rTo4NiBtURfhm6nOrguSBFKKbRtJQXqivldtvdDWdRj3ZfFWnVMzL+G  
22Yhc9YFNsfuOfeTwIA95w01n9/2be8fDwavSbn0qZMpU5s9QM9rCiSCWnSFOnuYE6RQuWqvOvt7  
5Dnv/jlCgDALnCrNcHjhLekTEpFyL2mdEjlbJTwwAMPyKf/8XPy2X/8vAzH6ChqB/K/fvx/yPYd2  
+R73vIj7BLjNRPgMa1BJ1h7sbain5j6uX3+HDmpUwVyCFUJtWGEFYgQ9c5BDkh0hgiPD0cCA4B+A  
pJqaieBNDBYYIeXyg5wPuE+ldBOSmGYZhyo6xpbB51DqUJiPhicBfSaMh2doBYLYe6Je++SYbjuU  
8+9UDOAcl+RVovzWistL8jMrexPUOfQrUWECY6qCD+G+arrRqEEaHu0tY1K2gwj80Szo4ph/6HHF  
Dize0GgxRQ1ojrFAUdkaUwQfcTEKIf/Kl+epXWtp8E0rKmWT0BD5ZEL3OPLMWXONJVPDb+cJUaWa  
CtUFMrEVpk4n9dqbnTB66P1C6/4oR/a9Rsf+9iTzzljnQxHVxP0gJod61y16owV8EiZCLZ6LMDK6  
KcDOF+S18vC9jPoKSuCy+eoC8R0CQ84qYlnRaSed7YN8T8sWHHPQH7xfntj71Nx/3yNyNKSxG  
0NbhZjoJnCc3w99+A9184nb5NWveY3Mb9psjqPi+onh8iFtjoT61eeAUgWTCwcs7+c8SA0fdh68+  
7qsH3iSGUHu6V9Z2DY3RGBqBvIQ9VJN4QbbdnFOqygeZf+AbO/4YxXvLhgZLffmjJMqhagFw8GcZ  
hXvW5EUMcWEUsVjd98pTz32sLzoFF8uB0cea3EpxTq+2sTxYJqwJoiXOUQQRaO142233mg/3wL/U  
9/V411hj1geHClqewMqMckawAybAmXoiKkmsc0w4DjaUmK+tTFMzULTVRhghfUdecOMBR30qRkqG  
ld48iRQWJlKvamC8FFgt8HptSvujJI0rU5snSYcVe6Zg3Wlw9+3rG2s/jbizHPKWL/zm1552XQ8e  
UVBGpSOZQjeTprY+i/qmkgZPCJ8f9eOrfJD73g3l25JqjImNZ8mhuNT1rT79j4u9913nzz2+JOEG  
u4LqejOnttlvLEuC8H4oJiwztUSHXFGfb1TnyqQvRGcx8/+wq/KZz7zGbn83T8hL7zgxtJcPSqj1  
SMhAx6wI8pIlyh+WEKtiMgO8TCAIIAA6oUTsrrvMXZA0QBKyG0tyHNFLVaSWpfYtBN6+gXXQy6ec  
lZ4reCQ5jfTeEayQPR2pQqltFUGGsntIGGPaoi6dHRav1WI8IWmmCgv4BDXhuvydz/1SfncF26Qt  
19xFXmj5NyaqmHiKXKqMqXs7JY6okIk3vfIg7L3qRVesy5jrmPTJpO2Rm1MMdHRR90VIHnu+OssY  
oW1ZKjZE0gmHJFZ82tqNT01kBMH8CeM6FnW4pinoZ7OLcPJKEZet+spjbjgk4C0ERuJtYqnGj3rC  
B/tItuIvMpbAkO4ppde8aNvzX7j9z5SPWeMdTgcXu0PjLxRo30hDSlQD3Hmpoihhp26TN531fvko  
hd/tWysHJDD990tt91+gzzy+H65+549IWIeUBQMvtdvQJhL1fs2bt7GmnJx6zYOYJ94/AnKZ3KBU  
6oa9ogchM0BdlqHFDHUT5//wm3yvLP/VnbuOp1REukpRzfACaOTGV5jmJxKg4ChqgGJONZmtHJQ+  
oNeqH0HynoREyRj8LcxgS0sTtK+7Lzk62Tr2bu5QwbGv3Z0f3Au25h+FSuHZZ7zyVGLycEIVHUGy  
yaJTS6+Pg+yGgicIbKChx7eK7c+uI+/H310rlz+nvfK88/frfe8h7S8ijDFxhc0A93VWwLDPXvui  
eQAYSdAnpqkjZzS6k810o594tY7J/NjTgrcr3XsES0VlZR0iO6N1Y66vhKvxXGM1argHmfUPg5Zw  
2gsOWR2uFy6cU10mh3xzQ0fb7wOVq1NFAOpNIRQsJwoTGYHNEJ3Pl2xOnMuC+Fanh++de9zwli/4  
5Wv/K5xMf0qb2Aofrlpwez0cSGVNHVA3DZ0hX/1135LFuchSmlR5KUVuTSktg/L48H4II9y5jnPI  
xxxGmq8pU3CmenK0cOytDAvCwt9WR1XsveJ/eHhBiNemGNTw+guSksjTC2NhwRe/ROf/JSMKgixZ  
fJ9b3ijzC0msRZOe3PUDC7WlySZSzRVRI0rGt2AgmL6yxFFRSMj/BBi80oZLeYqmE2c9nWXhSC9x  
DppvLYsg8UTJAufdf2JR0K9viqDTTtiulqXKr/i6CBqClvKmzLTCwLxGj5jMw119hgq+PLZa2+WA  
rVnKCMuu+leue/HL5cr3nW5vOrbXqsHFK9j0jQKTVQ6HZtf4Wt33H47a9XSm0BOVesgkmbGK04kS  
FBD9mZGLKnJVZWFprYsHjBCgXGhy53aJgLDANAU5u4BSfbG4RY4AxUVI/k+EZcjoOZwaIMJzxHI  
iIziZpXqrLhOzUrwjTYK+kgqzxp96ctNFQuLYtZT19f/ixNx33o5vv/rZXZ2VZXg3heUTOgvKXu  
qrBW/k8NAZVw+Hg/DX878jRdXnk0Sf13kcOyJXX38ltayeedCL3x7B5j7WheHDBaPbvfyhYVCz0x  
YN9KhgqSM7zS/McBXmN4i16YmiNgqX494QKBx//07+Uj/3JX8r73nMFA99yqsqEGOsgefexv2hxVI  
dhgsnRpLkRhboRL6ojCUVhPrusqXJs3/JoPNfJg0xbjxVak7yHalCORrB49IJt3ncMGk2oRd5Yd2  
8GqTTeJXXBbwIw0FvNmgeLU3XOfHDyyElm5fMLDRzfk5z/46/IHv/Mh9gLECe3WVIHMC8tsiACE9  
3zgvvvZreVunY4xdqNmTCS76SVnwOHaERkz2+heSbs7lkAGS4cxkkoT5/pW/DlkO0VduQePiCVcc  
xMiLCYBOi9003RABsjh3+eJKlM0tEN9vs1MuuvnwNP2JCp6tNhpRZnxXL7sOTFnnayuv6Eop7vZ6  
TWlv8ogS6ndpMoedy0mXlBbakfPju1rudx5/yPywCP7pDfYyt2n0+16iEBr9Jkgm5993m7ZG1Lfe  
/bcG+rXg7Ixm1CtgGwV25qmzZ2MHlsPXGODdpNkQXRHqj4t5Pa7H5QrfvIKefD+u51WjkmDOz56K  
Px5iMbDfwWAQq51IBBPTMsNHawHF613Elc7Onh905nPyjB7vk5/IaTeQz04kxVtBjnHNKrl20Jlw  
yftYfjIS1TBH+D6jeGa3Hj7Hda40wMtxlvFffjP/24vP8975anQoaim+8wOjF5G2aVXSXi1ZA6P  
87X7WGS2gh/dyNq/FrjihMpu79jOANL071bXLEpNGa9qim0Np3gYOva561NJP4vYIRw/ABAjme  
8w+m7SkAP2tpQJ+Hs9YBe6sSw3HYOAUYt0wktfxuftncSfhrQ51LwNK/t9+/3fT49pYv/0bvj4LH  
vEDUHooO3KaHBOQnaGP3lXjOYowQevBXiiIhuPEoqP5hRNCvXpIrrn2dtlz98OS97bIj127KE06C  
jXnU3v3y333yvjELFRG28JuTGihQMUegTvKx1M+aXtYWKapOvE49wRSJGHHntCLv/xd8kn/uzPQ  
4q6SIQUonttTZ/KRGQ8tGwAlSbAXcWDkNuhcugb3m/7ORe0HwBJCMQXq61Ou+Abmc4pOtAk9cN7I  
vWvrIZjJKJ6S6rObwv0t/JeCy33n6fHF0ba0Or0aXNOByAP+If1WUq19y0R97+Iz8sf/8/P6Wjk  
rQ2eKF2q++942YZV03sGEfQiUcqJyak6Uwa3BvMMTIDBYSSgMJwYAASi6sZU03hNvM3IZuoTG7Ve  
xmentLIOeZRQ6USYuIrPzrYZXZD00CcMBvQDrfMADJaTqzVsibZGpleks6ANhzrbOSB+fxVtfnju  
mYNNv5Nk6o+VxsLx8f9q3KsON6bM6FkdXx5vLMHx5oXjYInpmct1S6GFAjLkMbL63I4/AakD80HP

zjbt59C7whNYW24VCRB4yETi4sDV2k0TRXXxr9oa99aJ+Fns8Y6lujyrk/kw3/wh9TQ/f63/CgPC  
Q6210uoW4EN1jSqXVFBMgJHFD7qqDluAkFg29kX8BAnVDtUAWY4IkZPT3fL0sYQ1TWYah/jh/fS6  
IF7S0JBSKEPHz4ot9z1QBzwR6CB1WOZRXnc0wOHV+UXP/jLsue22+XH3nWFLICaZ3LiX3zwUSPW1  
9wCkHS6vHp9FTvLbUMmPK/+HNNcQEIHg0Gs/dgswggL+2MdgN8kfHa9TCOXc5l1/w9ExXXOPQ100  
I+GqiJw3NNjCDBGyrqJkMqynvKZJnG7uzewDJTdtJ8/7YoQNLqSRLnK2hDEs+s1Pf/svZDtbAGm5  
7iMrD/93vdk4UO+H1EINR9V6esibi6jAkA9iyZBvYpadH5uTh9eoSBvztyCh52ElHZ+YFFaSmBvA  
NlUWzcZad+BAwfk6NGjbDpBvhShT9MYjXSZTgSUjlUTWci5eVo3tDIYUGUbwR2sAfmSj370T+TWG  
77ALqLXRGLNpTJpjGCVTzkPpOEs/D6bQeACj0F7Yt1Vx0/mlrql3QgieRylpXtuGki6ssxvRXJ7  
iOykK6/6U5ZWdloGSgdp5EYsshXYgDc0YRo/Vef+nu5810/KfsP7ONMGNf+2KMPU0CuMf2J2lFOl  
nInpmOMLlKkKlCv/5WVNQNu5LGJxQ11cNaoulGzkwpXshGH/3Iaa1srkuMb/oRT1tfpzTiK+KfpO  
7fRsIx1qn+N9WvdmPhbGiOoj5A8M8iyufilPMX8N5A1dS9IwcP7jpu0+C1tbU3hQ99Lg4X0tKKH  
17TnLLsYFDjnl9t3+NYLi4u6N5Q0xLizc9Tinyth3QP0pdYe1GFaLu2MQo10pR/NlQx0BUQXWhZJ  
D1Hg0q1+aiVjaY7NiJhimlNkzppebDrwVl86i/+Qg+8SXV2my7laCmaJA9zBy9L52AnoZ3X6pLp  
7z/5ARsu64WtFASDOERp3kaKVWJAcAsQx1w394n5YY771XAPRyfKOger+Pvo3WrzW3jfc1Yn//EW  
98qt918bXhWjTz00ENESdX2GmJsFzqwpDUQdHRZP/ZyGhcifM/w0F6nQgyPErJc12F7acN//Z6m3  
07oartfyFzwWsRA94SZk9eexEdbXe0bBirrRwAgktnXCiOGkDBQTWOTSt+jbSQRRFDYImo4kqy9T  
5GxVSj+/PCBg6cel8b6tje+Idv3+ONXKqKLSAR2ALHxDB7WaxOvF2IUqPSHoC+Ih8tmjyVQN7UBu  
x0rU5KATiBAbCCFw5Jr/dX91XDPy6xxOXqlawQehavY5u+kpOFZXn/DtbJ8aH+km3mrhRG0Mmidp  
Ymtg1CYG5g/nAOfrfVqEgEFbSRwrKx69opjiBIk9WbaRhe7tMpmkZPhkFH18zfcTM1eRfCo6qED3  
10qxdNARx4pFK/Hz/bEOVW58sqfkb/4+CfkGyCclwL3F+OWRuVTvda4remTzqWx1xScM3WYtTfgY  
mtKrZsYzt1ZN9z5avKsxPbaqkyCJ4LzQQOwnyJN7ptEqjKaA4h5Bzg+y1T1lpmFlc1Mt7dq1JbnZ  
83PHUZIPstlSm29hcI7KR3QB74fAkTvuKxZn3z8sTdNp+ULiC7BoJ6H10S46NFKDuLnkDKFnwLVK  
KU3wY0cDBboiWkA6OrBCIK3pWEmbdsdKawO03EzNd11RBWLYlFgtNE40piIN7XN0oicm4GdlaIji  
qTKIsCA8DrJxOnbeTYv9aSITJWY+uJ9JuumeqizQ35GzDBTTSPzcFE7XnghQRZizkUzYVXLf4s6k  
ZeLJlZR6kYAp6/VdYvng/MbjxhVH374Ybn/4cfl+7VDF9OZVM9npIgYcIIPz7Q6X3Z1xOG4kt/5L  
x/hoU8ZWRtb6mxjI4NDcB5r6TQiIQxVMchZTH8ZtcqK8E/NpHSumlqnV39OJUiRTqccPSnhHGMwY  
J45s7b7i3+f2mtr4yqJTr4wWCKlgCpVHVHAIg/eaz9DWVoDiuoUGadtng3EURBBcmkcSqFmDZ9h5  
3FnrK//1ldlK+sbVyJ9cH2gXHCAs3Ez5ewuZW2aI7JkuUU91fJBqjMdvRHkz7THxixOTEXCMAW40  
pi68ucaNGr67IoiYlXSImpUyGs+OE2ttQsMkYqq1hrRUUkejblpPMEEmOU05X7j7w1B/12SoqIq/U  
saY3o2HfNhQLYAqH+rjOk2tgaS5z66LvpbGic+hN6WOC8rGuq7swMZSodRVjN2uZ0OVJNbwo/G6r  
K4sy+euu4XKgYopTtssAdeZK6mbJmyHVsHrdm1N5J4YeiJiCvtI9RxxMZ9sOEn+5gd1xA6yzRyT  
nSm2rNGm88zveRhVKtUNwlb/YDv7Y5+/DemBhBh89TXv8/nlLV6Tf4LkVHF1mvTemp8LhrfV0H5w  
5iAamah6TUcBK6bDSzrT6hzahQJlcyMqDa002OdjqZvCjXHC6DwMC6mHMF10VMAFDjgGZklWgdOw  
o2aJ6IpIehacZ5i+FBNi5Vx0dOWuzV0akMgqeKCGiF4naosMTFInpKjlFHRxI6zmooaq5teMhHjPS  
QeapsSAXqIsEY2pFR3txRdewFQbndfB0maxrceMfK04EKSNOYCb2AVG9EKftbhtu2w940KjkIFvm  
7QUwaaajoug1gqs+dqRW1LAr4o7L4Bw2jq7K/fffL488sV+jcWXXr/FassfX8ToM0d5Aeor5JeVMI  
2VSKX2P0a6srByRqEAOJqOSO0w0U5Dw+tf13kvWk53qg88wGqq7m0p6fd8r1BA8USveMTieIfW3Y  
PBIMjPqECvuNcqMmK6GI+wwK/Df72Ea+xygCHR01D5CEvsJHDFZqt7FAYt1kJGN8TlABrIfR1NIp  
9aOK2N9+1t+IAtpyHtJAgfDYrzBfSipUbHycPNKahhVWhONSsE2lYyy1OH7c1jcWxk4Ivwm106rP  
Ka/SW1euCLEMx5g1DhEvowLHQWkHoUy0yNSzVtGlbRvCn5V7KrSO9tAnDSCRuIMVSwi4bld/KKvY  
oSfPAtEzoQInHBWJmNGfygoZKUQB6b6srS6FCj7TjvRSbtkhi6qZqVJ0XGQLqgRwUd8EShZROH9  
6FM2FhfZVS95qY7eGC1k5m2kp7Gi2WEypQuWHs9C8OF0gauWRRoEa0c3wefZNGJlTxiCRX+oVKh+  
sZAFRHnnQ84ivE6r5VKGaKjzU6BOEqVSAeGDyIYyhjco2mhuGfLLKwDFsxsberNtlb3uImosLix  
HbMYo1mjJfB3o9jH98m0BU3byl3iYrAhXvBMoZw0bb7HL731HFlrMVo+trwx+7KWv5oJq2trcr84  
hLHK9QkYiMoYZo4aaaSFQkjAOwODx2dQAD5a9vajUNN1fq0ZxxR1EsaeevaVyQ2No808njtHliXH  
uOfsqzR2hZGGzu5ipJq6ibiT7WebDgTTD0qh//OPH2n7Dj1ZBprvz9okTxQI9xQ9X/ODXPTfkMEL  
kkru16Rk15wIUH/yCnqK4c8PXUCN65VO+dqqImPTNxxgARLAGAyZ5o2hPPDgI/LEk0e0xky0RheLQ  
pGonWWmBoiYA0gng3N6A6UuZ0YgmpFo1E/MCLQGVnmEYvXNDyroUav3KRFjRmD1ZwkJFSWijqQI  
a1JhGvmoFMV/wfT8iOobVr4kNud0Q3XKnBsYyxQXcw0HCAy6/hIzFAVteS1tNemeL3c0Wfx/6edd  
L1FSZFv16UHjy7f3Riuv1db70EmpZqKjKYT8k9Tazo0iQ7Eq0IBCpNws+cTTaFy83aVAdVz0xLOL  
YKpZGbKZlNiY4SebQDXTd+Z0tDCAwCrp6hLjvY1nctMuSGzw6n90MrV9g2xguiPqE/lpSaxtcSNX  
HLRxfRaoXxgspw6P0+4oa5Hwjo+n2oRp5pMEHFRCz+/5dQX6Gdri9RoqCpMrUAMn0uyH+R0M1uTw  
Z8rp4QUIqp+/gvXx06rd2kzi6jsA6RpNFTpdL55TxJlFqaTpg0z637ywsUghBZitaQQo9ChqbTCe  
9rP8nZXTiNxnw8hkFFDuNKZNPZ68ZnlBoJA+SgthU6tjsy2B0jP+GptcnS6HirsmXQU1PIID4cU  
Md+L5IIXGzNO3JNpxPuB1oa26Y0vrA34Rzp5p1gnJnwHsvHjbH+8H/43leEWu6lupgoiWrsRMjR+  
kyy+VA/jWHAeyJOUY8AgoYoWZQTrmM4dqboKQ4eBnHCEMTSJfKXQ1n3jjPypkcFdZ1cNQ2a6Z7r2  
FpnWPeXJmT5VAY0SBmB1Zu6hq5768q88cUX7TYthNRWWphsC65zvMYfTnOtu7PaIHOJzlaXduy0r

eQmoJ0lNkutbLFy1daqGNWA5NB0xlqVdrIdADFeXZU9d94re/cfjoewMgpbVOePkiVNDIzOsdI19  
Q28aVNBH6Md50SSDn/VoScidDU8N+XRTQEck0Xb7WgkG1wIBNLbRWV1Na0NMJMYvMK0RLpb5Y1k  
enSWHcaDSfM1kFtpNFXytfl+sta5+hzC/NcU4LPC5yxG1gmLR46lVlYZJe72iUgeJ2LUVJmUR2BB  
+uXwo+tHzfGurG6diVSWALnIS2Z23gFKnZNIX2xRU9VxpRYGyAqZwIM8FLwupPp6OkvbLtAa0tz5  
BjUkxqsNoG0G2rcVRPBooYwamhSo2prCCURjR4bSzK7fMxHBniAm+YHcv4LL4iIIIP/26ZzAhDQH  
EG9ihWOkJsBiTtV0a8sHM6lk05rxclqayzFx1G7OoSyMpWpH86AILSWKq1Tve6U1j9fd2OILJmly  
arWV5rMZ1d/NzVVCq9lOzzNmMrTgTtJlZRsrl9ZEhJz+8iR/TQ6B9srXlDaAVCkv4VitCM7qa39V  
Ok/PC/0GJyrmuc96x3YZvakMgogiPWVSy5WPFuoQYhk4GcC2R0ACT1h8JswcZILfCmu/Sr6ZybW  
dCMNv9U3CAzzDozviQdhdecHBegiDe+9jsuCUZ6Ge4Dd5+YNAgK+saIy3iAmGc6TG1q3U4dqkOWo  
zYQvA7PMxtB9NIOralub9bdEYHYaydWY3q64+CI1EHeYsN4q7+yuGiQnnlgXXA4Htjzzz1DBGvz1  
P1IjbkRwfr4WSy2yueMXWMSkES/C3PetOPUzpxURz9iWYRHVO/m0LA78LS6Nk9K4qOHw3W57fa75  
NDKGutNprxZEoEfXbUGJ4V75u0pX/fQRkikIax8XBHxxKIG5jDC9bUNyAbT4HRc06N6RZxTwtAKr  
duTqiUvpB3YJe49BAFG5Vg1kNMOWbx2xYuKf0f9CzUPRHu9ZrLZNNggRvW+/Ma2R2c311f2VLds  
jjOaQXXMq917QNJAEnmxDoaIoIssK//zx/6oPi6MdW119cr10TAYaqEHmZ1qxcYRONLcWh5S4Kyfc  
ZiOCOKqEXqz1NCmTeX07HZhlXlInOQ+AOBp30AR3nmsSH9C80k9YR7naYaoNf6qmNjHm6MGMIM5r  
VtkTiXtoqTdLzwniJc4COWXOFEBdXz2O+Ln7I0HoRNO89QwLnpGTvW24mlorgfRHnhz0qV/V1jO  
LGfQVQt1td15egR+cLNdxIW2CSldXZL/XxRC3jW4GobUWIKyIL/ta7rFv114HV3k11cc2pNK/z/5  
VCrcvSRt19zEEbsVjuYI1WCe27fx2gtMaBDOZrIwmBe5kBusI5t1XFckCmF4x9PJqxP4ZDm5xdZL  
kHLGZ+HHeWymklvnTHjxtjlsHbT39gBtnozRo8udW0puOOJ7WdG/5b28v8tDX7HD77p3PXhndT8  
NkOMregmEclv7GsWcf0JsGL9jIjoIcbNw2pTa+ReVKrpnH9X2OND6R1rj0bwdvcLp6q4FajoxHp8  
j7tKuhtdYwPrggcEG4Tz+IDBwqJQAHPGLFFkzYtErn4gudHkjYiQZLMKt171C44JlHkKgAaSL96i  
/OycOKp7WIog/1VcaMc3YnqCXNloo1sbGTkAPhyolH11lsRVVtieaKMaTbGnOWSdiRVvB4jUslqW  
PYCuPcnb6NwrSUGu8dWT4o10FJTDHjiqQO2NEzhk91sBCAYABrGG+OZ5VQZ77mw4afprhLTkyNsu  
EQM78VUa/YiWEhxGsQ47/MDNXKfj3bvzVPh0XeobOzk6DN1HBMz5HZMk3UVJtLWqVZJFTHHJzcOZ  
VR4rMnfHBFgOhw0311VdTaaJInnbGdzNr+CwnyuGklIexbyeT1M4bYwNS7V4yrIv9FhdKPr6lNDm  
+hDMQEsLq4qOjBB85gy1tcaWWqyd1Ld4JlqrM5qPdJR20rT6vLYw/GvIernlqxsWexzmzqGuzXXV  
/TaJg6ziJRMEXjmbFB+KuFx3EAnsnnX2UxCp3RAtY2YZNbTS2OKMxVT4cYWKVUm64LDPRTsAN87  
U23WAQx6l9jEp1miDyI1QIzPHKUhgRTBFI39WtiI8pJ6l3cLd0OZ6qZrIb0dzweyXwP4t2p6SK3h  
92VJyDH468PcDHev+qF+2LEcaNO0vCkk5qjHtVsSrmvkIvFGGYBzSdjJfG17BkhXSV2HN3lRGGTf  
j26TT2ZqVoJzKq17v73eK2iTqWfaNr6j15a5L30j5/1xvrW7/++LeHGvxmfEjjW0ob+kd1l1Dhge  
/GASLkpppaie4KrzLWmc22qi0prSdJTiIaaIHuvJGECJrJTYTtNV/RgCw6ouRiXCAR8Lv6qg/0  
tXk8bleBKY3EqMvrb7vHNCbTSgkRHSaCgrjeY6PpLEIXnSQSRpCrjplDMZLbLk1j/Yz/MT1k5e1  
y5oTYjh1EZCugakpMLChEuobgpr9cJKBp1J+EDam+ZCqTqyaIx5F5UbirKMEG0xFR9+jVwQWQiZq  
z668W13NIiUTUBE2+XlZa0/8zTySx3q6WssIm7bmjclumZAKgGU38tmVAMxqyYSqbBRFVhUjLgVX  
zPvLYT6NOFYyafEagRAROV7cyRyhtcky9mmiTr5fe9L5L3/Mxjn6Wirxmr9kT4/6GbGAY/s2Dz  
3pjXT+68rbRaLQETwWPh9WH8IA9Y0k0nJ8GQyot/QnebzidRCQJ66hSU7iJTKMMZWokbYp3YbVGq  
g9L9XOTqCukRldHr9kqF6QxpYWhsFvpDJ1UubT+/6t0BijTIabr+7zo/PMZOWuT4JxJHTs1b+PKj  
JEdoyiphZPOsLpU1fygWUz8LtknjYqr1UYO9ENmKCDtnan+00aoV6+76TaFYSZ9pa/hfbiDtlAyv  
NfinYPoUqOJ9QG88eSUQfxcAdBA7amg2IKoJJYx+w8dDg6ipvGmeRbxtT0uYJ7tqMbUONUNfn54A  
kEf7YTh+igYam7XqqwcrNacTQuL8LrEcj/jKh50LlHgz4BqxZUTmw0jLdIWYzSlSdvrUqVJMqtq8  
TSDfNf8orP98LpzMuC5Qb0aHM3wWW2sV737nd14PH47xjW4KZh1rY4nqv0rAAno+kVAzErWKqYd2  
9RxcxzGEQtz8/Tim905p0+JRTKrxVw93W9um8K0CDtdwKZzTAX3J7o3NZnt8npqRNU+7jhRUGKdG  
MoJ8LdwVy960SV6OI0EECVavG5KJY6JMFZQQLiBwkGU37xd03A0oIxY3RhWWWesen3czWN7e6iRX  
Onu2jpeM9FqMNTTrbpGDR1dJto8Dfhx4NfiyjoSmI3TKsuXl+jY3ywgoPJOFMqa11BBQiaIalzubYw  
HdD/YH30jxDjes4tqkzfnZC9Jq6ZTpV/q4qJ+Sgn6UthJBkb3JJJxHthEsEoGEBDSR2pes4loPsD  
IazRWm43fAaa9hLVCsirnJ9pbytrVtlibaByZlrkbbVw+5Z6BETrn+fSwz8kgGVlq394oc+0jyrj  
fXAk0++JtzkszFXpaqc1RTciM19yC20jA801RSEclh4aXaJK6KWOLLhHk14eEWtNKLED15bupjF9  
DRV7ic8veM7PT9Kswj2TkzBUGzREeCOiPaUcUlaVtWkQZmnj9vHwvM6dcdW2XXaL1EKrkp199k9K  
WIG6suRk/1NJfMDououbT7VOqNMOzUhnYeD0Bm20CXP8XubreGavz16ZFn+5fqB4wiC1+JR30o3N  
dTc10s7js74w10WS6qoByXSWyNPEu+Ga6ZBcP7cQJ586imTjRU21noOiI+c3jZST71QWnHA6RTqk  
ylJ5r1M4YbT8dSUHyqqb4xGE4MjppSchYPHtoKpAWvEqBKcBIOk2QEDz11j4TPP9fvWSW4icV3Ra  
5kuq/JN6EnSue91ZPKwGZfZGko7aKrGiAyKx18ER/Pz/9a282U31rX19bcVluFErYHDB0QLDLAoR  
twCV5MZkRMcq3Wq8iQh34JVCrllVTA67cw5p03aRExt4wgtn00yn3ZkLIkxODxTdOieOAZyYEV3  
RsJrkTW8Zk2p5paVfKpzJVatG50iRM+z+7zzqP8ibiMSWdHigMM2AzKM9tPA5DiRDeuhXuzuP2Uz  
hglmwHk43MUvOZSgQ0GsVTplYxrKBFVJ5NVufn2O+ToxshKhCam2z4fbXz+jD13U8VuNmh0dvNmQ  
BH4uR63A6jOLzezJRK76ixlQjRcW9sIBjW2iKRfS21VImfjTSersSaZrT0gBRCNKBDILTn14wdNI  
4x2MDedUuw95cY5VeJQZ1aTmBCuPfiaqS1UdibRQshWkk56rUffOMjW6FIHaX1mgyw6w0Yi/8fS6



lpFDXj99oy02cb3WAKp+aef1cb6jjf/wNnFdHpZV+kBAHOkRkCUwENTQHYS4tng1VRqrLgBrFcLF  
QNDpEQneFqOVQYyaUhUZ8SsK6ri++oK32dacLeLiZHSVvOXwt0dcH3ZJHJmELTOssHaGm3ysPmEE  
qrxua0uFs5Rr168m/NEp6aBAaSJfMxYROMjXnZOqtFqnBPiPRZPPi16G6bWJMa3OF5dpQiE0STq/  
fAgmcwIFAVxltfkmhtvaQ1Vsrh7xhtKSURiVZrSWocXTRLum2lsV2pH3oa1rxl1Y1vsSIHrabd00  
BnLwcOHolOA4fXzDrBetAsrnYjNe1MbNBF1aabcVJVqGfQGMqLLRlQYaj944wFWV2LZfppLVREbV  
ojAJHVAWrSnNWyrQuHpdGZzlobLoUtrpuH5FXaPuivMu2R8ovPRhMy6zbK2QZqzKdYbbdly4tFnt  
bGura6+rR5NMgedxw9flyr01U9l0PRlGfLehgCSB7Mk+97nXHGnCTzfpI6JLFOURnmohPVZPQnGB  
veBNjqiaawRRGND6oWuMGLnNrKoM9v1WR/TpW0Mzn9+nWqLvpQIrxheeCk4ivN3X9gaZKWURBJzu  
3rJpE5UHoURtZyoEfi+zxBtF7aZxhaI8FiTWNcRvK8mX4mqm5caoXlmyfmgAvZvuOk2ObS8artmk  
jjjVaV71FkKeq8MZidxsXBHFBxkCFG2kxMIIkHc0mLolzNNnVHKHDx0xNQIFYWFFBiOqxflJORp7  
+MkfxcRoGiAqDGg/Mnme1IMP4yWmFfDUCmjOqltTUDpnw/PuiD4YXFxy0wdqqltGmtOcKRJdzWGB  
FFbxYEOUFzaYDU7TjHF4GLHYyOvjCYDNbe83NXN//W9vNlQzD92i9dnU3Wh2+eUGiqJJKGYBNRA  
q8+TDU01BNws6puWHKnjIO0iWjhwiCtzTiKsejiY5XcIF+ZdO5XR8NWkULGA0NUsY5m3HjdaPolh  
S1KJD63rAyZVcOzv59/9mmytGWzws+SdikT0lN2MquOo8rIx6Kndjjf/Ak7w5fn7fXacY4bqqezn  
InaCovEBvdoDoF+hj21/3LtdRZJ2mvz1K9KdJ2kGBE/abefaURNlEHDTm6aL6q0Yq09TpqGyhL4  
GwPr6xa9lCHZugvWLiOiJge0/WlWlHJ7bpzFcharSdzcnhVqbDPZtVGCHZsVGG3bqWAFxgyF5ZZx  
ES2Nb+0yJS3m7qyLicSTBT6pY26UmMXRayvQS+zelavKc62PVXOWseDZ5A2Ldc16+XDpf1NP/1M2  
NCXLbLef889rw43eCd5lwZA963T7Z5VVWEg9G5OW+eAj5WmCxuNNTwURbtUXGAbgQktJlONN/GUz  
WhuhtpBQwWNGcwisaAKUaaxHSlVeIhJqlxVHChVWzd9HWreVjPwMxgzOaD28M4//wUGgseByHRlI  
9eYVqxtpCvalVmALEproJuhqu5sNVOobJGQuB2je20opqoworkibxqSzwSkh910/Q1ydhVi9aaRy  
TMtCRPOdjNLqcuompCLS+jIzD4XJeJonZpZLZ9ZzQCb1XU7/MbaOmtK3pveI1pm+N52419ihHbci  
pz3V8cwVPfIKhMNV9wuI2muCoj4xWXYuFfiG+ariBteXJzn992RqnREkkCyrBqIvEisWXPsbML6  
VsfxYFC2ny2K/B27NjNrZ2sYwlnHJzJcGHT/N89q411uL7x5onJZygeXVXW2dGtratI3R5tAUE0C  
7XpXF9hZtPJNKY0vX50+CB3CdVaP0pn+VH04uF7vSyzNDA15QjdAePgee8ItvqwhsAJaV1h4IzED  
o8P/2dmbmkSGSOLc4lcdNFFjKpqKBVfTXHAI1CLA5PMZFjY5NFuLKUvS05+ZfOu01UbqNF6m13Id  
rpLOl/Jzm7dwgRxLdgoEO7R4UPL8s9fuJXaSlx7aF5fkm4K2qpCJfXl0RDuyCzFWUhmVlK1j1IrQ  
dJorYmoh+sbjYaysbFhavW6PYGJbKo4XAcW4PX6wsI0qmHBedBxpyppwR8dj162Eeq3FQWLD4wuNs  
L60ZRHqUTSa+tancKplVHZItCZltEe54cqEuK+TJJIXSkAhHULYM3J5J909dmTThS7q5EKnfR25/  
vJVv//y8FlrrL/327+1NNzYeC29F4w0aaOLjtfUYGHMmRGcY+qGNRF5L/Id8cBAH5uMJ7bqwbS6b  
PaZpG0t4ewZUuVqRTAlvh+HclKteWuTYODms1QbTFU9MZxwGmU8faFyFD02YapYy51y4mY57cyzI  
jE889YVZDuL0tTbW9QPjJTKCEURqW6LO87STCNT8ndC8nul4AjvdtA6xT2SwwHFAxc0pMBIf5fXl  
oiS6q5SrIjMMgdgxicEs7ZlsEXNctx6VBXKEzdYDvMnLSjewVD1WhlIw8DnKTGvDmWmdRN54m1L  
mtx8QHry3H0xJHeaMzOMBxRzfNgyH70he8BBKGCvJXwsjB0beIj9so0nMuIX2HXtpe3kMhEm5fMo  
KxT3DVCb1J3v9Z11hGBpYSMop9nf/1M2dGXxVhvu/6m105H43ndp1p36gGkV6VG0loPPY230Fii+  
1hUbQBdtqanuRg9YVPG/ZmNM2SQImepedmKXlpb7TaSqKpIicvSVopUu7vtuj6xJVGNqUGktQuuN  
db9tcgCnHKiRovE6bzzzpV+iaClqohFIEbONRWGa7OU1ZrPwTGVNESAGQbbTyZhQZk1MO6c2GGFs  
21UZblQ2kpH0ZSWQujBGRxdXpMvhBQYnwt1qX/uRIHlOt7CtffYCB9EA6jguovGulwbwyA6RQydE  
c09WgFMkCoXWA3VYI98kLrdD/NGdIglk47sSiuu3e23uuayLwHDD6dFGRd14bmB5+qia4iW6PKqC  
qbS40rb4wpHyiylUlBz04xjNzjCHVODmfcMzMAGVNmMzFa75AlfUtVl6Tgbqbaz2Ye+c95bCff56  
me1sa6urbwGzaLkDHuYkjk7A2ss6iJKhJAiFZdNqcHyQeKmis7WOP+ydj+6vb5BG4cE+1I0tVVD5  
JawWiF93BJWK+A/SR3P2rgIX8SoeioMY+vrFkkDSTQzNLK2DvY+Iax007dL1TZj6LseF89HJzfh  
s+ZuxQp9W4ratROJ6OQt7lhK3bI7uFURBSJaVG+8brei6/UgCFmEFHQZ4WI7nx5ptkZTjVDm2HT  
J64flPnunlgxQyTKWqbwrbi4XVnc3obWTJrSm2sb6jEiTXoEpNQwSspoKEi6uhLcYkjCwYghFoijt  
RpnzfqdpIhSMxPqQVNrWCz1TaJDnwrOgWimeoyquLj+zk3zgli9PQg4Wky6mXIY/g8Ff/GBby9g  
xwXb9XVDILN56pp50lSyWJtcfOm5Wetsb7+m181Hw7lZUWlCHrdw6JemQyXpLB6wJEijaoaQgups  
d0FMDinwXGRb9mheSXWfcxYG2EjWUtVUk0iDq8xbqgrG6JXim6hKr/WhFEXQLcWU4+DMTTXEU1la  
W31LqYz1sH7bh705Oxznx+hZ013g1plbCKk7gsLBm4Q28421U5w+D0IxpqjIGF0TNBA8ihhkCnCA  
d3pGYSRcn21EOH+9dqBVFa0k55FppA3Q+z1WdeL3pd2x2hqCC3NjW3F1Do8CncnEYGEearSE704Z  
Z1f6YCYa7moEkSXNugqCo2hpYgNr3RVxVw2b4uegsFMR4z8UAqB8aQGu1TCem7Iq9K2oYc/x3qvw  
V11wL2Pitj1TyUKwXVHNN11HU1Hff9YHHDROsKeu0BZu+nYtpcUvbn5T3zgV369eaZs6Rkf3YRD9  
JLxdHpiZYglLlXyFjpS2TqLqakYP7St3nU4bxDVeMNG0OmJcDclWHSQNY8tmSSMz9yU25nVs3hgB  
35nks4M610QjVlei8xlrUJrrD8zR+/XM5KUSObPOMWHbN95Slys2+Z4t14xHD7U1N1Glmv5YjaKg  
zLYchLHVZVDDxv157ZZiGYRtYP+bFRQjQu55ebb5clDR5701+ymoB0YHdUX4YCaVsIl656Gur0n8  
aCawePwApureZEpDsJgjdeZd+pBb8B1GzKZybQ4aT/xlSTWOBqNlduKLjCj4lw/rm7kjLeeksOKO  
laYbS3I1q1bYzc4z1KbJkqNop1tBj6ZttDCWSZNZ4ds1s78/LE4cXbOheyglZCa/8IzaUvPeGQNH  
+ilo+G64ih9ZiXcLKQymDyUJpAlehC8kcFqNbHRhzRxxYPOvBqrZxNCzxBRVUV+SjU8NYqyjRTSC  
j3r9xJd7VjrGkTO1jId0GdikT+VqGbIzLFu4szFpUuQEVBcv7v0xRcQOVUW0zaNTC2NR/YAGMnK3

```

FLJkAKjy102KhIW3m+weYsCWhFVbZ9qTa5vwUdFGVWp2ntLLHXBFPiz//Q5JaJ3DCuVVpaErB8HI
nQU/TG7LKYq29rdkp4kx+yFSbQOxXOcjsZx1ylwvWT/WMc9yyp9bq5an6RR38n35tBh+za3NDEoo
hoF+KXTsogQTaCZMmvy+fLkoqhlvj/PmSpq16fhrm0c2BImnD1TKa3QzhLFwhsHnHQICqjp62bGy
XQZwsc6HzuiuSxf/pXf/v3VZ9KWnvHIOlgcvBwEclfar0wriAZpagK1EZJpxEiVUati5UXTij+jE
zqXqYRDItroce0gAL8HIQViQ8EN0VZGRHrXjMe0jrCPGkJOitGBI33coIioqXTjaNJ9YJ1VjYhem
5bmZfeFF7cdY2t8JV4oVvoZUJv65jlQ8HDQi/AlYl8XNqmoenNNKu+iSZXjunmYdBoTALFr7XEowv
+vOPfL4U4dmUjTppuKouTrspC4wwbZTxLFU4zW9dLqiPq8Ml+UzT/2stfQa2xGUKr6acAmTA4V0q
HNiXWBN0tBF1vFW0tQ2d58W47aGBEAmbYXhcJ94pgYLSrBpiVzXbg06A2JIZnsQ/1lYYxsZqKraT
n53juqqD12BuFj/J8c4NXzwNC/y3txPPdO29IxHloWFpUsG82uytjamxAg4mdT59UXBidZvmvL5H
FL5qI4ASm38AQ8LVk5tC5Y4KA8PbMkWCs+kKKl2kbM0t46uRfNm2udsMURWLTGZkf0dCSUdhoh3M
rstfLF9Ouc/7yzZvuPkjgq+1UL+poancGHphDVXqFWHazIBnjcY7pZt5+i272DASYlVldoQmawfD
h7fNhrKXnvWZCalbDRhdeM//MO12kVPtcvbyP+bp/MvxRylobo407VGk3ZKJapUJUK6M6oA+yUyf
xpfwKSZEI4Sxk2MgqIOmBbU2OySIQs6lLjJpLnCwqx615bLEIuo1xskmmvgM4yg5EuxVTXWVXdD
W9eH9e+Xd3mz00iMxEy/lmp1lTemct2Db+LV0oafPdrWJsyl/cO9frZJ59pW3rGI2sxHJ+6NL+gW
617yqgZDSdx1QMXGosq0pXl7LDZxZ81vVmldBdvrUnQejI/N5gRMosDb8fbJqoL5J62tu6y73D1j
eMu5xhFhd/Z8Shsp+GihSzqGM0En+NJLLw3GNODD99EA436tIm+sTdm8bUcCADZg7Wix1RoVeGDUq
9PJmuRz87K+fFD2PXC90cXZeXAYzoPrV0mVLuk6ALfe89dcs/DD+nnsUlnGjG149x0D5kZqkeM+
FlrbQ410gqlld190qnOwOMKxk5nlwZiCCexDrcDDdqmXdvMkQ7/s7ZGWWO8XPxbOATfKseF2uhIh
3MTnD43B/prA4c8sVqfdMuqisqCXSxW/z3br8gMO14/zQhnz189Kz/abRy2GQoRXXf+pw9/dPqsN
tY3ftfr+sPxSLVz8gGXTUGVHg+fLfJSWSVV1RyTwhhYn9uym2gc2JIdCctZxnoFy5J9hQFurjat9
O95kn6JZcLyJeuOmcZW22OJf854ZG+thAh5wuK8nHfB+cZnbNUCGzuknlZqaLbInWmTpbGxBA7lw
rYd4aCuS96blw7o3ILMbT6BQP7x2jKWALWHOxE7pBP5zD9+Xue3EQndNkrIlqlq05dqItjFjTb19
R+pGiScpu/q8ZlnZpKdvM70/tp4X7P2nsJYdE9MHjV4u4bAn8nTCDaITZ3URcoKjmx6HcqZ7rHJ4
1b4wlg2cc2HY3JdID7T63C9JOecdq8hmUFh5Xwuvmy5mzk5Eq6rGdxtLplDWAlE5K1fjhHo/xJgA
HavyI3ijlo8AAAAAE1FTkSuQmCC', height: 150, width: 180, top: 290, left: 50
}];

```

### Limitations of Image

The following features have some limitations in Image:

- Corner resizing option in the image element.
- Copy and paste the external image.

### Chart

A chart is a graphical representation of data, that organizes and represents a set of numerical or qualitative data. It mostly displays the selected range of data in terms of **x-axis** and **y-axis**. You can use the [allowChart](#) property to enable or disable the chart functionality.

\* The default value for the [allowChart](#) property is `true`.

### Types of chart

The following types of charts are available in the Spreadsheet.

- \* Column Chart
- \* Bar Chart
- \* Area Chart
- \* Line Chart
- \* Pie Chart
- \* Scatter Chart

### Insert Chart

You can insert the chart by using one of the following ways,

- Select the chart icon in the Ribbon toolbar under the Insert Tab.
- Use the [insertChart\(\)](#) method programmatically.

The available parameter in the [insertChart\(\)](#) method is,

Parameter	Type	Description
chart	<b>ChartModel</b>	Specifies the options to insert a chart in the spreadsheet.

The available arguments in the **ChartModel** are:

- type: Specifies the type of chart.
- theme: Specifies the theme of a chart.
- isSeriesInRows: Specifies to switch the row or a column.
- range: Specifies the selected range or specified range.
- id: Specifies the chart element id.
- markerSettings: Specifies the marker settings. The marker is used to provide information about the data points in the series and is currently only applicable to the line chart.

### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective, CellDirective, CellsDirective,
CellStyleModel, ChartModel } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective,
getFormatFromType, Spreadsheet } from '@syncfusion/ej2-react-spreadsheet';
import { chartData } from './datasource';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  const style = { backgroundColor: '#357cd2', color: '#fff', fontWeight:
'bold', textAlign: 'center', verticalAlign: 'middle' };
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ backgroundColor: '#357cd2', color:
'#fff', fontWeight: 'bold', textAlign: 'center' }, 'A3:F3');
      spreadsheet.numberFormat(getFormatFromType('Currency'),
'B4:F8');
      spreadsheet.merge('A1:F1');
      //Render Column chart
      spreadsheet.insertChart([
        { type: 'Column', theme: 'Bootstrap5Dark', range: 'A3:B6', id: 'column-chart' }
      ]);
      //Render Line chart with Marker
      spreadsheet.insertChart([
        { type: 'Line', range: 'A3:B6', markerSettings: { visible: true, shape: 'Circle', isFilled: false, size: 10, border: { width: 2, color: '#3cb371' } }, id: 'line-chart' }
      ]);
    }
  });
}
```

```

    }, []);
    return (
      <div>
        <SpreadsheetComponent ref={spreadsheetRef}>
          <SheetsDirective>
            <SheetDirective name='Book Sales'>
              <RowsDirective>
                <RowDirective height={30}>
                  <CellsDirective>
                    <CellDirective value='Book Sales 2016-
2020' style={style}></CellDirective>
                  </CellsDirective>
                </RowDirective>
                <RowDirective>
                  <CellsDirective>
                    <CellDirective
index={7}></CellDirective>
                  </CellsDirective>
                </RowDirective>
              </RowsDirective>
              <RangesDirective>
                <RangeDirective dataSource={chartData}
startCell='A3' ></RangeDirective>
              </RangesDirective>
              <ColumnsDirective>
                <ColumnDirective width={110}></ColumnDirective>
                <ColumnDirective width={100}></ColumnDirective>
                <ColumnDirective width={100}></ColumnDirective>
                <ColumnDirective width={100}></ColumnDirective>
                <ColumnDirective width={100}></ColumnDirective>
                <ColumnDirective width={100}></ColumnDirective>
              </ColumnsDirective>
            </SheetDirective>
          </SheetsDirective>
        </SpreadsheetComponent>
      </div>
    );
  };
  export default App;
  const root = createRoot(document.getElementById('root'));
  root.render(<App />);
  {% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective, CellDirective, CellsDirective,
CellStyleModel, ChartModel } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective,
getFormatFromType } from '@syncfusion/ej2-react-spreadsheet';
import { chartData } from './datasource';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);

```

```

const style: CellStyleModel = { backgroundColor: '#357cd2', color:
'#fff', fontWeight: 'bold', textAlign: 'center', verticalAlign: 'middle' };
React.useEffect(() => {
  let spreadsheet = spreadsheetRef.current;
  if (spreadsheet) {
    spreadsheet.cellFormat({ backgroundColor: '#357cd2', color:
'#fff', fontWeight: 'bold', textAlign: 'center' }, 'A3:F3');
    spreadsheet.numberFormat(getFormatFromType('Currency'),
'B4:F8');
    spreadsheet.merge('A1:F1');
    //Render Column chart
    spreadsheet.insertChart([{'type': 'Column', theme:
'Bootstrap5Dark', range: 'A3:B6', id: 'column-chart' }]);
    //Render Line chart with Marker
    spreadsheet.insertChart([{'type': 'Line', range: 'A3:B6',
markerSettings: {visible: true, shape: 'Circle', isFilled: false, size: 10,
border: {width: 2, color: '#3cb371'}}, id: 'line-chart' }]);
  }
}, []);
return (
  <div>
    <SpreadsheetComponent ref={spreadsheetRef}>
      <SheetsDirective>
        <SheetDirective name='Book Sales'>
          <RowsDirective>
            <RowDirective height={30}>
              <CellsDirective>
                <CellDirective value='Book Sales 2016-
2020' style={style}></CellDirective>
              </CellsDirective>
            </RowDirective>
            <RowDirective>
              <CellsDirective>
                <CellDirective
index={7}></CellDirective>
              </CellsDirective>
            </RowDirective>
          </RowsDirective>
          <RangesDirective>
            <RangeDirective dataSource={chartData}
startCell='A3' ></RangeDirective>
          </RangesDirective>
          <ColumnsDirective>
            <ColumnDirective width={110}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
          </ColumnsDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  </div>
);
};
export default App;

```



```
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}
```

### DATASOURCE.JSX

```
/**
 * Chart data source
 */
export let chartData = [
  { 'Book': 'Classics', 'Year 2016': 19033, 'Year 2017': 78453, 'Year
2018': 24354, 'Year 2019': 18757, 'Year 2020': 34343 },
  { 'Book': 'Mystery', 'Year 2016': 50400, 'Year 2017': 82311, 'Year
2018': 131003, 'Year 2019': 19899, 'Year 2020': 42200 },
  { 'Book': 'Romance', 'Year 2016': 18002, 'Year 2017': 49529, 'Year
2018': 79567, 'Year 2019': 12302, 'Year 2020': 21277 },
  { 'Book': 'Sci-Fi & Fantasy', 'Year 2016': 10033, 'Year 2017': 51200,
'Year 2018': 66211, 'Year 2019': 12899, 'Year 2020': 18779 },
  { 'Book': 'Horror', 'Year 2016': 23454, 'Year 2017': 78665, 'Year 2018':
81232, 'Year 2019': 19888, 'Year 2020': 20986 }
];
```

### DATASOURCE.TSX

```
/**
 * Chart data source
 */
export let chartData: Object[] = [
  { 'Book': 'Classics', 'Year 2016': 19033, 'Year 2017': 78453, 'Year
2018': 24354, 'Year 2019': 18757, 'Year 2020': 34343 },
  { 'Book': 'Mystery', 'Year 2016': 50400, 'Year 2017': 82311, 'Year
2018': 131003, 'Year 2019': 19899, 'Year 2020': 42200 },
  { 'Book': 'Romance', 'Year 2016': 18002, 'Year 2017': 49529, 'Year
2018': 79567, 'Year 2019': 12302, 'Year 2020': 21277 },
  { 'Book': 'Sci-Fi & Fantasy', 'Year 2016': 10033, 'Year 2017': 51200,
'Year 2018': 66211, 'Year 2019': 12899, 'Year 2020': 18779 },
  { 'Book': 'Horror', 'Year 2016': 23454, 'Year 2017': 78665, 'Year 2018':
81232, 'Year 2019': 19888, 'Year 2020': 20986 }
];
```

### Delete Chart

- If you want to delete the chart, just select the chart, and then press the Delete key.
- Use the [deleteChart\(\)](#) method programmatically.

The available parameter in the [deleteChart\(\)](#) method is,

Parameter	Type	Description
id	string	Specifies the id of the chart element to be deleted.

*Chart Customization*

Chart feature allows you to view and insert a chart in a spreadsheet, and you can change the height and width of the chart by resizing and moving it to another position.

- You can change the height and width of the chart by resizing.
- You can change the position of the chart by drag and drop.

**APP.JSX**

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective, CellDirective, CellsDirective
} from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective,
getFormatFromType } from '@syncfusion/ej2-react-spreadsheet';
import { chartData } from './datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  const style = { backgroundColor: '#357cd2', color: '#fff', fontWeight:
'bold', textAlign: 'center', verticalAlign: 'middle' };
  const chart = [{ type: 'Column', range: 'A3:F8' }];
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ backgroundColor: '#357cd2', color:
'#fff', fontWeight: 'bold', textAlign: 'center' }, 'A3:F3');
      spreadsheet.numberFormat(getFormatFromType('Currency'),
'B4:F8');
      spreadsheet.merge('A1:F1');
    }
  }, []);
  return (
    <div>
      <SpreadsheetComponent ref={spreadsheetRef} >
        <SheetsDirective>
          <SheetDirective name='Book Sales'>
            <RowsDirective>
              <RowDirective height={30}>
                <CellsDirective>
                  <CellDirective value='Book Sales 2016-
2020' style={style}></CellDirective>
                </CellsDirective>
              </RowDirective>
              <RowDirective>
                <CellsDirective>
                  <CellDirective index={7}
chart={chart}></CellDirective>
                </CellsDirective>
              </RowDirective>
            </RowsDirective>
            <RangesDirective>
              <RangeDirective dataSource={chartData}
startCell='A3' ></RangeDirective>
            </RangesDirective>
          </SheetDirective>
        </SheetsDirective>
      </SpreadsheetComponent>
    </div>
  );
}
```

```

        <ColumnsDirective>
            <ColumnDirective width={110}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
        </ColumnsDirective>
    </SheetDirective>
</SheetsDirective>
</SpreadsheetComponent>
</div>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective, CellDirective, CellsDirective,
CellStyleModel, ChartModel } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective,
getFormatFromType } from '@syncfusion/ej2-react-spreadsheet';
import { chartData } from './datasource';
function App() {
    const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
    const style: CellStyleModel = { backgroundColor: '#357cd2', color:
'#fff', fontWeight: 'bold', textAlign: 'center', verticalAlign: 'middle' };
    const chart: ChartModel[] = [{ type: 'Column', range: 'A3:F8' }];
    React.useEffect(() => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            spreadsheet.cellFormat({ backgroundColor: '#357cd2', color:
'#fff', fontWeight: 'bold', textAlign: 'center' }, 'A3:F3');
            spreadsheet.numberFormat(getFormatFromType('Currency'),
'B4:F8');
            spreadsheet.merge('A1:F1');
        }
    }, []);
    return (
        <div>
            <SpreadsheetComponent ref={spreadsheetRef}>
                <SheetsDirective>
                    <SheetDirective name='Book Sales'>
                        <RowsDirective>
                            <RowDirective height={30}>
                                <CellsDirective>
                                    <CellDirective value='Book Sales 2016-
2020' style={style}></CellDirective>
                                </CellsDirective>

```

```

        </RowDirective>
        <RowDirective>
            <CellsDirective>
                <CellDirective index={7}>
chart={chart}></CellDirective>
            </CellsDirective>
        </RowDirective>
    </RowsDirective>
    <RangesDirective>
        <RangeDirective dataSource={chartData}
startCell='A3' ></RangeDirective>
    </RangesDirective>
    <ColumnsDirective>
        <ColumnDirective width={110}></ColumnDirective>
        <ColumnDirective width={100}></ColumnDirective>
        <ColumnDirective width={100}></ColumnDirective>
        <ColumnDirective width={100}></ColumnDirective>
        <ColumnDirective width={100}></ColumnDirective>
        <ColumnDirective width={100}></ColumnDirective>
    </ColumnsDirective>
    </SheetDirective>
    </SheetsDirective>
    </SpreadsheetComponent>
</div>
);
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

### DATASOURCE.JSX

```

/**
 * Chart data source
 */
export let chartData = [
    { 'Book': 'Classics', 'Year 2016': 19033, 'Year 2017': 78453, 'Year
2018': 24354, 'Year 2019': 18757, 'Year 2020': 34343 },
    { 'Book': 'Mystery', 'Year 2016': 50400, 'Year 2017': 82311, 'Year
2018': 131003, 'Year 2019': 19899, 'Year 2020': 42200 },
    { 'Book': 'Romance', 'Year 2016': 18002, 'Year 2017': 49529, 'Year
2018': 79567, 'Year 2019': 12302, 'Year 2020': 21277 },
    { 'Book': 'Sci-Fi & Fantasy', 'Year 2016': 10033, 'Year 2017': 51200,
'Year 2018': 66211, 'Year 2019': 12899, 'Year 2020': 18779 },
    { 'Book': 'Horror', 'Year 2016': 23454, 'Year 2017': 78665, 'Year 2018':
81232, 'Year 2019': 19888, 'Year 2020': 20986 }
];

```

### DATASOURCE.TSX

```

/**
 * Chart data source
 */
export let chartData: Object[] = [

```

```

    { 'Book': 'Classics', 'Year 2016': 19033, 'Year 2017': 78453, 'Year
2018': 24354, 'Year 2019': 18757, 'Year 2020': 34343 },
    { 'Book': 'Mystery', 'Year 2016': 50400, 'Year 2017': 82311, 'Year
2018': 131003, 'Year 2019': 19899, 'Year 2020': 42200 },
    { 'Book': 'Romance', 'Year 2016': 18002, 'Year 2017': 49529, 'Year
2018': 79567, 'Year 2019': 12302, 'Year 2020': 21277 },
    { 'Book': 'Sci-Fi & Fantasy', 'Year 2016': 10033, 'Year 2017': 51200,
'Year 2018': 66211, 'Year 2019': 12899, 'Year 2020': 18779 },
    { 'Book': 'Horror', 'Year 2016': 23454, 'Year 2017': 78665, 'Year 2018':
81232, 'Year 2019': 19888, 'Year 2020': 20986 }
  ];

```

### Customization of line chart markers

Using the [actionBegin](#) event, you can change the shape, size, fill color, and border of the line chart marker. In the following example, you can see the modified marker appearance, such as shape and size, while creating the line chart with UI interaction.

### APP.JSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective, CellDirective, CellsDirective,
CellStyleModel, ChartModel } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective,
getFormatFromType, Spreadsheet } from '@syncfusion/ej2-react-spreadsheet';
import { chartData } from './datasource';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  const style = { backgroundColor: '#357cd2', color: '#fff', fontWeight:
'bold', textAlign: 'center', verticalAlign: 'middle' };
  const onActionBegin = (args) => {
    if (args.action === 'beforeInsertChart' &&
args.args.eventArgs.type.includes('Line')) {
      args.args.eventArgs.markerSettings.shape = 'Triangle';
      args.args.eventArgs.markerSettings.isFilled = false;
      args.args.eventArgs.markerSettings.size = 10;
    }
  }
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ backgroundColor: '#357cd2', color:
'#fff', fontWeight: 'bold', textAlign: 'center' }, 'A3:F3');
      spreadsheet.numberFormat(getFormatFromType('Currency'),
'B4:F8');
      spreadsheet.merge('A1:F1');
    }
  }, []);
  return (
    <div>
      <SpreadsheetComponent ref={spreadsheetRef}
actionBegin={onActionBegin}>
        <SheetsDirective>

```

```

        <SheetDirective name='Book Sales'>
          <RowsDirective>
            <RowDirective height={30}>
              <CellsDirective>
                <CellDirective value='Book Sales 2016-
2020' style={style}></CellDirective>
              </CellsDirective>
            </RowDirective>
            <RowDirective>
              <CellsDirective>
                <CellDirective
index={7}></CellDirective>
              </CellsDirective>
            </RowDirective>
          </RowsDirective>
          <RangesDirective>
            <RangeDirective dataSource={chartData}
startCell='A3' ></RangeDirective>
          </RangesDirective>
          <ColumnsDirective>
            <ColumnDirective width={110}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
          </ColumnsDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  </div>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

## APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective, CellDirective, CellsDirective,
CellStyleModel, ChartModel } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective,
getFormatFromType, Spreadsheet, BeforeChartEventArgs } from '@syncfusion/ej2-
react-spreadsheet';
import { chartData } from '../datasource';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  const style: CellStyleModel = { backgroundColor: '#357cd2', color:
'#fff', fontWeight: 'bold', textAlign: 'center', verticalAlign: 'middle' };
  const onActionBegin = (args: BeforeChartEventArgs) : void => {

```

```

        if (args.action === 'beforeInsertChart' &&
args.args.eventArgs.type.includes('Line')) {
            args.args.eventArgs.markerSettings.shape = 'Triangle';
            args.args.eventArgs.markerSettings.isFilled = false;
            args.args.eventArgs.markerSettings.size = 10;
        }
    }
    React.useEffect(() => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            spreadsheet.cellFormat({ backgroundColor: '#357cd2', color:
'#fff', fontWeight: 'bold', textAlign: 'center' }, 'A3:F3');
            spreadsheet.numberFormat(getFormatFromType('Currency'),
'B4:F8');
            spreadsheet.merge('A1:F1');
        }
    }, []);
    return (
        <div>
            <SpreadsheetComponent ref={spreadsheetRef}
actionBegin={onActionBegin}>
                <SheetsDirective>
                    <SheetDirective name='Book Sales'>
                        <RowsDirective>
                            <RowDirective height={30}>
                                <CellsDirective>
                                    <CellDirective value='Book Sales 2016-
2020' style={style}></CellDirective>
                                </CellsDirective>
                            </RowDirective>
                            <RowDirective>
                                <CellsDirective>
                                    <CellDirective
index={7}></CellDirective>
                                </CellsDirective>
                            </RowDirective>
                        </RowsDirective>
                        <RangesDirective>
                            <RangeDirective dataSource={chartData}
startCell='A3' ></RangeDirective>
                        </RangesDirective>
                        <ColumnsDirective>
                            <ColumnDirective width={110}></ColumnDirective>
                            <ColumnDirective width={100}></ColumnDirective>
                            <ColumnDirective width={100}></ColumnDirective>
                            <ColumnDirective width={100}></ColumnDirective>
                            <ColumnDirective width={100}></ColumnDirective>
                            <ColumnDirective width={100}></ColumnDirective>
                        </ColumnsDirective>
                    </SheetDirective>
                </SheetsDirective>
            </SpreadsheetComponent>
        </div>
    );
};
export default App;
const root = createRoot(document.getElementById('root')!);

```

```
root.render(<App />);
{% endraw %}
```

### DATASOURCE.JSX

```
/**
 * Chart data source
 */
export let chartData = [
  { 'Book': 'Classics', 'Year 2016': 19033, 'Year 2017': 78453, 'Year 2018': 24354, 'Year 2019': 18757, 'Year 2020': 34343 },
  { 'Book': 'Mystery', 'Year 2016': 50400, 'Year 2017': 82311, 'Year 2018': 131003, 'Year 2019': 19899, 'Year 2020': 42200 },
  { 'Book': 'Romance', 'Year 2016': 18002, 'Year 2017': 49529, 'Year 2018': 79567, 'Year 2019': 12302, 'Year 2020': 21277 },
  { 'Book': 'Sci-Fi & Fantasy', 'Year 2016': 10033, 'Year 2017': 51200, 'Year 2018': 66211, 'Year 2019': 12899, 'Year 2020': 18779 },
  { 'Book': 'Horror', 'Year 2016': 23454, 'Year 2017': 78665, 'Year 2018': 81232, 'Year 2019': 19888, 'Year 2020': 20986 }
];
```

### DATASOURCE.TSX

```
/**
 * Chart data source
 */
export let chartData: Object[] = [
  { 'Book': 'Classics', 'Year 2016': 19033, 'Year 2017': 78453, 'Year 2018': 24354, 'Year 2019': 18757, 'Year 2020': 34343 },
  { 'Book': 'Mystery', 'Year 2016': 50400, 'Year 2017': 82311, 'Year 2018': 131003, 'Year 2019': 19899, 'Year 2020': 42200 },
  { 'Book': 'Romance', 'Year 2016': 18002, 'Year 2017': 49529, 'Year 2018': 79567, 'Year 2019': 12302, 'Year 2020': 21277 },
  { 'Book': 'Sci-Fi & Fantasy', 'Year 2016': 10033, 'Year 2017': 51200, 'Year 2018': 66211, 'Year 2019': 12899, 'Year 2020': 18779 },
  { 'Book': 'Horror', 'Year 2016': 23454, 'Year 2017': 78665, 'Year 2018': 81232, 'Year 2019': 19888, 'Year 2020': 20986 }
];
```

### Limitations of Chart

The following features have some limitations in the Chart:

- Insert row/delete row between the chart data source will not reflect the chart.
- Copy/paste into the chart data source will not reflect the chart.
- Corner resizing option in chart element.

### Note

You can refer to our [React Spreadsheet](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Spreadsheet example](#) to know how to present and manipulate data.



## See Also

- [Formatting](#)
- [Rows and columns](#)
- [Hyperlink](#)
- [Sorting](#)

## Rows and columns in React Spreadsheet component

Spreadsheet is a tabular format consisting of rows and columns. The intersection point of rows and columns are called as cells. The list of operations that you can perform in rows and columns are,

- Insert
- Delete
- Show and Hide

## Insert

You can insert rows or columns anywhere in a spreadsheet. Use the [allowInsert](#) property to enable or disable the insert option in Spreadsheet.

## Row

The rows can be inserted in the following ways,

- Using [insertRow](#) method, you can insert the rows once the component is loaded.
- Using context menu, insert the empty rows in the desired position.

The following code example shows the options for inserting rows in the spreadsheet.

**APP.JSX**

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
    const spreadsheetRef = React.useRef(null);
    const rowsModel = [{
        index: 9, // Need to specify the index for the first row collection,
        the specified rows will be inserted in this index.
        cells: [{ value: '' }, { value: '8' }, { value: 'Northwoods
Cranberry Sauce' }, { value: '3' }, { value: '12 - 12 oz jars' },
{ value: '40.00' }, { value: '6' }, { value: 'false' }]
    },
    {
        cells: [{ value: '' }, { value: '9' }, { value: 'Mishi Kobe Niku' },
{ value: '4' }, { value: '18 - 500 g pkgs.' }, { value: '97.00' }, { value:
'29' }, { value: 'true' }]
    }
    ];
    const onCreate = () => {
```

```

    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
        // Applies style formatting before inserting the rows
        spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
    }, 'B1:H1');
        // inserting a empty row at 0th index
        spreadsheet.insertRow();
        // inserting 2 rows at the 9th index with data
        spreadsheet.insertRow(rowsModel);
        // Applies style formatting after the rows are inserted
        spreadsheet.cellFormat({ textAlign: 'center' }, 'B3:B12');
        spreadsheet.cellFormat({ textAlign: 'center' }, 'D3:D12');
        spreadsheet.cellFormat({ textAlign: 'center' }, 'F3:H12');
    }
};
return (
    <div>
        <SpreadsheetComponent ref={spreadsheetRef} created={onCreated}
showSheetTabs={false} showRibbon={false} showFormulaBar={false} >
            <SheetsDirective>
                <SheetDirective>
                    <RangesDirective>
                        <RangeDirective dataSource={data}
startCell={"B1"}></RangeDirective>
                    </RangesDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={20}></ColumnDirective>
                        <ColumnDirective width={90}></ColumnDirective>
                        <ColumnDirective width={220}></ColumnDirective>
                        <ColumnDirective width={90}></ColumnDirective>
                        <ColumnDirective width={140}></ColumnDirective>
                        <ColumnDirective width={90}></ColumnDirective>
                        <ColumnDirective width={100}></ColumnDirective>
                        <ColumnDirective width={100}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
            </SheetsDirective>
        </SpreadsheetComponent>
    </div>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowModel } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { data } from '../datasource';

```

```

function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  const rowsModel: RowModel[] = [{
    index: 9, // Need to specify the index for the first row collection,
the specified rows will be inserted in this index.
    cells: [{ value: '' }, { value: '8' }, { value: 'Northwoods'
Cranberry Sauce' }, { value: '3' }, { value: '12 - 12 oz jars' },
    { value: '40.00' }, { value: '6' }, { value: 'false' }]
  },
  {
    cells: [{ value: '' }, { value: '9' }, { value: 'Mishi Kobe Niku' },
    { value: '4' }, { value: '18 - 500 g pkgs.' }, { value: '97.00' }, { value:
    '29' }, { value: 'true' }]
  }
  ];
  const onCreate = (): void => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      // Applies style formatting before inserting the rows
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'B1:H1');
      // inserting a empty row at 0th index
      spreadsheet.insertRow();
      // inserting 2 rows at the 9th index with data
      spreadsheet.insertRow(rowsModel);
      // Applies style formatting after the rows are inserted
      spreadsheet.cellFormat({ textAlign: 'center' }, 'B3:B12');
      spreadsheet.cellFormat({ textAlign: 'center' }, 'D3:D12');
      spreadsheet.cellFormat({ textAlign: 'center' }, 'F3:H12');
    }
  };
  return (
    <div>
      <SpreadsheetComponent ref={spreadsheetRef} created={onCreate}
showSheetTabs={false} showRibbon={false} showFormulaBar={false} >
        <SheetsDirective>
          <SheetDirective>
            <RangesDirective>
              <RangeDirective dataSource={data}
startCell={"B1"}></RangeDirective>
            </RangesDirective>
            <ColumnsDirective>
              <ColumnDirective width={20}></ColumnDirective>
              <ColumnDirective width={90}></ColumnDirective>
              <ColumnDirective width={220}></ColumnDirective>
              <ColumnDirective width={90}></ColumnDirective>
              <ColumnDirective width={140}></ColumnDirective>
              <ColumnDirective width={90}></ColumnDirective>
              <ColumnDirective width={100}></ColumnDirective>
              <ColumnDirective width={100}></ColumnDirective>
            </ColumnsDirective>
          </SheetDirective>
        </SheetsDirective>
      </SpreadsheetComponent>
    </div>
  );
};
export default App;

```

```
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}
```

### DATASOURCE.JSX

```
/**
 * Insert row data source
 */
export let data = [
  {
    'Product Id': 1, 'Product Name': 'Chai', 'Supplier Id': 1, 'Quantity
Per Unit': '10 boxes x 20 bags', 'Unit Price': 18.00, 'Units In Stock': 39,
'Discontinued': false
  },
  {
    'Product Id': 2, 'Product Name': 'Chang', 'Supplier Id': 1,
'Quantity Per Unit': '24 - 12 oz bottles', 'Unit Price': 19.00, 'Units In
Stock': 17, 'Discontinued': true
  },
  {
    'Product Id': 3, 'Product Name': 'Aniseed Syrup', 'Supplier Id': 1,
'Quantity Per Unit': '12 - 550 ml bottles', 'Unit Price': 10.00, 'Units In
Stock': 13, 'Discontinued': false
  },
  {
    'Product Id': 4, 'Product Name': 'Chef Anton\'s Cajun Seasoning',
'Supplier Id': 2, 'Quantity Per Unit': '48 - 6 oz jars', 'Unit Price':
22.00, 'Units In Stock': 53, 'Discontinued': true
  },
  {
    'Product Id': 5, 'Product Name': 'Chef Anton\'s Gumbo Mix',
'Supplier Id': 2, 'Quantity Per Unit': '36 boxes', 'Unit Price': 21.35,
'Units In Stock': 0, 'Discontinued': true
  },
  {
    'Product Id': 6, 'Product Name': 'Grandma\'s Boysenberry Spread',
'Supplier Id': 3, 'Quantity Per Unit': '12 - 8 oz jars', 'Unit Price':
25.00, 'Units In Stock': 120, 'Discontinued': false
  },
  {
    'Product Id': 7, 'Product Name': 'Uncle Bob\'s Organic Dried Pears',
'Supplier Id': 3, 'Quantity Per Unit': '12 - 1 lb pkgs.', 'Unit Price':
30.00, 'Units In Stock': 15, 'Discontinued': true
  },
  {
    'Product Id': 10, 'Product Name': 'Queso Cabrales', 'Supplier Id':
5, 'Quantity Per Unit': '1 kg pkg.', 'Unit Price': 21.00, 'Units In Stock':
22, 'Discontinued': false
  }
];
```

### DATASOURCE.TSX

```
/**
 * Insert row data source
```

```

*/
export let data: Object[] = [
  {
    'Product Id': 1, 'Product Name': 'Chai', 'Supplier Id': 1, 'Quantity
Per Unit': '10 boxes x 20 bags', 'Unit Price': 18.00, 'Units In Stock': 39,
'Discontinued': false
  },
  {
    'Product Id': 2, 'Product Name': 'Chang', 'Supplier Id': 1,
'Quantity Per Unit': '24 - 12 oz bottles', 'Unit Price': 19.00, 'Units In
Stock': 17, 'Discontinued': true
  },
  {
    'Product Id': 3, 'Product Name': 'Aniseed Syrup', 'Supplier Id': 1,
'Quantity Per Unit': '12 - 550 ml bottles', 'Unit Price': 10.00, 'Units In
Stock': 13, 'Discontinued': false
  },
  {
    'Product Id': 4, 'Product Name': 'Chef Anton\'s Cajun Seasoning',
'Supplier Id': 2, 'Quantity Per Unit': '48 - 6 oz jars', 'Unit Price':
22.00, 'Units In Stock': 53, 'Discontinued': true
  },
  {
    'Product Id': 5, 'Product Name': 'Chef Anton\'s Gumbo Mix',
'Supplier Id': 2, 'Quantity Per Unit': '36 boxes', 'Unit Price': 21.35,
'Units In Stock': 0, 'Discontinued': true
  },
  {
    'Product Id': 6, 'Product Name': 'Grandma\'s Boysenberry Spread',
'Supplier Id': 3, 'Quantity Per Unit': '12 - 8 oz jars', 'Unit Price':
25.00, 'Units In Stock': 120, 'Discontinued': false
  },
  {
    'Product Id': 7, 'Product Name': 'Uncle Bob\'s Organic Dried Pears',
'Supplier Id': 3, 'Quantity Per Unit': '12 - 1 lb pkgs.', 'Unit Price':
30.00, 'Units In Stock': 15, 'Discontinued': true
  },
  {
    'Product Id': 10, 'Product Name': 'Queso Cabrales', 'Supplier Id':
5, 'Quantity Per Unit': '1 kg pkg.', 'Unit Price': 21.00, 'Units In Stock':
22, 'Discontinued': false
  }
];

```

### Column

The columns can be inserted in the following ways,

- Using [insertColumn](#) method, you can insert the columns once the component is loaded.
- Using context menu, insert the empty columns in the desired position.

The following code example shows the options for inserting columns in the spreadsheet.

### APP.JSX

```

{% raw %}
import * as React from 'react';

```

```

import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, getCellAddress } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { data } from '../datasource';
function App() {
    const spreadsheetRef = React.useRef(null);
    const cellsModel = [{ value: 'Unit Price', style: { fontWeight: 'bold',
textAlign: 'center' } }, { value: '18.00' },
    { value: '19.00' }, { value: '10.00' }, { value: '22.00' }, { value:
'21.35' }, { value: '25.00' }, { value: '30.00' },
    { value: '21.00' }, { value: '40.00' }, { value: '97.00' }];
    const onCreated = () => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            // Applies style formatting before inserting the column
            spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A2:G2');
            // inserting a empty column at 0th index
            spreadsheet.insertColumn();
            // inserting 1 column at the 5th index with column model
            spreadsheet.insertColumn([{ index: 5, width: 90 }]);
            let rowIndex = 1;
            // Updating the 5th column data
            cellsModel.forEach((cell) => {
                if (spreadsheet) {
                    spreadsheet.updateCell(cell, getCellAddress(rowIndex,
5)); rowIndex++;
                }
            });
            // Applies style formatting after the columns are inserted
            spreadsheet.cellFormat({ textAlign: 'center' }, 'B3:B12');
            spreadsheet.cellFormat({ textAlign: 'center' }, 'D3:D12');
            spreadsheet.cellFormat({ textAlign: 'center' }, 'F3:H12');
        }
    };
    return (
        <div>
            <SpreadsheetComponent ref={spreadsheetRef} created={onCreated}
showSheetTabs={false} showRibbon={false} showFormulaBar={false} >
                <SheetsDirective>
                    <SheetDirective>
                        <RangesDirective>
                            <RangeDirective dataSource={data}
startCell={"A2"}></RangeDirective>
                        </RangesDirective>
                        <ColumnsDirective>
                            <ColumnDirective width={90}></ColumnDirective>
                            <ColumnDirective width={220}></ColumnDirective>
                            <ColumnDirective width={90}></ColumnDirective>
                            <ColumnDirective width={140}></ColumnDirective>
                            <ColumnDirective width={90}></ColumnDirective>
                            <ColumnDirective width={100}></ColumnDirective>
                            <ColumnDirective width={100}></ColumnDirective>
                        </ColumnsDirective>
                    </SheetDirective>

```

```

        </SheetsDirective>
      </SpreadsheetComponent>
    </div>
  );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, CellModel, getCellAddress } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  const cellsModel: CellModel[] = [{ value: 'Unit Price', style: {
fontWeight: 'bold', textAlign: 'center' } }, { value: '18.00' },
{ value: '19.00' }, { value: '10.00' }, { value: '22.00' }, { value:
'21.35' }, { value: '25.00' }, { value: '30.00' },
{ value: '21.00' }, { value: '40.00' }, { value: '97.00' }];
  const onCreated = (): void => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      // Applies style formatting before inserting the column
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A2:G2');
      // inserting a empty column at 0th index
      spreadsheet.insertColumn();
      // inserting 1 column at the 5th index with column model
      spreadsheet.insertColumn([{ index: 5, width: 90 }]);
      let rowIndex = 1;
      // Updating the 5th column data
      cellsModel.forEach((cell: CellModel) => {
        if (spreadsheet) {
          spreadsheet.updateCell(cell, getCellAddress(rowIndex,
5)); rowIndex++;
        }
      });
      // Applies style formatting after the columns are inserted
      spreadsheet.cellFormat({ textAlign: 'center' }, 'B3:B12');
      spreadsheet.cellFormat({ textAlign: 'center' }, 'D3:D12');
      spreadsheet.cellFormat({ textAlign: 'center' }, 'F3:H12');
    }
  };
  return (
    <div>
      <SpreadsheetComponent ref={spreadsheetRef} created={onCreated}
showSheetTabs={false} showRibbon={false} showFormulaBar={false} >

```

```

        <SheetsDirective>
          <SheetDirective>
            <RangesDirective>
              <RangeDirective dataSource={data}
startCell={"A2"}></RangeDirective>
            </RangesDirective>
            <ColumnsDirective>
              <ColumnDirective width={90}></ColumnDirective>
              <ColumnDirective width={220}></ColumnDirective>
              <ColumnDirective width={90}></ColumnDirective>
              <ColumnDirective width={140}></ColumnDirective>
              <ColumnDirective width={90}></ColumnDirective>
              <ColumnDirective width={100}></ColumnDirective>
              <ColumnDirective width={100}></ColumnDirective>
            </ColumnsDirective>
          </SheetDirective>
        </SheetsDirective>
      </SpreadsheetComponent>
    </div>
  );
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

### DATASOURCE.JSX

```

/**
 * Insert column data source
 */
export let data = [
  {
    'Product Id': 1, 'Product Name': 'Chai', 'Supplier Id': 1, 'Quantity
Per Unit': '10 boxes x 20 bags', 'Units In Stock': 39, 'Discontinued': true
  },
  {
    'Product Id': 2, 'Product Name': 'Chang', 'Supplier Id': 1,
'Quantity Per Unit': '24 - 12 oz bottles', 'Units In Stock': 17,
'Discontinued': true
  },
  {
    'Product Id': 3, 'Product Name': 'Aniseed Syrup', 'Supplier Id': 1,
'Quantity Per Unit': '12 - 550 ml bottles', 'Units In Stock': 13,
'Discontinued': true
  },
  {
    'Product Id': 4, 'Product Name': 'Chef Anton\'s Cajun Seasoning',
'Supplier Id': 2, 'Quantity Per Unit': '48 - 6 oz jars', 'Units In Stock':
53, 'Discontinued': true
  },
  {
    'Product Id': 5, 'Product Name': 'Chef Anton\'s Gumbo Mix',
'Supplier Id': 2, 'Quantity Per Unit': '36 boxes', 'Units In Stock': 0,
'Discontinued': true
  },
]

```



```

    {
      'Product Id': 6, 'Product Name': 'Grandma\'s Boysenberry Spread',
      'Supplier Id': 3, 'Quantity Per Unit': '12 - 8 oz jars', 'Units In Stock':
      120, 'Discontinued': false
    },
    {
      'Product Id': 7, 'Product Name': 'Uncle Bob\'s Organic Dried Pears',
      'Supplier Id': 3, 'Quantity Per Unit': '12 - 1 lb pkgs.', 'Units In Stock':
      15, 'Discontinued': false
    },
    {
      'Product Id': 8, 'Product Name': 'Queso Cabrales', 'Supplier Id': 5,
      'Quantity Per Unit': '1 kg pkg.', 'Units In Stock': 22, 'Discontinued':
      false
    },
    {
      'Product Id': 9, 'Product Name': 'Northwoods Cranberry Sauce',
      'Supplier Id': 3, 'Quantity Per Unit': '12 - 12 oz jars', 'Units In Stock':
      6, 'Discontinued': false
    },
    {
      'Product Id': 10, 'Product Name': 'Mishi Kobe Niku', 'Supplier Id':
      4, 'Quantity Per Unit': '18 - 500 g pkgs.', 'Units In Stock': 29,
      'Discontinued': true
    }
  ];

```

#### DATASOURCE.TSX

```

/**
 * Insert column data source
 */
export let data: Object[] = [
  {
    'Product Id': 1, 'Product Name': 'Chai', 'Supplier Id': 1, 'Quantity
    Per Unit': '10 boxes x 20 bags', 'Units In Stock': 39, 'Discontinued': true
  },
  {
    'Product Id': 2, 'Product Name': 'Chang', 'Supplier Id': 1,
    'Quantity Per Unit': '24 - 12 oz bottles', 'Units In Stock': 17,
    'Discontinued': true
  },
  {
    'Product Id': 3, 'Product Name': 'Aniseed Syrup', 'Supplier Id': 1,
    'Quantity Per Unit': '12 - 550 ml bottles', 'Units In Stock': 13,
    'Discontinued': true
  },
  {
    'Product Id': 4, 'Product Name': 'Chef Anton\'s Cajun Seasoning',
    'Supplier Id': 2, 'Quantity Per Unit': '48 - 6 oz jars', 'Units In Stock':
    53, 'Discontinued': true
  },
  {
    'Product Id': 5, 'Product Name': 'Chef Anton\'s Gumbo Mix',
    'Supplier Id': 2, 'Quantity Per Unit': '36 boxes', 'Units In Stock': 0,
    'Discontinued': true
  }
];

```

```

    },
    {
      'Product Id': 6, 'Product Name': 'Grandma\'s Boysenberry Spread',
      'Supplier Id': 3, 'Quantity Per Unit': '12 - 8 oz jars', 'Units In Stock':
      120, 'Discontinued': false
    },
    {
      'Product Id': 7, 'Product Name': 'Uncle Bob\'s Organic Dried Pears',
      'Supplier Id': 3, 'Quantity Per Unit': '12 - 1 lb pkgs.', 'Units In Stock':
      15, 'Discontinued': false
    },
    {
      'Product Id': 8, 'Product Name': 'Queso Cabrales', 'Supplier Id': 5,
      'Quantity Per Unit': '1 kg pkg.', 'Units In Stock': 22, 'Discontinued':
      false
    },
    {
      'Product Id': 9, 'Product Name': 'Northwoods Cranberry Sauce',
      'Supplier Id': 3, 'Quantity Per Unit': '12 - 12 oz jars', 'Units In Stock':
      6, 'Discontinued': false
    },
    {
      'Product Id': 10, 'Product Name': 'Mishi Kobe Niku', 'Supplier Id':
      4, 'Quantity Per Unit': '18 - 500 g pkgs.', 'Units In Stock': 29,
      'Discontinued': true
    }
  ]];

```

### Delete

Delete support provides an option for deleting the rows and columns in the spreadsheet. Use [allowDelete](#) property to enable or disable the delete option in Spreadsheet.

The rows and columns can be deleted dynamically in the following ways,

- Using [delete](#) method, you can delete the loaded rows and columns.
- Using context menu, you can delete the selected rows and columns.

The following code example shows the delete operation of rows and columns in the spreadsheet.

### APP.JSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  const onCreated = () => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
    }, 'A1:H1');
    }
  };

```

```

        // deleting the rows from 8th to 10th index. To delete row, the
        // third argument of enum type is passed as 'Row', the last argument specifies
        // the sheet name or index in which the delete operation will perform. By
        // default, active sheet will be considered. It is applicable only for model
        // type Row and Column.
        spreadsheet.delete(8, 10, 'Row', 0); // startIndex, endIndex,
        // Row, sheet index
        // deleting the 2nd and 5th indexed columns
        spreadsheet.delete(2, 2, 'Column', 'Sheet2');
        spreadsheet.delete(5, 5, 'Column');
        spreadsheet.delete(0, 0, "Sheet"); // delete the first sheet.
        // sheet index starts from 0
        // Applies style formatting after deleted the rows and columns
        spreadsheet.cellFormat({ textAlign: 'center' }, 'A2:A8');
        spreadsheet.cellFormat({ textAlign: 'center' }, 'D2:G8');
    }
};
return (
    <div>
        <SpreadsheetComponent ref={spreadsheetRef} created={onCreated}
        showRibbon={false} showFormulaBar={false} allowDelete={true} >
            <SheetsDirective>
                <SheetDirective name={"Sheet1"}>
                    <RangesDirective>
                        <RangeDirective
dataSource={data}></RangeDirective>
                    </RangesDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={90}></ColumnDirective>
                        <ColumnDirective width={220}></ColumnDirective>
                        <ColumnDirective width={90}></ColumnDirective>
                        <ColumnDirective width={140}></ColumnDirective>
                        <ColumnDirective width={90}></ColumnDirective>
                        <ColumnDirective width={100}></ColumnDirective>
                        <ColumnDirective width={100}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
                <SheetDirective name={"Sheet2"}>
                    <RangesDirective>
                        <RangeDirective
dataSource={data}></RangeDirective>
                    </RangesDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={90}></ColumnDirective>
                        <ColumnDirective width={220}></ColumnDirective>
                        <ColumnDirective width={90}></ColumnDirective>
                        <ColumnDirective width={140}></ColumnDirective>
                        <ColumnDirective width={90}></ColumnDirective>
                        <ColumnDirective width={100}></ColumnDirective>
                        <ColumnDirective width={100}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
            </SheetsDirective>
        </SpreadsheetComponent>
    </div>
);
};

```

```
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}
```

### APP.TSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  const onCreated = (): void => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A1:H1');
      // deleting the rows from 8th to 10th index. To delete row, the
third argument of enum type is passed as 'Row', the last argument specifies
the sheet name or index in which the delete operation will perform. By
default, active sheet will be considered. It is applicable only for model
type Row and Column.
      spreadsheet.delete(8, 10, 'Row', 0); // startIndex, endIndex,
Row, sheet index
      // deleting the 2nd and 5th indexed columns
      spreadsheet.delete(2, 2, 'Column', 'Sheet2');
      spreadsheet.delete(5, 5, 'Column');
      spreadsheet.delete(0, 0, "Sheet"); // delete the first sheet.
sheet index starts from 0
      // Applies style formatting after deleted the rows and columns
      spreadsheet.cellFormat({ textAlign: 'center' }, 'A2:A8');
      spreadsheet.cellFormat({ textAlign: 'center' }, 'D2:G8');
    }
  };
  return (
    <div>
      <SpreadsheetComponent ref={spreadsheetRef} created={onCreated}
showRibbon={false} showFormulaBar={false} allowDelete={true} >
        <SheetsDirective>
          <SheetDirective name={"Sheet1"}>
            <RangesDirective>
              <RangeDirective
dataSource={data}></RangeDirective>
            </RangesDirective>
            <ColumnsDirective>
              <ColumnDirective width={90}></ColumnDirective>
              <ColumnDirective width={220}></ColumnDirective>
              <ColumnDirective width={90}></ColumnDirective>
              <ColumnDirective width={140}></ColumnDirective>
              <ColumnDirective width={90}></ColumnDirective>
              <ColumnDirective width={100}></ColumnDirective>
            </ColumnsDirective>
          </SheetDirective>
        </SheetsDirective>
      </SpreadsheetComponent>
    </div>
  );
}
```

```

        <ColumnDirective width={100}></ColumnDirective>
      </ColumnsDirective>
    </SheetDirective>
    <SheetDirective name={"Sheet2"}>
      <RangesDirective>
        <RangeDirective
dataSource={data}></RangeDirective>
      </RangesDirective>
      <ColumnsDirective>
        <ColumnDirective width={90}></ColumnDirective>
        <ColumnDirective width={220}></ColumnDirective>
        <ColumnDirective width={90}></ColumnDirective>
        <ColumnDirective width={140}></ColumnDirective>
        <ColumnDirective width={90}></ColumnDirective>
        <ColumnDirective width={100}></ColumnDirective>
        <ColumnDirective width={100}></ColumnDirective>
      </ColumnsDirective>
    </SheetDirective>
  </SheetsDirective>
</SpreadsheetComponent>
</div>
);
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

### DATASOURCE.JSX

```

/**
 * Delete row column data source
 */
export let data = [
  {
    'Product Id': 1, 'Product Name': 'Chai', 'Supplier Id': 1, 'Quantity
Per Unit': '10 boxes x 20 bags', 'Unit Price': 18.00, 'Units In Stock': 39,
'Discontinued': true
  },
  {
    'Product Id': 2, 'Product Name': 'Chang', 'Supplier Id': 1,
'Quantity Per Unit': '24 - 12 oz bottles', 'Unit Price': 19.00, 'Units In
Stock': 17, 'Discontinued': true
  },
  {
    'Product Id': 3, 'Product Name': 'Aniseed Syrup', 'Supplier Id': 1,
'Quantity Per Unit': '12 - 550 ml bottles', 'Unit Price': 10.00, 'Units In
Stock': 13, 'Discontinued': true
  },
  {
    'Product Id': 4, 'Product Name': 'Chef Anton\'s Cajun Seasoning',
'Supplier Id': 2, 'Quantity Per Unit': '48 - 6 oz jars', 'Unit Price':
22.00, 'Units In Stock': 53, 'Discontinued': true
  },
  {

```

```

    'Product Id': 5, 'Product Name': 'Chef Anton\'s Gumbo Mix',
    'Supplier Id': 2, 'Quantity Per Unit': '36 boxes', 'Unit Price': 21.35,
    'Units In Stock': 0, 'Discontinued': true
  },
  {
    'Product Id': 6, 'Product Name': 'Grandma\'s Boysenberry Spread',
    'Supplier Id': 3, 'Quantity Per Unit': '12 - 8 oz jars', 'Unit Price':
    25.00, 'Units In Stock': 120, 'Discontinued': false
  },
  {
    'Product Id': 7, 'Product Name': 'Uncle Bob\'s Organic Dried Pears',
    'Supplier Id': 3, 'Quantity Per Unit': '12 - 1 lb pkgs.', 'Unit Price':
    30.00, 'Units In Stock': 15, 'Discontinued': false
  },
  {
    'Product Id': 8, 'Product Name': 'Queso Cabrales', 'Supplier Id': 5,
    'Quantity Per Unit': '1 kg pkg.', 'Unit Price': 21.00, 'Units In Stock': 22,
    'Discontinued': false
  },
  {
    'Product Id': 9, 'Product Name': 'Northwoods Cranberry Sauce',
    'Supplier Id': 3, 'Quantity Per Unit': '12 - 12 oz jars', 'Unit Price':
    40.00, 'Units In Stock': 6, 'Discontinued': false
  },
  {
    'Product Id': 10, 'Product Name': 'Mishi Kobe Niku', 'Supplier Id':
    4, 'Quantity Per Unit': '18 - 500 g pkgs.', 'Unit Price': 97.00, 'Units In
    Stock': 29, 'Discontinued': true
  }
];

```

### DATASOURCE.TSX

```

/**
 * Delete row column data source
 */
export let data: Object[] = [
  {
    'Product Id': 1, 'Product Name': 'Chai', 'Supplier Id': 1, 'Quantity
    Per Unit': '10 boxes x 20 bags', 'Unit Price': 18.00, 'Units In Stock': 39,
    'Discontinued': true
  },
  {
    'Product Id': 2, 'Product Name': 'Chang', 'Supplier Id': 1,
    'Quantity Per Unit': '24 - 12 oz bottles', 'Unit Price': 19.00, 'Units In
    Stock': 17, 'Discontinued': true
  },
  {
    'Product Id': 3, 'Product Name': 'Aniseed Syrup', 'Supplier Id': 1,
    'Quantity Per Unit': '12 - 550 ml bottles', 'Unit Price': 10.00, 'Units In
    Stock': 13, 'Discontinued': true
  },
  {
    'Product Id': 4, 'Product Name': 'Chef Anton\'s Cajun Seasoning',
    'Supplier Id': 2, 'Quantity Per Unit': '48 - 6 oz jars', 'Unit Price':
    22.00, 'Units In Stock': 53, 'Discontinued': true
  }
];

```

```

    },
    {
      'Product Id': 5, 'Product Name': 'Chef Anton\'s Gumbo Mix',
      'Supplier Id': 2, 'Quantity Per Unit': '36 boxes', 'Unit Price': 21.35,
      'Units In Stock': 0, 'Discontinued': true
    },
    {
      'Product Id': 6, 'Product Name': 'Grandma\'s Boysenberry Spread',
      'Supplier Id': 3, 'Quantity Per Unit': '12 - 8 oz jars', 'Unit Price':
      25.00, 'Units In Stock': 120, 'Discontinued': false
    },
    {
      'Product Id': 7, 'Product Name': 'Uncle Bob\'s Organic Dried Pears',
      'Supplier Id': 3, 'Quantity Per Unit': '12 - 1 lb pkgs.', 'Unit Price':
      30.00, 'Units In Stock': 15, 'Discontinued': false
    },
    {
      'Product Id': 8, 'Product Name': 'Queso Cabrales', 'Supplier Id': 5,
      'Quantity Per Unit': '1 kg pkg.', 'Unit Price': 21.00, 'Units In Stock': 22,
      'Discontinued': false
    },
    {
      'Product Id': 9, 'Product Name': 'Northwoods Cranberry Sauce',
      'Supplier Id': 3, 'Quantity Per Unit': '12 - 12 oz jars', 'Unit Price':
      40.00, 'Units In Stock': 6, 'Discontinued': false
    },
    {
      'Product Id': 10, 'Product Name': 'Mishi Kobe Niku', 'Supplier Id':
      4, 'Quantity Per Unit': '18 - 500 g pkgs.', 'Unit Price': 97.00, 'Units In
      Stock': 29, 'Discontinued': true
    }
  ]];

```

### Limitations of insert and delete

The following features have some limitations in Insert/Delete:

- Insert row/column between the formatting applied cells.
- Insert row/column between the data validation.
- Insert row/column between the conditional formatting applied cells.
- Insert/delete row/column between the filter applied cells.

### Hide and show

You can show or hide the rows and columns in the spreadsheet through property binding, method, and context menu.

#### Row

The rows can be hidden or shown through the following ways,

- Using `hidden` property in row, you can hide/show the rows at initial load.
- Using `hideRow` method, you can hide the rows by specifying the start and end row index, set the last argument `hide` as `false` to unhide the hidden rows.
- Right-click on the row header and select the desired option from context menu

### Column

The columns can be hidden or shown through following ways,

- Using `hidden` property in columns, you can hide/show the columns at initial load.
- Using `hideColumn` method, you can hide the columns by specifying the start and end column index, set the last argument `hide` as `false` to unhide the hidden columns.
- Right-click on the column header and select the desired option from context menu

The following code example shows the hide/show rows and columns operation in the spreadsheet.

#### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective } from '@syncfusion/ej2-react-
spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A1:H1');
      // Unhide the 2nd index hidden column
      spreadsheet.hideColumn(1, 1, false);
      // Unhide the 3rd index hidden row
      spreadsheet.hideRow(3, 3, false);
      // Hiding the 6th index column
      spreadsheet.hideColumn(6);
      // Hiding the 8th and 9th index row
      spreadsheet.hideRow(8, 9);
      spreadsheet.cellFormat({ textAlign: 'center' }, 'D2:H11');
    }
  }, []);
  return (
    <div>
      <SpreadsheetComponent ref={spreadsheetRef} showSheetTabs={false}
showRibbon={false} showFormulaBar={false} >
        <SheetsDirective>
          <SheetDirective>
            <RangesDirective>
              <RangeDirective
dataSource={data}></RangeDirective>
            </RangesDirective>
            <RowsDirective>
              <RowDirective index={2}
hidden={true}></RowDirective>
              <RowDirective hidden={true}></RowDirective>
            </RowsDirective>
            <ColumnsDirective>
```



```

        <ColumnDirective width={150}></ColumnDirective>
        <ColumnDirective width={100}
hidden={true}></ColumnDirective>
        <ColumnDirective width={100}
hidden={true}></ColumnDirective>
        <ColumnDirective width={80}></ColumnDirective>
        <ColumnDirective width={80}></ColumnDirective>
        <ColumnDirective width={80}></ColumnDirective>
        <ColumnDirective width={80}></ColumnDirective>
        <ColumnDirective width={80}></ColumnDirective>
    </ColumnsDirective>
  </SheetDirective>
</SheetsDirective>
</SpreadsheetComponent>
</div>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

## APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective } from '@syncfusion/ej2-react-
spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { data } from '../datasource';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A1:H1');
      // Unhide the 2nd index hidden column
      spreadsheet.hideColumn(1, 1, false);
      // Unhide the 3rd index hidden row
      spreadsheet.hideRow(3, 3, false);
      // Hiding the 6th index column
      spreadsheet.hideColumn(6);
      // Hiding the 8th and 9th index row
      spreadsheet.hideRow(8, 9);
      spreadsheet.cellFormat({ textAlign: 'center' }, 'D2:H11');
    }
  }, []);
  return (
    <div>
      <SpreadsheetComponent ref={spreadsheetRef} showSheetTabs={false}
showRibbon={false} showFormulaBar={false} >
        <SheetsDirective>

```

```

        <SheetDirective>
          <RangesDirective>
            <RangeDirective
dataSource={data}></RangeDirective>
          </RangesDirective>
          <RowsDirective>
            <RowDirective index={2}
hidden={true}></RowDirective>
            <RowDirective hidden={true}></RowDirective>
          </RowsDirective>
          <ColumnsDirective>
            <ColumnDirective width={150}></ColumnDirective>
            <ColumnDirective width={100}
hidden={true}></ColumnDirective>
            <ColumnDirective width={100}
hidden={true}></ColumnDirective>
            <ColumnDirective width={80}></ColumnDirective>
            <ColumnDirective width={80}></ColumnDirective>
            <ColumnDirective width={80}></ColumnDirective>
            <ColumnDirective width={80}></ColumnDirective>
            <ColumnDirective width={80}></ColumnDirective>
          </ColumnsDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  </div>
);
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

### DATASOURCE.JSX

```

/**
 * Hide/show data source
 */
export let data = [
  { 'Item Name': 'Casual Shoes', 'Date': '02/14/2019', 'Time': '11:34:32 AM', 'Quantity': 10, 'Price': 20, 'Amount': '=D2*E2', 'Discount': 1, 'Profit': 10 },
  { 'Item Name': 'Sports Shoes', 'Date': '06/11/2019', 'Time': '05:56:32 AM', 'Quantity': 20, 'Price': 30, 'Amount': '=D3*E3', 'Discount': 5, 'Profit': 50 },
  { 'Item Name': 'Formal Shoes', 'Date': '07/27/2019', 'Time': '03:32:44 AM', 'Quantity': 20, 'Price': 15, 'Amount': '=D4*E4', 'Discount': 7, 'Profit': 27 },
  { 'Item Name': 'Sandals & Floaters', 'Date': '11/21/2019', 'Time': '06:23:54 AM', 'Quantity': 15, 'Price': 20, 'Amount': '=D5*E5', 'Discount': 11, 'Profit': 67 },
  { 'Item Name': 'Flip- Flops & Slippers', 'Date': '06/23/2019', 'Time': '12:43:59 AM', 'Quantity': 30, 'Price': 10, 'Amount': '=D6*E6', 'Discount': 10, 'Profit': 70 },

```

```

    { 'Item Name': 'Sneakers', 'Date': '07/22/2019', 'Time': '10:55:53 AM',
      'Quantity': 40, 'Price': 20, 'Amount': '=D7*E7', 'Discount': 13, 'Profit':
      66 },
    { 'Item Name': 'Running Shoes', 'Date': '02/04/2019', 'Time': '03:44:34
      AM', 'Quantity': 20, 'Price': 10, 'Amount': '=D8*E8', 'Discount': 3,
      'Profit': 14 },
    { 'Item Name': 'Loafers', 'Date': '11/30/2019', 'Time': '03:12:52 AM',
      'Quantity': 31, 'Price': 10, 'Amount': '=D9*E9', 'Discount': 6, 'Profit': 29
    },
    { 'Item Name': 'Cricket Shoes', 'Date': '07/09/2019', 'Time': '11:32:14
      AM', 'Quantity': 41, 'Price': 30, 'Amount': '=D10*E10', 'Discount': 12,
      'Profit': 166 },
    { 'Item Name': 'T-Shirts', 'Date': '10/31/2019', 'Time': '12:01:44 AM',
      'Quantity': 50, 'Price': 10, 'Amount': '=D11*E11', 'Discount': 9, 'Profit':
      55 }
  ];

```

### DATASOURCE.TSX

```

/**
 * Hide/show data source
 */
export let data: Object[] = [
  { 'Item Name': 'Casual Shoes', 'Date': '02/14/2019', 'Time': '11:34:32
    AM', 'Quantity': 10, 'Price': 20, 'Amount': '=D2*E2', 'Discount': 1,
    'Profit': 10 },
  { 'Item Name': 'Sports Shoes', 'Date': '06/11/2019', 'Time': '05:56:32
    AM', 'Quantity': 20, 'Price': 30, 'Amount': '=D3*E3', 'Discount': 5,
    'Profit': 50 },
  { 'Item Name': 'Formal Shoes', 'Date': '07/27/2019', 'Time': '03:32:44
    AM', 'Quantity': 20, 'Price': 15, 'Amount': '=D4*E4', 'Discount': 7,
    'Profit': 27 },
  { 'Item Name': 'Sandals & Floaters', 'Date': '11/21/2019', 'Time':
    '06:23:54 AM', 'Quantity': 15, 'Price': 20, 'Amount': '=D5*E5', 'Discount':
    11, 'Profit': 67 },
  { 'Item Name': 'Flip- Flops & Slippers', 'Date': '06/23/2019', 'Time':
    '12:43:59 AM', 'Quantity': 30, 'Price': 10, 'Amount': '=D6*E6', 'Discount':
    10, 'Profit': 70 },
  { 'Item Name': 'Sneakers', 'Date': '07/22/2019', 'Time': '10:55:53 AM',
    'Quantity': 40, 'Price': 20, 'Amount': '=D7*E7', 'Discount': 13, 'Profit':
    66 },
  { 'Item Name': 'Running Shoes', 'Date': '02/04/2019', 'Time': '03:44:34
    AM', 'Quantity': 20, 'Price': 10, 'Amount': '=D8*E8', 'Discount': 3,
    'Profit': 14 },
  { 'Item Name': 'Loafers', 'Date': '11/30/2019', 'Time': '03:12:52 AM',
    'Quantity': 31, 'Price': 10, 'Amount': '=D9*E9', 'Discount': 6, 'Profit': 29
  },
  { 'Item Name': 'Cricket Shoes', 'Date': '07/09/2019', 'Time': '11:32:14
    AM', 'Quantity': 41, 'Price': 30, 'Amount': '=D10*E10', 'Discount': 12,
    'Profit': 166 },
  { 'Item Name': 'T-Shirts', 'Date': '10/31/2019', 'Time': '12:01:44 AM',
    'Quantity': 50, 'Price': 10, 'Amount': '=D11*E11', 'Discount': 9, 'Profit':
    55 }
];

```

### Size

You can change the size of rows and columns in the spreadsheet by using [setRowsHeight](#) and [setColumnsWidth](#) methods.

### Row

You can change the height of single or multiple rows by using the [setRowsHeight](#) method.

You can provide the following type of ranges to the method:

- Single row range: ['2:2']
- Multiple rows range: ['1:100']
- Multiple rows with discontinuous range: ['1:10', '15:25', '30:40']
- Multiple rows with different sheets: [Sheet1!1:50, 'Sheet2!1:50', 'Sheet3!1:50']

The following code example shows how to change the height for single/multiple rows in the spreadsheet.

### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, } from '@syncfusion/ej2-react-spreadsheet';
import { defaultData } from './datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      // To change height for single row
      spreadsheet.setRowsHeight(40, ['2:2']);
      // To change height for multiple rows
      spreadsheet.setRowsHeight(40, ['4:8', '10:12']);
    }
  }, []);
  return (
    <SpreadsheetComponent ref={spreadsheetRef} >
      <SheetsDirective>
        <SheetDirective>
          <RangesDirective>
            <RangeDirective dataSource={defaultData}></RangeDirective>
          </RangesDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}
```

**APP.TSX**

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, } from '@syncfusion/ej2-react-spreadsheet';
import { defaultData } from '../datasource';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      // To change height for single row
      spreadsheet.setRowsHeight(40, ['2']);
      // To change height for multiple rows
      spreadsheet.setRowsHeight(40, ['4:8', '10:12']);
    }
  }, []);
  return (
    <SpreadsheetComponent ref={spreadsheetRef} >
      <SheetsDirective>
        <SheetDirective>
          <RangesDirective>
            <RangeDirective dataSource={defaultData}></RangeDirective>
          </RangesDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  );
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}
```

**DATASOURCE.JSX**

```
/**
 * Default data source
 */
export let defaultData = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
    Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
  { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
    Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
  { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
    AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
  { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
    '12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
    70 },
  { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
    Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
```

```

    { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
      Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
    { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
      Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
    { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
      Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
    { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
      Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
  ];
  /**
   * Grid datasource
   */
  export function getTradeData(dataCount) {
    let employees = [
      'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
      'Fuller', 'Leverling', 'Peacock',
      'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
      'Dodsworth', 'Anne',
      'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
      'Jack', 'Rose'
    ];
    let designation = ['Manager', 'CFO', 'Designer', 'Developer', 'Program
      Directory', 'System Analyst', 'Project Lead'];
    let mail = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
      'journrapide.com'];
    let location = ['UK', 'USA', 'Sweden', 'France', 'Canada', 'Argentina',
      'Austria', 'Germany', 'Mexico'];
    let status = ['Active', 'Inactive'];
    let trustworthiness = ['Perfect', 'Sufficient', 'Insufficient'];
    let tradeData = [];
    let address = [
      '59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do Paço, 67',
      '2, rue du Commerce', 'Boulevard Tirou, 255',
      'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
      Mercado, 12',
      'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
      'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
      Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
      'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
      43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
      'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
      10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
      'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
      7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
      'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
      'Taucherstraße 10', 'Taucherstraße 10',
      'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
      Circus', 'Av. dos Lusíadas, 23',
      'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
      123', '2817 Milton Dr.'];
    let employeeimg = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
      dataCount = parseInt(dataCount, 10);
    }
    for (let i = 1; i <= dataCount; i++) {
      let code = 10000;
      tradeData.push({
        'EmployeeID': code + i,

```

```

        'Employees': employees[Math.floor(Math.random() *
employees.length)] + ' ' + employees[Math.floor(Math.random() *
employees.length)],
        'Designation': designation[Math.floor(Math.random() *
designation.length)],
        'Location': location[Math.floor(Math.random() *
location.length)],
        'Status': status[Math.floor(Math.random() * status.length)],
        'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
        'Rating': Math.floor(Math.random() * 5),
        'Software': Math.floor(Math.random() * 100),
        'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
        'CurrentSalary': Math.floor((Math.random() * 100000)),
        'Address': address[Math.floor(Math.random() * address.length)],
    });
    let employee = 'Employees';
    let emp = tradeData[i - 1][employee];
    let sName = emp.substr(0, emp.indexOf(' ')).toLowerCase();
    let empmail = 'Mail';
    tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }
    return tradeData;
}
export let tradeData = [
    {
        "EmployeeID": 10001,
        "Employees": "Laura Nancy",
        "Designation": "Designer",
        "Location": "France",
        "Status": "Inactive",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
        "Software": 69,
        "EmployeeImg": "usermale",
        "CurrentSalary": 84194,
        "Address": "Taucherstraße 10",
        "Mail": "laura15@jourrapide.com"
    },
    {
        "EmployeeID": 10002,
        "Employees": "Zachery Van",
        "Designation": "CFO",
        "Location": "Canada",
        "Status": "Inactive",
        "Trustworthiness": "Insufficient",
        "Rating": 3,
        "Software": 99,
        "EmployeeImg": "usermale",
        "CurrentSalary": 55349,
        "Address": "5ª Ave. Los Palos Grandes",
        "Mail": "zachery109@sample.com"
    },
    {
        "EmployeeID": 10003,

```

```
"Employees": "Rose Fuller",
"Designation": "CFO",
"Location": "France",
"Status": "Active",
"Trustworthiness": "Insufficient",
"Rating": 1,
"Software": 1,
"EmployeeImg": "usermale",
"CurrentSalary": 16477,
"Address": "2817 Milton Dr.",
"Mail": "rose55@rpy.com"
},
{
  "EmployeeID": 10004,
  "Employees": "Jack Bergs",
  "Designation": "Manager",
  "Location": "Mexico",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 36,
  "EmployeeImg": "usermale",
  "CurrentSalary": 49040,
  "Address": "2, rue du Commerce",
  "Mail": "jack30@sample.com"
},
{
  "EmployeeID": 10005,
  "Employees": "Vinet Bergs",
  "Designation": "Program Directory",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 39,
  "EmployeeImg": "usermale",
  "CurrentSalary": 5495,
  "Address": "Rua da Panificadora, 12",
  "Mail": "vinet32@jourrapide.com"
},
{
  "EmployeeID": 10006,
  "Employees": "Buchanan Van",
  "Designation": "Designer",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 4,
  "Software": 78,
  "EmployeeImg": "usermale",
  "CurrentSalary": 42182,
  "Address": "24, place Kléber",
  "Mail": "buchanan18@mail.com"
},
{
  "EmployeeID": 10007,
  "Employees": "Dodsworth Nancy",
```



```

    "Designation": "Project Lead",
    "Location": "USA",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 35776,
    "Address": "Rua do Paço, 67",
    "Mail": "dodsworth84@mail.com"
  },
  {
    "EmployeeID": 10008,
    "Employees": "Laura Jack",
    "Designation": "Developer",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 89,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25108,
    "Address": "Rua da Panificadora, 12",
    "Mail": "laura82@mail.com"
  },
  {
    "EmployeeID": 10009,
    "Employees": "Anne Fuller",
    "Designation": "Program Directory",
    "Location": "Mexico",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 32568,
    "Address": "Gran Vía, 1",
    "Mail": "anne97@jourrapide.com"
  },
  {
    "EmployeeID": 10010,
    "Employees": "Buchanan Andrew",
    "Designation": "Designer",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 62,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 12320,
    "Address": "P.O. Box 555",
    "Mail": "buchanan50@jourrapide.com"
  },
  {
    "EmployeeID": 10011,
    "Employees": "Andrew Janet",
    "Designation": "System Analyst",

```

```

      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 3,
      "Software": 8,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 20890,
      "Address": "Starenweg 5",
      "Mail": "andrew63@mail.com"
    },
    {
      "EmployeeID": 10012,
      "Employees": "Margaret Tamer",
      "Designation": "System Analyst",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 4,
      "Software": 7,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 22337,
      "Address": "Magazinweg 7",
      "Mail": "margaret26@mail.com"
    },
    {
      "EmployeeID": 10013,
      "Employees": "Tamer Fuller",
      "Designation": "CFO",
      "Location": "Canada",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 3,
      "Software": 78,
      "EmployeeImg": "usermale",
      "CurrentSalary": 89181,
      "Address": "Taucherstraße 10",
      "Mail": "tamer40@arpy.com"
    },
    {
      "EmployeeID": 10014,
      "Employees": "Tamer Anne",
      "Designation": "CFO",
      "Location": "Sweden",
      "Status": "Active",
      "Trustworthiness": "Sufficient",
      "Rating": 0,
      "Software": 18,
      "EmployeeImg": "usermale",
      "CurrentSalary": 20998,
      "Address": "Taucherstraße 10",
      "Mail": "tamer68@arpy.com"
    },
    {
      "EmployeeID": 10015,
      "Employees": "Anton Davolio",
      "Designation": "Project Lead",
      "Location": "France",

```

```

    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 4,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 48232,
    "Address": "Luisenstr. 48",
    "Mail": "anton46@mail.com"
  },
  {
    "EmployeeID": 10016,
    "Employees": "Buchanan Buchanan",
    "Designation": "System Analyst",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "usermale",
    "CurrentSalary": 43041,
    "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
    "Mail": "buchanan68@mail.com"
  },
  {
    "EmployeeID": 10017,
    "Employees": "King Buchanan",
    "Designation": "Program Directory",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 44,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 25259,
    "Address": "Magazinweg 7",
    "Mail": "king80@jourrapide.com"
  },
  {
    "EmployeeID": 10018,
    "Employees": "Rose Michael",
    "Designation": "Project Lead",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 31,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 91156,
    "Address": "Fauntleroy Circus",
    "Mail": "rose75@mail.com"
  },
  {
    "EmployeeID": 10019,
    "Employees": "King Bergs",
    "Designation": "Developer",
    "Location": "Germany",
    "Status": "Active",

```

```

    "Trustworthiness": "Sufficient",
    "Rating": 2,
    "Software": 29,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 28826,
    "Address": "2817 Milton Dr.",
    "Mail": "king57@jourrapide.com"
  },
  {
    "EmployeeID": 10020,
    "Employees": "Davolio Fuller",
    "Designation": "Designer",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 3,
    "Software": 35,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 71035,
    "Address": "Gran Vía, 1",
    "Mail": "davolio29@arpy.com"
  },
  {
    "EmployeeID": 10021,
    "Employees": "Rose Rose",
    "Designation": "CFO",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 38,
    "EmployeeImg": "usermale",
    "CurrentSalary": 68123,
    "Address": "Rua do Mercado, 12",
    "Mail": "rose54@arpy.com"
  },
  {
    "EmployeeID": 10022,
    "Employees": "Andrew Michael",
    "Designation": "Program Directory",
    "Location": "UK",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 2,
    "Software": 61,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 75470,
    "Address": "2, rue du Commerce",
    "Mail": "andrew88@jourrapide.com"
  },
  {
    "EmployeeID": 10023,
    "Employees": "Davolio Kathryn",
    "Designation": "Manager",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Perfect",

```

```

    "Rating": 3,
    "Software": 25,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25234,
    "Address": "Hauptstr. 31",
    "Mail": "davolio42@sample.com"
  },
  {
    "EmployeeID": 10024,
    "Employees": "Anne Fleet",
    "Designation": "System Analyst",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 8341,
    "Address": "59 rue de l'Abbaye",
    "Mail": "anne86@arpy.com"
  },
  {
    "EmployeeID": 10025,
    "Employees": "Margaret Andrew",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 51,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84975,
    "Address": "P.O. Box 555",
    "Mail": "margaret41@arpy.com"
  },
  {
    "EmployeeID": 10026,
    "Employees": "Kathryn Laura",
    "Designation": "Project Lead",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 48,
    "EmployeeImg": "usermale",
    "CurrentSalary": 97282,
    "Address": "Avda. Azteca 123",
    "Mail": "kathryn82@rpy.com"
  },
  {
    "EmployeeID": 10027,
    "Employees": "Michael Michael",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 4,

```

```

    "Software": 16,
    "EmployeeImg": "usermale",
    "CurrentSalary": 4184,
    "Address": "Rua do Paço, 67",
    "Mail": "michael58@jourrapide.com"
  },
  {
    "EmployeeID": 10028,
    "Employees": "Leverling Vinet",
    "Designation": "Project Lead",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 57,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 38370,
    "Address": "59 rue de l'Abbaye",
    "Mail": "leverling102@sample.com"
  },
  {
    "EmployeeID": 10029,
    "Employees": "Rose Jack",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 46,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84790,
    "Address": "Rua do Mercado, 12",
    "Mail": "rose108@jourrapide.com"
  },
  {
    "EmployeeID": 10030,
    "Employees": "Vinet Van",
    "Designation": "Developer",
    "Location": "USA",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 40,
    "EmployeeImg": "usermale",
    "CurrentSalary": 71005,
    "Address": "Gran Vía, 1",
    "Mail": "vinet90@jourrapide.com"
  }
];

```

**DATASOURCE.TSX**

```

/**
 * Default data source
 */
export let defaultData: Object[] = [

```

```

    { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
      Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
    { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
      Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
    { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
      Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
    { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
      AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
    { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
      '12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
      70 },
    { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
      Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
    { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
      Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
    { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
      Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
    { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
      Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
    { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
      Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
  ];
  /**
   * Grid datasource
   */
  export function getTradeData(dataCount?: number): object {
    let employees: string[] = [
      'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
      'Fuller', 'Leverling', 'Peacock',
      'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
      'Dodsworth', 'Anne',
      'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
      'Jack', 'Rose'];
    let designation: string[] = ['Manager', 'CFO', 'Designer', 'Developer',
      'Program Directory', 'System Analyst', 'Project Lead'];
    let mail: string[] = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
      'jourrapide.com'];
    let location: string[] = ['UK', 'USA', 'Sweden', 'France', 'Canada',
      'Argentina', 'Austria', 'Germany', 'Mexico'];
    let status: string[] = ['Active', 'Inactive'];
    let trustworthiness: string[] = ['Perfect', 'Sufficient',
      'Insufficient'];
    let tradeData: Object[] = [];
    let address: string[] = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do
      Paço, 67', '2, rue du Commerce', 'Boulevard Tirou, 255',
      'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
      Mercado, 12',
      'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
      'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
      Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
      'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
      43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
      'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
      10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
      'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
      7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',

```

```

    'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
    'Taucherstraße 10', 'Taucherstraße 10',
    'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
    Circus', 'Av. dos Lusíadas, 23',
    'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
    123', '2817 Milton Dr.'];
    let employeeimg: string[] = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
        dataCount = parseInt(dataCount, 10);
    }
    for (let i: number = 1; i <= dataCount; i++) {
        let code: any = 10000;
        tradeData.push({
            'EmployeeID': code + i,
            'Employees':
                employees[Math.floor(Math.random() * employees.length)] + '
' + employees[Math.floor(Math.random() * employees.length)],
            'Designation': designation[Math.floor(Math.random() *
designation.length)],
            'Location': location[Math.floor(Math.random() *
location.length)],
            'Status': status[Math.floor(Math.random() * status.length)],
            'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
            'Rating': Math.floor(Math.random() * 5),
            'Software': Math.floor(Math.random() * 100),
            'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
            'CurrentSalary': Math.floor((Math.random() * 100000)),
            'Address': address[Math.floor(Math.random() * address.length)],
        });
        let employee: string = 'Employees';
        let emp: string = tradeData[i - 1][employee];
        let sName: string = emp.substr(0, emp.indexOf(' ')).toLowerCase();
        let empmail: string = 'Mail';
        tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }
    return tradeData;
}
export let tradeData: Object[] = [
    {
        "EmployeeID": 10001,
        "Employees": "Laura Nancy",
        "Designation": "Designer",
        "Location": "France",
        "Status": "Inactive",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
        "Software": 69,
        "EmployeeImg": "usermale",
        "CurrentSalary": 84194,
        "Address": "Taucherstraße 10",
        "Mail": "laura15@jourrapide.com"
    },
    {
        "EmployeeID": 10002,

```



```
"Employees": "Zachery Van",
"Designation": "CFO",
"Location": "Canada",
"Status": "Inactive",
"Trustworthiness": "Insufficient",
"Rating": 3,
"Software": 99,
"EmployeeImg": "usermale",
"CurrentSalary": 55349,
"Address": "5ª Ave. Los Palos Grandes",
"Mail": "zachery109@sample.com"
},
{
  "EmployeeID": 10003,
  "Employees": "Rose Fuller",
  "Designation": "CFO",
  "Location": "France",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 1,
  "EmployeeImg": "usermale",
  "CurrentSalary": 16477,
  "Address": "2817 Milton Dr.",
  "Mail": "rose55@rpy.com"
},
{
  "EmployeeID": 10004,
  "Employees": "Jack Bergs",
  "Designation": "Manager",
  "Location": "Mexico",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 36,
  "EmployeeImg": "usermale",
  "CurrentSalary": 49040,
  "Address": "2, rue du Commerce",
  "Mail": "jack30@sample.com"
},
{
  "EmployeeID": 10005,
  "Employees": "Vinet Bergs",
  "Designation": "Program Directory",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 39,
  "EmployeeImg": "usermale",
  "CurrentSalary": 5495,
  "Address": "Rua da Panificadora, 12",
  "Mail": "vinet32@jourrapide.com"
},
{
  "EmployeeID": 10006,
  "Employees": "Buchanan Van",
```

```
"Designation": "Designer",
"Location": "Germany",
"Status": "Active",
"Trustworthiness": "Insufficient",
"Rating": 4,
"Software": 78,
"EmployeeImg": "usermale",
"CurrentSalary": 42182,
"Address": "24, place Kléber",
"Mail": "buchanan18@mail.com"
},
{
  "EmployeeID": 10007,
  "Employees": "Dodsworth Nancy",
  "Designation": "Project Lead",
  "Location": "USA",
  "Status": "Inactive",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 0,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 35776,
  "Address": "Rua do Paço, 67",
  "Mail": "dodsworth84@mail.com"
},
{
  "EmployeeID": 10008,
  "Employees": "Laura Jack",
  "Designation": "Developer",
  "Location": "Austria",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 89,
  "EmployeeImg": "usermale",
  "CurrentSalary": 25108,
  "Address": "Rua da Panificadora, 12",
  "Mail": "laura82@mail.com"
},
{
  "EmployeeID": 10009,
  "Employees": "Anne Fuller",
  "Designation": "Program Directory",
  "Location": "Mexico",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 0,
  "Software": 19,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 32568,
  "Address": "Gran Vía, 1",
  "Mail": "anne97@jourrapide.com"
},
{
  "EmployeeID": 10010,
  "Employees": "Buchanan Andrew",
  "Designation": "Designer",
```

```

    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 62,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 12320,
    "Address": "P.O. Box 555",
    "Mail": "buchanan50@jourrapide.com"
  },
  {
    "EmployeeID": 10011,
    "Employees": "Andrew Janet",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 20890,
    "Address": "Starenweg 5",
    "Mail": "andrew63@mail.com"
  },
  {
    "EmployeeID": 10012,
    "Employees": "Margaret Tamer",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 7,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 22337,
    "Address": "Magazinweg 7",
    "Mail": "margaret26@mail.com"
  },
  {
    "EmployeeID": 10013,
    "Employees": "Tamer Fuller",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 89181,
    "Address": "Taucherstraße 10",
    "Mail": "tamer40@arpy.com"
  },
  {
    "EmployeeID": 10014,
    "Employees": "Tamer Anne",
    "Designation": "CFO",
    "Location": "Sweden",

```

```
"Status": "Active",
"Trustworthiness": "Sufficient",
"Rating": 0,
"Software": 18,
"EmployeeImg": "usermale",
"CurrentSalary": 20998,
"Address": "Taucherstraße 10",
"Mail": "tamer68@arpy.com"
},
{
  "EmployeeID": 10015,
  "Employees": "Anton Davolio",
  "Designation": "Project Lead",
  "Location": "France",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 4,
  "Software": 8,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 48232,
  "Address": "Luisenstr. 48",
  "Mail": "anton46@mail.com"
},
{
  "EmployeeID": 10016,
  "Employees": "Buchanan Buchanan",
  "Designation": "System Analyst",
  "Location": "Austria",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 19,
  "EmployeeImg": "usermale",
  "CurrentSalary": 43041,
  "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
  "Mail": "buchanan68@mail.com"
},
{
  "EmployeeID": 10017,
  "Employees": "King Buchanan",
  "Designation": "Program Directory",
  "Location": "Sweden",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 44,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 25259,
  "Address": "Magazinweg 7",
  "Mail": "king80@jourrapide.com"
},
{
  "EmployeeID": 10018,
  "Employees": "Rose Michael",
  "Designation": "Project Lead",
  "Location": "Canada",
  "Status": "Active",
```

```
"Trustworthiness": "Perfect",
"Rating": 4,
"Software": 31,
"EmployeeImg": "userfemale",
"CurrentSalary": 91156,
"Address": "Fauntleroy Circus",
"Mail": "rose75@mail.com"
},
{
  "EmployeeID": 10019,
  "Employees": "King Bergs",
  "Designation": "Developer",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 2,
  "Software": 29,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 28826,
  "Address": "2817 Milton Dr.",
  "Mail": "king57@jourrapide.com"
},
{
  "EmployeeID": 10020,
  "Employees": "Davolio Fuller",
  "Designation": "Designer",
  "Location": "Canada",
  "Status": "Inactive",
  "Trustworthiness": "Sufficient",
  "Rating": 3,
  "Software": 35,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 71035,
  "Address": "Gran Vía, 1",
  "Mail": "davolio29@arpy.com"
},
{
  "EmployeeID": 10021,
  "Employees": "Rose Rose",
  "Designation": "CFO",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 38,
  "EmployeeImg": "usermale",
  "CurrentSalary": 68123,
  "Address": "Rua do Mercado, 12",
  "Mail": "rose54@arpy.com"
},
{
  "EmployeeID": 10022,
  "Employees": "Andrew Michael",
  "Designation": "Program Directory",
  "Location": "UK",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
```

```
"Rating": 2,  
"Software": 61,  
"EmployeeImg": "userfemale",  
"CurrentSalary": 75470,  
"Address": "2, rue du Commerce",  
"Mail": "andrew88@jourrapide.com"  
},  
{  
  "EmployeeID": 10023,  
  "Employees": "Davolio Kathryn",  
  "Designation": "Manager",  
  "Location": "Germany",  
  "Status": "Active",  
  "Trustworthiness": "Perfect",  
  "Rating": 3,  
  "Software": 25,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 25234,  
  "Address": "Hauptstr. 31",  
  "Mail": "davolio42@sample.com"  
},  
{  
  "EmployeeID": 10024,  
  "Employees": "Anne Fleet",  
  "Designation": "System Analyst",  
  "Location": "UK",  
  "Status": "Active",  
  "Trustworthiness": "Perfect",  
  "Rating": 3,  
  "Software": 0,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 8341,  
  "Address": "59 rue de l'Abbaye",  
  "Mail": "anne86@arpy.com"  
},  
{  
  "EmployeeID": 10025,  
  "Employees": "Margaret Andrew",  
  "Designation": "System Analyst",  
  "Location": "Germany",  
  "Status": "Inactive",  
  "Trustworthiness": "Insufficient",  
  "Rating": 3,  
  "Software": 51,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 84975,  
  "Address": "P.O. Box 555",  
  "Mail": "margaret41@arpy.com"  
},  
{  
  "EmployeeID": 10026,  
  "Employees": "Kathryn Laura",  
  "Designation": "Project Lead",  
  "Location": "Austria",  
  "Status": "Active",  
  "Trustworthiness": "Insufficient",  
  "Rating": 3,
```

```
"Software": 48,  
"EmployeeImg": "usermale",  
"CurrentSalary": 97282,  
"Address": "Avda. Azteca 123",  
"Mail": "kathryn82@rpy.com"  
},  
{  
  "EmployeeID": 10027,  
  "Employees": "Michael Michael",  
  "Designation": "Developer",  
  "Location": "UK",  
  "Status": "Inactive",  
  "Trustworthiness": "Perfect",  
  "Rating": 4,  
  "Software": 16,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 4184,  
  "Address": "Rua do Paço, 67",  
  "Mail": "michael58@jourrapide.com"  
},  
{  
  "EmployeeID": 10028,  
  "Employees": "Leverling Vinet",  
  "Designation": "Project Lead",  
  "Location": "Germany",  
  "Status": "Inactive",  
  "Trustworthiness": "Perfect",  
  "Rating": 0,  
  "Software": 57,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 38370,  
  "Address": "59 rue de l'Abbaye",  
  "Mail": "leverling102@sample.com"  
},  
{  
  "EmployeeID": 10029,  
  "Employees": "Rose Jack",  
  "Designation": "Developer",  
  "Location": "UK",  
  "Status": "Active",  
  "Trustworthiness": "Perfect",  
  "Rating": 0,  
  "Software": 46,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 84790,  
  "Address": "Rua do Mercado, 12",  
  "Mail": "rosl08@jourrapide.com"  
},  
{  
  "EmployeeID": 10030,  
  "Employees": "Vinet Van",  
  "Designation": "Developer",  
  "Location": "USA",  
  "Status": "Active",  
  "Trustworthiness": "Sufficient",  
  "Rating": 0,  
  "Software": 40,
```

```

    "EmployeeImg": "usermale",
    "CurrentSalary": 71005,
    "Address": "Gran Vía, 1",
    "Mail": "vinet90@jourrapide.com"
  }
]

```

### Column

You can change the width of single or multiple columns by using the [setColumnsWidth](#) method.

You can provide the following type of ranges to the method:

- Single column range: ['F:F']
- Multiple columns range: ['A:F']
- Multiple columns with discontinuous range: ['A:C', 'G:I', 'K:M']
- Multiple columns with different sheets: [Sheet1!A:H, 'Sheet2!A:H', 'Sheet3!A:H']

The following code example shows how to change the width for single/multiple columns in the spreadsheet.

### APP.JSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective } from '@syncfusion/ej2-react-spreadsheet';
import { defaultData } from './datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      // To change width of single column
      spreadsheet.setColumnsWidth(100, ['F']);
      // To change width of multiple columns
      spreadsheet.setColumnsWidth(120, ['A:C', 'G:I', 'K:M']);
    }
  }, []);
  return (
    <SpreadsheetComponent ref={spreadsheetRef}>
      <SheetsDirective>
        <SheetDirective>
          <RangesDirective>
            <RangeDirective dataSource={defaultData}></RangeDirective>
          </RangesDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  );
};
export default App;
const root = createRoot(document.getElementById('root'));

```



```
root.render(<App />);
{% endraw %}
```

### APP.TSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective } from '@syncfusion/ej2-react-spreadsheet';
import { defaultData } from './datasource';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      // To change width of single column
      spreadsheet.setColumnsWidth(100, ['F']);
      // To change width of multiple columns
      spreadsheet.setColumnsWidth(120, ['A:C', 'G:I', 'K:M']);
    }
  }, []);
  return (
    <SpreadsheetComponent ref={spreadsheetRef}>
      <SheetsDirective>
        <SheetDirective>
          <RangesDirective>
            <RangeDirective dataSource={defaultData}></RangeDirective>
          </RangesDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  );
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}
```

### DATASOURCE.JSX

```
/**
 * Default data source
 */
export let defaultData = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
  { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
  { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
```

```

    { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
    '12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
    70 },
    { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
    Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
    { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
    Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
    { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
    Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
    { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
    Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
    { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
    Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
  ];
  /**
   * Grid datasource
   */
  export function getTradeData(dataCount) {
    let employees = [
      'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
      'Fuller', 'Leverling', 'Peacock',
      'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
      'Dodsworth', 'Anne',
      'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
      'Jack', 'Rose'
    ];
    let designation = ['Manager', 'CFO', 'Designer', 'Developer', 'Program
    Directory', 'System Analyst', 'Project Lead'];
    let mail = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
    'jourrapide.com'];
    let location = ['UK', 'USA', 'Sweden', 'France', 'Canada', 'Argentina',
    'Austria', 'Germany', 'Mexico'];
    let status = ['Active', 'Inactive'];
    let trustworthiness = ['Perfect', 'Sufficient', 'Insufficient'];
    let tradeData = [];
    let address = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do Paço, 67',
    '2, rue du Commerce', 'Boulevard Tirou, 255',
    'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
    Mercado, 12',
    'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
    'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
    Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
    'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
    43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
    'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
    10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
    'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
    7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
    'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
    'Taucherstraße 10', 'Taucherstraße 10',
    'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
    Circus', 'Av. dos Lusíadas, 23',
    'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
    123', '2817 Milton Dr.'];
    let employeeimg = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
      dataCount = parseInt(dataCount, 10);
    }
  }

```

```

    }
    for (let i = 1; i <= dataCount; i++) {
      let code = 10000;
      tradeData.push({
        'EmployeeID': code + i,
        'Employees': employees[Math.floor(Math.random() *
employees.length)] + ' ' + employees[Math.floor(Math.random() *
employees.length)],
        'Designation': designation[Math.floor(Math.random() *
designation.length)],
        'Location': location[Math.floor(Math.random() *
location.length)],
        'Status': status[Math.floor(Math.random() * status.length)],
        'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
        'Rating': Math.floor(Math.random() * 5),
        'Software': Math.floor(Math.random() * 100),
        'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
        'CurrentSalary': Math.floor((Math.random() * 100000)),
        'Address': address[Math.floor(Math.random() * address.length)],
      });
      let employee = 'Employees';
      let emp = tradeData[i - 1][employee];
      let sName = emp.substr(0, emp.indexOf(' ')).toLowerCase();
      let empmail = 'Mail';
      tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }
    return tradeData;
  }
  export let tradeData = [
    {
      "EmployeeID": 10001,
      "Employees": "Laura Nancy",
      "Designation": "Designer",
      "Location": "France",
      "Status": "Inactive",
      "Trustworthiness": "Sufficient",
      "Rating": 0,
      "Software": 69,
      "EmployeeImg": "usermale",
      "CurrentSalary": 84194,
      "Address": "Taucherstraße 10",
      "Mail": "laura15@jourrapide.com"
    },
    {
      "EmployeeID": 10002,
      "Employees": "Zachery Van",
      "Designation": "CFO",
      "Location": "Canada",
      "Status": "Inactive",
      "Trustworthiness": "Insufficient",
      "Rating": 3,
      "Software": 99,
      "EmployeeImg": "usermale",
      "CurrentSalary": 55349,

```

```

    "Address": "5a Ave. Los Palos Grandes",
    "Mail": "zachery109@sample.com"
  },
  {
    "EmployeeID": 10003,
    "Employees": "Rose Fuller",
    "Designation": "CFO",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 1,
    "EmployeeImg": "usermale",
    "CurrentSalary": 16477,
    "Address": "2817 Milton Dr.",
    "Mail": "rose55@rpy.com"
  },
  {
    "EmployeeID": 10004,
    "Employees": "Jack Bergs",
    "Designation": "Manager",
    "Location": "Mexico",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 36,
    "EmployeeImg": "usermale",
    "CurrentSalary": 49040,
    "Address": "2, rue du Commerce",
    "Mail": "jack30@sample.com"
  },
  {
    "EmployeeID": 10005,
    "Employees": "Vinet Bergs",
    "Designation": "Program Directory",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 39,
    "EmployeeImg": "usermale",
    "CurrentSalary": 5495,
    "Address": "Rua da Panificadora, 12",
    "Mail": "vinet32@jourrapide.com"
  },
  {
    "EmployeeID": 10006,
    "Employees": "Buchanan Van",
    "Designation": "Designer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 42182,
    "Address": "24, place Kléber",

```

```

    "Mail": "buchanan18@mail.com"
  },
  {
    "EmployeeID": 10007,
    "Employees": "Dodsworth Nancy",
    "Designation": "Project Lead",
    "Location": "USA",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 35776,
    "Address": "Rua do Paço, 67",
    "Mail": "dodsworth84@mail.com"
  },
  {
    "EmployeeID": 10008,
    "Employees": "Laura Jack",
    "Designation": "Developer",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 89,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25108,
    "Address": "Rua da Panificadora, 12",
    "Mail": "laura82@mail.com"
  },
  {
    "EmployeeID": 10009,
    "Employees": "Anne Fuller",
    "Designation": "Program Directory",
    "Location": "Mexico",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 32568,
    "Address": "Gran Vía, 1",
    "Mail": "anne97@jourrapide.com"
  },
  {
    "EmployeeID": 10010,
    "Employees": "Buchanan Andrew",
    "Designation": "Designer",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 62,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 12320,
    "Address": "P.O. Box 555",
    "Mail": "buchanan50@jourrapide.com"
  }

```

```

    },
    {
      "EmployeeID": 10011,
      "Employees": "Andrew Janet",
      "Designation": "System Analyst",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 3,
      "Software": 8,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 20890,
      "Address": "Starenweg 5",
      "Mail": "andrew63@mail.com"
    },
    {
      "EmployeeID": 10012,
      "Employees": "Margaret Tamer",
      "Designation": "System Analyst",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 4,
      "Software": 7,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 22337,
      "Address": "Magazinweg 7",
      "Mail": "margaret26@mail.com"
    },
    {
      "EmployeeID": 10013,
      "Employees": "Tamer Fuller",
      "Designation": "CFO",
      "Location": "Canada",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 3,
      "Software": 78,
      "EmployeeImg": "usermale",
      "CurrentSalary": 89181,
      "Address": "Taucherstraße 10",
      "Mail": "tamer40@arpy.com"
    },
    {
      "EmployeeID": 10014,
      "Employees": "Tamer Anne",
      "Designation": "CFO",
      "Location": "Sweden",
      "Status": "Active",
      "Trustworthiness": "Sufficient",
      "Rating": 0,
      "Software": 18,
      "EmployeeImg": "usermale",
      "CurrentSalary": 20998,
      "Address": "Taucherstraße 10",
      "Mail": "tamer68@arpy.com"
    },
  ],

```

```

{
  "EmployeeID": 10015,
  "Employees": "Anton Davolio",
  "Designation": "Project Lead",
  "Location": "France",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 4,
  "Software": 8,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 48232,
  "Address": "Luisenstr. 48",
  "Mail": "anton46@mail.com"
},
{
  "EmployeeID": 10016,
  "Employees": "Buchanan Buchanan",
  "Designation": "System Analyst",
  "Location": "Austria",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 19,
  "EmployeeImg": "usermale",
  "CurrentSalary": 43041,
  "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
  "Mail": "buchanan68@mail.com"
},
{
  "EmployeeID": 10017,
  "Employees": "King Buchanan",
  "Designation": "Program Directory",
  "Location": "Sweden",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 44,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 25259,
  "Address": "Magazinweg 7",
  "Mail": "king80@jourrapide.com"
},
{
  "EmployeeID": 10018,
  "Employees": "Rose Michael",
  "Designation": "Project Lead",
  "Location": "Canada",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 4,
  "Software": 31,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 91156,
  "Address": "Fauntleroy Circus",
  "Mail": "rose75@mail.com"
},
{

```

```
"EmployeeID": 10019,  
"Employees": "King Bergs",  
"Designation": "Developer",  
"Location": "Germany",  
"Status": "Active",  
"Trustworthiness": "Sufficient",  
"Rating": 2,  
"Software": 29,  
"EmployeeImg": "userfemale",  
"CurrentSalary": 28826,  
"Address": "2817 Milton Dr.",  
"Mail": "king57@jourrapide.com"  
},  
{  
  "EmployeeID": 10020,  
  "Employees": "Davolio Fuller",  
  "Designation": "Designer",  
  "Location": "Canada",  
  "Status": "Inactive",  
  "Trustworthiness": "Sufficient",  
  "Rating": 3,  
  "Software": 35,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 71035,  
  "Address": "Gran Vía, 1",  
  "Mail": "davolio29@arpy.com"  
},  
{  
  "EmployeeID": 10021,  
  "Employees": "Rose Rose",  
  "Designation": "CFO",  
  "Location": "Germany",  
  "Status": "Active",  
  "Trustworthiness": "Perfect",  
  "Rating": 3,  
  "Software": 38,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 68123,  
  "Address": "Rua do Mercado, 12",  
  "Mail": "rose54@arpy.com"  
},  
{  
  "EmployeeID": 10022,  
  "Employees": "Andrew Michael",  
  "Designation": "Program Directory",  
  "Location": "UK",  
  "Status": "Inactive",  
  "Trustworthiness": "Insufficient",  
  "Rating": 2,  
  "Software": 61,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 75470,  
  "Address": "2, rue du Commerce",  
  "Mail": "andrew88@jourrapide.com"  
},  
{  
  "EmployeeID": 10023,
```



```

    "Employees": "Davolio Kathryn",
    "Designation": "Manager",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 25,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25234,
    "Address": "Hauptstr. 31",
    "Mail": "davolio42@sample.com"
  },
  {
    "EmployeeID": 10024,
    "Employees": "Anne Fleet",
    "Designation": "System Analyst",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 8341,
    "Address": "59 rue de l'Abbaye",
    "Mail": "anne86@arpy.com"
  },
  {
    "EmployeeID": 10025,
    "Employees": "Margaret Andrew",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 51,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84975,
    "Address": "P.O. Box 555",
    "Mail": "margaret41@arpy.com"
  },
  {
    "EmployeeID": 10026,
    "Employees": "Kathryn Laura",
    "Designation": "Project Lead",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 48,
    "EmployeeImg": "usermale",
    "CurrentSalary": 97282,
    "Address": "Avda. Azteca 123",
    "Mail": "kathryn82@rpy.com"
  },
  {
    "EmployeeID": 10027,
    "Employees": "Michael Michael",

```

```

    "Designation": "Developer",
    "Location": "UK",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 16,
    "EmployeeImg": "usermale",
    "CurrentSalary": 4184,
    "Address": "Rua do Paço, 67",
    "Mail": "michael58@jourrapide.com"
  },
  {
    "EmployeeID": 10028,
    "Employees": "Leverling Vinet",
    "Designation": "Project Lead",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 57,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 38370,
    "Address": "59 rue de l'Abbaye",
    "Mail": "leverling102@sample.com"
  },
  {
    "EmployeeID": 10029,
    "Employees": "Rose Jack",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 46,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84790,
    "Address": "Rua do Mercado, 12",
    "Mail": "rosl08@jourrapide.com"
  },
  {
    "EmployeeID": 10030,
    "Employees": "Vinet Van",
    "Designation": "Developer",
    "Location": "USA",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 40,
    "EmployeeImg": "usermale",
    "CurrentSalary": 71005,
    "Address": "Gran Vía, 1",
    "Mail": "vinet90@jourrapide.com"
  }
];

```

**DATASOURCE.TSX**

```

/**
 * Default data source
 */
export let defaultData: Object[] = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
    Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
  { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
    Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
  { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
  { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
'12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
70 },
  { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
    Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
  { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
    Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
  { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
    Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
  { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
    Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
  { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
    Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
/**
 * Grid datasource
 */
export function getTradeData(dataCount?: number): object {
  let employees: string[] = [
    'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
    'Fuller', 'Leverling', 'Peacock',
    'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
    'Dodsworth', 'Anne',
    'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
    'Jack', 'Rose'];
  let designation: string[] = ['Manager', 'CFO', 'Designer', 'Developer',
    'Program Directory', 'System Analyst', 'Project Lead'];
  let mail: string[] = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
    'jourrapide.com'];
  let location: string[] = ['UK', 'USA', 'Sweden', 'France', 'Canada',
    'Argentina', 'Austria', 'Germany', 'Mexico'];
  let status: string[] = ['Active', 'Inactive'];
  let trustworthiness: string[] = ['Perfect', 'Sufficient',
    'Insufficient'];
  let tradeData: Object[] = [];
  let address: string[] = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do
Paço, 67', '2, rue du Commerce', 'Boulevard Tirou, 255',
    'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
Mercado, 12',
    'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
    'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
    'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',

```

```

    'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
    'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
    'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
    'Taucherstraße 10', 'Taucherstraße 10',
    'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
Circus', 'Av. dos Lusíadas, 23',
    'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
123', '2817 Milton Dr.'];
    let employeeimg: string[] = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
        dataCount = parseInt(dataCount, 10);
    }
    for (let i: number = 1; i <= dataCount; i++) {
        let code: any = 10000;
        tradeData.push({
            'EmployeeID': code + i,
            'Employees':
                employees[Math.floor(Math.random() * employees.length)] + '
' + employees[Math.floor(Math.random() * employees.length)],
            'Designation': designation[Math.floor(Math.random() *
designation.length)],
            'Location': location[Math.floor(Math.random() *
location.length)],
            'Status': status[Math.floor(Math.random() * status.length)],
            'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
            'Rating': Math.floor(Math.random() * 5),
            'Software': Math.floor(Math.random() * 100),
            'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
            'CurrentSalary': Math.floor((Math.random() * 100000)),
            'Address': address[Math.floor(Math.random() * address.length)],
        });
        let employee: string = 'Employees';
        let emp: string = tradeData[i - 1][employee];
        let sName: string = emp.substr(0, emp.indexOf(' ')).toLowerCase();
        let empmail: string = 'Mail';
        tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }
    return tradeData;
}
export let tradeData: Object[] = [
    {
        "EmployeeID": 10001,
        "Employees": "Laura Nancy",
        "Designation": "Designer",
        "Location": "France",
        "Status": "Inactive",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
        "Software": 69,
        "EmployeeImg": "usermale",
        "CurrentSalary": 84194,
        "Address": "Taucherstraße 10",
    }

```

```

    "Mail": "laura15@jourrapide.com"
  },
  {
    "EmployeeID": 10002,
    "Employees": "Zachery Van",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 99,
    "EmployeeImg": "usermale",
    "CurrentSalary": 55349,
    "Address": "5a Ave. Los Palos Grandes",
    "Mail": "zachery109@sample.com"
  },
  {
    "EmployeeID": 10003,
    "Employees": "Rose Fuller",
    "Designation": "CFO",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 1,
    "EmployeeImg": "usermale",
    "CurrentSalary": 16477,
    "Address": "2817 Milton Dr.",
    "Mail": "rose55@rpy.com"
  },
  {
    "EmployeeID": 10004,
    "Employees": "Jack Bergs",
    "Designation": "Manager",
    "Location": "Mexico",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 36,
    "EmployeeImg": "usermale",
    "CurrentSalary": 49040,
    "Address": "2, rue du Commerce",
    "Mail": "jack30@sample.com"
  },
  {
    "EmployeeID": 10005,
    "Employees": "Vinet Bergs",
    "Designation": "Program Directory",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 39,
    "EmployeeImg": "usermale",
    "CurrentSalary": 5495,
    "Address": "Rua da Panificadora, 12",
    "Mail": "vinet32@jourrapide.com"
  }

```

```
},
{
  "EmployeeID": 10006,
  "Employees": "Buchanan Van",
  "Designation": "Designer",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 4,
  "Software": 78,
  "EmployeeImg": "usermale",
  "CurrentSalary": 42182,
  "Address": "24, place Kléber",
  "Mail": "buchanan18@mail.com"
},
{
  "EmployeeID": 10007,
  "Employees": "Dodsworth Nancy",
  "Designation": "Project Lead",
  "Location": "USA",
  "Status": "Inactive",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 0,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 35776,
  "Address": "Rua do Paço, 67",
  "Mail": "dodsworth84@mail.com"
},
{
  "EmployeeID": 10008,
  "Employees": "Laura Jack",
  "Designation": "Developer",
  "Location": "Austria",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 89,
  "EmployeeImg": "usermale",
  "CurrentSalary": 25108,
  "Address": "Rua da Panificadora, 12",
  "Mail": "laura82@mail.com"
},
{
  "EmployeeID": 10009,
  "Employees": "Anne Fuller",
  "Designation": "Program Directory",
  "Location": "Mexico",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 0,
  "Software": 19,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 32568,
  "Address": "Gran Vía, 1",
  "Mail": "anne97@jourrapide.com"
},
},
```

```
{
  "EmployeeID": 10010,
  "Employees": "Buchanan Andrew",
  "Designation": "Designer",
  "Location": "Austria",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 62,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 12320,
  "Address": "P.O. Box 555",
  "Mail": "buchanan50@jourrapide.com"
},
{
  "EmployeeID": 10011,
  "Employees": "Andrew Janet",
  "Designation": "System Analyst",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 8,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 20890,
  "Address": "Starenweg 5",
  "Mail": "andrew63@mail.com"
},
{
  "EmployeeID": 10012,
  "Employees": "Margaret Tamer",
  "Designation": "System Analyst",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 4,
  "Software": 7,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 22337,
  "Address": "Magazinweg 7",
  "Mail": "margaret26@mail.com"
},
{
  "EmployeeID": 10013,
  "Employees": "Tamer Fuller",
  "Designation": "CFO",
  "Location": "Canada",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 78,
  "EmployeeImg": "usermale",
  "CurrentSalary": 89181,
  "Address": "Taucherstraße 10",
  "Mail": "tamer40@arpy.com"
},
{
```

```
"EmployeeID": 10014,  
"Employees": "Tamer Anne",  
"Designation": "CFO",  
"Location": "Sweden",  
"Status": "Active",  
"Trustworthiness": "Sufficient",  
"Rating": 0,  
"Software": 18,  
"EmployeeImg": "usermale",  
"CurrentSalary": 20998,  
"Address": "Taucherstraße 10",  
"Mail": "tamer68@arpy.com"  
},  
{  
  "EmployeeID": 10015,  
  "Employees": "Anton Davolio",  
  "Designation": "Project Lead",  
  "Location": "France",  
  "Status": "Active",  
  "Trustworthiness": "Sufficient",  
  "Rating": 4,  
  "Software": 8,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 48232,  
  "Address": "Luisenstr. 48",  
  "Mail": "anton46@mail.com"  
},  
{  
  "EmployeeID": 10016,  
  "Employees": "Buchanan Buchanan",  
  "Designation": "System Analyst",  
  "Location": "Austria",  
  "Status": "Inactive",  
  "Trustworthiness": "Perfect",  
  "Rating": 0,  
  "Software": 19,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 43041,  
  "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",  
  "Mail": "buchanan68@mail.com"  
},  
{  
  "EmployeeID": 10017,  
  "Employees": "King Buchanan",  
  "Designation": "Program Directory",  
  "Location": "Sweden",  
  "Status": "Active",  
  "Trustworthiness": "Sufficient",  
  "Rating": 0,  
  "Software": 44,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 25259,  
  "Address": "Magazinweg 7",  
  "Mail": "king80@jourrapide.com"  
},  
{  
  "EmployeeID": 10018,
```



```
"Employees": "Rose Michael",
"Designation": "Project Lead",
"Location": "Canada",
"Status": "Active",
"Trustworthiness": "Perfect",
"Rating": 4,
"Software": 31,
"EmployeeImg": "userfemale",
"CurrentSalary": 91156,
"Address": "Fauntleroy Circus",
"Mail": "rose75@mail.com"
},
{
  "EmployeeID": 10019,
  "Employees": "King Bergs",
  "Designation": "Developer",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 2,
  "Software": 29,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 28826,
  "Address": "2817 Milton Dr.",
  "Mail": "king57@jourrapide.com"
},
{
  "EmployeeID": 10020,
  "Employees": "Davolio Fuller",
  "Designation": "Designer",
  "Location": "Canada",
  "Status": "Inactive",
  "Trustworthiness": "Sufficient",
  "Rating": 3,
  "Software": 35,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 71035,
  "Address": "Gran Vía, 1",
  "Mail": "davolio29@arpy.com"
},
{
  "EmployeeID": 10021,
  "Employees": "Rose Rose",
  "Designation": "CFO",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 38,
  "EmployeeImg": "usermale",
  "CurrentSalary": 68123,
  "Address": "Rua do Mercado, 12",
  "Mail": "rose54@arpy.com"
},
{
  "EmployeeID": 10022,
  "Employees": "Andrew Michael",
```

```

    "Designation": "Program Directory",
    "Location": "UK",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 2,
    "Software": 61,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 75470,
    "Address": "2, rue du Commerce",
    "Mail": "andrew88@jourrapide.com"
  },
  {
    "EmployeeID": 10023,
    "Employees": "Davolio Kathryn",
    "Designation": "Manager",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 25,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25234,
    "Address": "Hauptstr. 31",
    "Mail": "davolio42@sample.com"
  },
  {
    "EmployeeID": 10024,
    "Employees": "Anne Fleet",
    "Designation": "System Analyst",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 8341,
    "Address": "59 rue de l'Abbaye",
    "Mail": "anne86@arpy.com"
  },
  {
    "EmployeeID": 10025,
    "Employees": "Margaret Andrew",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 51,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84975,
    "Address": "P.O. Box 555",
    "Mail": "margaret41@arpy.com"
  },
  {
    "EmployeeID": 10026,
    "Employees": "Kathryn Laura",
    "Designation": "Project Lead",

```

```
"Location": "Austria",
"Status": "Active",
"Trustworthiness": "Insufficient",
"Rating": 3,
"Software": 48,
"EmployeeImg": "usermale",
"CurrentSalary": 97282,
"Address": "Avda. Azteca 123",
"Mail": "kathryn82@rpy.com"
},
{
  "EmployeeID": 10027,
  "Employees": "Michael Michael",
  "Designation": "Developer",
  "Location": "UK",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 4,
  "Software": 16,
  "EmployeeImg": "usermale",
  "CurrentSalary": 4184,
  "Address": "Rua do Paço, 67",
  "Mail": "michael58@jourrapide.com"
},
{
  "EmployeeID": 10028,
  "Employees": "Leverling Vinet",
  "Designation": "Project Lead",
  "Location": "Germany",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 57,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 38370,
  "Address": "59 rue de l'Abbaye",
  "Mail": "leverling102@sample.com"
},
{
  "EmployeeID": 10029,
  "Employees": "Rose Jack",
  "Designation": "Developer",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 46,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 84790,
  "Address": "Rua do Mercado, 12",
  "Mail": "rose108@jourrapide.com"
},
{
  "EmployeeID": 10030,
  "Employees": "Vinet Van",
  "Designation": "Developer",
  "Location": "USA",
```

```

        "Status": "Active",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
        "Software": 40,
        "EmployeeImg": "usermale",
        "CurrentSalary": 71005,
        "Address": "Gran Vía, 1",
        "Mail": "vinet90@jourrapide.com"
    }
]

```

### Changing text in column headers

Using the [beforeCellRender](#) event, you can change the text in the column headers. In that event, you can use the `e-header-cell` class to identify the header cell element and update its text value.

The following code example shows how to change the text in the column headers.

#### APP.JSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
function App() {
    const spreadsheetRef = React.useRef(null);
    const beforeCellRenderHandler = (args) => {
        // Condition to check whether the rendered element is header cell.
        if (
            args.colIndex >= 0 &&
            args.colIndex <= 10 &&
            args.element.classList.contains('e-header-cell')
        ) {
            const text = 'custom header ' + args.colIndex.toString();
            // Add the custom text to the innerText of the element.
            args.element.innerText = text;
        }
    };
    return (
        <SpreadsheetComponent ref={spreadsheetRef}
        beforeCellRender={beforeCellRenderHandler}>
        </SpreadsheetComponent>
    );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

#### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, CellRenderEventArgs } from '@syncfusion/ej2-
react-spreadsheet';
function App() {

```

```

const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
const beforeCellRenderHandler = (args: CellRenderEventArgs): void => {
  // Condition to check whether the rendered element is header cell.
  if (
    args.colIndex >= 0 &&
    args.colIndex <= 10 &&
    args.element.classList.contains('e-header-cell')
  ) {
    const text = 'custom header ' + args.colIndex.toString();
    // Add the custom text to the innerText of the element.
    args.element.innerText = text;
  }
}
return (
  <SpreadsheetComponent ref={spreadsheetRef}
beforeCellRender={beforeCellRenderHandler}>
    </SpreadsheetComponent>
  );
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% enddraw %}

```

### Note

You can refer to our [React Spreadsheet](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Spreadsheet example](#) to know how to present and manipulate data.

### See Also

- [Hyperlink](#)
- [Sorting](#)
- [Filtering](#)

## Sort in React Spreadsheet component

Sorting helps arranging the data to a specific order in a selected range of cells. You can use the [allowSorting](#) property to enable or disable sorting functionality.

\* The default value for `allowSorting` property is `true`.

By default, the `sort` module is injected internally into Spreadsheet to perform sorting.

### Sort by cell value

In the active Spreadsheet, select a range of cells to sort by cell value. The range sort can be done by any of the following ways:

- Select the sort item in the Ribbon toolbar and choose the ascending or descending item.
- Right-click the sheet, select the sort item in the context menu and choose the ascending/descending item.
- Use the [sort\(\)](#) method programmatically.

The cell values can be sorted in the following orders:

- Ascending
- Descending

\* Ascending is the default order for sorting.

The `sort()` method with empty arguments will sort the selected range by active cell's column as sort column in ascending order.

\* The `beforeSort` event will be triggered before sorting the specified range.

\* The `sortComplete` event will be triggered after the sort action is completed successfully.

The following code example shows `sort` functionality in the Spreadsheet control.

#### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, SortEventArgs } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { defaultData } from './datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  const onDataBound = () => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet && spreadsheet.activeSheetIndex === 0 &&
!spreadsheet.isOpen) {
      spreadsheet.cellFormat({ fontWeight: 'bold' }, 'A1:H1');
      spreadsheet.sort({ containsHeader: true }, 'A1:H11');
    }
  };
  const onSortComplete = (args) => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.selectRange(args.range);
    }
  };
  return (
    <SpreadsheetComponent ref={spreadsheetRef} dataBound={onDataBound}
sortComplete={onSortComplete} >
      <SheetsDirective>
        <SheetDirective>
          <RangesDirective>
            <RangeDirective
dataSource={defaultData}></RangeDirective>
          </RangesDirective>
          <ColumnsDirective>
            <ColumnDirective width={180}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
          </ColumnsDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  );
}
```

```

        </SheetsDirective>
      </SpreadsheetComponent>
    );
  };
  export default App;
  const root = createRoot(document.getElementById('root'));
  root.render(<App />);
  {% endraw %}

```

## APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, SortEventArgs } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { defaultData } from './datasource';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  const onDataBound = (): void => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet && spreadsheet.activeSheetIndex === 0 &&
!spreadsheet.isOpen) {
      spreadsheet.cellFormat({ fontWeight: 'bold' }, 'A1:H1');
      spreadsheet.sort({ containsHeader: true }, 'A1:H11');
    }
  };
  const onSortComplete = (args: SortEventArgs): void => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.selectRange(args.range as string);
    }
  };
  return (
    <SpreadsheetComponent ref={spreadsheetRef} dataBound={onDataBound}
sortComplete={onSortComplete} >
      <SheetsDirective>
        <SheetDirective>
          <RangesDirective>
            <RangeDirective
dataSource={defaultData}></RangeDirective>
          </RangesDirective>
          <ColumnsDirective>
            <ColumnDirective width={180}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
          </ColumnsDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  );
};
export default App;
const root = createRoot(document.getElementById('root')!);

```

```
root.render(<App />);
{% endraw %}
```

### DATASOURCE.JSX

```
/**
 * Default data source
 */
export let defaultData = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
    Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
  { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
    Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
  { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
    AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
  { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
    '12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
    70 },
  { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
    Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
  { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
    Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
  { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
    Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
  { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
    Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
  { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
    Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];

/**
 * Grid datasource
 */
export function getTradeData(dataCount) {
  let employees = [
    'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
    'Fuller', 'Leverling', 'Peacock',
    'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
    'Dodsworth', 'Anne',
    'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
    'Jack', 'Rose'
  ];
  let designation = ['Manager', 'CFO', 'Designer', 'Developer', 'Program
    Directory', 'System Analyst', 'Project Lead'];
  let mail = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
    'jourrapide.com'];
  let location = ['UK', 'USA', 'Sweden', 'France', 'Canada', 'Argentina',
    'Austria', 'Germany', 'Mexico'];
  let status = ['Active', 'Inactive'];
  let trustworthiness = ['Perfect', 'Sufficient', 'Insufficient'];
  let tradeData = [];
  let address = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do Paço, 67',
    '2, rue du Commerce', 'Boulevard Tirou, 255',
    'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
    Mercado, 12',
```



```

    'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
    'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
    'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
    'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
    'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
    'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
    'Taucherstraße 10', 'Taucherstraße 10',
    'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
Circus', 'Av. dos Lusíadas, 23',
    'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
123', '2817 Milton Dr.'];
    let employeeimg = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
        dataCount = parseInt(dataCount, 10);
    }
    for (let i = 1; i <= dataCount; i++) {
        let code = 10000;
        tradeData.push({
            'EmployeeID': code + i,
            'Employees': employees[Math.floor(Math.random() *
employees.length)] + ' ' + employees[Math.floor(Math.random() *
employees.length)],
            'Designation': designation[Math.floor(Math.random() *
designation.length)],
            'Location': location[Math.floor(Math.random() *
location.length)],
            'Status': status[Math.floor(Math.random() * status.length)],
            'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
            'Rating': Math.floor(Math.random() * 5),
            'Software': Math.floor(Math.random() * 100),
            'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
            'CurrentSalary': Math.floor((Math.random() * 100000)),
            'Address': address[Math.floor(Math.random() * address.length)],
        });
        let employee = 'Employees';
        let emp = tradeData[i - 1][employee];
        let sName = emp.substr(0, emp.indexOf(' ')).toLowerCase();
        let empmail = 'Mail';
        tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }
    return tradeData;
}
export let tradeData = [
    {
        "EmployeeID": 10001,
        "Employees": "Laura Nancy",
        "Designation": "Designer",
        "Location": "France",
        "Status": "Inactive",
        "Trustworthiness": "Sufficient",

```

```

    "Rating": 0,
    "Software": 69,
    "EmployeeImg": "usermale",
    "CurrentSalary": 84194,
    "Address": "Taucherstraße 10",
    "Mail": "laura15@jourrapide.com"
  },
  {
    "EmployeeID": 10002,
    "Employees": "Zachery Van",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 99,
    "EmployeeImg": "usermale",
    "CurrentSalary": 55349,
    "Address": "5a Ave. Los Palos Grandes",
    "Mail": "zachery109@sample.com"
  },
  {
    "EmployeeID": 10003,
    "Employees": "Rose Fuller",
    "Designation": "CFO",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 1,
    "EmployeeImg": "usermale",
    "CurrentSalary": 16477,
    "Address": "2817 Milton Dr.",
    "Mail": "rose55@rpy.com"
  },
  {
    "EmployeeID": 10004,
    "Employees": "Jack Bergs",
    "Designation": "Manager",
    "Location": "Mexico",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 36,
    "EmployeeImg": "usermale",
    "CurrentSalary": 49040,
    "Address": "2, rue du Commerce",
    "Mail": "jack30@sample.com"
  },
  {
    "EmployeeID": 10005,
    "Employees": "Vinet Bergs",
    "Designation": "Program Directory",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,

```

```

    "Software": 39,
    "EmployeeImg": "usermale",
    "CurrentSalary": 5495,
    "Address": "Rua da Panificadora, 12",
    "Mail": "vinet32@jourrapide.com"
  },
  {
    "EmployeeID": 10006,
    "Employees": "Buchanan Van",
    "Designation": "Designer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 42182,
    "Address": "24, place Kléber",
    "Mail": "buchanan18@mail.com"
  },
  {
    "EmployeeID": 10007,
    "Employees": "Dodsworth Nancy",
    "Designation": "Project Lead",
    "Location": "USA",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 35776,
    "Address": "Rua do Paço, 67",
    "Mail": "dodsworth84@mail.com"
  },
  {
    "EmployeeID": 10008,
    "Employees": "Laura Jack",
    "Designation": "Developer",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 89,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25108,
    "Address": "Rua da Panificadora, 12",
    "Mail": "laura82@mail.com"
  },
  {
    "EmployeeID": 10009,
    "Employees": "Anne Fuller",
    "Designation": "Program Directory",
    "Location": "Mexico",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 0,
    "Software": 19,

```

```

      "EmployeeImg": "userfemale",
      "CurrentSalary": 32568,
      "Address": "Gran Vía, 1",
      "Mail": "anne97@jourrapide.com"
    },
    {
      "EmployeeID": 10010,
      "Employees": "Buchanan Andrew",
      "Designation": "Designer",
      "Location": "Austria",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 1,
      "Software": 62,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 12320,
      "Address": "P.O. Box 555",
      "Mail": "buchanan50@jourrapide.com"
    },
    {
      "EmployeeID": 10011,
      "Employees": "Andrew Janet",
      "Designation": "System Analyst",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 3,
      "Software": 8,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 20890,
      "Address": "Starenweg 5",
      "Mail": "andrew63@mail.com"
    },
    {
      "EmployeeID": 10012,
      "Employees": "Margaret Tamer",
      "Designation": "System Analyst",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 4,
      "Software": 7,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 22337,
      "Address": "Magazinweg 7",
      "Mail": "margaret26@mail.com"
    },
    {
      "EmployeeID": 10013,
      "Employees": "Tamer Fuller",
      "Designation": "CFO",
      "Location": "Canada",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 3,
      "Software": 78,
      "EmployeeImg": "usermale",

```

```
    "CurrentSalary": 89181,  
    "Address": "Taucherstraße 10",  
    "Mail": "tamer40@arpy.com"  
  },  
  {  
    "EmployeeID": 10014,  
    "Employees": "Tamer Anne",  
    "Designation": "CFO",  
    "Location": "Sweden",  
    "Status": "Active",  
    "Trustworthiness": "Sufficient",  
    "Rating": 0,  
    "Software": 18,  
    "EmployeeImg": "usermale",  
    "CurrentSalary": 20998,  
    "Address": "Taucherstraße 10",  
    "Mail": "tamer68@arpy.com"  
  },  
  {  
    "EmployeeID": 10015,  
    "Employees": "Anton Davolio",  
    "Designation": "Project Lead",  
    "Location": "France",  
    "Status": "Active",  
    "Trustworthiness": "Sufficient",  
    "Rating": 4,  
    "Software": 8,  
    "EmployeeImg": "userfemale",  
    "CurrentSalary": 48232,  
    "Address": "Luisenstr. 48",  
    "Mail": "anton46@mail.com"  
  },  
  {  
    "EmployeeID": 10016,  
    "Employees": "Buchanan Buchanan",  
    "Designation": "System Analyst",  
    "Location": "Austria",  
    "Status": "Inactive",  
    "Trustworthiness": "Perfect",  
    "Rating": 0,  
    "Software": 19,  
    "EmployeeImg": "usermale",  
    "CurrentSalary": 43041,  
    "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",  
    "Mail": "buchanan68@mail.com"  
  },  
  {  
    "EmployeeID": 10017,  
    "Employees": "King Buchanan",  
    "Designation": "Program Directory",  
    "Location": "Sweden",  
    "Status": "Active",  
    "Trustworthiness": "Sufficient",  
    "Rating": 0,  
    "Software": 44,  
    "EmployeeImg": "userfemale",  
    "CurrentSalary": 25259,
```

```

      "Address": "Magazinweg 7",
      "Mail": "king80@jourrapide.com"
    },
    {
      "EmployeeID": 10018,
      "Employees": "Rose Michael",
      "Designation": "Project Lead",
      "Location": "Canada",
      "Status": "Active",
      "Trustworthiness": "Perfect",
      "Rating": 4,
      "Software": 31,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 91156,
      "Address": "Fauntleroy Circus",
      "Mail": "rose75@mail.com"
    },
    {
      "EmployeeID": 10019,
      "Employees": "King Bergs",
      "Designation": "Developer",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Sufficient",
      "Rating": 2,
      "Software": 29,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 28826,
      "Address": "2817 Milton Dr.",
      "Mail": "king57@jourrapide.com"
    },
    {
      "EmployeeID": 10020,
      "Employees": "Davolio Fuller",
      "Designation": "Designer",
      "Location": "Canada",
      "Status": "Inactive",
      "Trustworthiness": "Sufficient",
      "Rating": 3,
      "Software": 35,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 71035,
      "Address": "Gran Vía, 1",
      "Mail": "davolio29@arpy.com"
    },
    {
      "EmployeeID": 10021,
      "Employees": "Rose Rose",
      "Designation": "CFO",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Perfect",
      "Rating": 3,
      "Software": 38,
      "EmployeeImg": "usermale",
      "CurrentSalary": 68123,
      "Address": "Rua do Mercado, 12",

```

```
      "Mail": "rose54@arpy.com"
    },
    {
      "EmployeeID": 10022,
      "Employees": "Andrew Michael",
      "Designation": "Program Directory",
      "Location": "UK",
      "Status": "Inactive",
      "Trustworthiness": "Insufficient",
      "Rating": 2,
      "Software": 61,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 75470,
      "Address": "2, rue du Commerce",
      "Mail": "andrew88@jourrapide.com"
    },
    {
      "EmployeeID": 10023,
      "Employees": "Davolio Kathryn",
      "Designation": "Manager",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Perfect",
      "Rating": 3,
      "Software": 25,
      "EmployeeImg": "usermale",
      "CurrentSalary": 25234,
      "Address": "Hauptstr. 31",
      "Mail": "davolio42@sample.com"
    },
    {
      "EmployeeID": 10024,
      "Employees": "Anne Fleet",
      "Designation": "System Analyst",
      "Location": "UK",
      "Status": "Active",
      "Trustworthiness": "Perfect",
      "Rating": 3,
      "Software": 0,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 8341,
      "Address": "59 rue de l'Abbaye",
      "Mail": "anne86@arpy.com"
    },
    {
      "EmployeeID": 10025,
      "Employees": "Margaret Andrew",
      "Designation": "System Analyst",
      "Location": "Germany",
      "Status": "Inactive",
      "Trustworthiness": "Insufficient",
      "Rating": 3,
      "Software": 51,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 84975,
      "Address": "P.O. Box 555",
      "Mail": "margaret41@arpy.com"
```

```
},
{
  "EmployeeID": 10026,
  "Employees": "Kathryn Laura",
  "Designation": "Project Lead",
  "Location": "Austria",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 48,
  "EmployeeImg": "usermale",
  "CurrentSalary": 97282,
  "Address": "Avda. Azteca 123",
  "Mail": "kathryn82@rpy.com"
},
{
  "EmployeeID": 10027,
  "Employees": "Michael Michael",
  "Designation": "Developer",
  "Location": "UK",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 4,
  "Software": 16,
  "EmployeeImg": "usermale",
  "CurrentSalary": 4184,
  "Address": "Rua do Paço, 67",
  "Mail": "michael58@jourrapide.com"
},
{
  "EmployeeID": 10028,
  "Employees": "Leverling Vinet",
  "Designation": "Project Lead",
  "Location": "Germany",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 57,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 38370,
  "Address": "59 rue de l'Abbaye",
  "Mail": "leverling102@sample.com"
},
{
  "EmployeeID": 10029,
  "Employees": "Rose Jack",
  "Designation": "Developer",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 46,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 84790,
  "Address": "Rua do Mercado, 12",
  "Mail": "roasel08@jourrapide.com"
},
}
```



```

{
  "EmployeeID": 10030,
  "Employees": "Vinet Van",
  "Designation": "Developer",
  "Location": "USA",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 40,
  "EmployeeImg": "usermale",
  "CurrentSalary": 71005,
  "Address": "Gran Vía, 1",
  "Mail": "vinet90@jourrapide.com"
}
];

```

### DATASOURCE.TSX

```

/**
 * Default data source
 */
export let defaultData: Object[] = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
    Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
  { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
    Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
  { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
  { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
'12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
70 },
  { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
    Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
  { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
    Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
  { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
    Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
  { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
    Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
  { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
    Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
/**
 * Grid datasource
 */
export function getTradeData(dataCount?: number): object {
  let employees: string[] = [
    'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
    'Fuller', 'Leverling', 'Peacock',
    'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
    'Dodsworth', 'Anne',
    'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
    'Jack', 'Rose'];

```

```

    let designation: string[] = ['Manager', 'CFO', 'Designer', 'Developer',
    'Program Directory', 'System Analyst', 'Project Lead'];
    let mail: string[] = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
    'jourrapide.com'];
    let location: string[] = ['UK', 'USA', 'Sweden', 'France', 'Canada',
    'Argentina', 'Austria', 'Germany', 'Mexico'];
    let status: string[] = ['Active', 'Inactive'];
    let trustworthiness: string[] = ['Perfect', 'Sufficient',
    'Insufficient'];
    let tradeData: Object[] = [];
    let address: string[] = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do
    Paço, 67', '2, rue du Commerce', 'Boulevard Tirou, 255',
    'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
    Mercado, 12',
    'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
    'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
    Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
    'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
    43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
    'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
    10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
    'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
    7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
    'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
    'Taucherstraße 10', 'Taucherstraße 10',
    'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
    Circus', 'Av. dos Lusíadas, 23',
    'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
    123', '2817 Milton Dr.'];
    let employeeimg: string[] = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
        dataCount = parseInt(dataCount, 10);
    }
    for (let i: number = 1; i <= dataCount; i++) {
        let code: any = 10000;
        tradeData.push({
            'EmployeeID': code + i,
            'Employees':
                employees[Math.floor(Math.random() * employees.length)] + '
' + employees[Math.floor(Math.random() * employees.length)],
            'Designation': designation[Math.floor(Math.random() *
    designation.length)],
            'Location': location[Math.floor(Math.random() *
    location.length)],
            'Status': status[Math.floor(Math.random() * status.length)],
            'Trustworthiness': trustworthiness[Math.floor(Math.random() *
    trustworthiness.length)],
            'Rating': Math.floor(Math.random() * 5),
            'Software': Math.floor(Math.random() * 100),
            'EmployeeImg': employeeimg[Math.floor(Math.random() *
    employeeimg.length)],
            'CurrentSalary': Math.floor((Math.random() * 100000)),
            'Address': address[Math.floor(Math.random() * address.length)],
        });
        let employee: string = 'Employees';
        let emp: string = tradeData[i - 1][employee];
        let sName: string = emp.substr(0, emp.indexOf(' ')).toLowerCase();

```

```

        let empmail: string = 'Mail';
        tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }
    return tradeData;
}
export let tradeData: Object[] = [
    {
        "EmployeeID": 10001,
        "Employees": "Laura Nancy",
        "Designation": "Designer",
        "Location": "France",
        "Status": "Inactive",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
        "Software": 69,
        "EmployeeImg": "usermale",
        "CurrentSalary": 84194,
        "Address": "Taucherstraße 10",
        "Mail": "laura15@jourrapide.com"
    },
    {
        "EmployeeID": 10002,
        "Employees": "Zachery Van",
        "Designation": "CFO",
        "Location": "Canada",
        "Status": "Inactive",
        "Trustworthiness": "Insufficient",
        "Rating": 3,
        "Software": 99,
        "EmployeeImg": "usermale",
        "CurrentSalary": 55349,
        "Address": "5a Ave. Los Palos Grandes",
        "Mail": "zachery109@sample.com"
    },
    {
        "EmployeeID": 10003,
        "Employees": "Rose Fuller",
        "Designation": "CFO",
        "Location": "France",
        "Status": "Active",
        "Trustworthiness": "Insufficient",
        "Rating": 1,
        "Software": 1,
        "EmployeeImg": "usermale",
        "CurrentSalary": 16477,
        "Address": "2817 Milton Dr.",
        "Mail": "rose55@rpy.com"
    },
    {
        "EmployeeID": 10004,
        "Employees": "Jack Bergs",
        "Designation": "Manager",
        "Location": "Mexico",
        "Status": "Inactive",
        "Trustworthiness": "Insufficient",
        "Rating": 3,
    }
]

```

```
"Software": 36,  
"EmployeeImg": "usermale",  
"CurrentSalary": 49040,  
"Address": "2, rue du Commerce",  
"Mail": "jack30@sample.com"  
},  
{  
  "EmployeeID": 10005,  
  "Employees": "Vinet Bergs",  
  "Designation": "Program Directory",  
  "Location": "UK",  
  "Status": "Active",  
  "Trustworthiness": "Insufficient",  
  "Rating": 1,  
  "Software": 39,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 5495,  
  "Address": "Rua da Panificadora, 12",  
  "Mail": "vinet32@jourrapide.com"  
},  
{  
  "EmployeeID": 10006,  
  "Employees": "Buchanan Van",  
  "Designation": "Designer",  
  "Location": "Germany",  
  "Status": "Active",  
  "Trustworthiness": "Insufficient",  
  "Rating": 4,  
  "Software": 78,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 42182,  
  "Address": "24, place Kléber",  
  "Mail": "buchanan18@mail.com"  
},  
{  
  "EmployeeID": 10007,  
  "Employees": "Dodsworth Nancy",  
  "Designation": "Project Lead",  
  "Location": "USA",  
  "Status": "Inactive",  
  "Trustworthiness": "Sufficient",  
  "Rating": 0,  
  "Software": 0,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 35776,  
  "Address": "Rua do Paço, 67",  
  "Mail": "dodsworth84@mail.com"  
},  
{  
  "EmployeeID": 10008,  
  "Employees": "Laura Jack",  
  "Designation": "Developer",  
  "Location": "Austria",  
  "Status": "Inactive",  
  "Trustworthiness": "Perfect",  
  "Rating": 3,  
  "Software": 89,
```

```

      "EmployeeImg": "usermale",
      "CurrentSalary": 25108,
      "Address": "Rua da Panificadora, 12",
      "Mail": "laura82@mail.com"
    },
    {
      "EmployeeID": 10009,
      "Employees": "Anne Fuller",
      "Designation": "Program Directory",
      "Location": "Mexico",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 0,
      "Software": 19,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 32568,
      "Address": "Gran Vía, 1",
      "Mail": "anne97@jourrapide.com"
    },
    {
      "EmployeeID": 10010,
      "Employees": "Buchanan Andrew",
      "Designation": "Designer",
      "Location": "Austria",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 1,
      "Software": 62,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 12320,
      "Address": "P.O. Box 555",
      "Mail": "buchanan50@jourrapide.com"
    },
    {
      "EmployeeID": 10011,
      "Employees": "Andrew Janet",
      "Designation": "System Analyst",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 3,
      "Software": 8,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 20890,
      "Address": "Starenweg 5",
      "Mail": "andrew63@mail.com"
    },
    {
      "EmployeeID": 10012,
      "Employees": "Margaret Tamer",
      "Designation": "System Analyst",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 4,
      "Software": 7,
      "EmployeeImg": "userfemale",

```

```
"CurrentSalary": 22337,  
"Address": "Magazinweg 7",  
"Mail": "margaret26@mail.com"  
},  
{  
  "EmployeeID": 10013,  
  "Employees": "Tamer Fuller",  
  "Designation": "CFO",  
  "Location": "Canada",  
  "Status": "Active",  
  "Trustworthiness": "Insufficient",  
  "Rating": 3,  
  "Software": 78,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 89181,  
  "Address": "Taucherstraße 10",  
  "Mail": "tamer40@arpy.com"  
},  
{  
  "EmployeeID": 10014,  
  "Employees": "Tamer Anne",  
  "Designation": "CFO",  
  "Location": "Sweden",  
  "Status": "Active",  
  "Trustworthiness": "Sufficient",  
  "Rating": 0,  
  "Software": 18,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 20998,  
  "Address": "Taucherstraße 10",  
  "Mail": "tamer68@arpy.com"  
},  
{  
  "EmployeeID": 10015,  
  "Employees": "Anton Davolio",  
  "Designation": "Project Lead",  
  "Location": "France",  
  "Status": "Active",  
  "Trustworthiness": "Sufficient",  
  "Rating": 4,  
  "Software": 8,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 48232,  
  "Address": "Luisenstr. 48",  
  "Mail": "anton46@mail.com"  
},  
{  
  "EmployeeID": 10016,  
  "Employees": "Buchanan Buchanan",  
  "Designation": "System Analyst",  
  "Location": "Austria",  
  "Status": "Inactive",  
  "Trustworthiness": "Perfect",  
  "Rating": 0,  
  "Software": 19,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 43041,
```

```

    "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
    "Mail": "buchanan68@mail.com"
  },
  {
    "EmployeeID": 10017,
    "Employees": "King Buchanan",
    "Designation": "Program Directory",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 44,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 25259,
    "Address": "Magazinweg 7",
    "Mail": "king80@jourrapide.com"
  },
  {
    "EmployeeID": 10018,
    "Employees": "Rose Michael",
    "Designation": "Project Lead",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 31,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 91156,
    "Address": "Fauntleroy Circus",
    "Mail": "rose75@mail.com"
  },
  {
    "EmployeeID": 10019,
    "Employees": "King Bergs",
    "Designation": "Developer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 2,
    "Software": 29,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 28826,
    "Address": "2817 Milton Dr.",
    "Mail": "king57@jourrapide.com"
  },
  {
    "EmployeeID": 10020,
    "Employees": "Davolio Fuller",
    "Designation": "Designer",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 3,
    "Software": 35,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 71035,
    "Address": "Gran Vía, 1",

```

```
"Mail": "davolio29@arpy.com"
},
{
  "EmployeeID": 10021,
  "Employees": "Rose Rose",
  "Designation": "CFO",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 38,
  "EmployeeImg": "usermale",
  "CurrentSalary": 68123,
  "Address": "Rua do Mercado, 12",
  "Mail": "rose54@arpy.com"
},
{
  "EmployeeID": 10022,
  "Employees": "Andrew Michael",
  "Designation": "Program Directory",
  "Location": "UK",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 2,
  "Software": 61,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 75470,
  "Address": "2, rue du Commerce",
  "Mail": "andrew88@jourrapide.com"
},
{
  "EmployeeID": 10023,
  "Employees": "Davolio Kathryn",
  "Designation": "Manager",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 25,
  "EmployeeImg": "usermale",
  "CurrentSalary": 25234,
  "Address": "Hauptstr. 31",
  "Mail": "davolio42@sample.com"
},
{
  "EmployeeID": 10024,
  "Employees": "Anne Fleet",
  "Designation": "System Analyst",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 0,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 8341,
  "Address": "59 rue de l'Abbaye",
  "Mail": "anne86@arpy.com"
```



```
},
{
  "EmployeeID": 10025,
  "Employees": "Margaret Andrew",
  "Designation": "System Analyst",
  "Location": "Germany",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 51,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 84975,
  "Address": "P.O. Box 555",
  "Mail": "margaret41@arpy.com"
},
{
  "EmployeeID": 10026,
  "Employees": "Kathryn Laura",
  "Designation": "Project Lead",
  "Location": "Austria",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 48,
  "EmployeeImg": "usermale",
  "CurrentSalary": 97282,
  "Address": "Avda. Azteca 123",
  "Mail": "kathryn82@rpy.com"
},
{
  "EmployeeID": 10027,
  "Employees": "Michael Michael",
  "Designation": "Developer",
  "Location": "UK",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 4,
  "Software": 16,
  "EmployeeImg": "usermale",
  "CurrentSalary": 4184,
  "Address": "Rua do Paço, 67",
  "Mail": "michael58@jourrapide.com"
},
{
  "EmployeeID": 10028,
  "Employees": "Leverling Vinet",
  "Designation": "Project Lead",
  "Location": "Germany",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 57,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 38370,
  "Address": "59 rue de l'Abbaye",
  "Mail": "leverling102@sample.com"
},
}
```

```

{
  "EmployeeID": 10029,
  "Employees": "Rose Jack",
  "Designation": "Developer",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 46,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 84790,
  "Address": "Rua do Mercado, 12",
  "Mail": "rose108@jourrapide.com"
},
{
  "EmployeeID": 10030,
  "Employees": "Vinet Van",
  "Designation": "Developer",
  "Location": "USA",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 40,
  "EmployeeImg": "usermale",
  "CurrentSalary": 71005,
  "Address": "Gran Vía, 1",
  "Mail": "vinet90@jourrapide.com"
}
]

```

### Data contains header

You can specify whether the selected range of cells contains header. To specify, you need to set the [containsHeader](#) property to `true` and pass it as `sortOption` arguments of the `sort()` method.

\* If the `containsHeader` property is not set and active cell column's first cell value type is differed from the second cell value type, the first row data in the range are marked as column headers.

You can also enable or disable this property using `beforeSort` event arguments,

```

`ts
const beforeSort = (args: BeforeSortEventArgs): void => {
  args.sortOptions.containsHeader = true;
}
`

```

In the custom sort dialog, the `Data contains header` checkbox is checked on load. Thus, the default value for `containsHeader` is `true` in custom sort dialog.

### Case sensitive sort

The default sort functionality of Spreadsheet is a case insensitive sorting. When you want to perform sorting with case sensitive, you need to set the [caseSensitive](#) property to `true` and pass it as `sortOption` arguments of the `sort()` method.

Case sensitive sorting is applicable only for cells with alphabets. In ascending order sorting with case sensitive enabled, the cells with lower case text will be placed above the cells with upper case text.

\* The default value for the `caseSensitive` property is `false`.

You can also enable or disable this property using `beforeSort` event arguments,

```
`ts
const beforeSort = (args: BeforeSortEventArgs): void => {
  args.sortOptions.caseSensitive = true;
}
`
```

In the custom sort dialog, the `Case sensitive` checkbox is unchecked on load as the default value is `false`.

### Sort multiple columns

When you want to perform sorting on multiple columns, it can be done by any of the following ways:

- Select the `Custom sort...` menu item from the Ribbon toolbar item or context menu item.
- Use the `sort()` method programmatically by providing sort criteria.

\* The current sorting functionality supports sorting based on cell values only.

### Custom sort dialog

The custom sort dialog helps sorting multiple columns in the selected range by utilizing the rich UI. This dialog will be appeared while choosing the `Custom sort...` from the Ribbon item or context menu item. By default, sort criteria with the first column name from the selected range will be appeared in the dialog on initial load and it cannot be removed.

You can add multiple criteria using the `Add Column` button at the bottom of the dialog. Thus, multiple columns can be specified with different sort order. The newly added sort criteria items can be removed using the `delete` icons at the end of each items.

You can refer to the [Data contains header](#) topic to learn more about `Data contains header` checkbox. To learn more about `Case sensitive` checkbox, you can refer to [Case sensitive sort](#) topic.

### Passing sort criteria manually

The multi-column sorting can also be performed manually by passing sort options to the `sort()` method programmatically. The `sortOption` have the following arguments:

- [sortDescriptors](#) – Sort criteria collection that holds the collection of field name, sort order, and [sortComparer](#).
- `containsHeader` – Boolean argument that specifies whether the range has headers in it.
- `caseSensitive` – Boolean argument that specifies whether the range needs to consider case.

\* All the arguments are optional.

\* When a `sortDescriptor` is specified without field, the field of the first `sortDescriptor` from the collection will be assigned from active cell's column name and others will be ignored. Hence, it will act as single column sorting.

### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { tradeData } from './datasource';
function App() {
    const spreadsheetRef = React.useRef(null);
    const onDataBound = () => {
        let spreadsheet = spreadsheetRef.current;
        let sortDescriptors = [
            { field: 'F', order: 'Ascending' },
            { field: 'E', order: 'Ascending' },
            { field: 'C', order: 'Descending' }
        ];
        if (spreadsheet && spreadsheet.activeSheetIndex === 0) {
            spreadsheet.sort({ sortDescriptors: sortDescriptors,
containsHeader: true }, 'A1:H30');
        }
    };
    const onSortComplete = (args) => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            spreadsheet.selectRange(args.range);
        }
    };
    return (
        <SpreadsheetComponent ref={spreadsheetRef} dataBound={onDataBound}
sortComplete={onSortComplete}>
            <SheetsDirective>
                <SheetDirective>
                    <RangesDirective>
                        <RangeDirective
dataSource={tradeData}></RangeDirective>
                    </RangesDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={90}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
            </SheetsDirective>
        </SpreadsheetComponent>
    );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}
```

**APP.TSX**

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, SortDescriptor, SortEventArgs } from '@syncfusion/ej2-
react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { tradeData } from './datasource';
function App() {
    const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
    const onDataBound = (): void => {
        let spreadsheet = spreadsheetRef.current;
        let sortDescriptors: SortDescriptor[] = [
            { field: 'F', order: 'Ascending' },
            { field: 'E', order: 'Ascending' },
            { field: 'C', order: 'Descending' }
        ];
        if (spreadsheet && spreadsheet.activeSheetIndex === 0) {
            spreadsheet.sort({ sortDescriptors: sortDescriptors,
containsHeader: true }, 'A1:H30');
        }
    };
    const onSortComplete = (args: SortEventArgs) => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            spreadsheet.selectRange(args.range as string);
        }
    };
    return (
        <SpreadsheetComponent ref={spreadsheetRef} dataBound={onDataBound}
sortComplete={onSortComplete}>
            <SheetsDirective>
                <SheetDirective>
                    <RangesDirective>
                        <RangeDirective
dataSource={tradeData}></RangeDirective>
                    </RangesDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={90}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
            </SheetsDirective>
        </SpreadsheetComponent>
    );
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

**DATASOURCE.JSX**

```

/**
 * Default data source
 */
export let defaultData = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
    Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
  { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
    Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
  { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
  { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
'12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
70 },
  { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
    Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
  { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
    Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
  { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
    Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
  { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
    Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
  { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
    Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
/**
 * Grid datasource
 */
export function getTradeData(dataCount) {
  let employees = [
    'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
    'Fuller', 'Leverling', 'Peacock',
    'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
    'Dodsworth', 'Anne',
    'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
    'Jack', 'Rose'
  ];
  let designation = ['Manager', 'CFO', 'Designer', 'Developer', 'Program
Directory', 'System Analyst', 'Project Lead'];
  let mail = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
'journrapide.com'];
  let location = ['UK', 'USA', 'Sweden', 'France', 'Canada', 'Argentina',
'Austria', 'Germany', 'Mexico'];
  let status = ['Active', 'Inactive'];
  let trustworthiness = ['Perfect', 'Sufficient', 'Insufficient'];
  let tradeData = [];
  let address = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do Paço, 67',
'2, rue du Commerce', 'Boulevard Tirou, 255',
'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
Mercado, 12',
'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',

```

```

'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
43', '5a Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
'Taucherstraße 10', 'Taucherstraße 10',
'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
Circus', 'Av. dos Lusíadas, 23',
'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
123', '2817 Milton Dr.'];
let employeeimg = ['usermale', 'userfemale'];
if (typeof dataCount === 'string') {
  dataCount = parseInt(dataCount, 10);
}
for (let i = 1; i <= dataCount; i++) {
  let code = 10000;
  tradeData.push({
    'EmployeeID': code + i,
    'Employees': employees[Math.floor(Math.random() *
employees.length)] + ' ' + employees[Math.floor(Math.random() *
employees.length)],
    'Designation': designation[Math.floor(Math.random() *
designation.length)],
    'Location': location[Math.floor(Math.random() *
location.length)],
    'Status': status[Math.floor(Math.random() * status.length)],
    'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
    'Rating': Math.floor(Math.random() * 5),
    'Software': Math.floor(Math.random() * 100),
    'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
    'CurrentSalary': Math.floor((Math.random() * 100000)),
    'Address': address[Math.floor(Math.random() * address.length)],
  });
  let employee = 'Employees';
  let emp = tradeData[i - 1][employee];
  let sName = emp.substr(0, emp.indexOf(' ')).toLowerCase();
  let empmail = 'Mail';
  tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
}
return tradeData;
}
export let tradeData = [
  {
    "EmployeeID": 10001,
    "Employees": "Laura Nancy",
    "Designation": "Designer",
    "Location": "France",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 69,
    "EmployeeImg": "usermale",
  }

```

```
"CurrentSalary": 84194,  
"Address": "Taucherstraße 10",  
"Mail": "laura15@jourrapide.com"  
},  
{  
  "EmployeeID": 10002,  
  "Employees": "Zachery Van",  
  "Designation": "CFO",  
  "Location": "Canada",  
  "Status": "Inactive",  
  "Trustworthiness": "Insufficient",  
  "Rating": 3,  
  "Software": 99,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 55349,  
  "Address": "5ª Ave. Los Palos Grandes",  
  "Mail": "zachery109@sample.com"  
},  
{  
  "EmployeeID": 10003,  
  "Employees": "Rose Fuller",  
  "Designation": "CFO",  
  "Location": "France",  
  "Status": "Active",  
  "Trustworthiness": "Insufficient",  
  "Rating": 1,  
  "Software": 1,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 16477,  
  "Address": "2817 Milton Dr.",  
  "Mail": "rose55@rpy.com"  
},  
{  
  "EmployeeID": 10004,  
  "Employees": "Jack Bergs",  
  "Designation": "Manager",  
  "Location": "Mexico",  
  "Status": "Inactive",  
  "Trustworthiness": "Insufficient",  
  "Rating": 3,  
  "Software": 36,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 49040,  
  "Address": "2, rue du Commerce",  
  "Mail": "jack30@sample.com"  
},  
{  
  "EmployeeID": 10005,  
  "Employees": "Vinet Bergs",  
  "Designation": "Program Directory",  
  "Location": "UK",  
  "Status": "Active",  
  "Trustworthiness": "Insufficient",  
  "Rating": 1,  
  "Software": 39,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 5495,
```



```

      "Address": "Rua da Panificadora, 12",
      "Mail": "vinet32@jourrapide.com"
    },
    {
      "EmployeeID": 10006,
      "Employees": "Buchanan Van",
      "Designation": "Designer",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 4,
      "Software": 78,
      "EmployeeImg": "usermale",
      "CurrentSalary": 42182,
      "Address": "24, place Kléber",
      "Mail": "buchanan18@mail.com"
    },
    {
      "EmployeeID": 10007,
      "Employees": "Dodsworth Nancy",
      "Designation": "Project Lead",
      "Location": "USA",
      "Status": "Inactive",
      "Trustworthiness": "Sufficient",
      "Rating": 0,
      "Software": 0,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 35776,
      "Address": "Rua do Paço, 67",
      "Mail": "dodsworth84@mail.com"
    },
    {
      "EmployeeID": 10008,
      "Employees": "Laura Jack",
      "Designation": "Developer",
      "Location": "Austria",
      "Status": "Inactive",
      "Trustworthiness": "Perfect",
      "Rating": 3,
      "Software": 89,
      "EmployeeImg": "usermale",
      "CurrentSalary": 25108,
      "Address": "Rua da Panificadora, 12",
      "Mail": "laura82@mail.com"
    },
    {
      "EmployeeID": 10009,
      "Employees": "Anne Fuller",
      "Designation": "Program Directory",
      "Location": "Mexico",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 0,
      "Software": 19,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 32568,
      "Address": "Gran Vía, 1",

```

```
    "Mail": "anne97@jourrapide.com"
  },
  {
    "EmployeeID": 10010,
    "Employees": "Buchanan Andrew",
    "Designation": "Designer",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 62,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 12320,
    "Address": "P.O. Box 555",
    "Mail": "buchanan50@jourrapide.com"
  },
  {
    "EmployeeID": 10011,
    "Employees": "Andrew Janet",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 20890,
    "Address": "Starenweg 5",
    "Mail": "andrew63@mail.com"
  },
  {
    "EmployeeID": 10012,
    "Employees": "Margaret Tamer",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 7,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 22337,
    "Address": "Magazinweg 7",
    "Mail": "margaret26@mail.com"
  },
  {
    "EmployeeID": 10013,
    "Employees": "Tamer Fuller",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 89181,
    "Address": "Taucherstraße 10",
    "Mail": "tamer40@arpy.com"
```

```

    },
    {
      "EmployeeID": 10014,
      "Employees": "Tamer Anne",
      "Designation": "CFO",
      "Location": "Sweden",
      "Status": "Active",
      "Trustworthiness": "Sufficient",
      "Rating": 0,
      "Software": 18,
      "EmployeeImg": "usermale",
      "CurrentSalary": 20998,
      "Address": "Taucherstraße 10",
      "Mail": "tamer68@arpy.com"
    },
    {
      "EmployeeID": 10015,
      "Employees": "Anton Davolio",
      "Designation": "Project Lead",
      "Location": "France",
      "Status": "Active",
      "Trustworthiness": "Sufficient",
      "Rating": 4,
      "Software": 8,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 48232,
      "Address": "Luisenstr. 48",
      "Mail": "anton46@mail.com"
    },
    {
      "EmployeeID": 10016,
      "Employees": "Buchanan Buchanan",
      "Designation": "System Analyst",
      "Location": "Austria",
      "Status": "Inactive",
      "Trustworthiness": "Perfect",
      "Rating": 0,
      "Software": 19,
      "EmployeeImg": "usermale",
      "CurrentSalary": 43041,
      "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
      "Mail": "buchanan68@mail.com"
    },
    {
      "EmployeeID": 10017,
      "Employees": "King Buchanan",
      "Designation": "Program Directory",
      "Location": "Sweden",
      "Status": "Active",
      "Trustworthiness": "Sufficient",
      "Rating": 0,
      "Software": 44,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 25259,
      "Address": "Magazinweg 7",
      "Mail": "king80@jourrapide.com"
    },
  ],

```

```
{
  "EmployeeID": 10018,
  "Employees": "Rose Michael",
  "Designation": "Project Lead",
  "Location": "Canada",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 4,
  "Software": 31,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 91156,
  "Address": "Fauntleroy Circus",
  "Mail": "rose75@mail.com"
},
{
  "EmployeeID": 10019,
  "Employees": "King Bergs",
  "Designation": "Developer",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 2,
  "Software": 29,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 28826,
  "Address": "2817 Milton Dr.",
  "Mail": "king57@jourrapide.com"
},
{
  "EmployeeID": 10020,
  "Employees": "Davolio Fuller",
  "Designation": "Designer",
  "Location": "Canada",
  "Status": "Inactive",
  "Trustworthiness": "Sufficient",
  "Rating": 3,
  "Software": 35,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 71035,
  "Address": "Gran Vía, 1",
  "Mail": "davolio29@arpy.com"
},
{
  "EmployeeID": 10021,
  "Employees": "Rose Rose",
  "Designation": "CFO",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 38,
  "EmployeeImg": "usermale",
  "CurrentSalary": 68123,
  "Address": "Rua do Mercado, 12",
  "Mail": "rose54@arpy.com"
},
{
```

```
    "EmployeeID": 10022,  
    "Employees": "Andrew Michael",  
    "Designation": "Program Directory",  
    "Location": "UK",  
    "Status": "Inactive",  
    "Trustworthiness": "Insufficient",  
    "Rating": 2,  
    "Software": 61,  
    "EmployeeImg": "userfemale",  
    "CurrentSalary": 75470,  
    "Address": "2, rue du Commerce",  
    "Mail": "andrew88@jourrapide.com"  
  },  
  {  
    "EmployeeID": 10023,  
    "Employees": "Davolio Kathryn",  
    "Designation": "Manager",  
    "Location": "Germany",  
    "Status": "Active",  
    "Trustworthiness": "Perfect",  
    "Rating": 3,  
    "Software": 25,  
    "EmployeeImg": "usermale",  
    "CurrentSalary": 25234,  
    "Address": "Hauptstr. 31",  
    "Mail": "davolio42@sample.com"  
  },  
  {  
    "EmployeeID": 10024,  
    "Employees": "Anne Fleet",  
    "Designation": "System Analyst",  
    "Location": "UK",  
    "Status": "Active",  
    "Trustworthiness": "Perfect",  
    "Rating": 3,  
    "Software": 0,  
    "EmployeeImg": "userfemale",  
    "CurrentSalary": 8341,  
    "Address": "59 rue de l'Abbaye",  
    "Mail": "anne86@arpy.com"  
  },  
  {  
    "EmployeeID": 10025,  
    "Employees": "Margaret Andrew",  
    "Designation": "System Analyst",  
    "Location": "Germany",  
    "Status": "Inactive",  
    "Trustworthiness": "Insufficient",  
    "Rating": 3,  
    "Software": 51,  
    "EmployeeImg": "userfemale",  
    "CurrentSalary": 84975,  
    "Address": "P.O. Box 555",  
    "Mail": "margaret41@arpy.com"  
  },  
  {  
    "EmployeeID": 10026,
```

```
    "Employees": "Kathryn Laura",
    "Designation": "Project Lead",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 48,
    "EmployeeImg": "usermale",
    "CurrentSalary": 97282,
    "Address": "Avda. Azteca 123",
    "Mail": "kathryn82@rpy.com"
  },
  {
    "EmployeeID": 10027,
    "Employees": "Michael Michael",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 16,
    "EmployeeImg": "usermale",
    "CurrentSalary": 4184,
    "Address": "Rua do Paço, 67",
    "Mail": "michael58@jourrapide.com"
  },
  {
    "EmployeeID": 10028,
    "Employees": "Leverling Vinet",
    "Designation": "Project Lead",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 57,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 38370,
    "Address": "59 rue de l'Abbaye",
    "Mail": "leverling102@sample.com"
  },
  {
    "EmployeeID": 10029,
    "Employees": "Rose Jack",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 46,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84790,
    "Address": "Rua do Mercado, 12",
    "Mail": "rose108@jourrapide.com"
  },
  {
    "EmployeeID": 10030,
    "Employees": "Vinet Van",
```

```

        "Designation": "Developer",
        "Location": "USA",
        "Status": "Active",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
        "Software": 40,
        "EmployeeImg": "usermale",
        "CurrentSalary": 71005,
        "Address": "Gran Vía, 1",
        "Mail": "vinet90@jourrapide.com"
    }
];

```

### DATASOURCE.TSX

```

/**
 * Default data source
 */
export let defaultData: Object[] = [
    { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
      Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
    { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
      Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
    { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
      Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
    { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
      AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
    { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
      '12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
      70 },
    { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
      Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
    { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
      Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
    { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
      Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
    { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
      Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
    { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
      Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
/**
 * Grid datasource
 */
export function getTradeData(dataCount?: number): object {
    let employees: string[] = [
        'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
        'Fuller', 'Leverling', 'Peacock',
        'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
        'Dodsworth', 'Anne',
        'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
        'Jack', 'Rose'];
    let designation: string[] = ['Manager', 'CFO', 'Designer', 'Developer',
        'Program Directory', 'System Analyst', 'Project Lead'];
    let mail: string[] = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
        'jourrapide.com'];

```

```

    let location: string[] = ['UK', 'USA', 'Sweden', 'France', 'Canada',
    'Argentina', 'Austria', 'Germany', 'Mexico'];
    let status: string[] = ['Active', 'Inactive'];
    let trustworthiness: string[] = ['Perfect', 'Sufficient',
    'Insufficient'];
    let tradeData: Object[] = [];
    let address: string[] = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do
    Paço, 67', '2, rue du Commerce', 'Boulevard Tirou, 255',
    'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
    Mercado, 12',
    'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
    'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
    Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
    'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
    43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
    'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
    10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
    'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
    7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
    'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
    'Taucherstraße 10', 'Taucherstraße 10',
    'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
    Circus', 'Av. dos Lusíadas, 23',
    'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
    123', '2817 Milton Dr.'];
    let employeeimg: string[] = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
        dataCount = parseInt(dataCount, 10);
    }
    for (let i: number = 1; i <= dataCount; i++) {
        let code: any = 10000;
        tradeData.push({
            'EmployeeID': code + i,
            'Employees':
                employees[Math.floor(Math.random() * employees.length)] + '
' + employees[Math.floor(Math.random() * employees.length)],
            'Designation': designation[Math.floor(Math.random() *
            designation.length)],
            'Location': location[Math.floor(Math.random() *
            location.length)],
            'Status': status[Math.floor(Math.random() * status.length)],
            'Trustworthiness': trustworthiness[Math.floor(Math.random() *
            trustworthiness.length)],
            'Rating': Math.floor(Math.random() * 5),
            'Software': Math.floor(Math.random() * 100),
            'EmployeeImg': employeeimg[Math.floor(Math.random() *
            employeeimg.length)],
            'CurrentSalary': Math.floor((Math.random() * 100000)),
            'Address': address[Math.floor(Math.random() * address.length)],
        });
        let employee: string = 'Employees';
        let emp: string = tradeData[i - 1][employee];
        let sName: string = emp.substr(0, emp.indexOf(' ')).toLowerCase();
        let empmail: string = 'Mail';
        tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }

```



```
    return tradeData;
  }
export let tradeData: Object[] = [
  {
    "EmployeeID": 10001,
    "Employees": "Laura Nancy",
    "Designation": "Designer",
    "Location": "France",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 69,
    "EmployeeImg": "usermale",
    "CurrentSalary": 84194,
    "Address": "Taucherstraße 10",
    "Mail": "laura15@jourrapide.com"
  },
  {
    "EmployeeID": 10002,
    "Employees": "Zachery Van",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 99,
    "EmployeeImg": "usermale",
    "CurrentSalary": 55349,
    "Address": "5a Ave. Los Palos Grandes",
    "Mail": "zachery109@sample.com"
  },
  {
    "EmployeeID": 10003,
    "Employees": "Rose Fuller",
    "Designation": "CFO",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 1,
    "EmployeeImg": "usermale",
    "CurrentSalary": 16477,
    "Address": "2817 Milton Dr.",
    "Mail": "rose55@rpy.com"
  },
  {
    "EmployeeID": 10004,
    "Employees": "Jack Bergs",
    "Designation": "Manager",
    "Location": "Mexico",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 36,
    "EmployeeImg": "usermale",
    "CurrentSalary": 49040,
    "Address": "2, rue du Commerce",
```

```
"Mail": "jack30@sample.com"
},
{
  "EmployeeID": 10005,
  "Employees": "Vinet Bergs",
  "Designation": "Program Directory",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 39,
  "EmployeeImg": "usermale",
  "CurrentSalary": 5495,
  "Address": "Rua da Panificadora, 12",
  "Mail": "vinet32@jourrapide.com"
},
{
  "EmployeeID": 10006,
  "Employees": "Buchanan Van",
  "Designation": "Designer",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 4,
  "Software": 78,
  "EmployeeImg": "usermale",
  "CurrentSalary": 42182,
  "Address": "24, place Kléber",
  "Mail": "buchanan18@mail.com"
},
{
  "EmployeeID": 10007,
  "Employees": "Dodsworth Nancy",
  "Designation": "Project Lead",
  "Location": "USA",
  "Status": "Inactive",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 0,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 35776,
  "Address": "Rua do Paço, 67",
  "Mail": "dodsworth84@mail.com"
},
{
  "EmployeeID": 10008,
  "Employees": "Laura Jack",
  "Designation": "Developer",
  "Location": "Austria",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 89,
  "EmployeeImg": "usermale",
  "CurrentSalary": 25108,
  "Address": "Rua da Panificadora, 12",
  "Mail": "laura82@mail.com"
```

```
},
{
  "EmployeeID": 10009,
  "Employees": "Anne Fuller",
  "Designation": "Program Directory",
  "Location": "Mexico",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 0,
  "Software": 19,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 32568,
  "Address": "Gran Vía, 1",
  "Mail": "anne97@jourrapide.com"
},
{
  "EmployeeID": 10010,
  "Employees": "Buchanan Andrew",
  "Designation": "Designer",
  "Location": "Austria",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 62,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 12320,
  "Address": "P.O. Box 555",
  "Mail": "buchanan50@jourrapide.com"
},
{
  "EmployeeID": 10011,
  "Employees": "Andrew Janet",
  "Designation": "System Analyst",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 8,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 20890,
  "Address": "Starenweg 5",
  "Mail": "andrew63@mail.com"
},
{
  "EmployeeID": 10012,
  "Employees": "Margaret Tamer",
  "Designation": "System Analyst",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 4,
  "Software": 7,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 22337,
  "Address": "Magazinweg 7",
  "Mail": "margaret26@mail.com"
},
}
```

```

{
  "EmployeeID": 10013,
  "Employees": "Tamer Fuller",
  "Designation": "CFO",
  "Location": "Canada",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 78,
  "EmployeeImg": "usermale",
  "CurrentSalary": 89181,
  "Address": "Taucherstraße 10",
  "Mail": "tamer40@arpy.com"
},
{
  "EmployeeID": 10014,
  "Employees": "Tamer Anne",
  "Designation": "CFO",
  "Location": "Sweden",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 18,
  "EmployeeImg": "usermale",
  "CurrentSalary": 20998,
  "Address": "Taucherstraße 10",
  "Mail": "tamer68@arpy.com"
},
{
  "EmployeeID": 10015,
  "Employees": "Anton Davolio",
  "Designation": "Project Lead",
  "Location": "France",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 4,
  "Software": 8,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 48232,
  "Address": "Luisenstr. 48",
  "Mail": "anton46@mail.com"
},
{
  "EmployeeID": 10016,
  "Employees": "Buchanan Buchanan",
  "Designation": "System Analyst",
  "Location": "Austria",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 19,
  "EmployeeImg": "usermale",
  "CurrentSalary": 43041,
  "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
  "Mail": "buchanan68@mail.com"
},
{

```

```
"EmployeeID": 10017,  
"Employees": "King Buchanan",  
"Designation": "Program Directory",  
"Location": "Sweden",  
"Status": "Active",  
"Trustworthiness": "Sufficient",  
"Rating": 0,  
"Software": 44,  
"EmployeeImg": "userfemale",  
"CurrentSalary": 25259,  
"Address": "Magazinweg 7",  
"Mail": "king80@jourrapide.com"  
},  
{  
  "EmployeeID": 10018,  
  "Employees": "Rose Michael",  
  "Designation": "Project Lead",  
  "Location": "Canada",  
  "Status": "Active",  
  "Trustworthiness": "Perfect",  
  "Rating": 4,  
  "Software": 31,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 91156,  
  "Address": "Fauntleroy Circus",  
  "Mail": "rose75@mail.com"  
},  
{  
  "EmployeeID": 10019,  
  "Employees": "King Bergs",  
  "Designation": "Developer",  
  "Location": "Germany",  
  "Status": "Active",  
  "Trustworthiness": "Sufficient",  
  "Rating": 2,  
  "Software": 29,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 28826,  
  "Address": "2817 Milton Dr.",  
  "Mail": "king57@jourrapide.com"  
},  
{  
  "EmployeeID": 10020,  
  "Employees": "Davolio Fuller",  
  "Designation": "Designer",  
  "Location": "Canada",  
  "Status": "Inactive",  
  "Trustworthiness": "Sufficient",  
  "Rating": 3,  
  "Software": 35,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 71035,  
  "Address": "Gran Vía, 1",  
  "Mail": "davolio29@arpy.com"  
},  
{  
  "EmployeeID": 10021,
```

```

    "Employees": "Rose Rose",
    "Designation": "CFO",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 38,
    "EmployeeImg": "usermale",
    "CurrentSalary": 68123,
    "Address": "Rua do Mercado, 12",
    "Mail": "rose54@arpy.com"
  },
  {
    "EmployeeID": 10022,
    "Employees": "Andrew Michael",
    "Designation": "Program Directory",
    "Location": "UK",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 2,
    "Software": 61,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 75470,
    "Address": "2, rue du Commerce",
    "Mail": "andrew88@jourrapide.com"
  },
  {
    "EmployeeID": 10023,
    "Employees": "Davolio Kathryn",
    "Designation": "Manager",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 25,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25234,
    "Address": "Hauptstr. 31",
    "Mail": "davolio42@sample.com"
  },
  {
    "EmployeeID": 10024,
    "Employees": "Anne Fleet",
    "Designation": "System Analyst",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 8341,
    "Address": "59 rue de l'Abbaye",
    "Mail": "anne86@arpy.com"
  },
  {
    "EmployeeID": 10025,
    "Employees": "Margaret Andrew",

```

```
"Designation": "System Analyst",
"Location": "Germany",
"Status": "Inactive",
"Trustworthiness": "Insufficient",
"Rating": 3,
"Software": 51,
"EmployeeImg": "userfemale",
"CurrentSalary": 84975,
"Address": "P.O. Box 555",
"Mail": "margaret41@arpy.com"
},
{
  "EmployeeID": 10026,
  "Employees": "Kathryn Laura",
  "Designation": "Project Lead",
  "Location": "Austria",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 48,
  "EmployeeImg": "usermale",
  "CurrentSalary": 97282,
  "Address": "Avda. Azteca 123",
  "Mail": "kathryn82@rpy.com"
},
{
  "EmployeeID": 10027,
  "Employees": "Michael Michael",
  "Designation": "Developer",
  "Location": "UK",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 4,
  "Software": 16,
  "EmployeeImg": "usermale",
  "CurrentSalary": 4184,
  "Address": "Rua do Paço, 67",
  "Mail": "michael58@jourrapide.com"
},
{
  "EmployeeID": 10028,
  "Employees": "Leverling Vinet",
  "Designation": "Project Lead",
  "Location": "Germany",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 57,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 38370,
  "Address": "59 rue de l'Abbaye",
  "Mail": "leverling102@sample.com"
},
{
  "EmployeeID": 10029,
  "Employees": "Rose Jack",
  "Designation": "Developer",
```

```

      "Location": "UK",
      "Status": "Active",
      "Trustworthiness": "Perfect",
      "Rating": 0,
      "Software": 46,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 84790,
      "Address": "Rua do Mercado, 12",
      "Mail": "rose108@jourrapide.com"
    },
    {
      "EmployeeID": 10030,
      "Employees": "Vinet Van",
      "Designation": "Developer",
      "Location": "USA",
      "Status": "Active",
      "Trustworthiness": "Sufficient",
      "Rating": 0,
      "Software": 40,
      "EmployeeImg": "usermale",
      "CurrentSalary": 71005,
      "Address": "Gran Vía, 1",
      "Mail": "vinet90@jourrapide.com"
    }
  ]

```

### Custom sort comparer

The [sortDescriptor](#) holds the [sortComparer](#) property, which is a function and it is used to customize the sort comparer for specific sort criteria. Each `sortDescriptor` can be customized using the custom sort comparer function.

By customizing sort comparer, you can define the sort action as desired.

\* The `sortComparer` is an optional property of `sortDescriptor`.

For custom sort comparer example, refer to the [Sort a range by custom list](#) in the `how-to` section.

### Known error validations

The following errors have been handled for sorting,

- *Out of range validation:* When the selected range is not a used range of the active sheet, it is considered as invalid and the out of range alert with the message `Select a cell or range inside the used range and try again` will be displayed. No sort will be performed if the range is invalid.
- *Empty field validation:* When the sort criteria does not have a column selected (empty) in the custom sort dialog, it will become invalid, and an error message `Sort criteria column should not be empty` will be displayed on `OK` button click.
- *Duplicate field validation:* When the column names of added sort criteria are repeated more than once in the custom sort dialog, it will become invalid and an error message `is mentioned more than once. Duplicate columns must be removed` will be displayed on `OK` button click.



## Limitations

- Sorting is not supported with formula contained cells.

## Note

You can refer to our [React Spreadsheet](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Spreadsheet example](#) to know how to present and manipulate data.

## See Also

- [Sort a range by custom list](#)
- [Hyperlink](#)
- [Filtering](#)
- [Undo Redo](#)

## Filter in React Spreadsheet component

Filtering helps you to view specific rows in the spreadsheet by hiding the other rows. You can use the [allowFiltering](#) property to enable or disable filtering functionality.

\* The default value for `allowFiltering` property is `true`.

By default, the `Filter` module is injected internally into Spreadsheet to perform filtering.

## Apply filter on UI

In the active sheet, select a range of cells to filter by value of the cell. The filtering can be done by any of the following ways:

- Select the filter item in the Ribbon toolbar.
- Right-click the sheet, select the filter item in the context menu.
- Use the [applyFilter\(\)](#) method programmatically.
- Use `Ctrl + Shift + L` keyboard shortcut to apply the filter.

\* Use `Alt + Up/Down` keyboard shortcut to open the filter dialog.

## Filter by criteria

The [applyFilter\(\)](#) method will apply the filter UI, based on the predicate and range given in the arguments.

\* The [beforeFilter](#) event will be triggered before filtering the specified range.

\* The [filterComplete](#) event will be triggered after the filter action is completed successfully.

The following code example shows `filter` functionality in the Spreadsheet control.

## APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
```

```

import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { tradeData } from './datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  const onDataBound = () => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      if (spreadsheet.activeSheetIndex === 0) {
        let departments = ['Sweden', 'Canada', 'UK'];
        let predicateList = []
        departments.forEach((department) => { predicateList.push({
field: 'D', predicate: 'or', operator: 'equal', value: department }); })
        spreadsheet.applyFilter(predicateList);
      }
    }
  };
  return (
    <SpreadsheetComponent ref={spreadsheetRef} dataBound={onDataBound}>
      <SheetsDirective>
        <SheetDirective>
          <RangesDirective>
            <RangeDirective
dataSource={tradeData}></RangeDirective>
          </RangesDirective>
          <ColumnsDirective>
            <ColumnDirective width={90}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
          </ColumnsDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { PredicateModel } from '@syncfusion/ej2-react-grids';
import { tradeData } from './datasource';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  const onDataBound = (): void => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {

```

```

        if (spreadsheet.activeSheetIndex === 0) {
            let departments: string[] = ['Sweden', 'Canada', 'UK'];
            let predicateList: PredicateModel[] = []
            departments.forEach((department: string) => {
                predicateList.push({ field: 'D', predicate: 'or', operator: 'equal', value:
                department }); })
            spreadsheet.applyFilter(predicateList);
        }
    };
    return (
        <SpreadsheetComponent ref={spreadsheetRef} dataBound={onDataBound}>
            <SheetsDirective>
                <SheetDirective>
                    <RangesDirective>
                        <RangeDirective
dataSource={tradeData}></RangeDirective>
                    </RangesDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={90}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
            </SheetsDirective>
        </SpreadsheetComponent>
    );
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

### DATASOURCE.JSX

```

/**
 * Default data source
 */
export let defaultData = [
    { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
    { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
    Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
    { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
    Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
    { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
    { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
'12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
70 },
    { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
    Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
    { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
    Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
    { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
    Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },

```

```

    { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
      Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
    { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
      Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
  ];
  /**
   * Grid datasource
   */
  export function getTradeData(dataCount) {
    let employees = [
      'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
      'Fuller', 'Leverling', 'Peacock',
      'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
      'Dodsworth', 'Anne',
      'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
      'Jack', 'Rose'
    ];
    let designation = ['Manager', 'CFO', 'Designer', 'Developer', 'Program
      Directory', 'System Analyst', 'Project Lead'];
    let mail = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
      'jourrapide.com'];
    let location = ['UK', 'USA', 'Sweden', 'France', 'Canada', 'Argentina',
      'Austria', 'Germany', 'Mexico'];
    let status = ['Active', 'Inactive'];
    let trustworthiness = ['Perfect', 'Sufficient', 'Insufficient'];
    let tradeData = [];
    let address = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do Paço, 67',
      '2, rue du Commerce', 'Boulevard Tirou, 255',
      'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
      Mercado, 12',
      'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
      'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
      Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
      'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
      43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
      'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
      10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
      'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
      7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
      'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
      'Taucherstraße 10', 'Taucherstraße 10',
      'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
      Circus', 'Av. dos Lusíadas, 23',
      'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
      123', '2817 Milton Dr.'];
    let employeeimg = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
      dataCount = parseInt(dataCount, 10);
    }
    for (let i = 1; i <= dataCount; i++) {
      let code = 10000;
      tradeData.push({
        'EmployeeID': code + i,
        'Employees': employees[Math.floor(Math.random() *
          employees.length)] + ' ' + employees[Math.floor(Math.random() *
          employees.length)],

```

```

        'Designation': designation[Math.floor(Math.random() *
designation.length)],
        'Location': location[Math.floor(Math.random() *
location.length)],
        'Status': status[Math.floor(Math.random() * status.length)],
        'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
        'Rating': Math.floor(Math.random() * 5),
        'Software': Math.floor(Math.random() * 100),
        'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
        'CurrentSalary': Math.floor((Math.random() * 100000)),
        'Address': address[Math.floor(Math.random() * address.length)],
    });
    let employee = 'Employees';
    let emp = tradeData[i - 1][employee];
    let sName = emp.substr(0, emp.indexOf(' ')).toLowerCase();
    let empmail = 'Mail';
    tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }
    return tradeData;
}
export let tradeData = [
    {
        "EmployeeID": 10001,
        "Employees": "Laura Nancy",
        "Designation": "Designer",
        "Location": "France",
        "Status": "Inactive",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
        "Software": 69,
        "EmployeeImg": "usermale",
        "CurrentSalary": 84194,
        "Address": "Taucherstraße 10",
        "Mail": "laura15@jourrapide.com"
    },
    {
        "EmployeeID": 10002,
        "Employees": "Zachery Van",
        "Designation": "CFO",
        "Location": "Canada",
        "Status": "Inactive",
        "Trustworthiness": "Insufficient",
        "Rating": 3,
        "Software": 99,
        "EmployeeImg": "usermale",
        "CurrentSalary": 55349,
        "Address": "5a Ave. Los Palos Grandes",
        "Mail": "zachery109@sample.com"
    },
    {
        "EmployeeID": 10003,
        "Employees": "Rose Fuller",
        "Designation": "CFO",
        "Location": "France",

```

```

    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 1,
    "EmployeeImg": "usermale",
    "CurrentSalary": 16477,
    "Address": "2817 Milton Dr.",
    "Mail": "rose55@rpy.com"
  },
  {
    "EmployeeID": 10004,
    "Employees": "Jack Bergs",
    "Designation": "Manager",
    "Location": "Mexico",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 36,
    "EmployeeImg": "usermale",
    "CurrentSalary": 49040,
    "Address": "2, rue du Commerce",
    "Mail": "jack30@sample.com"
  },
  {
    "EmployeeID": 10005,
    "Employees": "Vinet Bergs",
    "Designation": "Program Directory",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 39,
    "EmployeeImg": "usermale",
    "CurrentSalary": 5495,
    "Address": "Rua da Panificadora, 12",
    "Mail": "vinet32@jourrapide.com"
  },
  {
    "EmployeeID": 10006,
    "Employees": "Buchanan Van",
    "Designation": "Designer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 42182,
    "Address": "24, place Kléber",
    "Mail": "buchanan18@mail.com"
  },
  {
    "EmployeeID": 10007,
    "Employees": "Dodsworth Nancy",
    "Designation": "Project Lead",
    "Location": "USA",
    "Status": "Inactive",

```

```

    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 35776,
    "Address": "Rua do Paço, 67",
    "Mail": "dodsworth84@mail.com"
  },
  {
    "EmployeeID": 10008,
    "Employees": "Laura Jack",
    "Designation": "Developer",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 89,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25108,
    "Address": "Rua da Panificadora, 12",
    "Mail": "laura82@mail.com"
  },
  {
    "EmployeeID": 10009,
    "Employees": "Anne Fuller",
    "Designation": "Program Directory",
    "Location": "Mexico",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 32568,
    "Address": "Gran Vía, 1",
    "Mail": "anne97@jourrapide.com"
  },
  {
    "EmployeeID": 10010,
    "Employees": "Buchanan Andrew",
    "Designation": "Designer",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 62,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 12320,
    "Address": "P.O. Box 555",
    "Mail": "buchanan50@jourrapide.com"
  },
  {
    "EmployeeID": 10011,
    "Employees": "Andrew Janet",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",

```

```
    "Rating": 3,  
    "Software": 8,  
    "EmployeeImg": "userfemale",  
    "CurrentSalary": 20890,  
    "Address": "Starenweg 5",  
    "Mail": "andrew63@mail.com"  
  },  
  {  
    "EmployeeID": 10012,  
    "Employees": "Margaret Tamer",  
    "Designation": "System Analyst",  
    "Location": "Germany",  
    "Status": "Active",  
    "Trustworthiness": "Insufficient",  
    "Rating": 4,  
    "Software": 7,  
    "EmployeeImg": "userfemale",  
    "CurrentSalary": 22337,  
    "Address": "Magazinweg 7",  
    "Mail": "margaret26@mail.com"  
  },  
  {  
    "EmployeeID": 10013,  
    "Employees": "Tamer Fuller",  
    "Designation": "CFO",  
    "Location": "Canada",  
    "Status": "Active",  
    "Trustworthiness": "Insufficient",  
    "Rating": 3,  
    "Software": 78,  
    "EmployeeImg": "usermale",  
    "CurrentSalary": 89181,  
    "Address": "Taucherstraße 10",  
    "Mail": "tamer40@arpy.com"  
  },  
  {  
    "EmployeeID": 10014,  
    "Employees": "Tamer Anne",  
    "Designation": "CFO",  
    "Location": "Sweden",  
    "Status": "Active",  
    "Trustworthiness": "Sufficient",  
    "Rating": 0,  
    "Software": 18,  
    "EmployeeImg": "usermale",  
    "CurrentSalary": 20998,  
    "Address": "Taucherstraße 10",  
    "Mail": "tamer68@arpy.com"  
  },  
  {  
    "EmployeeID": 10015,  
    "Employees": "Anton Davolio",  
    "Designation": "Project Lead",  
    "Location": "France",  
    "Status": "Active",  
    "Trustworthiness": "Sufficient",  
    "Rating": 4,
```



```

      "Software": 8,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 48232,
      "Address": "Luisenstr. 48",
      "Mail": "anton46@mail.com"
    },
    {
      "EmployeeID": 10016,
      "Employees": "Buchanan Buchanan",
      "Designation": "System Analyst",
      "Location": "Austria",
      "Status": "Inactive",
      "Trustworthiness": "Perfect",
      "Rating": 0,
      "Software": 19,
      "EmployeeImg": "usermale",
      "CurrentSalary": 43041,
      "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
      "Mail": "buchanan68@mail.com"
    },
    {
      "EmployeeID": 10017,
      "Employees": "King Buchanan",
      "Designation": "Program Directory",
      "Location": "Sweden",
      "Status": "Active",
      "Trustworthiness": "Sufficient",
      "Rating": 0,
      "Software": 44,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 25259,
      "Address": "Magazinweg 7",
      "Mail": "king80@jourrapide.com"
    },
    {
      "EmployeeID": 10018,
      "Employees": "Rose Michael",
      "Designation": "Project Lead",
      "Location": "Canada",
      "Status": "Active",
      "Trustworthiness": "Perfect",
      "Rating": 4,
      "Software": 31,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 91156,
      "Address": "Fauntleroy Circus",
      "Mail": "rose75@mail.com"
    },
    {
      "EmployeeID": 10019,
      "Employees": "King Bergs",
      "Designation": "Developer",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Sufficient",
      "Rating": 2,
      "Software": 29,

```

```
"EmployeeImg": "userfemale",
"CurrentSalary": 28826,
"Address": "2817 Milton Dr.",
"Mail": "king57@jourrapide.com"
},
{
  "EmployeeID": 10020,
  "Employees": "Davolio Fuller",
  "Designation": "Designer",
  "Location": "Canada",
  "Status": "Inactive",
  "Trustworthiness": "Sufficient",
  "Rating": 3,
  "Software": 35,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 71035,
  "Address": "Gran Vía, 1",
  "Mail": "davolio29@arpy.com"
},
{
  "EmployeeID": 10021,
  "Employees": "Rose Rose",
  "Designation": "CFO",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 38,
  "EmployeeImg": "usermale",
  "CurrentSalary": 68123,
  "Address": "Rua do Mercado, 12",
  "Mail": "rose54@arpy.com"
},
{
  "EmployeeID": 10022,
  "Employees": "Andrew Michael",
  "Designation": "Program Directory",
  "Location": "UK",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 2,
  "Software": 61,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 75470,
  "Address": "2, rue du Commerce",
  "Mail": "andrew88@jourrapide.com"
},
{
  "EmployeeID": 10023,
  "Employees": "Davolio Kathryn",
  "Designation": "Manager",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 25,
  "EmployeeImg": "usermale",
```

```
"CurrentSalary": 25234,  
"Address": "Hauptstr. 31",  
"Mail": "davolio42@sample.com"  
},  
{  
  "EmployeeID": 10024,  
  "Employees": "Anne Fleet",  
  "Designation": "System Analyst",  
  "Location": "UK",  
  "Status": "Active",  
  "Trustworthiness": "Perfect",  
  "Rating": 3,  
  "Software": 0,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 8341,  
  "Address": "59 rue de lAbbaye",  
  "Mail": "anne86@arpy.com"  
},  
{  
  "EmployeeID": 10025,  
  "Employees": "Margaret Andrew",  
  "Designation": "System Analyst",  
  "Location": "Germany",  
  "Status": "Inactive",  
  "Trustworthiness": "Insufficient",  
  "Rating": 3,  
  "Software": 51,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 84975,  
  "Address": "P.O. Box 555",  
  "Mail": "margaret41@arpy.com"  
},  
{  
  "EmployeeID": 10026,  
  "Employees": "Kathryn Laura",  
  "Designation": "Project Lead",  
  "Location": "Austria",  
  "Status": "Active",  
  "Trustworthiness": "Insufficient",  
  "Rating": 3,  
  "Software": 48,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 97282,  
  "Address": "Avda. Azteca 123",  
  "Mail": "kathryn82@rpy.com"  
},  
{  
  "EmployeeID": 10027,  
  "Employees": "Michael Michael",  
  "Designation": "Developer",  
  "Location": "UK",  
  "Status": "Inactive",  
  "Trustworthiness": "Perfect",  
  "Rating": 4,  
  "Software": 16,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 4184,
```

```

      "Address": "Rua do Paço, 67",
      "Mail": "michael58@jourrapide.com"
    },
    {
      "EmployeeID": 10028,
      "Employees": "Leverling Vinet",
      "Designation": "Project Lead",
      "Location": "Germany",
      "Status": "Inactive",
      "Trustworthiness": "Perfect",
      "Rating": 0,
      "Software": 57,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 38370,
      "Address": "59 rue de l'Abbaye",
      "Mail": "leverling102@sample.com"
    },
    {
      "EmployeeID": 10029,
      "Employees": "Rose Jack",
      "Designation": "Developer",
      "Location": "UK",
      "Status": "Active",
      "Trustworthiness": "Perfect",
      "Rating": 0,
      "Software": 46,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 84790,
      "Address": "Rua do Mercado, 12",
      "Mail": "rose108@jourrapide.com"
    },
    {
      "EmployeeID": 10030,
      "Employees": "Vinet Van",
      "Designation": "Developer",
      "Location": "USA",
      "Status": "Active",
      "Trustworthiness": "Sufficient",
      "Rating": 0,
      "Software": 40,
      "EmployeeImg": "usermale",
      "CurrentSalary": 71005,
      "Address": "Gran Vía, 1",
      "Mail": "vinet90@jourrapide.com"
    }
  ],
];

```

### DATASOURCE.TSX

```

/**
 * Default data source
 */
export let defaultData: Object[] = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },

```

```

    { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
      Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
    { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
      Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
    { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
    { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
'12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
70 },
    { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
      Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
    { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
      Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
    { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
      Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
    { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
      Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
    { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
      Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
  ];
  /**
   * Grid datasource
   */
  export function getTradeData(dataCount?: number): object {
    let employees: string[] = [
      'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
      'Fuller', 'Leverling', 'Peacock',
      'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
      'Dodsworth', 'Anne',
      'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
      'Jack', 'Rose'];
    let designation: string[] = ['Manager', 'CFO', 'Designer', 'Developer',
      'Program Directory', 'System Analyst', 'Project Lead'];
    let mail: string[] = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
      'jourrapide.com'];
    let location: string[] = ['UK', 'USA', 'Sweden', 'France', 'Canada',
      'Argentina', 'Austria', 'Germany', 'Mexico'];
    let status: string[] = ['Active', 'Inactive'];
    let trustworthiness: string[] = ['Perfect', 'Sufficient',
      'Insufficient'];
    let tradeData: Object[] = [];
    let address: string[] = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do
Paço, 67', '2, rue du Commerce', 'Boulevard Tirou, 255',
      'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
Mercado, 12',
      'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
      'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
      'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
      'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
      'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
      'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
      'Taucherstraße 10', 'Taucherstraße 10',

```

```

    'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
Circus', 'Av. dos Lusíadas, 23',
    'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
123', '2817 Milton Dr.'];
    let employeeimg: string[] = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
        dataCount = parseInt(dataCount, 10);
    }
    for (let i: number = 1; i <= dataCount; i++) {
        let code: any = 10000;
        tradeData.push({
            'EmployeeID': code + i,
            'Employees':
                employees[Math.floor(Math.random() * employees.length)] + '
' + employees[Math.floor(Math.random() * employees.length)],
            'Designation': designation[Math.floor(Math.random() *
designation.length)],
            'Location': location[Math.floor(Math.random() *
location.length)],
            'Status': status[Math.floor(Math.random() * status.length)],
            'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
            'Rating': Math.floor(Math.random() * 5),
            'Software': Math.floor(Math.random() * 100),
            'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
            'CurrentSalary': Math.floor((Math.random() * 100000)),
            'Address': address[Math.floor(Math.random() * address.length)],
        });
        let employee: string = 'Employees';
        let emp: string = tradeData[i - 1][employee];
        let sName: string = emp.substr(0, emp.indexOf(' ')).toLowerCase();
        let empmail: string = 'Mail';
        tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }
    return tradeData;
}
export let tradeData: Object[] = [
    {
        "EmployeeID": 10001,
        "Employees": "Laura Nancy",
        "Designation": "Designer",
        "Location": "France",
        "Status": "Inactive",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
        "Software": 69,
        "EmployeeImg": "usermale",
        "CurrentSalary": 84194,
        "Address": "Taucherstraße 10",
        "Mail": "laura15@jourrapide.com"
    },
    {
        "EmployeeID": 10002,
        "Employees": "Zachery Van",
        "Designation": "CFO",

```

```

    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 99,
    "EmployeeImg": "usermale",
    "CurrentSalary": 55349,
    "Address": "5ª Ave. Los Palos Grandes",
    "Mail": "zachery109@sample.com"
  },
  {
    "EmployeeID": 10003,
    "Employees": "Rose Fuller",
    "Designation": "CFO",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 1,
    "EmployeeImg": "usermale",
    "CurrentSalary": 16477,
    "Address": "2817 Milton Dr.",
    "Mail": "rose55@rpy.com"
  },
  {
    "EmployeeID": 10004,
    "Employees": "Jack Bergs",
    "Designation": "Manager",
    "Location": "Mexico",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 36,
    "EmployeeImg": "usermale",
    "CurrentSalary": 49040,
    "Address": "2, rue du Commerce",
    "Mail": "jack30@sample.com"
  },
  {
    "EmployeeID": 10005,
    "Employees": "Vinet Bergs",
    "Designation": "Program Directory",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 39,
    "EmployeeImg": "usermale",
    "CurrentSalary": 5495,
    "Address": "Rua da Panificadora, 12",
    "Mail": "vinet32@jourrapide.com"
  },
  {
    "EmployeeID": 10006,
    "Employees": "Buchanan Van",
    "Designation": "Designer",
    "Location": "Germany",

```

```
"Status": "Active",
"Trustworthiness": "Insufficient",
"Rating": 4,
"Software": 78,
"EmployeeImg": "usermale",
"CurrentSalary": 42182,
"Address": "24, place Kléber",
"Mail": "buchanan18@mail.com"
},
{
  "EmployeeID": 10007,
  "Employees": "Dodsworth Nancy",
  "Designation": "Project Lead",
  "Location": "USA",
  "Status": "Inactive",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 0,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 35776,
  "Address": "Rua do Paço, 67",
  "Mail": "dodsworth84@mail.com"
},
{
  "EmployeeID": 10008,
  "Employees": "Laura Jack",
  "Designation": "Developer",
  "Location": "Austria",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 89,
  "EmployeeImg": "usermale",
  "CurrentSalary": 25108,
  "Address": "Rua da Panificadora, 12",
  "Mail": "laura82@mail.com"
},
{
  "EmployeeID": 10009,
  "Employees": "Anne Fuller",
  "Designation": "Program Directory",
  "Location": "Mexico",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 0,
  "Software": 19,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 32568,
  "Address": "Gran Vía, 1",
  "Mail": "anne97@jourrapide.com"
},
{
  "EmployeeID": 10010,
  "Employees": "Buchanan Andrew",
  "Designation": "Designer",
  "Location": "Austria",
  "Status": "Active",
```



```
"Trustworthiness": "Insufficient",
"Rating": 1,
"Software": 62,
"EmployeeImg": "userfemale",
"CurrentSalary": 12320,
"Address": "P.O. Box 555",
"Mail": "buchanan50@jourrapide.com"
},
{
  "EmployeeID": 10011,
  "Employees": "Andrew Janet",
  "Designation": "System Analyst",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 8,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 20890,
  "Address": "Starenweg 5",
  "Mail": "andrew63@mail.com"
},
{
  "EmployeeID": 10012,
  "Employees": "Margaret Tamer",
  "Designation": "System Analyst",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 4,
  "Software": 7,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 22337,
  "Address": "Magazinweg 7",
  "Mail": "margaret26@mail.com"
},
{
  "EmployeeID": 10013,
  "Employees": "Tamer Fuller",
  "Designation": "CFO",
  "Location": "Canada",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 78,
  "EmployeeImg": "usermale",
  "CurrentSalary": 89181,
  "Address": "Taucherstraße 10",
  "Mail": "tamer40@arpy.com"
},
{
  "EmployeeID": 10014,
  "Employees": "Tamer Anne",
  "Designation": "CFO",
  "Location": "Sweden",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
```

```
"Rating": 0,
"Software": 18,
"EmployeeImg": "usermale",
"CurrentSalary": 20998,
"Address": "Taucherstraße 10",
"Mail": "tamer68@arpy.com"
},
{
  "EmployeeID": 10015,
  "Employees": "Anton Davolio",
  "Designation": "Project Lead",
  "Location": "France",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 4,
  "Software": 8,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 48232,
  "Address": "Luisenstr. 48",
  "Mail": "anton46@mail.com"
},
{
  "EmployeeID": 10016,
  "Employees": "Buchanan Buchanan",
  "Designation": "System Analyst",
  "Location": "Austria",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 19,
  "EmployeeImg": "usermale",
  "CurrentSalary": 43041,
  "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
  "Mail": "buchanan68@mail.com"
},
{
  "EmployeeID": 10017,
  "Employees": "King Buchanan",
  "Designation": "Program Directory",
  "Location": "Sweden",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 44,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 25259,
  "Address": "Magazinweg 7",
  "Mail": "king80@jourrapide.com"
},
{
  "EmployeeID": 10018,
  "Employees": "Rose Michael",
  "Designation": "Project Lead",
  "Location": "Canada",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 4,
```

```
"Software": 31,  
"EmployeeImg": "userfemale",  
"CurrentSalary": 91156,  
"Address": "Fauntleroy Circus",  
"Mail": "rose75@mail.com"  
},  
{  
  "EmployeeID": 10019,  
  "Employees": "King Bergs",  
  "Designation": "Developer",  
  "Location": "Germany",  
  "Status": "Active",  
  "Trustworthiness": "Sufficient",  
  "Rating": 2,  
  "Software": 29,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 28826,  
  "Address": "2817 Milton Dr.",  
  "Mail": "king57@jourrapide.com"  
},  
{  
  "EmployeeID": 10020,  
  "Employees": "Davolio Fuller",  
  "Designation": "Designer",  
  "Location": "Canada",  
  "Status": "Inactive",  
  "Trustworthiness": "Sufficient",  
  "Rating": 3,  
  "Software": 35,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 71035,  
  "Address": "Gran Vía, 1",  
  "Mail": "davolio29@arpy.com"  
},  
{  
  "EmployeeID": 10021,  
  "Employees": "Rose Rose",  
  "Designation": "CFO",  
  "Location": "Germany",  
  "Status": "Active",  
  "Trustworthiness": "Perfect",  
  "Rating": 3,  
  "Software": 38,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 68123,  
  "Address": "Rua do Mercado, 12",  
  "Mail": "rose54@arpy.com"  
},  
{  
  "EmployeeID": 10022,  
  "Employees": "Andrew Michael",  
  "Designation": "Program Directory",  
  "Location": "UK",  
  "Status": "Inactive",  
  "Trustworthiness": "Insufficient",  
  "Rating": 2,  
  "Software": 61,
```

```
"EmployeeImg": "userfemale",
"CurrentSalary": 75470,
"Address": "2, rue du Commerce",
"Mail": "andrew88@jourrapide.com"
},
{
  "EmployeeID": 10023,
  "Employees": "Davolio Kathryn",
  "Designation": "Manager",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 25,
  "EmployeeImg": "usermale",
  "CurrentSalary": 25234,
  "Address": "Hauptstr. 31",
  "Mail": "davolio42@sample.com"
},
{
  "EmployeeID": 10024,
  "Employees": "Anne Fleet",
  "Designation": "System Analyst",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 0,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 8341,
  "Address": "59 rue de l'Abbaye",
  "Mail": "anne86@arpy.com"
},
{
  "EmployeeID": 10025,
  "Employees": "Margaret Andrew",
  "Designation": "System Analyst",
  "Location": "Germany",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 51,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 84975,
  "Address": "P.O. Box 555",
  "Mail": "margaret41@arpy.com"
},
{
  "EmployeeID": 10026,
  "Employees": "Kathryn Laura",
  "Designation": "Project Lead",
  "Location": "Austria",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 48,
  "EmployeeImg": "usermale",
```

```
"CurrentSalary": 97282,  
"Address": "Avda. Azteca 123",  
"Mail": "kathryn82@rpy.com"  
},  
{  
  "EmployeeID": 10027,  
  "Employees": "Michael Michael",  
  "Designation": "Developer",  
  "Location": "UK",  
  "Status": "Inactive",  
  "Trustworthiness": "Perfect",  
  "Rating": 4,  
  "Software": 16,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 4184,  
  "Address": "Rua do Paço, 67",  
  "Mail": "michael58@jourrapide.com"  
},  
{  
  "EmployeeID": 10028,  
  "Employees": "Leverling Vinet",  
  "Designation": "Project Lead",  
  "Location": "Germany",  
  "Status": "Inactive",  
  "Trustworthiness": "Perfect",  
  "Rating": 0,  
  "Software": 57,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 38370,  
  "Address": "59 rue de l'Abbaye",  
  "Mail": "leverling102@sample.com"  
},  
{  
  "EmployeeID": 10029,  
  "Employees": "Rose Jack",  
  "Designation": "Developer",  
  "Location": "UK",  
  "Status": "Active",  
  "Trustworthiness": "Perfect",  
  "Rating": 0,  
  "Software": 46,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 84790,  
  "Address": "Rua do Mercado, 12",  
  "Mail": "rose108@jourrapide.com"  
},  
{  
  "EmployeeID": 10030,  
  "Employees": "Vinet Van",  
  "Designation": "Developer",  
  "Location": "USA",  
  "Status": "Active",  
  "Trustworthiness": "Sufficient",  
  "Rating": 0,  
  "Software": 40,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 71005,
```

```
"Address": "Gran Vía, 1",  
"Mail": "vinet90@jourrapide.com"  
}  
]
```

### Filter by cell value

To apply a filter for a cell value, right-click the cell and choose filter -> **Filter By Selected Cell's Value** option from the menu. It applies the filter based on the value of the selected cell in the current sheet.

### Clear filter

After applying filter to a certain column, you may want to clear it to make all filtered rows visible again. It can be done in the following ways,

- Choose **Clear** option in ribbon toolbar under **Filter and Sort**. It clears the filters applied in the spreadsheet for all fields.
- Use the [clearFilter\(\)](#) method programmatically, to clear the applied filters in spreadsheet for all fields.

### Clear filter on a field

After filtering, you can clear/reset the filter for a field alone. It can be done in the following ways,

- Click filter icon in the column's header and then choose **Clear Filter** option from the filter dialog.
- You can right-click on a filtered column cell and choose **Clear Filter from** . option from the context menu.
- Use the [clearFilter\(field\)](#) method programmatically, to clear the filter in a particular column.

### Reapply filter

When you want to reapply the filter after some changes happened in the rows. It can be done in the following ways,

- You can choose **Reapply** option in ribbon toolbar under **Filter and Sort** to reapply the filtered columns again.
- You can right-click on a filtered cell and choose **Reapply** option from the context menu. It reapplies the filters again in the Spreadsheet for all the fields.

### Known error validations

The following errors have been handled for filtering,

- *Out of range validation:* When the selected range is not a used range of the active sheet, it is considered as invalid and the out of range alert with the message **Select a cell or range inside the used range and try again** will be displayed. No filter will be performed if the range is invalid.

### Limitations

The following features have some limitations in Filter:

- Insert/delete row/column between the filter applied cells.
- Merge cells with filter.
- Copy/cut paste the filter applied cells.

#### Note

You can refer to our [React Spreadsheet](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Spreadsheet example](#) to know how to present and manipulate data.

#### See Also

- [Sorting](#)
- [Hyperlink](#)
- [Undo Redo](#)

#### Link in React Spreadsheet component

Hyperlink is used to navigate to web links or cell reference within the sheet or to other sheets in Spreadsheet. You can use the [allowHyperlink](#) property to enable or disable hyperlink functionality.

\* The default value for `allowHyperlink` property is `true`.

#### Insert Link

You can insert a hyperlink in a worksheet cell for quick access to related information.

##### User Interface:

In the active spreadsheet, click the cell where you want to create a hyperlink. Insert hyperlink can be done by any of the following ways:

- Select the INSERT tab in the Ribbon toolbar and choose the `Link` item.
- Right-click the cell and then click Hyperlink item in the context menu, or you can press Ctrl+K.
- Use the [addHyperlink\(\)](#) method programmatically.

#### Edit Hyperlink

You can change an existing hyperlink in your workbook by changing its destination or the text that is used to represent it.

##### User Interface:

In the active spreadsheet, Select the cell that contains the hyperlink that you want to change. Editing the hyperlink while opening the dialog can be done by any one of the following ways:

- Select the INSERT tab in the Ribbon toolbar and choose the `Link` item.
- Right-click the cell and then click Edit Hyperlink item in the context menu, or you can press Ctrl+K.

In the Edit Link dialog box, make the changes that you want, and click UPDATE.

#### Remove Hyperlink

Performing this operation remove a single hyperlink without losing the display text.

**User Interface:**

In the active spreadsheet, click the cell where you want to remove a hyperlink. remove hyperlink can be done by any of the following ways:

- Right-click the cell and then click Remove Hyperlink item in the context menu.
- Use the [removeHyperlink\(\)](#) method programmatically.

**How to change target attribute**

There is an event named `beforeHyperlinkClick` which triggers only on clicking hyperlink. You can customize where to open the hyperlink by using the `target` property in the arguments of that event.

**APP.JSX**

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
CellDirective, RowDirective } from '@syncfusion/ej2-react-spreadsheet';
import { ColumnsDirective, ColumnDirective, CellsDirective, RowsDirective }
from '@syncfusion/ej2-react-spreadsheet';
function App() {
  const onBeforeHyperlinkClick = (args) => {
    args.target = '_self'; //change target attribute
  };
  return (
    <SpreadsheetComponent beforeHyperlinkClick={onBeforeHyperlinkClick}>
      <SheetsDirective>
        <SheetDirective selectedRange='D13' name='PriceDetails'>
          <RowsDirective>
            <RowDirective>
              <CellsDirective>
                <CellDirective value='Item Name'
                </CellDirective>
                <CellDirective value='Quantity'
                </CellDirective>
                <CellDirective value='Price'
                </CellDirective>
                <CellDirective value='Amount'
                </CellDirective>
                <CellDirective value='Stock Detail'
                </CellDirective>
                <CellDirective value='Website'
                </CellDirective>
              </CellsDirective>
            </RowDirective>
            <RowDirective>
              <CellsDirective>
                <CellDirective value='Casual Shoes'
                </CellDirective>
                <CellDirective value='10' ></CellDirective>
                <CellDirective value='20' ></CellDirective>
                <CellDirective value='200' ></CellDirective>
                <CellDirective value='OUT OF STOCK'
                </CellDirective>
              </CellsDirective>
            </RowDirective>
          </RowsDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  );
}
```



```

                                <CellDirective value='Amazon'
hyperlink='https://www.amazon.com/'></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective>
                                    <CellsDirective>
                                        <CellDirective value='Sports Shoes'
></CellDirective>
                                        <CellDirective value='20' ></CellDirective>
                                        <CellDirective value='30' ></CellDirective>
                                        <CellDirective value='600'></CellDirective>
                                        <CellDirective value='IN STOCK'
hyperlink='Stock!A2:B2' ></CellDirective>
                                        <CellDirective value='Overstock'
hyperlink='https://www.overstock.com/'></CellDirective>
                                        </CellsDirective>
                                        </RowDirective>
                                        <RowDirective>
                                            <CellsDirective>
                                                <CellDirective value='Formal Shoes'
></CellDirective>
                                                <CellDirective value='20' ></CellDirective>
                                                <CellDirective value='15' ></CellDirective>
                                                <CellDirective value='300'></CellDirective>
                                                <CellDirective value='IN STOCK'
hyperlink='Stock!A3:B3' ></CellDirective>
                                                <CellDirective value='AliExpress'
hyperlink='https://www.aliexpress.com/'></CellDirective>
                                                </CellsDirective>
                                                </RowDirective>
                                                <RowDirective>
                                                    <CellsDirective>
                                                        <CellDirective value='Sandals & Floaters'
></CellDirective>
                                                        <CellDirective value='15' ></CellDirective>
                                                        <CellDirective value='20' ></CellDirective>
                                                        <CellDirective value='300' ></CellDirective>
                                                        <CellDirective value='OUT OF STOCK'
></CellDirective>
                                                        <CellDirective value='AliBaba'
hyperlink='https://www.alibaba.com/'></CellDirective>
                                                        </CellsDirective>
                                                        </RowDirective>
                                                        <RowDirective>
                                                            <CellsDirective>
                                                                <CellDirective value='Flip-Flops & Slippers'
></CellDirective>
                                                                <CellDirective value='30' ></CellDirective>
                                                                <CellDirective value='10' ></CellDirective>
                                                                <CellDirective value='300'></CellDirective>
                                                                <CellDirective value='IN STOCK'
hyperlink='Stock!A4:B4' ></CellDirective>
                                                                <CellDirective value='Taobao'
hyperlink='https://taobao.com/'></CellDirective>
                                                                </CellsDirective>
                                                                </RowDirective>
                                                                </RowsDirective>

```

```

        <ColumnsDirective>
            <ColumnDirective width={110}></ColumnDirective>
            <ColumnDirective width={115}></ColumnDirective>
            <ColumnDirective width={110}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={110}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
        </ColumnsDirective>
    </SheetDirective>
    <SheetDirective selectedRange='D13' name='Stock'>
        <RowsDirective>
            <RowDirective>
                <CellsDirective>
                    <CellDirective value='Item Name'
></CellDirective>
                    <CellDirective value='Available Count'
></CellDirective>
                </CellsDirective>
            </RowDirective>
            <RowDirective>
                <CellsDirective>
                    <CellDirective value='Casual Shoes'
></CellDirective>
                    <CellDirective value='10' ></CellDirective>
                </CellsDirective>
            </RowDirective>
            <RowDirective>
                <CellsDirective>
                    <CellDirective value='Sports Shoes'
></CellDirective>
                    <CellDirective value='20' ></CellDirective>
                </CellsDirective>
            </RowDirective>
            <RowDirective>
                <CellsDirective>
                    <CellDirective value='Formal Shoes'
></CellDirective>
                    <CellDirective value='20' ></CellDirective>
                </CellsDirective>
            </RowDirective>
            <RowDirective>
                <CellsDirective>
                    <CellDirective value='Sandals & Floaters'
></CellDirective>
                    <CellDirective value='15' ></CellDirective>
                </CellsDirective>
            </RowDirective>
            <RowDirective>
                <CellsDirective>
                    <CellDirective value='Flip-Flops & Slippers'
></CellDirective>
                    <CellDirective value='30' ></CellDirective>
                </CellsDirective>
            </RowDirective>
        </RowsDirective>
    </ColumnsDirective>
    <ColumnDirective width={110}></ColumnDirective>

```

```

        <ColumnDirective width={115}></ColumnDirective>
      </ColumnsDirective>
    </SheetDirective>
  </SheetsDirective>
</SpreadsheetComponent>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
CellDirective, RowDirective, BeforeHyperlinkArgs } from '@syncfusion/ej2-
react-spreadsheet';
import { ColumnsDirective, ColumnDirective, CellsDirective, RowsDirective }
from '@syncfusion/ej2-react-spreadsheet';
function App() {
  const onBeforeHyperlinkClick = (args: BeforeHyperlinkArgs): void => {
    args.target = '_self'; //change target attribute
  };
  return (
    <SpreadsheetComponent beforeHyperlinkClick={onBeforeHyperlinkClick}>
      <SheetsDirective>
        <SheetDirective selectedRange='D13' name='PriceDetails'>
          <RowsDirective>
            <RowDirective>
              <CellsDirective>
                <CellDirective value='Item Name'
></CellDirective>
                <CellDirective value='Quantity'
></CellDirective>
                <CellDirective value='Price'
></CellDirective>
                <CellDirective value='Amount'
></CellDirective>
                <CellDirective value='Stock Detail'
></CellDirective>
                <CellDirective value='Website'
></CellDirective>
              </CellsDirective>
            </RowDirective>
            <RowDirective>
              <CellsDirective>
                <CellDirective value='Casual Shoes'
></CellDirective>
                <CellDirective value='10' ></CellDirective>
                <CellDirective value='20' ></CellDirective>
                <CellDirective value='200' ></CellDirective>
                <CellDirective value='OUT OF STOCK'
></CellDirective>
              </CellsDirective>
            </RowDirective>
          </RowsDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  );
}

```

```

                                <CellDirective value='Amazon'
hyperlink='https://www.amazon.com/'></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective>
                                <CellsDirective>
                                <CellDirective value='Sports Shoes'
></CellDirective>
                                <CellDirective value='20' ></CellDirective>
                                <CellDirective value='30' ></CellDirective>
                                <CellDirective value='600'></CellDirective>
                                <CellDirective value='IN STOCK'
hyperlink='Stock!A2:B2' ></CellDirective>
                                <CellDirective value='Overstock'
hyperlink='https://www.overstock.com/'></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective>
                                <CellsDirective>
                                <CellDirective value='Formal Shoes'
></CellDirective>
                                <CellDirective value='20' ></CellDirective>
                                <CellDirective value='15' ></CellDirective>
                                <CellDirective value='300'></CellDirective>
                                <CellDirective value='IN STOCK'
hyperlink='Stock!A3:B3' ></CellDirective>
                                <CellDirective value='AliExpress'
hyperlink='https://www.aliexpress.com/'></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective>
                                <CellsDirective>
                                <CellDirective value='Sandals & Floaters'
></CellDirective>
                                <CellDirective value='15' ></CellDirective>
                                <CellDirective value='20' ></CellDirective>
                                <CellDirective value='300' ></CellDirective>
                                <CellDirective value='OUT OF STOCK'
></CellDirective>
                                <CellDirective value='AliBaba'
hyperlink='https://www.alibaba.com/'></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                <RowDirective>
                                <CellsDirective>
                                <CellDirective value='Flip-Flops & Slippers'
></CellDirective>
                                <CellDirective value='30' ></CellDirective>
                                <CellDirective value='10' ></CellDirective>
                                <CellDirective value='300'></CellDirective>
                                <CellDirective value='IN STOCK'
hyperlink='Stock!A4:B4' ></CellDirective>
                                <CellDirective value='Taobao'
hyperlink='https://taobao.com/'></CellDirective>
                                </CellsDirective>
                                </RowDirective>
                                </RowsDirective>

```

```

        <ColumnsDirective>
            <ColumnDirective width={110}></ColumnDirective>
            <ColumnDirective width={115}></ColumnDirective>
            <ColumnDirective width={110}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={110}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
        </ColumnsDirective>
    </SheetDirective>
    <SheetDirective selectedRange='D13' name='Stock'>
        <RowsDirective>
            <RowDirective>
                <CellsDirective>
                    <CellDirective value='Item Name'
></CellDirective>
                    <CellDirective value='Available Count'
></CellDirective>
                </CellsDirective>
            </RowDirective>
            <RowDirective>
                <CellsDirective>
                    <CellDirective value='Casual Shoes'
></CellDirective>
                    <CellDirective value='10' ></CellDirective>
                </CellsDirective>
            </RowDirective>
            <RowDirective>
                <CellsDirective>
                    <CellDirective value='Sports Shoes'
></CellDirective>
                    <CellDirective value='20' ></CellDirective>
                </CellsDirective>
            </RowDirective>
            <RowDirective>
                <CellsDirective>
                    <CellDirective value='Formal Shoes'
></CellDirective>
                    <CellDirective value='20' ></CellDirective>
                </CellsDirective>
            </RowDirective>
            <RowDirective>
                <CellsDirective>
                    <CellDirective value='Sandals & Floaters'
></CellDirective>
                    <CellDirective value='15' ></CellDirective>
                </CellsDirective>
            </RowDirective>
            <RowDirective>
                <CellsDirective>
                    <CellDirective value='Flip-Flops & Slippers'
></CellDirective>
                    <CellDirective value='30' ></CellDirective>
                </CellsDirective>
            </RowDirective>
        </RowsDirective>
    </ColumnsDirective>
    <ColumnDirective width={110}></ColumnDirective>

```

```
        <ColumnDirective width={115}></ColumnDirective>
      </ColumnsDirective>
    </SheetDirective>
  </SheetsDirective>
</SpreadsheetComponent>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% enddraw %}
```

## Limitations

- Inserting hyperlink not supported for multiple ranges.

## Note

You can refer to our [React Spreadsheet](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Spreadsheet example](#) to know how to present and manipulate data.

## See Also

- [Sorting](#)
- [Filtering](#)
- [Undo Redo](#)

## Clipboard in React Spreadsheet component

The Spreadsheet provides support for the clipboard operations (cut, copy, and paste). Clipboard operations can be enabled or disabled by setting the [enableClipboard](#) property in Spreadsheet.

By default, the `enableClipboard` property is true.

## Cut

It is used to cut the data from selected range of cells, rows or columns in a spreadsheet and make it available in the clipboard.

### User Interface:

Cut can be done in one of the following ways.

- Using Cut button in the Ribbon's HOME tab to perform cut operation.
- Using Cut option in the Context Menu.
- Using `Ctrl + X` | `Command + X` keyboard shortcut.
- Using the [cut](#) method.

## Copy

It is used to copy the data from selected range of cells, rows or columns in a spreadsheet and make it available in the clipboard.

### User Interface:

Copy can be done in one of the following ways.

- Using Copy button in the Ribbon's HOME tab to perform copy operation.
- Using Copy option in the Context Menu.
- Using **Ctrl + C** | **Command + C** keyboard shortcut.
- Using the [copy](#) method.

### Paste

It is used to paste the clipboard data to the selected range, rows or columns. You have the following options in Paste,

- **Paste Special** - You can paste the values with formatting.
- **Paste** - You can paste only the values without formatting.

It also performs for external clipboard operation. If you perform cut and paste, clipboard data will be cleared, whereas in copy and paste the clipboard contents will be maintained. If you perform paste inside the copied range, the clipboard data will be cleared.

### User Interface:

Paste can be done in one of the following ways.

- Using Paste button in the Ribbon's HOME tab to perform paste operation.
- Using Paste option in the Context Menu.
- Using **Ctrl + V** | **Command + V** keyboard shortcut.
- Using the [paste](#) method.

If you use the Keyboard shortcut key for cut (**Ctrl + X**) / copy (**Ctrl + C**) from other sources, you should use **Ctrl + V** shortcut while pasting into the spreadsheet.

### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { DropDownButtonComponent } from '@syncfusion/ej2-react-
splitbuttons';
import { defaultData } from './datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  const items = [
    { text: "Cut" },
    { text: "Copy" },
    { text: "Paste" }
  ];
  const itemSelect = (args) => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {

```

```

        if (args.item.text === 'Cut')
            spreadsheet.cut();
        if (args.item.text === 'Copy')
            spreadsheet.copy();
        if (args.item.text === 'Paste')
            spreadsheet.paste();
    }
};
React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
        spreadsheet.cellFormat({ fontWeight: 'bold', textAlign:
'center', verticalAlign: 'middle' }, 'A1:H1');
    }
}, []);
return (
    <div>
        <DropDownButtonComponent items={items} select={itemSelect}>
Clipboard </DropDownButtonComponent>
        <SpreadsheetComponent ref={spreadsheetRef}
enableClipboard={true}>
            <SheetsDirective>
                <SheetDirective>
                    <RangesDirective>
                        <RangeDirective
dataSource={defaultData}></RangeDirective>
                    </RangesDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                        <ColumnDirective width={92}></ColumnDirective>
                        <ColumnDirective width={96}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
            </SheetsDirective>
        </SpreadsheetComponent>
    </div>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, MenuSelectEventArgs } from '@syncfusion/ej2-react-
spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { DropDownButtonComponent } from '@syncfusion/ej2-react-
splitbuttons';
import { defaultData } from './datasource';

```



```

function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  const items = [
    { text: "Cut" },
    { text: "Copy" },
    { text: "Paste" }
  ];
  const itemSelect = (args: MenuSelectEventArgs): void => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      if (args.item.text === 'Cut')
        spreadsheet.cut();
      if (args.item.text === 'Copy')
        spreadsheet.copy();
      if (args.item.text === 'Paste')
        spreadsheet.paste();
    }
  };
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign:
'center', verticalAlign: 'middle' }, 'A1:H1');
    }
  }, []);
  return (
    <div>
      <DropDownButtonComponent items={items} select={itemSelect}>
Clipboard </DropDownButtonComponent>
      <SpreadsheetComponent ref={spreadsheetRef}
enableClipboard={true}>
        <SheetsDirective>
          <SheetDirective>
            <RangesDirective>
              <RangeDirective
dataSource={defaultData}></RangeDirective>
            </RangesDirective>
            <ColumnsDirective>
              <ColumnDirective width={130}></ColumnDirective>
              <ColumnDirective width={92}></ColumnDirective>
              <ColumnDirective width={96}></ColumnDirective>
            </ColumnsDirective>
          </SheetDirective>
        </SheetsDirective>
      </SpreadsheetComponent>
    </div>
  );
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

### DATASOURCE.JSX

```
/**
```

```

* Default data source
*/
export let defaultData = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
    Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
  { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
    Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
  { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54 AM',
    Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
  { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time: '12:43:59 AM',
    Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit: 70 },
  { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
    Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
  { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
    Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
  { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
    Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
  { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
    Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
  { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
    Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
/**
 * Grid datasource
 */
export function getTradeData(dataCount) {
  let employees = [
    'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
    'Fuller', 'Leverling', 'Peacock',
    'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
    'Dodsworth', 'Anne',
    'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
    'Jack', 'Rose'
  ];
  let designation = ['Manager', 'CFO', 'Designer', 'Developer', 'Program
    Directory', 'System Analyst', 'Project Lead'];
  let mail = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
    'jourrapide.com'];
  let location = ['UK', 'USA', 'Sweden', 'France', 'Canada', 'Argentina',
    'Austria', 'Germany', 'Mexico'];
  let status = ['Active', 'Inactive'];
  let trustworthiness = ['Perfect', 'Sufficient', 'Insufficient'];
  let tradeData = [];
  let address = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do Paço, 67',
    '2, rue du Commerce', 'Boulevard Tirou, 255',
    'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
    Mercado, 12',
    'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
    'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
    Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
    'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
    43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
    'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
    10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',

```

```

    'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
    'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
    'Taucherstraße 10', 'Taucherstraße 10',
    'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
Circus', 'Av. dos Lusíadas, 23',
    'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
123', '2817 Milton Dr.'];
    let employeeimg = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
        dataCount = parseInt(dataCount, 10);
    }
    for (let i = 1; i <= dataCount; i++) {
        let code = 10000;
        tradeData.push({
            'EmployeeID': code + i,
            'Employees': employees[Math.floor(Math.random() *
employees.length)] + ' ' + employees[Math.floor(Math.random() *
employees.length)],
            'Designation': designation[Math.floor(Math.random() *
designation.length)],
            'Location': location[Math.floor(Math.random() *
location.length)],
            'Status': status[Math.floor(Math.random() * status.length)],
            'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
            'Rating': Math.floor(Math.random() * 5),
            'Software': Math.floor(Math.random() * 100),
            'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
            'CurrentSalary': Math.floor((Math.random() * 100000)),
            'Address': address[Math.floor(Math.random() * address.length)],
        });
        let employee = 'Employees';
        let emp = tradeData[i - 1][employee];
        let sName = emp.substr(0, emp.indexOf(' ')).toLowerCase();
        let empmail = 'Mail';
        tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }
    return tradeData;
}
export let tradeData = [
    {
        "EmployeeID": 10001,
        "Employees": "Laura Nancy",
        "Designation": "Designer",
        "Location": "France",
        "Status": "Inactive",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
        "Software": 69,
        "EmployeeImg": "usermale",
        "CurrentSalary": 84194,
        "Address": "Taucherstraße 10",
        "Mail": "laura15@jourrapide.com"
    },

```

```
{
  "EmployeeID": 10002,
  "Employees": "Zachery Van",
  "Designation": "CFO",
  "Location": "Canada",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 99,
  "EmployeeImg": "usermale",
  "CurrentSalary": 55349,
  "Address": "5a Ave. Los Palos Grandes",
  "Mail": "zachery109@sample.com"
},
{
  "EmployeeID": 10003,
  "Employees": "Rose Fuller",
  "Designation": "CFO",
  "Location": "France",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 1,
  "EmployeeImg": "usermale",
  "CurrentSalary": 16477,
  "Address": "2817 Milton Dr.",
  "Mail": "rose55@rpy.com"
},
{
  "EmployeeID": 10004,
  "Employees": "Jack Bergs",
  "Designation": "Manager",
  "Location": "Mexico",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 36,
  "EmployeeImg": "usermale",
  "CurrentSalary": 49040,
  "Address": "2, rue du Commerce",
  "Mail": "jack30@sample.com"
},
{
  "EmployeeID": 10005,
  "Employees": "Vinet Bergs",
  "Designation": "Program Directory",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 39,
  "EmployeeImg": "usermale",
  "CurrentSalary": 5495,
  "Address": "Rua da Panificadora, 12",
  "Mail": "vinet32@jourrapide.com"
},
{
```

```
"EmployeeID": 10006,  
"Employees": "Buchanan Van",  
"Designation": "Designer",  
"Location": "Germany",  
"Status": "Active",  
"Trustworthiness": "Insufficient",  
"Rating": 4,  
"Software": 78,  
"EmployeeImg": "usermale",  
"CurrentSalary": 42182,  
"Address": "24, place Kléber",  
"Mail": "buchanan18@mail.com"  
},  
{  
  "EmployeeID": 10007,  
  "Employees": "Dodsworth Nancy",  
  "Designation": "Project Lead",  
  "Location": "USA",  
  "Status": "Inactive",  
  "Trustworthiness": "Sufficient",  
  "Rating": 0,  
  "Software": 0,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 35776,  
  "Address": "Rua do Paço, 67",  
  "Mail": "dodsworth84@mail.com"  
},  
{  
  "EmployeeID": 10008,  
  "Employees": "Laura Jack",  
  "Designation": "Developer",  
  "Location": "Austria",  
  "Status": "Inactive",  
  "Trustworthiness": "Perfect",  
  "Rating": 3,  
  "Software": 89,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 25108,  
  "Address": "Rua da Panificadora, 12",  
  "Mail": "laura82@mail.com"  
},  
{  
  "EmployeeID": 10009,  
  "Employees": "Anne Fuller",  
  "Designation": "Program Directory",  
  "Location": "Mexico",  
  "Status": "Active",  
  "Trustworthiness": "Insufficient",  
  "Rating": 0,  
  "Software": 19,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 32568,  
  "Address": "Gran Vía, 1",  
  "Mail": "anne97@jourrapide.com"  
},  
{  
  "EmployeeID": 10010,
```

```

    "Employees": "Buchanan Andrew",
    "Designation": "Designer",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 62,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 12320,
    "Address": "P.O. Box 555",
    "Mail": "buchanan50@jourrapide.com"
  },
  {
    "EmployeeID": 10011,
    "Employees": "Andrew Janet",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 20890,
    "Address": "Starenweg 5",
    "Mail": "andrew63@mail.com"
  },
  {
    "EmployeeID": 10012,
    "Employees": "Margaret Tamer",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 7,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 22337,
    "Address": "Magazinweg 7",
    "Mail": "margaret26@mail.com"
  },
  {
    "EmployeeID": 10013,
    "Employees": "Tamer Fuller",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 89181,
    "Address": "Taucherstraße 10",
    "Mail": "tamer40@arpy.com"
  },
  {
    "EmployeeID": 10014,
    "Employees": "Tamer Anne",

```

```
"Designation": "CFO",
"Location": "Sweden",
"Status": "Active",
"Trustworthiness": "Sufficient",
"Rating": 0,
"Software": 18,
"EmployeeImg": "usermale",
"CurrentSalary": 20998,
"Address": "Taucherstraße 10",
"Mail": "tamer68@arpy.com"
},
{
  "EmployeeID": 10015,
  "Employees": "Anton Davolio",
  "Designation": "Project Lead",
  "Location": "France",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 4,
  "Software": 8,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 48232,
  "Address": "Luisenstr. 48",
  "Mail": "anton46@mail.com"
},
{
  "EmployeeID": 10016,
  "Employees": "Buchanan Buchanan",
  "Designation": "System Analyst",
  "Location": "Austria",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 19,
  "EmployeeImg": "usermale",
  "CurrentSalary": 43041,
  "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
  "Mail": "buchanan68@mail.com"
},
{
  "EmployeeID": 10017,
  "Employees": "King Buchanan",
  "Designation": "Program Directory",
  "Location": "Sweden",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 44,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 25259,
  "Address": "Magazinweg 7",
  "Mail": "king80@jourrapide.com"
},
{
  "EmployeeID": 10018,
  "Employees": "Rose Michael",
  "Designation": "Project Lead",
```

```

      "Location": "Canada",
      "Status": "Active",
      "Trustworthiness": "Perfect",
      "Rating": 4,
      "Software": 31,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 91156,
      "Address": "Fauntleroy Circus",
      "Mail": "rose75@mail.com"
    },
    {
      "EmployeeID": 10019,
      "Employees": "King Bergs",
      "Designation": "Developer",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Sufficient",
      "Rating": 2,
      "Software": 29,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 28826,
      "Address": "2817 Milton Dr.",
      "Mail": "king57@jourrapide.com"
    },
    {
      "EmployeeID": 10020,
      "Employees": "Davolio Fuller",
      "Designation": "Designer",
      "Location": "Canada",
      "Status": "Inactive",
      "Trustworthiness": "Sufficient",
      "Rating": 3,
      "Software": 35,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 71035,
      "Address": "Gran Vía, 1",
      "Mail": "davolio29@arpy.com"
    },
    {
      "EmployeeID": 10021,
      "Employees": "Rose Rose",
      "Designation": "CFO",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Perfect",
      "Rating": 3,
      "Software": 38,
      "EmployeeImg": "usermale",
      "CurrentSalary": 68123,
      "Address": "Rua do Mercado, 12",
      "Mail": "rose54@arpy.com"
    },
    {
      "EmployeeID": 10022,
      "Employees": "Andrew Michael",
      "Designation": "Program Directory",
      "Location": "UK",

```



```
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 2,
    "Software": 61,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 75470,
    "Address": "2, rue du Commerce",
    "Mail": "andrew88@jourrapide.com"
  },
  {
    "EmployeeID": 10023,
    "Employees": "Davolio Kathryn",
    "Designation": "Manager",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 25,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25234,
    "Address": "Hauptstr. 31",
    "Mail": "davolio42@sample.com"
  },
  {
    "EmployeeID": 10024,
    "Employees": "Anne Fleet",
    "Designation": "System Analyst",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 8341,
    "Address": "59 rue de l'Abbaye",
    "Mail": "anne86@arpy.com"
  },
  {
    "EmployeeID": 10025,
    "Employees": "Margaret Andrew",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 51,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84975,
    "Address": "P.O. Box 555",
    "Mail": "margaret41@arpy.com"
  },
  {
    "EmployeeID": 10026,
    "Employees": "Kathryn Laura",
    "Designation": "Project Lead",
    "Location": "Austria",
    "Status": "Active",
```

```
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 48,
    "EmployeeImg": "usermale",
    "CurrentSalary": 97282,
    "Address": "Avda. Azteca 123",
    "Mail": "kathryn82@rpy.com"
  },
  {
    "EmployeeID": 10027,
    "Employees": "Michael Michael",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 16,
    "EmployeeImg": "usermale",
    "CurrentSalary": 4184,
    "Address": "Rua do Paço, 67",
    "Mail": "michael58@jourrapide.com"
  },
  {
    "EmployeeID": 10028,
    "Employees": "Leverling Vinet",
    "Designation": "Project Lead",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 57,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 38370,
    "Address": "59 rue de l'Abbaye",
    "Mail": "leverling102@sample.com"
  },
  {
    "EmployeeID": 10029,
    "Employees": "Rose Jack",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 46,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84790,
    "Address": "Rua do Mercado, 12",
    "Mail": "rosl08@jourrapide.com"
  },
  {
    "EmployeeID": 10030,
    "Employees": "Vinet Van",
    "Designation": "Developer",
    "Location": "USA",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
```

```

        "Rating": 0,
        "Software": 40,
        "EmployeeImg": "usermale",
        "CurrentSalary": 71005,
        "Address": "Gran Via, 1",
        "Mail": "vinet90@jourrapide.com"
    }
];

```

**DATASOURCE.TSX**

```

/**
 * Default data source
 */
export let defaultData: Object[] = [
    { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
    { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
    Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
    { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
    Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
    { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
    AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
    { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
    '12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
    70 },
    { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
    Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
    { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
    Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
    { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
    Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
    { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
    Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
    { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
    Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
/**
 * Grid datasource
 */
export function getTradeData(dataCount?: number): object {
    let employees: string[] = [
        'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
        'Fuller', 'Leverling', 'Peacock',
        'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
        'Dodsworth', 'Anne',
        'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
        'Jack', 'Rose'];
    let designation: string[] = ['Manager', 'CFO', 'Designer', 'Developer',
    'Program Directory', 'System Analyst', 'Project Lead'];
    let mail: string[] = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
    'jourrapide.com'];
    let location: string[] = ['UK', 'USA', 'Sweden', 'France', 'Canada',
    'Argentina', 'Austria', 'Germany', 'Mexico'];
    let status: string[] = ['Active', 'Inactive'];

```

```

    let trustworthiness: string[] = ['Perfect', 'Sufficient',
    'Insufficient'];
    let tradeData: Object[] = [];
    let address: string[] = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do
    Paço, 67', '2, rue du Commerce', 'Boulevard Tirou, 255',
    'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
    Mercado, 12',
    'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
    'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
    Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
    'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
    43', '5a Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
    'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
    10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
    'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
    7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
    'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
    'Taucherstraße 10', 'Taucherstraße 10',
    'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
    Circus', 'Av. dos Lusíadas, 23',
    'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
    123', '2817 Milton Dr.'];
    let employeeimg: string[] = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
        dataCount = parseInt(dataCount, 10);
    }
    for (let i: number = 1; i <= dataCount; i++) {
        let code: any = 10000;
        tradeData.push({
            'EmployeeID': code + i,
            'Employees':
                employees[Math.floor(Math.random() * employees.length)] + '
' + employees[Math.floor(Math.random() * employees.length)],
            'Designation': designation[Math.floor(Math.random() *
designation.length)],
            'Location': location[Math.floor(Math.random() *
location.length)],
            'Status': status[Math.floor(Math.random() * status.length)],
            'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
            'Rating': Math.floor(Math.random() * 5),
            'Software': Math.floor(Math.random() * 100),
            'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
            'CurrentSalary': Math.floor((Math.random() * 100000)),
            'Address': address[Math.floor(Math.random() * address.length)],
        });
        let employee: string = 'Employees';
        let emp: string = tradeData[i - 1][employee];
        let sName: string = emp.substr(0, emp.indexOf(' ')).toLowerCase();
        let empmail: string = 'Mail';
        tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }
    return tradeData;
}
export let tradeData: Object[] = [

```

```
{
  "EmployeeID": 10001,
  "Employees": "Laura Nancy",
  "Designation": "Designer",
  "Location": "France",
  "Status": "Inactive",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 69,
  "EmployeeImg": "usermale",
  "CurrentSalary": 84194,
  "Address": "Taucherstraße 10",
  "Mail": "laura15@jourrapide.com"
},
{
  "EmployeeID": 10002,
  "Employees": "Zachery Van",
  "Designation": "CFO",
  "Location": "Canada",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 99,
  "EmployeeImg": "usermale",
  "CurrentSalary": 55349,
  "Address": "5ª Ave. Los Palos Grandes",
  "Mail": "zachery109@sample.com"
},
{
  "EmployeeID": 10003,
  "Employees": "Rose Fuller",
  "Designation": "CFO",
  "Location": "France",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 1,
  "EmployeeImg": "usermale",
  "CurrentSalary": 16477,
  "Address": "2817 Milton Dr.",
  "Mail": "rose55@rpy.com"
},
{
  "EmployeeID": 10004,
  "Employees": "Jack Bergs",
  "Designation": "Manager",
  "Location": "Mexico",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 36,
  "EmployeeImg": "usermale",
  "CurrentSalary": 49040,
  "Address": "2, rue du Commerce",
  "Mail": "jack30@sample.com"
},
{
```

```
"EmployeeID": 10005,  
"Employees": "Vinet Bergs",  
"Designation": "Program Directory",  
"Location": "UK",  
"Status": "Active",  
"Trustworthiness": "Insufficient",  
"Rating": 1,  
"Software": 39,  
"EmployeeImg": "usermale",  
"CurrentSalary": 5495,  
"Address": "Rua da Panificadora, 12",  
"Mail": "vinet32@jourrapide.com"  
},  
{  
  "EmployeeID": 10006,  
  "Employees": "Buchanan Van",  
  "Designation": "Designer",  
  "Location": "Germany",  
  "Status": "Active",  
  "Trustworthiness": "Insufficient",  
  "Rating": 4,  
  "Software": 78,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 42182,  
  "Address": "24, place Kléber",  
  "Mail": "buchanan18@mail.com"  
},  
{  
  "EmployeeID": 10007,  
  "Employees": "Dodsworth Nancy",  
  "Designation": "Project Lead",  
  "Location": "USA",  
  "Status": "Inactive",  
  "Trustworthiness": "Sufficient",  
  "Rating": 0,  
  "Software": 0,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 35776,  
  "Address": "Rua do Paço, 67",  
  "Mail": "dodsworth84@mail.com"  
},  
{  
  "EmployeeID": 10008,  
  "Employees": "Laura Jack",  
  "Designation": "Developer",  
  "Location": "Austria",  
  "Status": "Inactive",  
  "Trustworthiness": "Perfect",  
  "Rating": 3,  
  "Software": 89,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 25108,  
  "Address": "Rua da Panificadora, 12",  
  "Mail": "laura82@mail.com"  
},  
{  
  "EmployeeID": 10009,
```

```

    "Employees": "Anne Fuller",
    "Designation": "Program Directory",
    "Location": "Mexico",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 32568,
    "Address": "Gran Vía, 1",
    "Mail": "anne97@jourrapide.com"
  },
  {
    "EmployeeID": 10010,
    "Employees": "Buchanan Andrew",
    "Designation": "Designer",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 62,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 12320,
    "Address": "P.O. Box 555",
    "Mail": "buchanan50@jourrapide.com"
  },
  {
    "EmployeeID": 10011,
    "Employees": "Andrew Janet",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 20890,
    "Address": "Starenweg 5",
    "Mail": "andrew63@mail.com"
  },
  {
    "EmployeeID": 10012,
    "Employees": "Margaret Tamer",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 7,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 22337,
    "Address": "Magazinweg 7",
    "Mail": "margaret26@mail.com"
  },
  {
    "EmployeeID": 10013,
    "Employees": "Tamer Fuller",

```

```

    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 89181,
    "Address": "Taucherstraße 10",
    "Mail": "tamer40@arpy.com"
  },
  {
    "EmployeeID": 10014,
    "Employees": "Tamer Anne",
    "Designation": "CFO",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 18,
    "EmployeeImg": "usermale",
    "CurrentSalary": 20998,
    "Address": "Taucherstraße 10",
    "Mail": "tamer68@arpy.com"
  },
  {
    "EmployeeID": 10015,
    "Employees": "Anton Davolio",
    "Designation": "Project Lead",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 4,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 48232,
    "Address": "Luisenstr. 48",
    "Mail": "anton46@mail.com"
  },
  {
    "EmployeeID": 10016,
    "Employees": "Buchanan Buchanan",
    "Designation": "System Analyst",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "usermale",
    "CurrentSalary": 43041,
    "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
    "Mail": "buchanan68@mail.com"
  },
  {
    "EmployeeID": 10017,
    "Employees": "King Buchanan",
    "Designation": "Program Directory",

```



```
"Location": "Sweden",
"Status": "Active",
"Trustworthiness": "Sufficient",
"Rating": 0,
"Software": 44,
"EmployeeImg": "userfemale",
"CurrentSalary": 25259,
"Address": "Magazinweg 7",
"Mail": "king80@jourrapide.com"
},
{
  "EmployeeID": 10018,
  "Employees": "Rose Michael",
  "Designation": "Project Lead",
  "Location": "Canada",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 4,
  "Software": 31,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 91156,
  "Address": "Fauntleroy Circus",
  "Mail": "rose75@mail.com"
},
{
  "EmployeeID": 10019,
  "Employees": "King Bergs",
  "Designation": "Developer",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 2,
  "Software": 29,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 28826,
  "Address": "2817 Milton Dr.",
  "Mail": "king57@jourrapide.com"
},
{
  "EmployeeID": 10020,
  "Employees": "Davolio Fuller",
  "Designation": "Designer",
  "Location": "Canada",
  "Status": "Inactive",
  "Trustworthiness": "Sufficient",
  "Rating": 3,
  "Software": 35,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 71035,
  "Address": "Gran Via, 1",
  "Mail": "davolio29@arpy.com"
},
{
  "EmployeeID": 10021,
  "Employees": "Rose Rose",
  "Designation": "CFO",
  "Location": "Germany",
```

```
"Status": "Active",
"Trustworthiness": "Perfect",
"Rating": 3,
"Software": 38,
"EmployeeImg": "usermale",
"CurrentSalary": 68123,
"Address": "Rua do Mercado, 12",
"Mail": "rose54@arpy.com"
},
{
  "EmployeeID": 10022,
  "Employees": "Andrew Michael",
  "Designation": "Program Directory",
  "Location": "UK",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 2,
  "Software": 61,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 75470,
  "Address": "2, rue du Commerce",
  "Mail": "andrew88@jourrapide.com"
},
{
  "EmployeeID": 10023,
  "Employees": "Davolio Kathryn",
  "Designation": "Manager",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 25,
  "EmployeeImg": "usermale",
  "CurrentSalary": 25234,
  "Address": "Hauptstr. 31",
  "Mail": "davolio42@sample.com"
},
{
  "EmployeeID": 10024,
  "Employees": "Anne Fleet",
  "Designation": "System Analyst",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 0,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 8341,
  "Address": "59 rue de l'Abbaye",
  "Mail": "anne86@arpy.com"
},
{
  "EmployeeID": 10025,
  "Employees": "Margaret Andrew",
  "Designation": "System Analyst",
  "Location": "Germany",
  "Status": "Inactive",
```

```
"Trustworthiness": "Insufficient",
"Rating": 3,
"Software": 51,
"EmployeeImg": "userfemale",
"CurrentSalary": 84975,
"Address": "P.O. Box 555",
"Mail": "margaret41@arpy.com"
},
{
  "EmployeeID": 10026,
  "Employees": "Kathryn Laura",
  "Designation": "Project Lead",
  "Location": "Austria",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 48,
  "EmployeeImg": "usermale",
  "CurrentSalary": 97282,
  "Address": "Avda. Azteca 123",
  "Mail": "kathryn82@rpy.com"
},
{
  "EmployeeID": 10027,
  "Employees": "Michael Michael",
  "Designation": "Developer",
  "Location": "UK",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 4,
  "Software": 16,
  "EmployeeImg": "usermale",
  "CurrentSalary": 4184,
  "Address": "Rua do Paço, 67",
  "Mail": "michael58@jourrapide.com"
},
{
  "EmployeeID": 10028,
  "Employees": "Leverling Vinet",
  "Designation": "Project Lead",
  "Location": "Germany",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 57,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 38370,
  "Address": "59 rue de l'Abbaye",
  "Mail": "leverling102@sample.com"
},
{
  "EmployeeID": 10029,
  "Employees": "Rose Jack",
  "Designation": "Developer",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Perfect",
```

```

    "Rating": 0,
    "Software": 46,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84790,
    "Address": "Rua do Mercado, 12",
    "Mail": "rose108@jourrapide.com"
  },
  {
    "EmployeeID": 10030,
    "Employees": "Vinet Van",
    "Designation": "Developer",
    "Location": "USA",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 40,
    "EmployeeImg": "usermale",
    "CurrentSalary": 71005,
    "Address": "Gran Vía, 1",
    "Mail": "vinet90@jourrapide.com"
  }
]

```

### Prevent the paste functionality

The following example shows, how to prevent the paste action in spreadsheet. In [actionBegin](#) event, you can set `cancel` argument as false in paste request type.

### APP.JSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { DropDownButtonComponent } from '@syncfusion/ej2-react-
splitbuttons';
import { defaultData } from '../datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  const items = [
    { text: 'Cut' },
    { text: 'Copy' },
    { text: 'Paste' }
  ];
  const onActionBegin = (pasteArgs) => {
    if (pasteArgs.args.action === 'clipboard' &&
    pasteArgs.args.eventArgs.requestType === 'paste') {
      pasteArgs.args.eventArgs.cancel = true;
    }
  };
  const itemSelect = (args) => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      if (args.item.text === 'Cut')

```

```

        spreadsheet.cut();
        if (args.item.text === 'Copy')
            spreadsheet.copy();
        if (args.item.text === 'Paste')
            spreadsheet.paste();
    }
};
React.useEffect(() => {
    if (spreadsheetRef.current) {
        spreadsheetRef.current.cellFormat({ fontWeight: 'bold',
textAlign: 'center', verticalAlign: 'middle' }, 'A1:H1');
    }
}, []);
return (
    <div>
        <DropDownButtonComponent items={items} select={itemSelect}>
Clipboard </DropDownButtonComponent>
        <SpreadsheetComponent ref={spreadsheetRef}
actionBegin={onActionBegin} enableClipboard={true} >
            <SheetsDirective>
                <SheetDirective>
                    <RangesDirective>
                        <RangeDirective
dataSource={defaultData}></RangeDirective>
                    </RangesDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                        <ColumnDirective width={92}></ColumnDirective>
                        <ColumnDirective width={96}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
            </SheetsDirective>
        </SpreadsheetComponent>
    </div>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

**APP.TSX**

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, MenuSelectEventArgs } from '@syncfusion/ej2-react-
spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { DropDownButtonComponent } from '@syncfusion/ej2-react-
splitbuttons';
import { defaultData } from './datasource';
function App() {
    const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);

```

```

const items = [
  { text: 'Cut' },
  { text: 'Copy' },
  { text: 'Paste' }
];

const onActionBegin = (pasteArgs: any): void => {
  if (pasteArgs.args.action === 'clipboard' &&
    pasteArgs.args.eventArgs.requestType === 'paste') {
    pasteArgs.args.eventArgs.cancel = true;
  }
};

const itemSelect = (args: MenuSelectEventArgs): void => {
  let spreadsheet = spreadsheetRef.current;
  if (spreadsheet) {
    if (args.item.text === 'Cut')
      spreadsheet.cut();
    if (args.item.text === 'Copy')
      spreadsheet.copy();
    if (args.item.text === 'Paste')
      spreadsheet.paste();
  }
};

React.useEffect(() => {
  if (spreadsheetRef.current) {
    spreadsheetRef.current.cellFormat({ fontWeight: 'bold',
    textAlign: 'center', verticalAlign: 'middle' }, 'A1:H1');
  }
}, []);

return (
  <div>
    <DropDownButtonComponent items={items} select={itemSelect}>
      Clipboard </DropDownButtonComponent>
    <SpreadsheetComponent ref={spreadsheetRef}
      actionBegin={onActionBegin} enableClipboard={true} >
      <SheetsDirective>
        <SheetDirective>
          <RangesDirective>
            <RangeDirective
              dataSource={defaultData}></RangeDirective>
          </RangesDirective>
          <ColumnsDirective>
            <ColumnDirective width={130}></ColumnDirective>
            <ColumnDirective width={92}></ColumnDirective>
            <ColumnDirective width={96}></ColumnDirective>
          </ColumnsDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  </div>
);
};

export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

**DATASOURCE.JSX**

```

/**
 * Default data source
 */
export let defaultData = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
    Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
  { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
    Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
  { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
  { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
'12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
70 },
  { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
    Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
  { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
    Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
  { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
    Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
  { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
    Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
  { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
    Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
/**
 * Grid datasource
 */
export function getTradeData(dataCount) {
  let employees = [
    'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
    'Fuller', 'Leverling', 'Peacock',
    'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
    'Dodsworth', 'Anne',
    'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
    'Jack', 'Rose'
  ];
  let designation = ['Manager', 'CFO', 'Designer', 'Developer', 'Program
Directory', 'System Analyst', 'Project Lead'];
  let mail = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
'journrapide.com'];
  let location = ['UK', 'USA', 'Sweden', 'France', 'Canada', 'Argentina',
'Austria', 'Germany', 'Mexico'];
  let status = ['Active', 'Inactive'];
  let trustworthiness = ['Perfect', 'Sufficient', 'Insufficient'];
  let tradeData = [];
  let address = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do Paço, 67',
'2, rue du Commerce', 'Boulevard Tirou, 255',
'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
Mercado, 12',
'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',

```

```

'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
43', '5a Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
'Taucherstraße 10', 'Taucherstraße 10',
'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
Circus', 'Av. dos Lusíadas, 23',
'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
123', '2817 Milton Dr.'];
let employeeimg = ['usermale', 'userfemale'];
if (typeof dataCount === 'string') {
  dataCount = parseInt(dataCount, 10);
}
for (let i = 1; i <= dataCount; i++) {
  let code = 10000;
  tradeData.push({
    'EmployeeID': code + i,
    'Employees': employees[Math.floor(Math.random() *
employees.length)] + ' ' + employees[Math.floor(Math.random() *
employees.length)],
    'Designation': designation[Math.floor(Math.random() *
designation.length)],
    'Location': location[Math.floor(Math.random() *
location.length)],
    'Status': status[Math.floor(Math.random() * status.length)],
    'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
    'Rating': Math.floor(Math.random() * 5),
    'Software': Math.floor(Math.random() * 100),
    'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
    'CurrentSalary': Math.floor((Math.random() * 100000)),
    'Address': address[Math.floor(Math.random() * address.length)],
  });
  let employee = 'Employees';
  let emp = tradeData[i - 1][employee];
  let sName = emp.substr(0, emp.indexOf(' ')).toLowerCase();
  let empmail = 'Mail';
  tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
}
return tradeData;
}
export let tradeData = [
  {
    "EmployeeID": 10001,
    "Employees": "Laura Nancy",
    "Designation": "Designer",
    "Location": "France",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 69,
    "EmployeeImg": "usermale",
  }

```



```
"CurrentSalary": 84194,  
"Address": "Taucherstraße 10",  
"Mail": "laura15@jourrapide.com"  
},  
{  
  "EmployeeID": 10002,  
  "Employees": "Zachery Van",  
  "Designation": "CFO",  
  "Location": "Canada",  
  "Status": "Inactive",  
  "Trustworthiness": "Insufficient",  
  "Rating": 3,  
  "Software": 99,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 55349,  
  "Address": "5ª Ave. Los Palos Grandes",  
  "Mail": "zachery109@sample.com"  
},  
{  
  "EmployeeID": 10003,  
  "Employees": "Rose Fuller",  
  "Designation": "CFO",  
  "Location": "France",  
  "Status": "Active",  
  "Trustworthiness": "Insufficient",  
  "Rating": 1,  
  "Software": 1,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 16477,  
  "Address": "2817 Milton Dr.",  
  "Mail": "rose55@rpy.com"  
},  
{  
  "EmployeeID": 10004,  
  "Employees": "Jack Bergs",  
  "Designation": "Manager",  
  "Location": "Mexico",  
  "Status": "Inactive",  
  "Trustworthiness": "Insufficient",  
  "Rating": 3,  
  "Software": 36,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 49040,  
  "Address": "2, rue du Commerce",  
  "Mail": "jack30@sample.com"  
},  
{  
  "EmployeeID": 10005,  
  "Employees": "Vinet Bergs",  
  "Designation": "Program Directory",  
  "Location": "UK",  
  "Status": "Active",  
  "Trustworthiness": "Insufficient",  
  "Rating": 1,  
  "Software": 39,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 5495,
```

```
      "Address": "Rua da Panificadora, 12",
      "Mail": "vinet32@jourrapide.com"
    },
    {
      "EmployeeID": 10006,
      "Employees": "Buchanan Van",
      "Designation": "Designer",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 4,
      "Software": 78,
      "EmployeeImg": "usermale",
      "CurrentSalary": 42182,
      "Address": "24, place Kléber",
      "Mail": "buchanan18@mail.com"
    },
    {
      "EmployeeID": 10007,
      "Employees": "Dodsworth Nancy",
      "Designation": "Project Lead",
      "Location": "USA",
      "Status": "Inactive",
      "Trustworthiness": "Sufficient",
      "Rating": 0,
      "Software": 0,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 35776,
      "Address": "Rua do Paço, 67",
      "Mail": "dodsworth84@mail.com"
    },
    {
      "EmployeeID": 10008,
      "Employees": "Laura Jack",
      "Designation": "Developer",
      "Location": "Austria",
      "Status": "Inactive",
      "Trustworthiness": "Perfect",
      "Rating": 3,
      "Software": 89,
      "EmployeeImg": "usermale",
      "CurrentSalary": 25108,
      "Address": "Rua da Panificadora, 12",
      "Mail": "laura82@mail.com"
    },
    {
      "EmployeeID": 10009,
      "Employees": "Anne Fuller",
      "Designation": "Program Directory",
      "Location": "Mexico",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 0,
      "Software": 19,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 32568,
      "Address": "Gran Vía, 1",
```

```

    "Mail": "anne97@jourrapide.com"
  },
  {
    "EmployeeID": 10010,
    "Employees": "Buchanan Andrew",
    "Designation": "Designer",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 62,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 12320,
    "Address": "P.O. Box 555",
    "Mail": "buchanan50@jourrapide.com"
  },
  {
    "EmployeeID": 10011,
    "Employees": "Andrew Janet",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 20890,
    "Address": "Starenweg 5",
    "Mail": "andrew63@mail.com"
  },
  {
    "EmployeeID": 10012,
    "Employees": "Margaret Tamer",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 7,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 22337,
    "Address": "Magazinweg 7",
    "Mail": "margaret26@mail.com"
  },
  {
    "EmployeeID": 10013,
    "Employees": "Tamer Fuller",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 89181,
    "Address": "Taucherstraße 10",
    "Mail": "tamer40@arpy.com"
  }

```

```
},
{
  "EmployeeID": 10014,
  "Employees": "Tamer Anne",
  "Designation": "CFO",
  "Location": "Sweden",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 18,
  "EmployeeImg": "usermale",
  "CurrentSalary": 20998,
  "Address": "Taucherstraße 10",
  "Mail": "tamer68@arpy.com"
},
{
  "EmployeeID": 10015,
  "Employees": "Anton Davolio",
  "Designation": "Project Lead",
  "Location": "France",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 4,
  "Software": 8,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 48232,
  "Address": "Luisenstr. 48",
  "Mail": "anton46@mail.com"
},
{
  "EmployeeID": 10016,
  "Employees": "Buchanan Buchanan",
  "Designation": "System Analyst",
  "Location": "Austria",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 19,
  "EmployeeImg": "usermale",
  "CurrentSalary": 43041,
  "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
  "Mail": "buchanan68@mail.com"
},
{
  "EmployeeID": 10017,
  "Employees": "King Buchanan",
  "Designation": "Program Directory",
  "Location": "Sweden",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 44,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 25259,
  "Address": "Magazinweg 7",
  "Mail": "king80@jourrapide.com"
},
}
```

```

{
  "EmployeeID": 10018,
  "Employees": "Rose Michael",
  "Designation": "Project Lead",
  "Location": "Canada",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 4,
  "Software": 31,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 91156,
  "Address": "Fauntleroy Circus",
  "Mail": "rose75@mail.com"
},
{
  "EmployeeID": 10019,
  "Employees": "King Bergs",
  "Designation": "Developer",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 2,
  "Software": 29,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 28826,
  "Address": "2817 Milton Dr.",
  "Mail": "king57@jourrapide.com"
},
{
  "EmployeeID": 10020,
  "Employees": "Davolio Fuller",
  "Designation": "Designer",
  "Location": "Canada",
  "Status": "Inactive",
  "Trustworthiness": "Sufficient",
  "Rating": 3,
  "Software": 35,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 71035,
  "Address": "Gran Vía, 1",
  "Mail": "davolio29@arpy.com"
},
{
  "EmployeeID": 10021,
  "Employees": "Rose Rose",
  "Designation": "CFO",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 38,
  "EmployeeImg": "usermale",
  "CurrentSalary": 68123,
  "Address": "Rua do Mercado, 12",
  "Mail": "rose54@arpy.com"
},
{

```

```
    "EmployeeID": 10022,  
    "Employees": "Andrew Michael",  
    "Designation": "Program Directory",  
    "Location": "UK",  
    "Status": "Inactive",  
    "Trustworthiness": "Insufficient",  
    "Rating": 2,  
    "Software": 61,  
    "EmployeeImg": "userfemale",  
    "CurrentSalary": 75470,  
    "Address": "2, rue du Commerce",  
    "Mail": "andrew88@jourrapide.com"  
  },  
  {  
    "EmployeeID": 10023,  
    "Employees": "Davolio Kathryn",  
    "Designation": "Manager",  
    "Location": "Germany",  
    "Status": "Active",  
    "Trustworthiness": "Perfect",  
    "Rating": 3,  
    "Software": 25,  
    "EmployeeImg": "usermale",  
    "CurrentSalary": 25234,  
    "Address": "Hauptstr. 31",  
    "Mail": "davolio42@sample.com"  
  },  
  {  
    "EmployeeID": 10024,  
    "Employees": "Anne Fleet",  
    "Designation": "System Analyst",  
    "Location": "UK",  
    "Status": "Active",  
    "Trustworthiness": "Perfect",  
    "Rating": 3,  
    "Software": 0,  
    "EmployeeImg": "userfemale",  
    "CurrentSalary": 8341,  
    "Address": "59 rue de l'Abbaye",  
    "Mail": "anne86@arpy.com"  
  },  
  {  
    "EmployeeID": 10025,  
    "Employees": "Margaret Andrew",  
    "Designation": "System Analyst",  
    "Location": "Germany",  
    "Status": "Inactive",  
    "Trustworthiness": "Insufficient",  
    "Rating": 3,  
    "Software": 51,  
    "EmployeeImg": "userfemale",  
    "CurrentSalary": 84975,  
    "Address": "P.O. Box 555",  
    "Mail": "margaret41@arpy.com"  
  },  
  {  
    "EmployeeID": 10026,
```

```
    "Employees": "Kathryn Laura",
    "Designation": "Project Lead",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 48,
    "EmployeeImg": "usermale",
    "CurrentSalary": 97282,
    "Address": "Avda. Azteca 123",
    "Mail": "kathryn82@rpy.com"
  },
  {
    "EmployeeID": 10027,
    "Employees": "Michael Michael",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 16,
    "EmployeeImg": "usermale",
    "CurrentSalary": 4184,
    "Address": "Rua do Paço, 67",
    "Mail": "michael58@jourrapide.com"
  },
  {
    "EmployeeID": 10028,
    "Employees": "Leverling Vinet",
    "Designation": "Project Lead",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 57,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 38370,
    "Address": "59 rue de l'Abbaye",
    "Mail": "leverling102@sample.com"
  },
  {
    "EmployeeID": 10029,
    "Employees": "Rose Jack",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 46,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84790,
    "Address": "Rua do Mercado, 12",
    "Mail": "rose108@jourrapide.com"
  },
  {
    "EmployeeID": 10030,
    "Employees": "Vinet Van",
```

```

        "Designation": "Developer",
        "Location": "USA",
        "Status": "Active",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
        "Software": 40,
        "EmployeeImg": "usermale",
        "CurrentSalary": 71005,
        "Address": "Gran Vía, 1",
        "Mail": "vinet90@jourrapide.com"
    }
];

```

### DATASOURCE.TSX

```

/**
 * Default data source
 */
export let defaultData: Object[] = [
    { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
      Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
    { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
      Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
    { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
      Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
    { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
    { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
'12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
70 },
    { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
      Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
    { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
      Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
    { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
      Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
    { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
      Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
    { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
      Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
/**
 * Grid datasource
 */
export function getTradeData(dataCount?: number): object {
    let employees: string[] = [
        'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
        'Fuller', 'Leverling', 'Peacock',
        'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
        'Dodsworth', 'Anne',
        'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
        'Jack', 'Rose'];
    let designation: string[] = ['Manager', 'CFO', 'Designer', 'Developer',
        'Program Directory', 'System Analyst', 'Project Lead'];
    let mail: string[] = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
        'jourrapide.com'];
}

```



```

    let location: string[] = ['UK', 'USA', 'Sweden', 'France', 'Canada',
    'Argentina', 'Austria', 'Germany', 'Mexico'];
    let status: string[] = ['Active', 'Inactive'];
    let trustworthiness: string[] = ['Perfect', 'Sufficient',
    'Insufficient'];
    let tradeData: Object[] = [];
    let address: string[] = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do
    Paço, 67', '2, rue du Commerce', 'Boulevard Tirou, 255',
    'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
    Mercado, 12',
    'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
    'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
    Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
    'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
    43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
    'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
    10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
    'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
    7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
    'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
    'Taucherstraße 10', 'Taucherstraße 10',
    'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
    Circus', 'Av. dos Lusíadas, 23',
    'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
    123', '2817 Milton Dr.'];
    let employeeimg: string[] = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
        dataCount = parseInt(dataCount, 10);
    }
    for (let i: number = 1; i <= dataCount; i++) {
        let code: any = 10000;
        tradeData.push({
            'EmployeeID': code + i,
            'Employees':
                employees[Math.floor(Math.random() * employees.length)] + '
' + employees[Math.floor(Math.random() * employees.length)],
            'Designation': designation[Math.floor(Math.random() *
            designation.length)],
            'Location': location[Math.floor(Math.random() *
            location.length)],
            'Status': status[Math.floor(Math.random() * status.length)],
            'Trustworthiness': trustworthiness[Math.floor(Math.random() *
            trustworthiness.length)],
            'Rating': Math.floor(Math.random() * 5),
            'Software': Math.floor(Math.random() * 100),
            'EmployeeImg': employeeimg[Math.floor(Math.random() *
            employeeimg.length)],
            'CurrentSalary': Math.floor((Math.random() * 100000)),
            'Address': address[Math.floor(Math.random() * address.length)],
        });
        let employee: string = 'Employees';
        let emp: string = tradeData[i - 1][employee];
        let sName: string = emp.substr(0, emp.indexOf(' ')).toLowerCase();
        let empmail: string = 'Mail';
        tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
        + 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }

```

```
    return tradeData;
  }
export let tradeData: Object[] = [
  {
    "EmployeeID": 10001,
    "Employees": "Laura Nancy",
    "Designation": "Designer",
    "Location": "France",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 69,
    "EmployeeImg": "usermale",
    "CurrentSalary": 84194,
    "Address": "Taucherstraße 10",
    "Mail": "laura15@jourrapide.com"
  },
  {
    "EmployeeID": 10002,
    "Employees": "Zachery Van",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 99,
    "EmployeeImg": "usermale",
    "CurrentSalary": 55349,
    "Address": "5a Ave. Los Palos Grandes",
    "Mail": "zachery109@sample.com"
  },
  {
    "EmployeeID": 10003,
    "Employees": "Rose Fuller",
    "Designation": "CFO",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 1,
    "EmployeeImg": "usermale",
    "CurrentSalary": 16477,
    "Address": "2817 Milton Dr.",
    "Mail": "rose55@rpy.com"
  },
  {
    "EmployeeID": 10004,
    "Employees": "Jack Bergs",
    "Designation": "Manager",
    "Location": "Mexico",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 36,
    "EmployeeImg": "usermale",
    "CurrentSalary": 49040,
    "Address": "2, rue du Commerce",
  }
]
```

```
"Mail": "jack30@sample.com"
},
{
  "EmployeeID": 10005,
  "Employees": "Vinet Bergs",
  "Designation": "Program Directory",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 39,
  "EmployeeImg": "usermale",
  "CurrentSalary": 5495,
  "Address": "Rua da Panificadora, 12",
  "Mail": "vinet32@jourrapide.com"
},
{
  "EmployeeID": 10006,
  "Employees": "Buchanan Van",
  "Designation": "Designer",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 4,
  "Software": 78,
  "EmployeeImg": "usermale",
  "CurrentSalary": 42182,
  "Address": "24, place Kléber",
  "Mail": "buchanan18@mail.com"
},
{
  "EmployeeID": 10007,
  "Employees": "Dodsworth Nancy",
  "Designation": "Project Lead",
  "Location": "USA",
  "Status": "Inactive",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 0,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 35776,
  "Address": "Rua do Paço, 67",
  "Mail": "dodsworth84@mail.com"
},
{
  "EmployeeID": 10008,
  "Employees": "Laura Jack",
  "Designation": "Developer",
  "Location": "Austria",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 89,
  "EmployeeImg": "usermale",
  "CurrentSalary": 25108,
  "Address": "Rua da Panificadora, 12",
  "Mail": "laura82@mail.com"
```

```
},
{
  "EmployeeID": 10009,
  "Employees": "Anne Fuller",
  "Designation": "Program Directory",
  "Location": "Mexico",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 0,
  "Software": 19,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 32568,
  "Address": "Gran Vía, 1",
  "Mail": "anne97@jourrapide.com"
},
{
  "EmployeeID": 10010,
  "Employees": "Buchanan Andrew",
  "Designation": "Designer",
  "Location": "Austria",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 62,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 12320,
  "Address": "P.O. Box 555",
  "Mail": "buchanan50@jourrapide.com"
},
{
  "EmployeeID": 10011,
  "Employees": "Andrew Janet",
  "Designation": "System Analyst",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 8,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 20890,
  "Address": "Starenweg 5",
  "Mail": "andrew63@mail.com"
},
{
  "EmployeeID": 10012,
  "Employees": "Margaret Tamer",
  "Designation": "System Analyst",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 4,
  "Software": 7,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 22337,
  "Address": "Magazinweg 7",
  "Mail": "margaret26@mail.com"
},
}
```

```
{
  "EmployeeID": 10013,
  "Employees": "Tamer Fuller",
  "Designation": "CFO",
  "Location": "Canada",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 78,
  "EmployeeImg": "usermale",
  "CurrentSalary": 89181,
  "Address": "Taucherstraße 10",
  "Mail": "tamer40@arpy.com"
},
{
  "EmployeeID": 10014,
  "Employees": "Tamer Anne",
  "Designation": "CFO",
  "Location": "Sweden",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 18,
  "EmployeeImg": "usermale",
  "CurrentSalary": 20998,
  "Address": "Taucherstraße 10",
  "Mail": "tamer68@arpy.com"
},
{
  "EmployeeID": 10015,
  "Employees": "Anton Davolio",
  "Designation": "Project Lead",
  "Location": "France",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 4,
  "Software": 8,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 48232,
  "Address": "Luisenstr. 48",
  "Mail": "anton46@mail.com"
},
{
  "EmployeeID": 10016,
  "Employees": "Buchanan Buchanan",
  "Designation": "System Analyst",
  "Location": "Austria",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 19,
  "EmployeeImg": "usermale",
  "CurrentSalary": 43041,
  "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
  "Mail": "buchanan68@mail.com"
},
{
```

```
"EmployeeID": 10017,  
"Employees": "King Buchanan",  
"Designation": "Program Directory",  
"Location": "Sweden",  
"Status": "Active",  
"Trustworthiness": "Sufficient",  
"Rating": 0,  
"Software": 44,  
"EmployeeImg": "userfemale",  
"CurrentSalary": 25259,  
"Address": "Magazinweg 7",  
"Mail": "king80@jourrapide.com"  
},  
{  
  "EmployeeID": 10018,  
  "Employees": "Rose Michael",  
  "Designation": "Project Lead",  
  "Location": "Canada",  
  "Status": "Active",  
  "Trustworthiness": "Perfect",  
  "Rating": 4,  
  "Software": 31,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 91156,  
  "Address": "Fauntleroy Circus",  
  "Mail": "rose75@mail.com"  
},  
{  
  "EmployeeID": 10019,  
  "Employees": "King Bergs",  
  "Designation": "Developer",  
  "Location": "Germany",  
  "Status": "Active",  
  "Trustworthiness": "Sufficient",  
  "Rating": 2,  
  "Software": 29,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 28826,  
  "Address": "2817 Milton Dr.",  
  "Mail": "king57@jourrapide.com"  
},  
{  
  "EmployeeID": 10020,  
  "Employees": "Davolio Fuller",  
  "Designation": "Designer",  
  "Location": "Canada",  
  "Status": "Inactive",  
  "Trustworthiness": "Sufficient",  
  "Rating": 3,  
  "Software": 35,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 71035,  
  "Address": "Gran Vía, 1",  
  "Mail": "davolio29@arpy.com"  
},  
{  
  "EmployeeID": 10021,
```

```
"Employees": "Rose Rose",
"Designation": "CFO",
"Location": "Germany",
"Status": "Active",
"Trustworthiness": "Perfect",
"Rating": 3,
"Software": 38,
"EmployeeImg": "usermale",
"CurrentSalary": 68123,
"Address": "Rua do Mercado, 12",
"Mail": "rose54@arpy.com"
},
{
  "EmployeeID": 10022,
  "Employees": "Andrew Michael",
  "Designation": "Program Directory",
  "Location": "UK",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 2,
  "Software": 61,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 75470,
  "Address": "2, rue du Commerce",
  "Mail": "andrew88@jourrapide.com"
},
{
  "EmployeeID": 10023,
  "Employees": "Davolio Kathryn",
  "Designation": "Manager",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 25,
  "EmployeeImg": "usermale",
  "CurrentSalary": 25234,
  "Address": "Hauptstr. 31",
  "Mail": "davolio42@sample.com"
},
{
  "EmployeeID": 10024,
  "Employees": "Anne Fleet",
  "Designation": "System Analyst",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 0,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 8341,
  "Address": "59 rue de l'Abbaye",
  "Mail": "anne86@arpy.com"
},
{
  "EmployeeID": 10025,
  "Employees": "Margaret Andrew",
```

```

    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 51,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84975,
    "Address": "P.O. Box 555",
    "Mail": "margaret41@arpy.com"
  },
  {
    "EmployeeID": 10026,
    "Employees": "Kathryn Laura",
    "Designation": "Project Lead",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 48,
    "EmployeeImg": "usermale",
    "CurrentSalary": 97282,
    "Address": "Avda. Azteca 123",
    "Mail": "kathryn82@rpy.com"
  },
  {
    "EmployeeID": 10027,
    "Employees": "Michael Michael",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 16,
    "EmployeeImg": "usermale",
    "CurrentSalary": 4184,
    "Address": "Rua do Paço, 67",
    "Mail": "michael58@jourrapide.com"
  },
  {
    "EmployeeID": 10028,
    "Employees": "Leverling Vinet",
    "Designation": "Project Lead",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 57,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 38370,
    "Address": "59 rue de l'Abbaye",
    "Mail": "leverling102@sample.com"
  },
  {
    "EmployeeID": 10029,
    "Employees": "Rose Jack",
    "Designation": "Developer",

```



```

    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 46,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84790,
    "Address": "Rua do Mercado, 12",
    "Mail": "rose108@jourrapide.com"
  },
  {
    "EmployeeID": 10030,
    "Employees": "Vinet Van",
    "Designation": "Developer",
    "Location": "USA",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 40,
    "EmployeeImg": "usermale",
    "CurrentSalary": 71005,
    "Address": "Gran Vía, 1",
    "Mail": "vinet90@jourrapide.com"
  }
]

```

## Limitations

- External clipboard is not fully supported while copying data from another source and pasting into a spreadsheet, it only works with basic supports (Values, Number, cell, and Text formatting).
- If you copy =SUM(A2,B2) and paste, the formula reference will change depending on the pasted cell address but we don't have support for nested formula(formula reference will be same).
- Clipboard is not supported with conditional formatting (values only pasting).
- We have limitation while copying the whole sheet data and pasting it into another sheet.

## Note

You can refer to our [React Spreadsheet](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Spreadsheet example](#) to know how to present and manipulate data.

## Selection in React Spreadsheet component

Selection provides interactive support to highlight the cell, row, or column that you select. Selection can be done through Mouse, Touch, or Keyboard interaction. To enable selection, set `mode` as `Single` / `Multiple` in [selectionSettings](#). If you set `mode` to `None`, it disables the UI selection.

\* The default value for `mode` in `selectionSettings` is `Multiple`.

You have the following options in Selection,

- Cell selection
- Row selection

- Column selection

### Cell selection

Cell selection is used to select a single or multiple cells. It can be performed using the [selectRange](#) method.

#### User Interface:

- Click on a cell to select it (or) use the **arrow** keys to navigate to it and select it.
- To select a range, select a cell, then use the left mouse button to select and drag over to other cells (or) use the **Shift + arrow** keys to select the range.
- To select non-adjacent cells and cell ranges, hold **Ctrl** and select the cells.

You can quickly locate and select specific cells or ranges by entering their names or cell references in the Name box, which is located to the left of the formula bar, and also select named or unnamed cells or ranges by using the Go To (**Ctrl+G**) command.

### Row selection

Row selection is used to select a single or multiple rows.

#### User Interface:

You can perform row selection in any of the following ways,

- By clicking the row header.
- To select multiple rows, select a row header with the left mouse button and drag over to other row headers (or) use the **Shift + arrow** keys to select multiple rows.
- To select non-adjacent rows, hold **Ctrl** and select the row header.
- You can also use the [selectRange](#) method for row selection.

The following sample shows the row selection in the spreadsheet, here selecting the 5th row using the [selectRange](#) method.

#### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, getRangeAddress } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { budgetData } from './datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign:
'center', verticalAlign: 'middle' }, 'A1:D1');
      let colCount = spreadsheet.getActiveSheet().colCount;
      spreadsheet.selectRange(getRangeAddress([4, 0, 4, colCount]));
    }
  });
}
```

```

    }, []);
    return (
      <div>
        <SpreadsheetComponent ref={spreadsheetRef} selectionSettings={{
mode: 'Multiple' }}>
          <SheetsDirective>
            <SheetDirective>
              <RangesDirective>
                <RangeDirective
dataSource={budgetData}></RangeDirective>
              </RangesDirective>
              <ColumnsDirective>
                <ColumnDirective width={130}></ColumnDirective>
                <ColumnDirective width={92}></ColumnDirective>
                <ColumnDirective width={96}></ColumnDirective>
              </ColumnsDirective>
            </SheetDirective>
          </SheetsDirective>
        </SpreadsheetComponent>
      </div>
    );
  };
  export default App;
  const root = createRoot(document.getElementById('root'));
  root.render(<App />);
  {% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, getRangeAddress } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { budgetData } from './datasource';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign:
'center', verticalAlign: 'middle' }, 'A1:D1');
      let colCount: number = spreadsheet.getActiveSheet().colCount as
number;
      spreadsheet.selectRange(getRangeAddress([4, 0, 4, colCount]));
    }
  }, []);
  return (
    <div>
      <SpreadsheetComponent ref={spreadsheetRef} selectionSettings={{
mode: 'Multiple' }}>
        <SheetsDirective>
          <SheetDirective>
            <RangesDirective>

```

```

                                <RangeDirective
dataSource={budgetData}></RangeDirective>
                                </RangesDirective>
                                <ColumnsDirective>
                                    <ColumnDirective width={130}></ColumnDirective>
                                    <ColumnDirective width={92}></ColumnDirective>
                                    <ColumnDirective width={96}></ColumnDirective>
                                </ColumnsDirective>
                            </SheetDirective>
                        </SheetsDirective>
                    </SpreadsheetComponent>
                </div>
            );
        };
    export default App;
    const root = createRoot(document.getElementById('root')!);
    root.render(<App />);
    {% endraw %}

```

### DATASOURCE.JSX

```

/**
 * Default data source
 */
export let budgetData = [
    {
        "Expense Type": "Housing",
        "Projected Cost": 7000,
        "Actual Cost": 7500,
        "Difference": -500,
    },
    {
        "Expense Type": "Transportation",
        "Projected Cost": 500,
        "Actual Cost": 500,
        "Difference": 0,
    },
    {
        "Expense Type": "Insurance",
        "Projected Cost": 1000,
        "Actual Cost": 1000,
        "Difference": 0,
    },
    {
        "Expense Type": "Food",
        "Projected Cost": 2000,
        "Actual Cost": 1800,
        "Difference": 200,
    },
    {
        "Expense Type": "Pets",
        "Projected Cost": 300,
        "Actual Cost": 200,
        "Difference": 100,
    },
    {

```

```

        "Expense Type": "Personel Care",
        "Projected Cost": 500,
        "Actual Cost": 500,
        "Difference": 0,
    }, {
        "Expense Type": "Loan",
        "Projected Cost": 1000,
        "Actual Cost": 1000,
        "Difference": 0,
    }, {
        "Expense Type": "Tax",
        "Projected Cost": 200,
        "Actual Cost": 200,
        "Difference": 0,
    }, {
        "Expense Type": "Savings",
        "Projected Cost": 1000,
        "Actual Cost": 900,
        "Difference": 100,
    },
    {
        "Expense Type": "Total",
        "Projected Cost": 13500,
        "Actual Cost": 13600,
        "Difference": -100,
    }
];

```

### DATASOURCE.TSX

```

/**
 * Default data source
 */
export let budgetData: Object[] = [
    {
        "Expense Type": "Housing",
        "Projected Cost": 7000,
        "Actual Cost": 7500,
        "Difference": -500,
    },
    {
        "Expense Type": "Transportation",
        "Projected Cost": 500,
        "Actual Cost": 500,
        "Difference": 0,
    },
    {
        "Expense Type": "Insurance",
        "Projected Cost": 1000,
        "Actual Cost": 1000,
        "Difference": 0,
    },
    {
        "Expense Type": "Food",
        "Projected Cost": 2000,
        "Actual Cost": 1800,
    },
];

```

```

    "Difference": 200,
  },
  {
    "Expense Type": "Pets",
    "Projected Cost": 300,
    "Actual Cost": 200,
    "Difference": 100,
  },
  {
    "Expense Type": "Personel Care",
    "Projected Cost": 500,
    "Actual Cost": 500,
    "Difference": 0,
  }, {
    "Expense Type": "Loan",
    "Projected Cost": 1000,
    "Actual Cost": 1000,
    "Difference": 0,
  }, {
    "Expense Type": "Tax",
    "Projected Cost": 200,
    "Actual Cost": 200,
    "Difference": 0,
  }, {
    "Expense Type": "Savings",
    "Projected Cost": 1000,
    "Actual Cost": 900,
    "Difference": 100,
  },
  {
    "Expense Type": "Total",
    "Projected Cost": 13500,
    "Actual Cost": 13600,
    "Difference": -100,
  }
];

```

### Column selection

Column selection is used to select a single or multiple columns.

#### User Interface:

You can perform column selection in any of the following ways,

- By clicking the column header.
- To select multiple columns, select a column header with the left mouse button and drag over to other column headers (or) use the **Shift + arrow** keys to select the multiple columns.
- To select non-adjacent columns, hold **Ctrl** and select the column header.
- You can also use the **selectRange** method for row selection.

The following sample shows the column selection in the spreadsheet, here selecting the 3rd column using the **selectRange** method.

#### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, getRangeAddress } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { budgetData } from '../datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign:
'center', verticalAlign: 'middle' }, 'A1:D1');
      let rowCount = spreadsheet.getActiveSheet().rowCount;
      spreadsheet.selectRange(getRangeAddress([0, 2, rowCount, 2]));
    }
  }, []);
  return (
    <div>
      <SpreadsheetComponent ref={spreadsheetRef} selectionSettings={{
mode: 'Multiple' }}>
        <SheetsDirective>
          <SheetDirective>
            <RangesDirective>
              <RangeDirective
dataSource={budgetData}></RangeDirective>
            </RangesDirective>
            <ColumnsDirective>
              <ColumnDirective width={130}></ColumnDirective>
              <ColumnDirective width={92}></ColumnDirective>
              <ColumnDirective width={96}></ColumnDirective>
            </ColumnsDirective>
          </SheetDirective>
        </SheetsDirective>
      </SpreadsheetComponent>
    </div>
  );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}
```

**APP.TSX**

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, getRangeAddress } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { budgetData } from '../datasource';
function App() {
```

```

    const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
    React.useEffect(() => {
      let spreadsheet = spreadsheetRef.current;
      if (spreadsheet) {
        spreadsheet.cellFormat({ fontWeight: 'bold', textAlign:
'center', verticalAlign: 'middle' }, 'A1:D1');
        let rowCount: number = spreadsheet.getActiveSheet().rowCount as
number;
        spreadsheet.selectRange(getRangeAddress([0, 2, rowCount, 2]));
      }
    }, []);
    return (
      <div>
        <SpreadsheetComponent ref={spreadsheetRef} selectionSettings={{
mode: 'Multiple' }}>
          <SheetsDirective>
            <SheetDirective>
              <RangesDirective>
                <RangeDirective
dataSource={budgetData}></RangeDirective>
              </RangesDirective>
              <ColumnsDirective>
                <ColumnDirective width={130}></ColumnDirective>
                <ColumnDirective width={92}></ColumnDirective>
                <ColumnDirective width={96}></ColumnDirective>
              </ColumnsDirective>
            </SheetDirective>
          </SheetsDirective>
        </SpreadsheetComponent>
      </div>
    );
  };
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

## DATASOURCE.JSX

```

/**
 * Default data source
 */
export let budgetData = [
  {
    "Expense Type": "Housing",
    "Projected Cost": 7000,
    "Actual Cost": 7500,
    "Difference": -500,
  },
  {
    "Expense Type": "Transportation",
    "Projected Cost": 500,
    "Actual Cost": 500,
    "Difference": 0,
  },
  {

```



```

        "Expense Type": "Insurance",
        "Projected Cost": 1000,
        "Actual Cost": 1000,
        "Difference": 0,
    },
    {
        "Expense Type": "Food",
        "Projected Cost": 2000,
        "Actual Cost": 1800,
        "Difference": 200,
    },
    {
        "Expense Type": "Pets",
        "Projected Cost": 300,
        "Actual Cost": 200,
        "Difference": 100,
    },
    {
        "Expense Type": "Personel Care",
        "Projected Cost": 500,
        "Actual Cost": 500,
        "Difference": 0,
    }, {
        "Expense Type": "Loan",
        "Projected Cost": 1000,
        "Actual Cost": 1000,
        "Difference": 0,
    }, {
        "Expense Type": "Tax",
        "Projected Cost": 200,
        "Actual Cost": 200,
        "Difference": 0,
    }, {
        "Expense Type": "Savings",
        "Projected Cost": 1000,
        "Actual Cost": 900,
        "Difference": 100,
    },
    {
        "Expense Type": "Total",
        "Projected Cost": 13500,
        "Actual Cost": 13600,
        "Difference": -100,
    }
];

```

### DATASOURCE.TSX

```

/**
 * Default data source
 */
export let budgetData: Object[] = [
    {
        "Expense Type": "Housing",
        "Projected Cost": 7000,
        "Actual Cost": 7500,
    }
];

```

```
"Difference": -500,
},
{
  "Expense Type": "Transportation",
  "Projected Cost": 500,
  "Actual Cost": 500,
  "Difference": 0,
},
{
  "Expense Type": "Insurance",
  "Projected Cost": 1000,
  "Actual Cost": 1000,
  "Difference": 0,
},
{
  "Expense Type": "Food",
  "Projected Cost": 2000,
  "Actual Cost": 1800,
  "Difference": 200,
},
{
  "Expense Type": "Pets",
  "Projected Cost": 300,
  "Actual Cost": 200,
  "Difference": 100,
},
{
  "Expense Type": "Personel Care",
  "Projected Cost": 500,
  "Actual Cost": 500,
  "Difference": 0,
},
{
  "Expense Type": "Loan",
  "Projected Cost": 1000,
  "Actual Cost": 1000,
  "Difference": 0,
},
{
  "Expense Type": "Tax",
  "Projected Cost": 200,
  "Actual Cost": 200,
  "Difference": 0,
},
{
  "Expense Type": "Savings",
  "Projected Cost": 1000,
  "Actual Cost": 900,
  "Difference": 100,
},
{
  "Expense Type": "Total",
  "Projected Cost": 13500,
  "Actual Cost": 13600,
  "Difference": -100,
}
];
```

### How to remove selection in the spreadsheet

The following sample shows, how to remove the selection in the spreadsheet. Here changing the **mode** as **None** in [selectionSettings](#) to disable's the UI selection.

#### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { budgetData } from '../datasource';
function App() {
    const spreadsheetRef = React.useRef(null);
    React.useEffect(() => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            spreadsheet.cellFormat({ fontWeight: 'bold', textAlign:
'center', verticalAlign: 'middle' }, 'A1:D1');
        }
    }, []);
    const onEdit = (args) => {
        args.cancel = true;
    };
    return (
        <div>
            <SpreadsheetComponent ref={spreadsheetRef} cellEdit={onEdit}
selectionSettings={{ mode: 'None' }}>
                <SheetsDirective>
                    <SheetDirective>
                        <RangesDirective>
                            <RangeDirective
dataSource={budgetData}></RangeDirective>
                        </RangesDirective>
                        <ColumnsDirective>
                            <ColumnDirective width={130}></ColumnDirective>
                            <ColumnDirective width={92}></ColumnDirective>
                            <ColumnDirective width={96}></ColumnDirective>
                        </ColumnsDirective>
                    </SheetDirective>
                </SheetsDirective>
            </SpreadsheetComponent>
        </div>
    );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}
```

#### APP.TSX

```
{% raw %}
import * as React from 'react';
```

```

import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, CellEditEventArgs } from '@syncfusion/ej2-react-
spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { budgetData } from './datasource';
function App() {
    const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
    React.useEffect(() => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            spreadsheet.cellFormat({ fontWeight: 'bold', textAlign:
'center', verticalAlign: 'middle' }, 'A1:D1');
        }
    }, []);
    const onEdit = (args: CellEditEventArgs): void => {
        args.cancel = true;
    };
    return (
        <div>
            <SpreadsheetComponent ref={spreadsheetRef} cellEdit={onEdit}
selectionSettings={{ mode: 'None' }}>
                <SheetsDirective>
                    <SheetDirective>
                        <RangesDirective>
                            <RangeDirective
dataSource={budgetData}></RangeDirective>
                        </RangesDirective>
                        <ColumnsDirective>
                            <ColumnDirective width={130}></ColumnDirective>
                            <ColumnDirective width={92}></ColumnDirective>
                            <ColumnDirective width={96}></ColumnDirective>
                        </ColumnsDirective>
                    </SheetDirective>
                </SheetsDirective>
            </SpreadsheetComponent>
        </div>
    );
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

### DATASOURCE.JSX

```

/**
 * Default data source
 */
export let budgetData = [
    {
        "Expense Type": "Housing",
        "Projected Cost": 7000,
        "Actual Cost": 7500,
        "Difference": -500,
    }
];

```

```

    },
    {
      "Expense Type": "Transportation",
      "Projected Cost": 500,
      "Actual Cost": 500,
      "Difference": 0,
    },
    {
      "Expense Type": "Insurance",
      "Projected Cost": 1000,
      "Actual Cost": 1000,
      "Difference": 0,
    },
    {
      "Expense Type": "Food",
      "Projected Cost": 2000,
      "Actual Cost": 1800,
      "Difference": 200,
    },
    {
      "Expense Type": "Pets",
      "Projected Cost": 300,
      "Actual Cost": 200,
      "Difference": 100,
    },
    {
      "Expense Type": "Personel Care",
      "Projected Cost": 500,
      "Actual Cost": 500,
      "Difference": 0,
    },
    {
      "Expense Type": "Loan",
      "Projected Cost": 1000,
      "Actual Cost": 1000,
      "Difference": 0,
    },
    {
      "Expense Type": "Tax",
      "Projected Cost": 200,
      "Actual Cost": 200,
      "Difference": 0,
    },
    {
      "Expense Type": "Savings",
      "Projected Cost": 1000,
      "Actual Cost": 900,
      "Difference": 100,
    },
    {
      "Expense Type": "Total",
      "Projected Cost": 13500,
      "Actual Cost": 13600,
      "Difference": -100,
    }
  ];

```

**DATASOURCE.TSX**

```
/**
 * Default data source
 */
export let budgetData: Object[] = [
  {
    "Expense Type": "Housing",
    "Projected Cost": 7000,
    "Actual Cost": 7500,
    "Difference": -500,
  },
  {
    "Expense Type": "Transportation",
    "Projected Cost": 500,
    "Actual Cost": 500,
    "Difference": 0,
  },
  {
    "Expense Type": "Insurance",
    "Projected Cost": 1000,
    "Actual Cost": 1000,
    "Difference": 0,
  },
  {
    "Expense Type": "Food",
    "Projected Cost": 2000,
    "Actual Cost": 1800,
    "Difference": 200,
  },
  {
    "Expense Type": "Pets",
    "Projected Cost": 300,
    "Actual Cost": 200,
    "Difference": 100,
  },
  {
    "Expense Type": "Personel Care",
    "Projected Cost": 500,
    "Actual Cost": 500,
    "Difference": 0,
  },
  {
    "Expense Type": "Loan",
    "Projected Cost": 1000,
    "Actual Cost": 1000,
    "Difference": 0,
  },
  {
    "Expense Type": "Tax",
    "Projected Cost": 200,
    "Actual Cost": 200,
    "Difference": 0,
  },
  {
    "Expense Type": "Savings",
    "Projected Cost": 1000,
    "Actual Cost": 900,
    "Difference": 100,
  },
  {
    "Expense Type": "Total",
```

```
"Projected Cost": 13500,  
"Actual Cost": 13600,  
"Difference": -100,  
}  
];
```

## Limitations

- We have a limitation while performing the Select All(ctrl + A). You can do this only by clicking the Select All button at the top left corner.

## Note

You can refer to our [React Spreadsheet](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Spreadsheet example](#) to know how to present and manipulate data.

## Scrolling in React Spreadsheet component

Scrolling helps you to move quickly to different areas of the worksheet. It moves faster if we use horizontal and vertical scroll bars. Scrolling can be enabled by setting the [allowScrolling](#) as true.

By default, the [allowScrolling](#) property is true.

You have the following options in Scrolling by using [scrollSettings](#).

- Finite scrolling.
- Virtual scrolling.

## Finite Scrolling

Finite scrolling supports two type of modes in scrolling. You can use the [isFinite](#) property in [scrollSettings](#) to specify the mode of scrolling.

- Finite - This mode does not create a new row/column when the scrollbar reaches the end. This can be achieved by setting the [isFinite](#) property as true.
- Infinite - This mode creates a new row/column when the scrollbar reaches the end. This can be achieved by setting the [isFinite](#) property as false.

By Default, the [isFinite](#) property is false.

## Virtual Scrolling

- Virtual scrolling allows you to load data that you require (load data based on viewport size) without buffering the entire huge database. You can set the [enableVirtualization](#) property in [scrollSettings](#) as true.

In virtual scrolling [enableVirtualization](#) is set to true means, it allows you to load the spreadsheet data while scrolling.

By Default, the [enableVirtualization](#) property is true.

**User Interface:**

You can scroll through the worksheet using one of the following ways,

- Using the **arrow** keys.
- Using the Horizontal and Vertical **scroll** bars.
- Using the **mouse** wheel.

**Finite scrolling with defined rows and columns**

If you want to perform scrolling with defined rows and columns, you must define **rowCount** and **colCount** in the **sheets** property and set **isFinite** as true and **enableVirtualization** as false in **scrollSettings**.

The following code example shows the finite scrolling with defined rows and columns in the spreadsheet. Here, we used **rowCount** as 20 and **colCount** as 20, after reaching the 20th row or 20th column you can't able to scroll.

**APP.JSX**

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { defaultData } from './datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  const scrollSettings = { isFinite: true };
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign:
'center', verticalAlign: 'middle' }, 'A1:G1');
    }
  }, []);
  return (
    <div>
      <SpreadsheetComponent ref={spreadsheetRef}
scrollSettings={scrollSettings}>
        <SheetsDirective>
          <SheetDirective rowCount={9} colCount={7}>
            <RangesDirective>
              <RangeDirective
dataSource={defaultData}></RangeDirective>
            </RangesDirective>
            <ColumnsDirective>
              <ColumnDirective width={130}></ColumnDirective>
              <ColumnDirective width={92}></ColumnDirective>
              <ColumnDirective width={96}></ColumnDirective>
              <ColumnDirective width={130}></ColumnDirective>
              <ColumnDirective width={92}></ColumnDirective>
              <ColumnDirective width={96}></ColumnDirective>
              <ColumnDirective width={96}></ColumnDirective>
            </ColumnsDirective>
          </SheetDirective>
        </SheetsDirective>
      </SpreadsheetComponent>
    </div>
  );
}
```



```

        </ColumnsDirective>
      </SheetDirective>
    </SheetsDirective>
  </SpreadsheetComponent>
</div>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, ScrollSettingsModel } from '@syncfusion/ej2-react-
spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { defaultData } from './datasource';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  const scrollSettings: ScrollSettingsModel = { isFinite: true };
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign:
'center', verticalAlign: 'middle' }, 'A1:G1');
    }
  }, []);
  return (
    <div>
      <SpreadsheetComponent ref={spreadsheetRef}
scrollSettings={scrollSettings}>
        <SheetsDirective>
          <SheetDirective rowCount={9} colCount={7}>
            <RangesDirective>
              <RangeDirective
dataSource={defaultData}></RangeDirective>
            </RangesDirective>
            <ColumnsDirective>
              <ColumnDirective width={130}></ColumnDirective>
              <ColumnDirective width={92}></ColumnDirective>
              <ColumnDirective width={96}></ColumnDirective>
              <ColumnDirective width={130}></ColumnDirective>
              <ColumnDirective width={92}></ColumnDirective>
              <ColumnDirective width={96}></ColumnDirective>
              <ColumnDirective width={96}></ColumnDirective>
            </ColumnsDirective>
          </SheetDirective>
        </SheetsDirective>
      </SpreadsheetComponent>
    </div>

```

```

    );
  };
  export default App;
  const root = createRoot(document.getElementById('root')!);
  root.render(<App />);
  {% endraw %}

```

### DATASOURCE.JSX

```

/**
 * Default data source
 */
export let defaultData = [
  {
    'Product Id': 1, 'Product Name': 'Chai', 'Supplier Id': 1, 'Quantity
    Per Unit': '10 boxes x 20 bags', 'Unit Price': 18.00, 'Units In Stock': 39,
    'Discontinued': false
  },
  {
    'Product Id': 2, 'Product Name': 'Chang', 'Supplier Id': 1,
    'Quantity Per Unit': '24 - 12 oz bottles', 'Unit Price': 19.00, 'Units In
    Stock': 17, 'Discontinued': true
  },
  {
    'Product Id': 3, 'Product Name': 'Aniseed Syrup', 'Supplier Id': 1,
    'Quantity Per Unit': '12 - 550 ml bottles', 'Unit Price': 10.00, 'Units In
    Stock': 13, 'Discontinued': false
  },
  {
    'Product Id': 4, 'Product Name': 'Chef Anton\'s Cajun Seasoning',
    'Supplier Id': 2, 'Quantity Per Unit': '48 - 6 oz jars', 'Unit Price':
    22.00, 'Units In Stock': 53, 'Discontinued': true
  },
  {
    'Product Id': 5, 'Product Name': 'Chef Anton\'s Gumbo Mix',
    'Supplier Id': 2, 'Quantity Per Unit': '36 boxes', 'Unit Price': 21.35,
    'Units In Stock': 0, 'Discontinued': true
  },
  {
    'Product Id': 6, 'Product Name': 'Grandma\'s Boysenberry Spread',
    'Supplier Id': 3, 'Quantity Per Unit': '12 - 8 oz jars', 'Unit Price':
    25.00, 'Units In Stock': 120, 'Discontinued': false
  },
  {
    'Product Id': 7, 'Product Name': 'Uncle Bob\'s Organic Dried Pears',
    'Supplier Id': 3, 'Quantity Per Unit': '12 - 1 lb pkgs.', 'Unit Price':
    30.00, 'Units In Stock': 15, 'Discontinued': true
  },
  {
    'Product Id': 10, 'Product Name': 'Queso Cabrales', 'Supplier Id':
    5, 'Quantity Per Unit': '1 kg pkg.', 'Unit Price': 21.00, 'Units In Stock':
    22, 'Discontinued': false
  }
];

```

**DATASOURCE.TSX**

```

/**
 * Default data source
 */
export let defaultData: Object[] = [
  {
    'Product Id': 1, 'Product Name': 'Chai', 'Supplier Id': 1, 'Quantity Per
    Unit': '10 boxes x 20 bags', 'Unit Price': 18.00, 'Units In Stock': 39,
    'Discontinued': false
  },
  {
    'Product Id': 2, 'Product Name': 'Chang', 'Supplier Id': 1, 'Quantity
    Per Unit': '24 - 12 oz bottles', 'Unit Price': 19.00, 'Units In Stock': 17,
    'Discontinued': true
  },
  {
    'Product Id': 3, 'Product Name': 'Aniseed Syrup', 'Supplier Id': 1,
    'Quantity Per Unit': '12 - 550 ml bottles', 'Unit Price': 10.00, 'Units In
    Stock': 13, 'Discontinued': false
  },
  {
    'Product Id': 4, 'Product Name': 'Chef Anton\'s Cajun Seasoning',
    'Supplier Id': 2, 'Quantity Per Unit': '48 - 6 oz jars', 'Unit Price':
    22.00, 'Units In Stock': 53, 'Discontinued': true
  },
  {
    'Product Id': 5, 'Product Name': 'Chef Anton\'s Gumbo Mix', 'Supplier
    Id': 2, 'Quantity Per Unit': '36 boxes', 'Unit Price': 21.35, 'Units In
    Stock': 0, 'Discontinued': true
  },
  {
    'Product Id': 6, 'Product Name': 'Grandma\'s Boysenberry Spread',
    'Supplier Id': 3, 'Quantity Per Unit': '12 - 8 oz jars', 'Unit Price':
    25.00, 'Units In Stock': 120, 'Discontinued': false
  },
  {
    'Product Id': 7, 'Product Name': 'Uncle Bob\'s Organic Dried Pears',
    'Supplier Id': 3, 'Quantity Per Unit': '12 - 1 lb pkgs.', 'Unit Price':
    30.00, 'Units In Stock': 15, 'Discontinued': true
  },
  {
    'Product Id': 10, 'Product Name': 'Queso Cabrales', 'Supplier Id': 5,
    'Quantity Per Unit': '1 kg pkg.', 'Unit Price': 21.00, 'Units In Stock': 22,
    'Discontinued': false
  }
];

```

**Note**

You can refer to our [React Spreadsheet](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Spreadsheet example](#) to know how to present and manipulate data.

**Protect sheet in React Spreadsheet component**

Sheet protection helps you to prevent the users from modifying the data in the spreadsheet.

## Protect Sheet

Protect sheet feature helps you to prevent the unknown users from accidentally changing, editing, moving, or deleting data in a spreadsheet. And you can also protect the sheet with password. You can use the [isProtected](#) property to enable or disable the Protecting functionality.

\* The default value for `isProtected` property is `false`.

By default in protected sheet, selecting, formatting, inserting, deleting functionalities are disabled. To enable some of the above said functionalities

the `protectSettings` options are used in a protected spreadsheet.

The available `protectSettings` options in spreadsheet are,

Options	Uses
-----	-----
Select Cells	Used to perform Cell Selection.
Format Cells	Used to perform Cell formatting.
Format Rows	Used to perform Row formatting.
Format Columns	Used to perform Column formatting.
Insert Link	Used to perform Hyperlink Insertions.

\* The default value for all `protectSettings` options are `false`.

By default, the `Protect Sheet` module is injected internally into the Spreadsheet to perform sheet protection function.

### User Interface:

In the active Spreadsheet, the sheet protection can be done by any of the following ways:

- Select the Protect Sheet item in the Ribbon toolbar under the Data Tab, and then select your desired options.
- Right-click the sheet tab, select the Protect Sheet item in the context menu, and then select your desired options.
- Use the [protectSheet\(\)](#) method programmatically.

The following example shows `Protect Sheet` functionality in the Spreadsheet control.

### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { budgetData, salaryData } from './datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
```

```

const dataBound = () => {
  let spreadsheet = spreadsheetRef.current;
  if (spreadsheet) {
    spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A1:D1');
    spreadsheet.cellFormat({ fontWeight: 'bold' }, 'A11:D11');
    spreadsheet.protectSheet(1, { selectCells: false },
"syncfusion"); // protect sheet with password
  }
};
return (
  <div>
    <SpreadsheetComponent ref={spreadsheetRef} dataBound={dataBound}
allowAutoFill={false} >
      <SheetsDirective>
        <SheetDirective name={"Budget"} isProtected={true}
protectSettings={{ selectCells: true }}>
          <RangesDirective>
            <RangeDirective
dataSource={budgetData}></RangeDirective>
          </RangesDirective>
          <ColumnsDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
          </ColumnsDirective>
        </SheetDirective>
        <SheetDirective name={"Salary"}>
          <RangesDirective>
            <RangeDirective
dataSource={salaryData}></RangeDirective>
          </RangesDirective>
          <ColumnsDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
          </ColumnsDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  </div>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

**APP.TSX**

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';

```

```

import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { budgetData, salaryData } from '../datasource';
function App() {
    const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
    const dataBound = (): void => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A1:D1');
            spreadsheet.cellFormat({ fontWeight: 'bold' }, 'A11:D11');
            spreadsheet.protectSheet(1, { selectCells: false },
"syncfusion"); // protect sheet with password
        }
    };
    return (
        <div>
            <SpreadsheetComponent ref={spreadsheetRef} dataBound={dataBound}
allowAutoFill={false} >
                <SheetsDirective>
                    <SheetDirective name={"Budget"} isProtected={true}
protectSettings={{ selectCells: true }}>
                        <RangesDirective>
                            <RangeDirective
dataSource={budgetData}></RangeDirective>
                        </RangesDirective>
                        <ColumnsDirective>
                            <ColumnDirective width={100}></ColumnDirective>
                            <ColumnDirective width={100}></ColumnDirective>
                            <ColumnDirective width={100}></ColumnDirective>
                            <ColumnDirective width={100}></ColumnDirective>
                        </ColumnsDirective>
                    </SheetDirective>
                    <SheetDirective name={"Salary"}>
                        <RangesDirective>
                            <RangeDirective
dataSource={salaryData}></RangeDirective>
                        </RangesDirective>
                        <ColumnsDirective>
                            <ColumnDirective width={100}></ColumnDirective>
                            <ColumnDirective width={100}></ColumnDirective>
                            <ColumnDirective width={100}></ColumnDirective>
                            <ColumnDirective width={100}></ColumnDirective>
                        </ColumnsDirective>
                    </SheetDirective>
                </SheetsDirective>
            </SpreadsheetComponent>
        </div>
    );
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% enddraw %}

```

**DATASOURCE.JSX**

```
export let budgetData = [  
  {  
    "Expense Type": "Housing",  
    "Projected Cost": 7000,  
    "Actual Cost": 7500,  
    "Difference": -500,  
  },  
  {  
    "Expense Type": "Transportation",  
    "Projected Cost": 500,  
    "Actual Cost": 500,  
    "Difference": 0,  
  },  
  {  
    "Expense Type": "Insurance",  
    "Projected Cost": 1000,  
    "Actual Cost": 1000,  
    "Difference": 0,  
  },  
  {  
    "Expense Type": "Food",  
    "Projected Cost": 2000,  
    "Actual Cost": 1800,  
    "Difference": 200,  
  },  
  {  
    "Expense Type": "Pets",  
    "Projected Cost": 300,  
    "Actual Cost": 200,  
    "Difference": 100,  
  },  
  {  
    "Expense Type": "Personel Care",  
    "Projected Cost": 500,  
    "Actual Cost": 500,  
    "Difference": 0,  
  }, {  
    "Expense Type": "Loan",  
    "Projected Cost": 1000,  
    "Actual Cost": 1000,  
    "Difference": 0,  
  }, {  
    "Expense Type": "Tax",  
    "Projected Cost": 200,  
    "Actual Cost": 200,  
    "Difference": 0,  
  }, {  
    "Expense Type": "Savings",  
    "Projected Cost": 1000,  
    "Actual Cost": 900,  
    "Difference": 100,  
  },  
  {  
    
```

```

        "Expense Type": "Total",
        "Projected Cost": 13500,
        "Actual Cost": 13600,
        "Difference": -100,
    }
];
export let salaryData = [
    {
        "Earnings": "Basic",
        "Credit Amount": 20000,
        "Deductions": "Provident Fund",
        "Debit Amount": 2400,
    },
    {
        "Earnings": "HRA",
        "Credit Amount": 8000,
        "Deductions": "ESI",
        "Debit Amount": 0,
    },
    {
        "Earnings": "Special Allowance",
        "Credit Amount": 25000,
        "Deductions": "Professional Tax",
        "Debit Amount": 200,
    },
    {
        "Earnings": "Incentives",
        "Credit Amount": 2000,
        "Deductions": "TDS",
        "Debit Amount": 2750,
    },
    {
        "Earnings": "Bonus",
        "Credit Amount": 1500,
        "Deductions": "Other Deduction",
        "Debit Amount": 0,
    },
    {
        "Earnings": "Total Earnings",
        "Credit Amount": 56500,
        "Deductions": "Total Deductions",
        "Debit Amount": 5350,
    }
];

```

**DATASOURCE.TSX**

```

export let budgetData: Object[] = [
    {
        "Expense Type": "Housing",
        "Projected Cost": 7000,
        "Actual Cost": 7500,
        "Difference": -500,
    },
    {
        "Expense Type": "Transportation",
    }
];

```



```

    "Projected Cost": 500,
    "Actual Cost": 500,
    "Difference": 0,
  },
  {
    "Expense Type": "Insurance",
    "Projected Cost": 1000,
    "Actual Cost": 1000,
    "Difference": 0,
  },
  {
    "Expense Type": "Food",
    "Projected Cost": 2000,
    "Actual Cost": 1800,
    "Difference": 200,
  },
  {
    "Expense Type": "Pets",
    "Projected Cost": 300,
    "Actual Cost": 200,
    "Difference": 100,
  },
  {
    "Expense Type": "Personel Care",
    "Projected Cost": 500,
    "Actual Cost": 500,
    "Difference": 0,
  },
  {
    "Expense Type": "Loan",
    "Projected Cost": 1000,
    "Actual Cost": 1000,
    "Difference": 0,
  },
  {
    "Expense Type": "Tax",
    "Projected Cost": 200,
    "Actual Cost": 200,
    "Difference": 0,
  },
  {
    "Expense Type": "Savings",
    "Projected Cost": 1000,
    "Actual Cost": 900,
    "Difference": 100,
  },
  {
    "Expense Type": "Total",
    "Projected Cost": 13500,
    "Actual Cost": 13600,
    "Difference": -100,
  }
]
export let salaryData: Object[] = [
  {
    "Earnings": "Basic",
    "Credit Amount": 20000,
    "Deductions": "Provident Fund",
    "Debit Amount": 2400,
  },

```

```

{
  "Earnings": "HRA",
  "Credit Amount": 8000,
  "Deductions": "ESI",
  "Debit Amount": 0,
},
{
  "Earnings": "Special Allowance",
  "Credit Amount": 25000,
  "Deductions": "Professional Tax",
  "Debit Amount": 200,
},
{
  "Earnings": "Incentives",
  "Credit Amount": 2000,
  "Deductions": "TDS",
  "Debit Amount": 2750,
},
{
  "Earnings": "Bonus",
  "Credit Amount": 1500,
  "Deductions": "Other Deduction",
  "Debit Amount": 0,
},
{
  "Earnings": "Total Earnings",
  "Credit Amount": 56500,
  "Deductions": "Total Deductions",
  "Debit Amount": 5350,
}
]

```

### Limitations of Protect sheet

- Password encryption is not supported

### Unprotect Sheet

Unprotect sheet is used to enable all the functionalities that are already disabled in a protected spreadsheet.

#### User Interface:

In the active Spreadsheet, the sheet Unprotection can be done by any of the following ways:

- Select the **Unprotect Sheet** item in the Ribbon toolbar under the Data Tab.
- Right-click the sheet tab, select the **Unprotect Sheet** item in the context menu.
- Use the [unprotectSheet\(\)](#) method programmatically.

### Unlock the particular cells in the protected sheet

In protected spreadsheet, to make some particular cell or range of cells are editable, you can use [lockCells\(\)](#) method, with the parameter **range** and **isLocked** property as false.

### APP.JSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, ColumnsDirective, ColumnDirective, RangeDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { DialogComponent } from '@syncfusion/ej2-react-popups';
import { budgetData, salaryData } from './datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  const dialogRef = React.useRef(null);
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A1:D1');
      spreadsheet.cellFormat({ fontWeight: 'bold' }, 'A11:D11');
    }
  }, []);
  const lockCells = () => {
    let spreadsheet = spreadsheetRef.current;
    let dialog = dialogRef.current;
    if (spreadsheet) {
      spreadsheet.lockCells('A1:F3', false);
    }
    if (dialog) {
      dialog.hide();
    }
  };
  const showDlg = () => {
    let dialog = dialogRef.current;
    if (dialog) {
      dialog.show();
    }
  };
  const protectSettings = { selectCells: true };
  const btn = [{ click: lockCells, buttonModel: { content: 'Ok',
isPrimary: true } }];
  return (
    <div>
      <button onClick={showDlg}>Unlock cells</button>
      <SpreadsheetComponent ref={spreadsheetRef}>
        <SheetsDirective>
          <SheetDirective name={"Budget"} isProtected={true}
protectSettings={protectSettings}>
            <RangesDirective>
              <RangeDirective
dataSource={budgetData}></RangeDirective>
            </RangesDirective>
            <ColumnsDirective>
              <ColumnDirective width={100}></ColumnDirective>
              <ColumnDirective width={100}></ColumnDirective>
              <ColumnDirective width={100}></ColumnDirective>
              <ColumnDirective width={100}></ColumnDirective>
            </ColumnsDirective>
          </SheetDirective>
          <SheetDirective name={"Salary"}>

```

```

        <RangesDirective>
          <RangeDirective
dataSource={salaryData}></RangeDirective>
        </RangesDirective>
        <ColumnsDirective>
          <ColumnDirective width={100}></ColumnDirective>
          <ColumnDirective width={100}></ColumnDirective>
          <ColumnDirective width={100}></ColumnDirective>
          <ColumnDirective width={100}></ColumnDirective>
        </ColumnsDirective>
      </SheetDirective>
    </SheetsDirective>
  </SpreadsheetComponent>
  <DialogComponent ref={dialogRef} header={'Spreadsheet'}
target={document.getElementById('spreadsheet')} content={'"A1:F3" range of
cells has been unlocked.'}
    showCloseIcon={true} isModal={true} visible={false}
width={'500px'} buttons={btn}>
    </DialogComponent>
  </div>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

**APP.TSX**

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, ColumnsDirective, ColumnDirective, RangeDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { ButtonPropsModel, DialogComponent } from '@syncfusion/ej2-react-
popups';
import { budgetData, salaryData } from '../datasource';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  const dialogRef = React.useRef<DialogComponent>(null);
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A1:D1');
      spreadsheet.cellFormat({ fontWeight: 'bold' }, 'A11:D11');
    }
  }, []);
  const lockCells = (): void => {
    let spreadsheet = spreadsheetRef.current;
    let dialog = dialogRef.current;
    if (spreadsheet) {
      spreadsheet.lockCells('A1:F3', false);
    }
    if (dialog) {

```

```

        dialog.hide();
    }
};
const showDlg = (): void => {
    let dialog = dialogRef.current;
    if (dialog) {
        dialog.show();
    }
};
const protectSettings = { selectCells: true };
const btn: ButtonPropsModel[] = [{ click: lockCells, buttonModel: {
content: 'Ok', isPrimary: true } }];
return (
    <div>
        <button onClick={showDlg}>Unlock cells</button>
        <SpreadsheetComponent ref={spreadsheetRef}>
            <SheetsDirective>
                <SheetDirective name={"Budget"} isProtected={true}
protectSettings={protectSettings}>
                    <RangesDirective>
                        <RangeDirective
dataSource={budgetData}></RangeDirective>
                    </RangesDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={100}></ColumnDirective>
                        <ColumnDirective width={100}></ColumnDirective>
                        <ColumnDirective width={100}></ColumnDirective>
                        <ColumnDirective width={100}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
                <SheetDirective name={"Salary"}>
                    <RangesDirective>
                        <RangeDirective
dataSource={salaryData}></RangeDirective>
                    </RangesDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={100}></ColumnDirective>
                        <ColumnDirective width={100}></ColumnDirective>
                        <ColumnDirective width={100}></ColumnDirective>
                        <ColumnDirective width={100}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
            </SheetsDirective>
        </SpreadsheetComponent>
        <DialogComponent ref={dialogRef} header={'Spreadsheet'}
target={document.getElementById('spreadsheet') as HTMLElement}
content={"A1:F3" range of cells has been unlocked.}
        showCloseIcon={true} isModal={true} visible={false}
width={'500px'} buttons={btn}>
    </DialogComponent>
    </div>
);
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% enddraw %}

```

**DATASOURCE.JSX**

```
export let budgetData = [
  {
    "Expense Type": "Housing",
    "Projected Cost": 7000,
    "Actual Cost": 7500,
    "Difference": -500,
  },
  {
    "Expense Type": "Transportation",
    "Projected Cost": 500,
    "Actual Cost": 500,
    "Difference": 0,
  },
  {
    "Expense Type": "Insurance",
    "Projected Cost": 1000,
    "Actual Cost": 1000,
    "Difference": 0,
  },
  {
    "Expense Type": "Food",
    "Projected Cost": 2000,
    "Actual Cost": 1800,
    "Difference": 200,
  },
  {
    "Expense Type": "Pets",
    "Projected Cost": 300,
    "Actual Cost": 200,
    "Difference": 100,
  },
  {
    "Expense Type": "Personel Care",
    "Projected Cost": 500,
    "Actual Cost": 500,
    "Difference": 0,
  },
  {
    "Expense Type": "Loan",
    "Projected Cost": 1000,
    "Actual Cost": 1000,
    "Difference": 0,
  },
  {
    "Expense Type": "Tax",
    "Projected Cost": 200,
    "Actual Cost": 200,
    "Difference": 0,
  },
  {
    "Expense Type": "Savings",
    "Projected Cost": 1000,
    "Actual Cost": 900,
    "Difference": 100,
  },
  {
    {
```

```

        "Expense Type": "Total",
        "Projected Cost": 13500,
        "Actual Cost": 13600,
        "Difference": -100,
    }
];
export let salaryData = [
    {
        "Earnings": "Basic",
        "Credit Amount": 20000,
        "Deductions": "Provident Fund",
        "Debit Amount": 2400,
    },
    {
        "Earnings": "HRA",
        "Credit Amount": 8000,
        "Deductions": "ESI",
        "Debit Amount": 0,
    },
    {
        "Earnings": "Special Allowance",
        "Credit Amount": 25000,
        "Deductions": "Professional Tax",
        "Debit Amount": 200,
    },
    {
        "Earnings": "Incentives",
        "Credit Amount": 2000,
        "Deductions": "TDS",
        "Debit Amount": 2750,
    },
    {
        "Earnings": "Bonus",
        "Credit Amount": 1500,
        "Deductions": "Other Deduction",
        "Debit Amount": 0,
    },
    {
        "Earnings": "Total Earnings",
        "Credit Amount": 56500,
        "Deductions": "Total Deductions",
        "Debit Amount": 5350,
    }
];

```

**DATASOURCE.TSX**

```

export let budgetData: Object[] = [
    {
        "Expense Type": "Housing",
        "Projected Cost": 7000,
        "Actual Cost": 7500,
        "Difference": -500,
    },
    {
        "Expense Type": "Transportation",
    }
];

```

```

    "Projected Cost": 500,
    "Actual Cost": 500,
    "Difference": 0,
  },
  {
    "Expense Type": "Insurance",
    "Projected Cost": 1000,
    "Actual Cost": 1000,
    "Difference": 0,
  },
  {
    "Expense Type": "Food",
    "Projected Cost": 2000,
    "Actual Cost": 1800,
    "Difference": 200,
  },
  {
    "Expense Type": "Pets",
    "Projected Cost": 300,
    "Actual Cost": 200,
    "Difference": 100,
  },
  {
    "Expense Type": "Personel Care",
    "Projected Cost": 500,
    "Actual Cost": 500,
    "Difference": 0,
  },
  {
    "Expense Type": "Loan",
    "Projected Cost": 1000,
    "Actual Cost": 1000,
    "Difference": 0,
  },
  {
    "Expense Type": "Tax",
    "Projected Cost": 200,
    "Actual Cost": 200,
    "Difference": 0,
  },
  {
    "Expense Type": "Savings",
    "Projected Cost": 1000,
    "Actual Cost": 900,
    "Difference": 100,
  },
  {
    "Expense Type": "Total",
    "Projected Cost": 13500,
    "Actual Cost": 13600,
    "Difference": -100,
  }
]
export let salaryData: Object[] = [
  {
    "Earnings": "Basic",
    "Credit Amount": 20000,
    "Deductions": "Provident Fund",
    "Debit Amount": 2400,
  },

```



```

    {
      "Earnings": "HRA",
      "Credit Amount": 8000,
      "Deductions": "ESI",
      "Debit Amount": 0,
    },
    {
      "Earnings": "Special Allowance",
      "Credit Amount": 25000,
      "Deductions": "Professional Tax",
      "Debit Amount": 200,
    },
    {
      "Earnings": "Incentives",
      "Credit Amount": 2000,
      "Deductions": "TDS",
      "Debit Amount": 2750,
    },
    {
      "Earnings": "Bonus",
      "Credit Amount": 1500,
      "Deductions": "Other Deduction",
      "Debit Amount": 0,
    },
    {
      "Earnings": "Total Earnings",
      "Credit Amount": 56500,
      "Deductions": "Total Deductions",
      "Debit Amount": 5350,
    }
  ]

```

### Protect Workbook

Protect workbook feature helps you to protect the workbook so that users cannot insert, delete, rename, hide the sheets in the spreadsheet. You can use the [password](#) property to protect workbook with password. You can use the [isProtected](#) property to protect or unprotect the workbook without the password.

The default value for `isProtected` property is `false`.

### User Interface:

In the active Spreadsheet, you can protect the worksheet by selecting the Data tab in the Ribbon toolbar and choosing the **Protect Workbook** item. Then, enter the password and confirm it and click on OK.

The following example shows **Protect Workbook** by using the [isProtected](#) property in the Spreadsheet control.

### APP.JSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';

```

```

import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
  return (
    <SpreadsheetComponent isProtected={true}>
      <SheetsDirective>
        <SheetDirective>
          <RangesDirective>
            <RangeDirective dataSource={data}></RangeDirective>
          </RangesDirective>
          <ColumnsDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={110}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={180}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
          </ColumnsDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
  return (
    <SpreadsheetComponent isProtected={true}>
      <SheetsDirective>
        <SheetDirective>
          <RangesDirective>
            <RangeDirective dataSource={data}></RangeDirective>
          </RangesDirective>
          <ColumnsDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={110}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={180}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
          </ColumnsDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

```

        </SheetsDirective>
      </SpreadsheetComponent>
    );
  };
  export default App;
  const root = createRoot(document.getElementById('root'));
  root.render(<App />);
  {% endraw %}

```

### DATASOURCE.JSX

```

/**
 * Default data source
 */
export let data = [{
  OrderID: 10248,
  CustomerID: 'VINET',
  EmployeeID: 5,
  ShipName: 'Vins et alcools Chevalier',
  ShipCity: 'Reims',
  ShipAddress: '59 rue de l'Abbaye'
},
{
  OrderID: 10249,
  CustomerID: 'TOMSP',
  EmployeeID: 6,
  ShipName: 'Toms Spezialitäten',
  ShipCity: 'Münster',
  ShipAddress: 'Luisenstr. 48'
},
{
  OrderID: 10250,
  CustomerID: 'HANAR',
  EmployeeID: 4,
  ShipName: 'Hanari Carnes',
  ShipCity: 'Rio de Janeiro',
  ShipAddress: 'Rua do Paço, 67'
},
{
  OrderID: 10251,
  CustomerID: 'VICTE',
  EmployeeID: 3,
  ShipName: 'Victuailles en stock',
  ShipCity: 'Lyon',
  ShipAddress: '2, rue du Commerce'
},
{
  OrderID: 10252,
  CustomerID: 'SUPRD',
  EmployeeID: 4,
  ShipName: 'Suprêmes délices',
  ShipCity: 'Charleroi',
  ShipAddress: 'Boulevard Tirou, 255'
}
]];

```

**DATASOURCE.TSX**

```

/**
 * Default data source
 */
export let data: Object[] = [{
  OrderID: 10248,
  CustomerID: 'VINET',
  EmployeeID: 5,
  ShipName: 'Vins et alcools Chevalier',
  ShipCity: 'Reims',
  ShipAddress: '59 rue de l'Abbaye'
},
{
  OrderID: 10249,
  CustomerID: 'TOMSP',
  EmployeeID: 6,
  ShipName: 'Toms Spezialitäten',
  ShipCity: 'Münster',
  ShipAddress: 'Luisenstr. 48'
},
{
  OrderID: 10250,
  CustomerID: 'HANAR',
  EmployeeID: 4,
  ShipName: 'Hanari Carnes',
  ShipCity: 'Rio de Janeiro',
  ShipAddress: 'Rua do Paço, 67'
},
{
  OrderID: 10251,
  CustomerID: 'VICTE',
  EmployeeID: 3,
  ShipName: 'Victuailles en stock',
  ShipCity: 'Lyon',
  ShipAddress: '2, rue du Commerce'
},
{
  OrderID: 10252,
  CustomerID: 'SUPRD',
  EmployeeID: 4,
  ShipName: 'Suprêmes délices',
  ShipCity: 'Charleroi',
  ShipAddress: 'Boulevard Tirou, 255'
}
];

```

The following example shows **Protect Workbook** by using the [password](#) property in the Spreadsheet control. To unprotect the workbook, click the unprotect workbook button in the data tab and provide the password as syncfusion in the dialog box.

**APP.JSX**

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';

```

```

import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { data } from '../datasource';
function App() {
    return (
        <SpreadsheetComponent password='syncfusion' >
            <SheetsDirective>
                <SheetDirective>
                    <RangesDirective>
                        <RangeDirective dataSource={data}></RangeDirective>
                    </RangesDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={100}></ColumnDirective>
                        <ColumnDirective width={110}></ColumnDirective>
                        <ColumnDirective width={100}></ColumnDirective>
                        <ColumnDirective width={180}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
            </SheetsDirective>
        </SpreadsheetComponent>
    );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

**APP.TSX**

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { data } from '../datasource';
function App() {
    return (
        <SpreadsheetComponent password='syncfusion' >
            <SheetsDirective>
                <SheetDirective>
                    <RangesDirective>
                        <RangeDirective dataSource={data}></RangeDirective>
                    </RangesDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={100}></ColumnDirective>
                        <ColumnDirective width={110}></ColumnDirective>
                        <ColumnDirective width={100}></ColumnDirective>
                        <ColumnDirective width={180}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
            </SheetsDirective>
        </SpreadsheetComponent>
    );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

```

        </ColumnsDirective>
      </SheetDirective>
    </SheetsDirective>
  </SpreadsheetComponent>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### DATASOURCE.JSX

```

/**
 * Default data source
 */
export let data = [{
  OrderID: 10248,
  CustomerID: 'VINET',
  EmployeeID: 5,
  ShipName: 'Vins et alcools Chevalier',
  ShipCity: 'Reims',
  ShipAddress: '59 rue de l'Abbaye'
},
{
  OrderID: 10249,
  CustomerID: 'TOMSP',
  EmployeeID: 6,
  ShipName: 'Toms Spezialitäten',
  ShipCity: 'Münster',
  ShipAddress: 'Luisenstr. 48'
},
{
  OrderID: 10250,
  CustomerID: 'HANAR',
  EmployeeID: 4,
  ShipName: 'Hanari Carnes',
  ShipCity: 'Rio de Janeiro',
  ShipAddress: 'Rua do Paço, 67'
},
{
  OrderID: 10251,
  CustomerID: 'VICTE',
  EmployeeID: 3,
  ShipName: 'Victuailles en stock',
  ShipCity: 'Lyon',
  ShipAddress: '2, rue du Commerce'
},
{
  OrderID: 10252,
  CustomerID: 'SUPRD',
  EmployeeID: 4,
  ShipName: 'Suprêmes délices',
  ShipCity: 'Charleroi',
  ShipAddress: 'Boulevard Tirou, 255'
}
]];

```

**DATASOURCE.TSX**

```

/**
 * Default data source
 */
export let data: Object[] = [{
  OrderID: 10248,
  CustomerID: 'VINET',
  EmployeeID: 5,
  ShipName: 'Vins et alcools Chevalier',
  ShipCity: 'Reims',
  ShipAddress: '59 rue de l'Abbaye'
},
{
  OrderID: 10249,
  CustomerID: 'TOMSP',
  EmployeeID: 6,
  ShipName: 'Toms Spezialitäten',
  ShipCity: 'Münster',
  ShipAddress: 'Luisenstr. 48'
},
{
  OrderID: 10250,
  CustomerID: 'HANAR',
  EmployeeID: 4,
  ShipName: 'Hanari Carnes',
  ShipCity: 'Rio de Janeiro',
  ShipAddress: 'Rua do Paço, 67'
},
{
  OrderID: 10251,
  CustomerID: 'VICTE',
  EmployeeID: 3,
  ShipName: 'Victuailles en stock',
  ShipCity: 'Lyon',
  ShipAddress: '2, rue du Commerce'
},
{
  OrderID: 10252,
  CustomerID: 'SUPRD',
  EmployeeID: 4,
  ShipName: 'Suprêmes délices',
  ShipCity: 'Charleroi',
  ShipAddress: 'Boulevard Tirou, 255'
}
];

```

**Unprotect Workbook**

Unprotect Workbook is used to enable the insert, delete, rename, move, copy, hide or unhide sheets feature in the spreadsheet.

**User Interface:**

In the active Spreadsheet, the workbook Unprotection can be done in any of the following ways:

- Select the **Unprotect Workbook** item in the Ribbon toolbar under the Data Tab and provide the valid password in the dialog box.

#### Note

You can refer to our [React Spreadsheet](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Spreadsheet example](#) to know how to present and manipulate data.

#### See Also

- [Hyperlink](#)

### Searching in React Spreadsheet component

Find and Replace helps you to search for target text and replace the founded text with alternative text within the sheet or workbook. You can use the [allowFindAndReplace](#) property to enable or disable Find and Replace functionality.

\* The default value for **allowFindAndReplace** property is **true**.

#### Find

Find is used to select the matched contents of a cell within sheet or workbook. It is extremely useful when working with large sets of data source.

##### User Interface:

Find can be done by any of the following ways:

- Select the Search icon in the Ribbon toolbar or using **Ctrl + F** key to open the Find dialog.
- Using find Next and find Previous buttons to search the given value in workbook.
- Select the option button in Find dialog to open the Find and Replace dialog then select the below properties for enhanced searching.

\* **Search within**: To search the target in a sheet (default) or in an entire workbook.

\* **Search by**: It enhance the search, either By Rows (default), or By Columns.

\* **Match case**: To find the matched value with case sensitive.

\* **Match exact cell contents**: To find the exact matched cell value with entire match cases.

- Using [find\(\)](#) method to perform find operation.

#### Replace

Replace is used to change the find contents of a cell within sheet or workbook. Replace All is used to change all the matched contents of a cell within sheet or workbook.

##### User Interface:

Replace can be done by any of the following ways:

- Using **Ctrl + H** key to open the Find and Replace dialog.
- Using Replace button to change the found value in sheet or workbook.



- Using ReplaceAll button to change all the found value's in sheet or workbook.
- Using [replace\(\)](#) method to perform replace operation by passing the argument `args.replaceby` as `replace`.
- Using [replace\(\)](#) method to perform replace all operation by passing the argument `args.replaceby` as `replaceall`.

### Go to

Go to is used to navigate to a specific cell address in the sheet or workbook.

### User Interface:

- Using `Ctrl + G` key to open the Go To dialog.
- Using [goTo\(\)](#) method to perform Go To operation.

In the following sample, searching can be done by following ways:

- Select the Home tab in the Ribbon toolbar, and then select the Search icon.
- Enter any value in the search textbox.
- Select the next (or) previous button to find the entered value in the spreadsheet.
- You can have more options to find values by selecting the more options in the search toolbar.

### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { budgetData, defaultData } from '../datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  const onDataBound = () => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A1:D1');
      spreadsheet.cellFormat({ fontWeight: 'bold' }, 'A11:D11');
    }
  };
  return (
    <SpreadsheetComponent allowFindAndReplace={false}
ref={spreadsheetRef} dataBound={onDataBound}>
      <SheetsDirective>
        <SheetDirective name='Price Details'>
          <RangesDirective>
            <RangeDirective
dataSource={defaultData}></RangeDirective>
          </RangesDirective>
          <ColumnsDirective>
            <ColumnDirective width={130}></ColumnDirective>
            <ColumnDirective width={92}></ColumnDirective>
          </ColumnsDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  );
}
```

```

        <ColumnDirective width={96}></ColumnDirective>
      </ColumnsDirective>
    </SheetDirective>
    <SheetDirective name='Budget'>
      <RangesDirective>
        <RangeDirective
dataSource={budgetData}></RangeDirective>
      </RangesDirective>
      <ColumnsDirective>
        <ColumnDirective width={130}></ColumnDirective>
        <ColumnDirective width={92}></ColumnDirective>
        <ColumnDirective width={96}></ColumnDirective>
      </ColumnsDirective>
    </SheetDirective>
  </SheetsDirective>
</SpreadsheetComponent>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { budgetData, defaultData } from '../datasource';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  const onDataBound = (): void => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A1:D1');
      spreadsheet.cellFormat({ fontWeight: 'bold' }, 'A11:D11');
    }
  };
  return (
    <SpreadsheetComponent allowFindAndReplace={false}
ref={spreadsheetRef} dataBound={onDataBound}>
      <SheetsDirective>
        <SheetDirective name='Price Details'>
          <RangesDirective>
            <RangeDirective
dataSource={defaultData}></RangeDirective>
          </RangesDirective>
          <ColumnsDirective>
            <ColumnDirective width={130}></ColumnDirective>
            <ColumnDirective width={92}></ColumnDirective>
            <ColumnDirective width={96}></ColumnDirective>

```

```

        </ColumnsDirective>
      </SheetDirective>
      <SheetDirective name='Budget'>
        <RangesDirective>
          <RangeDirective
dataSource={budgetData}></RangeDirective>
        </RangesDirective>
        <ColumnsDirective>
          <ColumnDirective width={130}></ColumnDirective>
          <ColumnDirective width={92}></ColumnDirective>
          <ColumnDirective width={96}></ColumnDirective>
        </ColumnsDirective>
      </SheetDirective>
    </SheetsDirective>
  </SpreadsheetComponent>
);
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

### DATASOURCE.JSX

```

/**
 * Default data source
 */
export let defaultData = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
  { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
  { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
  { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
'12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
70 },
  { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
  { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
  { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
  { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
  { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
export let budgetData = [
  {
    "Expense Type": "Housing",
    "Projected Cost": 7000,
    "Actual Cost": 7500,
    "Difference": -500,

```

```

    },
    {
      "Expense Type": "Transportation",
      "Projected Cost": 500,
      "Actual Cost": 500,
      "Difference": 0,
    },
    {
      "Expense Type": "Insurance",
      "Projected Cost": 1000,
      "Actual Cost": 1000,
      "Difference": 0,
    },
    {
      "Expense Type": "Food",
      "Projected Cost": 2000,
      "Actual Cost": 1800,
      "Difference": 200,
    },
    {
      "Expense Type": "Pets",
      "Projected Cost": 300,
      "Actual Cost": 200,
      "Difference": 100,
    },
    {
      "Expense Type": "Personel Care",
      "Projected Cost": 500,
      "Actual Cost": 500,
      "Difference": 0,
    },
    {
      "Expense Type": "Loan",
      "Projected Cost": 1000,
      "Actual Cost": 1000,
      "Difference": 0,
    },
    {
      "Expense Type": "Tax",
      "Projected Cost": 200,
      "Actual Cost": 200,
      "Difference": 0,
    },
    {
      "Expense Type": "Savings",
      "Projected Cost": 1000,
      "Actual Cost": 900,
      "Difference": 100,
    },
    {
      "Expense Type": "Total",
      "Projected Cost": 13500,
      "Actual Cost": 13600,
      "Difference": -100,
    }
  ];

```

**DATASOURCE.TSX**

```

/**
 * Default data source
 */
export let defaultData: Object[] = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
    Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
  { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
    Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
  { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
  { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
'12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
70 },
  { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
    Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
  { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
    Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
  { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
    Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
  { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
    Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
  { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
    Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
export let budgetData: Object[] = [
  {
    "Expense Type": "Housing",
    "Projected Cost": 7000,
    "Actual Cost": 7500,
    "Difference": -500,
  },
  {
    "Expense Type": "Transportation",
    "Projected Cost": 500,
    "Actual Cost": 500,
    "Difference": 0,
  },
  {
    "Expense Type": "Insurance",
    "Projected Cost": 1000,
    "Actual Cost": 1000,
    "Difference": 0,
  },
  {
    "Expense Type": "Food",
    "Projected Cost": 2000,
    "Actual Cost": 1800,
    "Difference": 200,
  },
  {
    "Expense Type": "Pets",
    "Projected Cost": 300,
    "Actual Cost": 200,
    "Difference": 100,
  },
];

```

```

{
  "Expense Type": "Personel Care",
  "Projected Cost": 500,
  "Actual Cost": 500,
  "Difference": 0,
}, {
  "Expense Type": "Loan",
  "Projected Cost": 1000,
  "Actual Cost": 1000,
  "Difference": 0,
}, {
  "Expense Type": "Tax",
  "Projected Cost": 200,
  "Actual Cost": 200,
  "Difference": 0,
}, {
  "Expense Type": "Savings",
  "Projected Cost": 1000,
  "Actual Cost": 900,
  "Difference": 100,
},
{
  "Expense Type": "Total",
  "Projected Cost": 13500,
  "Actual Cost": 13600,
  "Difference": -100,
}
]

```

## Limitations

- Undo/redo for Replace All is not supported in this feature.

## Note

You can refer to our [React Spreadsheet](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Spreadsheet example](#) to know how to present and manipulate data.

## Undo redo in React Spreadsheet component

**Undo** option helps you to undo the last action performed and **Redo** option helps you to do the same action which is reverted in the Spreadsheet. You can use the [allowUndoRedo](#) property to enable or disable undo redo functionality in spreadsheet.

\* The default value for `allowUndoRedo` property is `true`.

By default, the **UndoRedo** module is injected internally into Spreadsheet to perform undo redo.

## Undo

It reverses the last action you performed with Spreadsheet. Undo can be done by any of the following ways:

- Select the undo item from HOME tab in Ribbon toolbar.
- Use `Ctrl + Z` keyboard shortcut to perform the undo.

- Use the [undo](#) method programmatically.

### Redo

It reverses the last undo action you performed with Spreadsheet. Redo can be done by any of the following ways:

- Select the redo item from HOME tab in Ribbon toolbar.
- Use **Ctrl + Y** keyboard shortcut to perform the redo.
- Use the [redo](#) method programmatically.

### Update custom actions in UndoRedo collection

You can update your own custom actions in UndoRedo collection, by using the [updateUndoRedoCollection](#) method. And also customize the undo redo operations of your custom action by using **actionComplete** event.

The following code example shows **How to update and customize your own actions for undo redo functionality in the Spreadsheet control.**

### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { getRangeIndexes } from '@syncfusion/ej2-react-spreadsheet';
import { addClass, removeClass } from '@syncfusion/ej2-base';
import { defaultData } from './datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  const updateCollection = () => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      let cell = spreadsheet.getActiveSheet().activeCell;
      let cellIdx = getRangeIndexes(cell);
      let Element = spreadsheet.getCell(cellIdx[0], cellIdx[1]);
      if (!Element.classList.contains("customClass")) {
        Element.classList.add('customClass'); // To add the custom
class in active cell element
        spreadsheet.updateUndoRedoCollection({ eventArgs: { class:
'customClass', rowIdx: cellIdx[0], colIdx: cellIdx[1], action: 'customCSS' }
}); // To update the undo redo collection
      }
    }
  };
  const onActionComplete = (args) => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      let actionEvents = args;
      if (actionEvents.eventArgs.action === "customCSS") {

```

```

        let Element =
        spreadsheet.getCell(actionEvents.eventArgs.rowIdx,
        actionEvents.eventArgs.colIdx);
        if (actionEvents.isUndoRedo && actionEvents.isUndo) {
            removeClass([Element], 'customClass'); // To remove the
custom class in undo action
        } else {
            addClass([Element], 'customClass'); // To add the custom
class in redo action
        }
    }
};
return (
    <div>
        <button onClick={updateCollection}>add/remove Class</button>
        <SpreadsheetComponent
            ref={spreadsheetRef} actionComplete={onActionComplete} >
            <SheetsDirective>
                <SheetDirective>
                    <RangesDirective>
                        <RangeDirective
dataSource={defaultData}></RangeDirective>
                    </RangesDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={180}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
            </SheetsDirective>
        </SpreadsheetComponent>
    </div>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

## APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { getRangeIndexes } from '@syncfusion/ej2-react-spreadsheet';
import { addClass, removeClass } from '@syncfusion/ej2-base';
import { defaultData } from './datasource';
function App() {
    const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
    const updateCollection = (): void => {
        let spreadsheet = spreadsheetRef.current;

```



```

        if (spreadsheet) {
            let cell: string = spreadsheet.getActiveSheet().activeCell as
string;
            let cellIdx: number[] = getRangeIndexes(cell);
            let Element: HTMLElement = spreadsheet.getCell(cellIdx[0],
cellIdx[1]);
            if (!Element.classList.contains("customClass")) {
                Element.classList.add('customClass'); // To add the custom
class in active cell element
                spreadsheet.updateUndoRedoCollection({ eventArgs: { class:
'customClass', rowIdx: cellIdx[0], colIdx: cellIdx[1], action: 'customCSS' }
}); // To update the undo redo collection
            }
        }
    };
    const onActionComplete = (args: any) => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            let actionEvents = args;
            if (actionEvents.eventArgs.action === "customCSS") {
                let Element: HTMLElement =
spreadsheet.getCell(actionEvents.eventArgs.rowIdx,
actionEvents.eventArgs.colIdx);
                if (actionEvents.isUndoRedo && actionEvents.isUndo) {
                    removeClass([Element], 'customClass'); // To remove the
custom class in undo action
                } else {
                    addClass([Element], 'customClass'); // To add the custom
class in redo action
                }
            }
        }
    };
    return (
        <div>
            <button onClick={updateCollection}>add/remove Class</button>
            <SpreadsheetComponent
                ref={spreadsheetRef} actionComplete={onActionComplete} >
                <SheetsDirective>
                    <SheetDirective>
                        <RangesDirective>
                            <RangeDirective
dataSource={defaultData}></RangeDirective>
                        </RangesDirective>
                        <ColumnsDirective>
                            <ColumnDirective width={180}></ColumnDirective>
                            <ColumnDirective width={130}></ColumnDirective>
                            <ColumnDirective width={130}></ColumnDirective>
                        </ColumnsDirective>
                    </SheetDirective>
                </SheetsDirective>
            </SpreadsheetComponent>
        </div>
    );
};
export default App;
const root = createRoot(document.getElementById('root')!);

```

```
root.render(<App />);
{% endraw %}
```

## DATASOURCE.JSX

```
/**
 * Default data source
 */
export let defaultData = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
    Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
  { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
    Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
  { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
  { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
'12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
70 },
  { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
    Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
  { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
    Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
  { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
    Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
  { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
    Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
  { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
    Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];

/**
 * Grid datasource
 */
export function getTradeData(dataCount) {
  let employees = [
    'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
    'Fuller', 'Leverling', 'Peacock',
    'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
    'Dodsworth', 'Anne',
    'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
    'Jack', 'Rose'
  ];
  let designation = ['Manager', 'CFO', 'Designer', 'Developer', 'Program
Directory', 'System Analyst', 'Project Lead'];
  let mail = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
'jourrapide.com'];
  let location = ['UK', 'USA', 'Sweden', 'France', 'Canada', 'Argentina',
'Austria', 'Germany', 'Mexico'];
  let status = ['Active', 'Inactive'];
  let trustworthiness = ['Perfect', 'Sufficient', 'Insufficient'];
  let tradeData = [];
  let address = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do Paço, 67',
'2, rue du Commerce', 'Boulevard Tirou, 255',
'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
Mercado, 12',
```

```

        'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
        'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
        'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
        'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
        'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
        'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
        'Taucherstraße 10', 'Taucherstraße 10',
        'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
Circus', 'Av. dos Lusíadas, 23',
        'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
123', '2817 Milton Dr.'];
    let employeeimg = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
        dataCount = parseInt(dataCount, 10);
    }
    for (let i = 1; i <= dataCount; i++) {
        let code = 10000;
        tradeData.push({
            'EmployeeID': code + i,
            'Employees': employees[Math.floor(Math.random() *
employees.length)] + ' ' + employees[Math.floor(Math.random() *
employees.length)],
            'Designation': designation[Math.floor(Math.random() *
designation.length)],
            'Location': location[Math.floor(Math.random() *
location.length)],
            'Status': status[Math.floor(Math.random() * status.length)],
            'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
            'Rating': Math.floor(Math.random() * 5),
            'Software': Math.floor(Math.random() * 100),
            'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
            'CurrentSalary': Math.floor((Math.random() * 100000)),
            'Address': address[Math.floor(Math.random() * address.length)],
        });
        let employee = 'Employees';
        let emp = tradeData[i - 1][employee];
        let sName = emp.substr(0, emp.indexOf(' ')).toLowerCase();
        let empmail = 'Mail';
        tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }
    return tradeData;
}
export let tradeData = [
    {
        "EmployeeID": 10001,
        "Employees": "Laura Nancy",
        "Designation": "Designer",
        "Location": "France",
        "Status": "Inactive",
        "Trustworthiness": "Sufficient",
    }

```

```
    "Rating": 0,
    "Software": 69,
    "EmployeeImg": "usermale",
    "CurrentSalary": 84194,
    "Address": "Taucherstraße 10",
    "Mail": "laura15@jourrapide.com"
  },
  {
    "EmployeeID": 10002,
    "Employees": "Zachery Van",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 99,
    "EmployeeImg": "usermale",
    "CurrentSalary": 55349,
    "Address": "5a Ave. Los Palos Grandes",
    "Mail": "zachery109@sample.com"
  },
  {
    "EmployeeID": 10003,
    "Employees": "Rose Fuller",
    "Designation": "CFO",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 1,
    "EmployeeImg": "usermale",
    "CurrentSalary": 16477,
    "Address": "2817 Milton Dr.",
    "Mail": "rose55@rpy.com"
  },
  {
    "EmployeeID": 10004,
    "Employees": "Jack Bergs",
    "Designation": "Manager",
    "Location": "Mexico",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 36,
    "EmployeeImg": "usermale",
    "CurrentSalary": 49040,
    "Address": "2, rue du Commerce",
    "Mail": "jack30@sample.com"
  },
  {
    "EmployeeID": 10005,
    "Employees": "Vinet Bergs",
    "Designation": "Program Directory",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
```

```

    "Software": 39,
    "EmployeeImg": "usermale",
    "CurrentSalary": 5495,
    "Address": "Rua da Panificadora, 12",
    "Mail": "vinet32@jourrapide.com"
  },
  {
    "EmployeeID": 10006,
    "Employees": "Buchanan Van",
    "Designation": "Designer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 42182,
    "Address": "24, place Kléber",
    "Mail": "buchanan18@mail.com"
  },
  {
    "EmployeeID": 10007,
    "Employees": "Dodsworth Nancy",
    "Designation": "Project Lead",
    "Location": "USA",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 35776,
    "Address": "Rua do Paço, 67",
    "Mail": "dodsworth84@mail.com"
  },
  {
    "EmployeeID": 10008,
    "Employees": "Laura Jack",
    "Designation": "Developer",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 89,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25108,
    "Address": "Rua da Panificadora, 12",
    "Mail": "laura82@mail.com"
  },
  {
    "EmployeeID": 10009,
    "Employees": "Anne Fuller",
    "Designation": "Program Directory",
    "Location": "Mexico",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 0,
    "Software": 19,

```

```
    "EmployeeImg": "userfemale",
    "CurrentSalary": 32568,
    "Address": "Gran Vía, 1",
    "Mail": "anne97@jourrapide.com"
  },
  {
    "EmployeeID": 10010,
    "Employees": "Buchanan Andrew",
    "Designation": "Designer",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 62,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 12320,
    "Address": "P.O. Box 555",
    "Mail": "buchanan50@jourrapide.com"
  },
  {
    "EmployeeID": 10011,
    "Employees": "Andrew Janet",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 20890,
    "Address": "Starenweg 5",
    "Mail": "andrew63@mail.com"
  },
  {
    "EmployeeID": 10012,
    "Employees": "Margaret Tamer",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 7,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 22337,
    "Address": "Magazinweg 7",
    "Mail": "margaret26@mail.com"
  },
  {
    "EmployeeID": 10013,
    "Employees": "Tamer Fuller",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 78,
    "EmployeeImg": "usermale",
```

```
"CurrentSalary": 89181,  
"Address": "Taucherstraße 10",  
"Mail": "tamer40@arpy.com"  
},  
{  
  "EmployeeID": 10014,  
  "Employees": "Tamer Anne",  
  "Designation": "CFO",  
  "Location": "Sweden",  
  "Status": "Active",  
  "Trustworthiness": "Sufficient",  
  "Rating": 0,  
  "Software": 18,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 20998,  
  "Address": "Taucherstraße 10",  
  "Mail": "tamer68@arpy.com"  
},  
{  
  "EmployeeID": 10015,  
  "Employees": "Anton Davolio",  
  "Designation": "Project Lead",  
  "Location": "France",  
  "Status": "Active",  
  "Trustworthiness": "Sufficient",  
  "Rating": 4,  
  "Software": 8,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 48232,  
  "Address": "Luisenstr. 48",  
  "Mail": "anton46@mail.com"  
},  
{  
  "EmployeeID": 10016,  
  "Employees": "Buchanan Buchanan",  
  "Designation": "System Analyst",  
  "Location": "Austria",  
  "Status": "Inactive",  
  "Trustworthiness": "Perfect",  
  "Rating": 0,  
  "Software": 19,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 43041,  
  "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",  
  "Mail": "buchanan68@mail.com"  
},  
{  
  "EmployeeID": 10017,  
  "Employees": "King Buchanan",  
  "Designation": "Program Directory",  
  "Location": "Sweden",  
  "Status": "Active",  
  "Trustworthiness": "Sufficient",  
  "Rating": 0,  
  "Software": 44,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 25259,
```

```
      "Address": "Magazinweg 7",
      "Mail": "king80@jourrapide.com"
    },
    {
      "EmployeeID": 10018,
      "Employees": "Rose Michael",
      "Designation": "Project Lead",
      "Location": "Canada",
      "Status": "Active",
      "Trustworthiness": "Perfect",
      "Rating": 4,
      "Software": 31,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 91156,
      "Address": "Fauntleroy Circus",
      "Mail": "rose75@mail.com"
    },
    {
      "EmployeeID": 10019,
      "Employees": "King Bergs",
      "Designation": "Developer",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Sufficient",
      "Rating": 2,
      "Software": 29,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 28826,
      "Address": "2817 Milton Dr.",
      "Mail": "king57@jourrapide.com"
    },
    {
      "EmployeeID": 10020,
      "Employees": "Davolio Fuller",
      "Designation": "Designer",
      "Location": "Canada",
      "Status": "Inactive",
      "Trustworthiness": "Sufficient",
      "Rating": 3,
      "Software": 35,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 71035,
      "Address": "Gran Vía, 1",
      "Mail": "davolio29@arpy.com"
    },
    {
      "EmployeeID": 10021,
      "Employees": "Rose Rose",
      "Designation": "CFO",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Perfect",
      "Rating": 3,
      "Software": 38,
      "EmployeeImg": "usermale",
      "CurrentSalary": 68123,
      "Address": "Rua do Mercado, 12",
```



```
      "Mail": "rose54@arpy.com"
    },
    {
      "EmployeeID": 10022,
      "Employees": "Andrew Michael",
      "Designation": "Program Directory",
      "Location": "UK",
      "Status": "Inactive",
      "Trustworthiness": "Insufficient",
      "Rating": 2,
      "Software": 61,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 75470,
      "Address": "2, rue du Commerce",
      "Mail": "andrew88@jourrapide.com"
    },
    {
      "EmployeeID": 10023,
      "Employees": "Davolio Kathryn",
      "Designation": "Manager",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Perfect",
      "Rating": 3,
      "Software": 25,
      "EmployeeImg": "usermale",
      "CurrentSalary": 25234,
      "Address": "Hauptstr. 31",
      "Mail": "davolio42@sample.com"
    },
    {
      "EmployeeID": 10024,
      "Employees": "Anne Fleet",
      "Designation": "System Analyst",
      "Location": "UK",
      "Status": "Active",
      "Trustworthiness": "Perfect",
      "Rating": 3,
      "Software": 0,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 8341,
      "Address": "59 rue de l'Abbaye",
      "Mail": "anne86@arpy.com"
    },
    {
      "EmployeeID": 10025,
      "Employees": "Margaret Andrew",
      "Designation": "System Analyst",
      "Location": "Germany",
      "Status": "Inactive",
      "Trustworthiness": "Insufficient",
      "Rating": 3,
      "Software": 51,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 84975,
      "Address": "P.O. Box 555",
      "Mail": "margaret41@arpy.com"
```

```
{
  "EmployeeID": 10026,
  "Employees": "Kathryn Laura",
  "Designation": "Project Lead",
  "Location": "Austria",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 48,
  "EmployeeImg": "usermale",
  "CurrentSalary": 97282,
  "Address": "Avda. Azteca 123",
  "Mail": "kathryn82@rpy.com"
},
{
  "EmployeeID": 10027,
  "Employees": "Michael Michael",
  "Designation": "Developer",
  "Location": "UK",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 4,
  "Software": 16,
  "EmployeeImg": "usermale",
  "CurrentSalary": 4184,
  "Address": "Rua do Paço, 67",
  "Mail": "michael58@jourrapide.com"
},
{
  "EmployeeID": 10028,
  "Employees": "Leverling Vinet",
  "Designation": "Project Lead",
  "Location": "Germany",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 57,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 38370,
  "Address": "59 rue de l'Abbaye",
  "Mail": "leverling102@sample.com"
},
{
  "EmployeeID": 10029,
  "Employees": "Rose Jack",
  "Designation": "Developer",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 46,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 84790,
  "Address": "Rua do Mercado, 12",
  "Mail": "rosl08@jourrapide.com"
}
```

```

{
  "EmployeeID": 10030,
  "Employees": "Vinet Van",
  "Designation": "Developer",
  "Location": "USA",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 40,
  "EmployeeImg": "usermale",
  "CurrentSalary": 71005,
  "Address": "Gran Vía, 1",
  "Mail": "vinet90@jourrapide.com"
}
];

```

### DATASOURCE.TSX

```

/**
 * Default data source
 */
export let defaultData: Object[] = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
    Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
  { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
    Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
  { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
  { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
'12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
70 },
  { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
    Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
  { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
    Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
  { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
    Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
  { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
    Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
  { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
    Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
/**
 * Grid datasource
 */
export function getTradeData(dataCount?: number): object {
  let employees: string[] = [
    'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
    'Fuller', 'Leverling', 'Peacock',
    'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
    'Dodsworth', 'Anne',
    'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
    'Jack', 'Rose'];

```

```

    let designation: string[] = ['Manager', 'CFO', 'Designer', 'Developer',
    'Program Directory', 'System Analyst', 'Project Lead'];
    let mail: string[] = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
    'jourrapide.com'];
    let location: string[] = ['UK', 'USA', 'Sweden', 'France', 'Canada',
    'Argentina', 'Austria', 'Germany', 'Mexico'];
    let status: string[] = ['Active', 'Inactive'];
    let trustworthiness: string[] = ['Perfect', 'Sufficient',
    'Insufficient'];
    let tradeData: Object[] = [];
    let address: string[] = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do
    Paço, 67', '2, rue du Commerce', 'Boulevard Tirou, 255',
    'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
    Mercado, 12',
    'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
    'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
    Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
    'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
    43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
    'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
    10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
    'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
    7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
    'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
    'Taucherstraße 10', 'Taucherstraße 10',
    'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
    Circus', 'Av. dos Lusíadas, 23',
    'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
    123', '2817 Milton Dr.'];
    let employeeimg: string[] = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
        dataCount = parseInt(dataCount, 10);
    }
    for (let i: number = 1; i <= dataCount; i++) {
        let code: any = 10000;
        tradeData.push({
            'EmployeeID': code + i,
            'Employees':
                employees[Math.floor(Math.random() * employees.length)] + '
' + employees[Math.floor(Math.random() * employees.length)],
            'Designation': designation[Math.floor(Math.random() *
    designation.length)],
            'Location': location[Math.floor(Math.random() *
    location.length)],
            'Status': status[Math.floor(Math.random() * status.length)],
            'Trustworthiness': trustworthiness[Math.floor(Math.random() *
    trustworthiness.length)],
            'Rating': Math.floor(Math.random() * 5),
            'Software': Math.floor(Math.random() * 100),
            'EmployeeImg': employeeimg[Math.floor(Math.random() *
    employeeimg.length)],
            'CurrentSalary': Math.floor((Math.random() * 100000)),
            'Address': address[Math.floor(Math.random() * address.length)],
        });
        let employee: string = 'Employees';
        let emp: string = tradeData[i - 1][employee];
        let sName: string = emp.substr(0, emp.indexOf(' ')).toLowerCase();

```

```

        let empmail: string = 'Mail';
        tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }
    return tradeData;
}
export let tradeData: Object[] = [
    {
        "EmployeeID": 10001,
        "Employees": "Laura Nancy",
        "Designation": "Designer",
        "Location": "France",
        "Status": "Inactive",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
        "Software": 69,
        "EmployeeImg": "usermale",
        "CurrentSalary": 84194,
        "Address": "Taucherstraße 10",
        "Mail": "laura15@jourrapide.com"
    },
    {
        "EmployeeID": 10002,
        "Employees": "Zachery Van",
        "Designation": "CFO",
        "Location": "Canada",
        "Status": "Inactive",
        "Trustworthiness": "Insufficient",
        "Rating": 3,
        "Software": 99,
        "EmployeeImg": "usermale",
        "CurrentSalary": 55349,
        "Address": "5a Ave. Los Palos Grandes",
        "Mail": "zachery109@sample.com"
    },
    {
        "EmployeeID": 10003,
        "Employees": "Rose Fuller",
        "Designation": "CFO",
        "Location": "France",
        "Status": "Active",
        "Trustworthiness": "Insufficient",
        "Rating": 1,
        "Software": 1,
        "EmployeeImg": "usermale",
        "CurrentSalary": 16477,
        "Address": "2817 Milton Dr.",
        "Mail": "rose55@rpy.com"
    },
    {
        "EmployeeID": 10004,
        "Employees": "Jack Bergs",
        "Designation": "Manager",
        "Location": "Mexico",
        "Status": "Inactive",
        "Trustworthiness": "Insufficient",
        "Rating": 3,

```

```
"Software": 36,  
"EmployeeImg": "usermale",  
"CurrentSalary": 49040,  
"Address": "2, rue du Commerce",  
"Mail": "jack30@sample.com"  
},  
{  
  "EmployeeID": 10005,  
  "Employees": "Vinet Bergs",  
  "Designation": "Program Directory",  
  "Location": "UK",  
  "Status": "Active",  
  "Trustworthiness": "Insufficient",  
  "Rating": 1,  
  "Software": 39,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 5495,  
  "Address": "Rua da Panificadora, 12",  
  "Mail": "vinet32@jourrapide.com"  
},  
{  
  "EmployeeID": 10006,  
  "Employees": "Buchanan Van",  
  "Designation": "Designer",  
  "Location": "Germany",  
  "Status": "Active",  
  "Trustworthiness": "Insufficient",  
  "Rating": 4,  
  "Software": 78,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 42182,  
  "Address": "24, place Kléber",  
  "Mail": "buchanan18@mail.com"  
},  
{  
  "EmployeeID": 10007,  
  "Employees": "Dodsworth Nancy",  
  "Designation": "Project Lead",  
  "Location": "USA",  
  "Status": "Inactive",  
  "Trustworthiness": "Sufficient",  
  "Rating": 0,  
  "Software": 0,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 35776,  
  "Address": "Rua do Paço, 67",  
  "Mail": "dodsworth84@mail.com"  
},  
{  
  "EmployeeID": 10008,  
  "Employees": "Laura Jack",  
  "Designation": "Developer",  
  "Location": "Austria",  
  "Status": "Inactive",  
  "Trustworthiness": "Perfect",  
  "Rating": 3,  
  "Software": 89,
```

```
"EmployeeImg": "usermale",
"CurrentSalary": 25108,
"Address": "Rua da Panificadora, 12",
"Mail": "laura82@mail.com"
},
{
  "EmployeeID": 10009,
  "Employees": "Anne Fuller",
  "Designation": "Program Directory",
  "Location": "Mexico",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 0,
  "Software": 19,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 32568,
  "Address": "Gran Vía, 1",
  "Mail": "anne97@jourrapide.com"
},
{
  "EmployeeID": 10010,
  "Employees": "Buchanan Andrew",
  "Designation": "Designer",
  "Location": "Austria",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 1,
  "Software": 62,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 12320,
  "Address": "P.O. Box 555",
  "Mail": "buchanan50@jourrapide.com"
},
{
  "EmployeeID": 10011,
  "Employees": "Andrew Janet",
  "Designation": "System Analyst",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 8,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 20890,
  "Address": "Starenweg 5",
  "Mail": "andrew63@mail.com"
},
{
  "EmployeeID": 10012,
  "Employees": "Margaret Tamer",
  "Designation": "System Analyst",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 4,
  "Software": 7,
  "EmployeeImg": "userfemale",
```

```
"CurrentSalary": 22337,  
"Address": "Magazinweg 7",  
"Mail": "margaret26@mail.com"  
},  
{  
  "EmployeeID": 10013,  
  "Employees": "Tamer Fuller",  
  "Designation": "CFO",  
  "Location": "Canada",  
  "Status": "Active",  
  "Trustworthiness": "Insufficient",  
  "Rating": 3,  
  "Software": 78,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 89181,  
  "Address": "Taucherstraße 10",  
  "Mail": "tamer40@arpy.com"  
},  
{  
  "EmployeeID": 10014,  
  "Employees": "Tamer Anne",  
  "Designation": "CFO",  
  "Location": "Sweden",  
  "Status": "Active",  
  "Trustworthiness": "Sufficient",  
  "Rating": 0,  
  "Software": 18,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 20998,  
  "Address": "Taucherstraße 10",  
  "Mail": "tamer68@arpy.com"  
},  
{  
  "EmployeeID": 10015,  
  "Employees": "Anton Davolio",  
  "Designation": "Project Lead",  
  "Location": "France",  
  "Status": "Active",  
  "Trustworthiness": "Sufficient",  
  "Rating": 4,  
  "Software": 8,  
  "EmployeeImg": "userfemale",  
  "CurrentSalary": 48232,  
  "Address": "Luisenstr. 48",  
  "Mail": "anton46@mail.com"  
},  
{  
  "EmployeeID": 10016,  
  "Employees": "Buchanan Buchanan",  
  "Designation": "System Analyst",  
  "Location": "Austria",  
  "Status": "Inactive",  
  "Trustworthiness": "Perfect",  
  "Rating": 0,  
  "Software": 19,  
  "EmployeeImg": "usermale",  
  "CurrentSalary": 43041,
```



```
"Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
"Mail": "buchanan68@mail.com"
},
{
  "EmployeeID": 10017,
  "Employees": "King Buchanan",
  "Designation": "Program Directory",
  "Location": "Sweden",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 44,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 25259,
  "Address": "Magazinweg 7",
  "Mail": "king80@jourrapide.com"
},
{
  "EmployeeID": 10018,
  "Employees": "Rose Michael",
  "Designation": "Project Lead",
  "Location": "Canada",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 4,
  "Software": 31,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 91156,
  "Address": "Fauntleroy Circus",
  "Mail": "rose75@mail.com"
},
{
  "EmployeeID": 10019,
  "Employees": "King Bergs",
  "Designation": "Developer",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 2,
  "Software": 29,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 28826,
  "Address": "2817 Milton Dr.",
  "Mail": "king57@jourrapide.com"
},
{
  "EmployeeID": 10020,
  "Employees": "Davolio Fuller",
  "Designation": "Designer",
  "Location": "Canada",
  "Status": "Inactive",
  "Trustworthiness": "Sufficient",
  "Rating": 3,
  "Software": 35,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 71035,
  "Address": "Gran Vía, 1",
```

```
"Mail": "davolio29@arpy.com"
},
{
  "EmployeeID": 10021,
  "Employees": "Rose Rose",
  "Designation": "CFO",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 38,
  "EmployeeImg": "usermale",
  "CurrentSalary": 68123,
  "Address": "Rua do Mercado, 12",
  "Mail": "rose54@arpy.com"
},
{
  "EmployeeID": 10022,
  "Employees": "Andrew Michael",
  "Designation": "Program Directory",
  "Location": "UK",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 2,
  "Software": 61,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 75470,
  "Address": "2, rue du Commerce",
  "Mail": "andrew88@jourrapide.com"
},
{
  "EmployeeID": 10023,
  "Employees": "Davolio Kathryn",
  "Designation": "Manager",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 25,
  "EmployeeImg": "usermale",
  "CurrentSalary": 25234,
  "Address": "Hauptstr. 31",
  "Mail": "davolio42@sample.com"
},
{
  "EmployeeID": 10024,
  "Employees": "Anne Fleet",
  "Designation": "System Analyst",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 0,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 8341,
  "Address": "59 rue de l'Abbaye",
  "Mail": "anne86@arpy.com"
```

```
},
{
  "EmployeeID": 10025,
  "Employees": "Margaret Andrew",
  "Designation": "System Analyst",
  "Location": "Germany",
  "Status": "Inactive",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 51,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 84975,
  "Address": "P.O. Box 555",
  "Mail": "margaret41@arpy.com"
},
{
  "EmployeeID": 10026,
  "Employees": "Kathryn Laura",
  "Designation": "Project Lead",
  "Location": "Austria",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 48,
  "EmployeeImg": "usermale",
  "CurrentSalary": 97282,
  "Address": "Avda. Azteca 123",
  "Mail": "kathryn82@rpy.com"
},
{
  "EmployeeID": 10027,
  "Employees": "Michael Michael",
  "Designation": "Developer",
  "Location": "UK",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 4,
  "Software": 16,
  "EmployeeImg": "usermale",
  "CurrentSalary": 4184,
  "Address": "Rua do Paço, 67",
  "Mail": "michael58@jourrapide.com"
},
{
  "EmployeeID": 10028,
  "Employees": "Leverling Vinet",
  "Designation": "Project Lead",
  "Location": "Germany",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 57,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 38370,
  "Address": "59 rue de l'Abbaye",
  "Mail": "leverling102@sample.com"
},
}
```

```

{
  "EmployeeID": 10029,
  "Employees": "Rose Jack",
  "Designation": "Developer",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 46,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 84790,
  "Address": "Rua do Mercado, 12",
  "Mail": "rose108@jourrapide.com"
},
{
  "EmployeeID": 10030,
  "Employees": "Vinet Van",
  "Designation": "Developer",
  "Location": "USA",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 40,
  "EmployeeImg": "usermale",
  "CurrentSalary": 71005,
  "Address": "Gran Vía, 1",
  "Mail": "vinet90@jourrapide.com"
}
]

```

### Note

You can refer to our [React Spreadsheet](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Spreadsheet example](#) to know how to present and manipulate data.

### See Also

- [Sorting](#)
- [Filtering](#)
- [Hyperlink](#)

### Accessibility in React Spreadsheet component

The Spreadsheet component followed the accessibility guidelines and standards, including [ADA](#), [Section 508](#), [WCAG 2.2](#) standards, and [WCAG roles](#) that are commonly used to evaluate accessibility.

The accessibility compliance for the Spreadsheet component is outlined below.

| Accessibility Criteria | Compatibility |

| -- | -- |

| [WCAG 2.2](#) Support |  |

| [Section 508](#) Support |  |

| Screen Reader Support |  |

| Right-To-Left Support |  |

| Color Contrast |  |

| Mobile Device Support |  |

| Keyboard Navigation Support |  |

| [Accessibility Checker](#) Validation |  |

| [Axe-core](#) Accessibility Validation |  |

<style>

.post .post-content img {

display: inline-block;

margin: 0.5em 0;

}

</style>

<div> - All features of the component meet the requirement.</div>

<div> - Some features of the component do not meet the requirement.</div>

<div> - The component does not meet the requirement.</div>

#### WAI-ARIA attributes

The Spreadsheet component followed the [WAI-ARIA](#) patterns to meet the accessibility. The following ARIA attributes are used in the Spreadsheet component:

| Attributes | Purpose |

|-----|-----|

| **grid** (role) | This role is added to the spreadsheet content table and describes the collection of rows and columns. |

| **gridcell** (role) | This role is added to the cell element and describes the rows **<td>** element. |

| **rowheader** (role) | This role is added to the row header and describes the header of the rows. |

- | **colheader** (role) | This role is added to the column header and describes the header of the columns. |
- | **aria-rowindex** (attribute) | This attribute describes the table's row index in the spreadsheet. |
- | **aria-colindex** (attribute) | This attribute describes the table's column index in the spreadsheet. |
- | **aria-selected** (attribute) | This attribute describes an item's (cell, menu, checkbox, etc.) current selected state in the spreadsheet. |
- | **aria-rowcount** (attribute) | This attribute describes the total number of rows in the table. |
- | **aria-colcount** (attribute) | This attribute describes the total number of columns in the table. |
- | **aria-busy** (attribute) | This attribute describes a currently updated or modified element. |
- | **aria-label** (attribute) | This attribute describes the accessible name for the interactive elements. |
- | **textbox** (role) | This role is assigned to the textbox that accepts text input. |
- | **menu** (role) | This role has been added to the menu and describes the menu items. |
- | **aria-expanded** (attribute) | This attribute describes the control (for example, dropdown) is expanded or collapsed. |
- | **aria-multiline** (attribute) | This attribute defines what the Alt + Enter key does in the spreadsheet editor. |

### Keyboard interaction

The Spreadsheet component followed the [keyboard interaction](#) guideline, making it easy for people who use assistive technologies (AT) and those who completely rely on keyboard navigation. The following keyboard shortcuts are supported by the Spreadsheet component.

- | Press | To do this |
- |-----|-----|
- | Up arrow | Navigate from the active cell to the previous cell in the same column. |
- | Down arrow | Navigate from the active cell to the next cell in the same column. |
- | Left arrow | Navigate from the active cell to the previous cell in the same row. |
- | Right arrow | Navigate from the active cell to the next cell in the same row. |
- | Tab | Navigate the active cell to the next cell in the same row. |
- | Shift + Tab | Navigate the active cell to the previous cell in the same row. |
- | Home | Moves the selection to starting column in worksheet. |
- | Ctrl + Home | Move the selection to the first visible cell on a worksheet. |
- | Shift + Home | Extend the cell selection to the first column of a worksheet. |
- | Ctrl + Shift + Home | Extend the selection of cells to the beginning of the worksheet. |
- | Ctrl + End | Move to the last cell on a worksheet, right most last column and last row cell. |
- | Page Up | Move page up. |
- | Page Down | Move page down. |

- | Shift + Page Up | Perform page up by selecting all cells between. |
- | Shift + Page Down | Perform page down by selecting all cells between. |
- | Ctrl + Up | Navigate to the non-blank cell before the active cell in the same column. |
- | Ctrl + Down | Navigate to the last non-blank cell in the same column as the active cell. |
- | Ctrl + Left | Navigate to the non-blank cell before the active cell in the same row. |
- | Ctrl + Right | Navigate to the last non-blank cell in the same row as the active cell. |
- | Shift + Up | Extend the selection of cells to the previous row. |
- | Shift + Down | Extend the selection of cells to the next row. |
- | Shift + Left | Extend the selection of cells to the previous column. |
- | Shift + Right | Extend the selection of cells to the next column. |
- | Ctrl + Shift + Up | Extend the cell selection to the previous non-blank cell in the same column as the active cell. |
- | Ctrl + Shift + Down | Extend the cell selection to the last non-blank cell in the same column as the active cell. |
- | Ctrl + Shift + Left | Extend the cell selection to the previous non-blank cell in the same row as the active cell. |
- | Ctrl + Shift + Right | Extend the cell selection to the last non-blank cell in the same row as the active cell. |
- | Enter | Navigate the active cell to the next cell in the same column. |
- | Shift + Enter | Navigate to the previous cell in the same column from the active cell. |
- | Alt + Enter | While editing, add a new line. |
- | Enter | Complete the cell editing and select the cell below in the same column. |
- | Shift + Enter | Complete the cell editing and select the cell above in the same column. |
- | Tab | Complete the cell editing and select the next cell in the same row. |
- | Shift + Tab | Complete the cell editing and select the previous cell in the same row. |
- | Alt | Focus on the active ribbon tab. |
- | Left | Move the focus to the previous items in the ribbon content. |
- | Right | Move the focus to the next items in the ribbon content. |
- | Alt + Down | Open the ribbon dropdown menu. |
- | Esc / Alt + Up | Close the ribbon dropdown menu. |

### Ensuring accessibility

The Spreadsheet component's accessibility levels are ensured through an [accessibility-checker](#) and [axe-core](#) software tools during automated testing.

The accessibility compliance of the Spreadsheet component is shown in the following sample. Open the [sample](#) in a new window to evaluate the accessibility of the Spreadsheet component with accessibility tools.

See also

- [Accessibility in Syncfusion React components](#)

### Keyboard shortcuts in React Spreadsheet component

The keyboard shortcuts supported in the spreadsheet are,

Shortcut	Description
----- -----	
Ctrl + O	Displays dialog to open a file.
Ctrl + S / Alt + F2	Saves the workbook.
F2	Enables edit mode.
ESC	Cancel edit mode and discard the changes.
Backspace and SPACE	Clears content of the active cell and enables edit mode.
Ctrl + C	Copies the selected cells.
Ctrl + X	Cuts the selected cells.
Ctrl + V	Paste the clipboard(cut or copied) content in the new selected range.
Ctrl + B	Applies or removes <b>bold</b> formatting.
Ctrl + I	Applies or removes <i>italic</i> formatting.
Ctrl + U	Applies or removes <u>underline</u> .
Ctrl + 5	Applies or removes <del>strikethrough</del> .
Ctrl + Z	Reverses (Undo) the last action.
Ctrl + Y	Repeats (Redo) the last reversed action.
Ctrl + K	It opens the <b>Insert Hyperlink</b> dialog for adding new hyperlink to a cell. If the selected cell already contains hyperlink, it opens the <b>Edit Hyperlink</b> dialog.
Ctrl + F / Shift + F5	Opens <b>Find</b> dialog.
Ctrl + H	Opens <b>Find and Replace</b> dialog.
Ctrl + G	Opens <b>GoTo</b> dialog, which helps to navigate to cell.
Ctrl + Shift + L	Applies filter to the first row of the selected range or used range.
Alt + F	Opens the <b>File</b> menu.
Alt + H	Go to <b>Home</b> tab.
Alt + N	Go to <b>Insert</b> tab.
Alt + M	Go to <b>Formulas</b> tab.
Alt + A	Go to <b>Data</b> tab.
Alt + W	Go to <b>View</b> tab.



- | Spacebar or Enter | Activate a selected button. |
- | Ctrl + F8 | Expand or collapse the ribbon content. |
- | Ctrl + Shift + U | Expand or collapse the formula bar. |
- | Ctrl + 9 | Hide the selected row. |
- | Ctrl + 0 | Hide the selected column. |
- | Ctrl + & | Apply an outline border to the selected cells. |
- | Ctrl + Shift + & | Apply an outline border to the cells that you've chosen. |
- | Ctrl + Shift + ~ | Apply the **General** number format. |
- | Ctrl + Shift + \$ | Apply the **Currency** format with two decimal places (negative numbers in parentheses). |
- | Shift + F10 | Open the context menu. |
- | Ctrl + % | Apply the **Percentage** format with no decimal places. |
- | Ctrl + ^ | Apply the **Scientific** number format with two decimal places. |
- | Ctrl + Shift + # | Apply the **Date** format with the day, month, and year. |
- | Ctrl + Shift + @ | Apply the **Time** format with the hour and minute, and AM or PM. |
- | Ctrl + Shift + ! | Apply the **Number** format with two decimal places, thousands separator, and minus sign (-) for negative values. |
- | Ctrl + Spacebar | Select an entire column in a worksheet. |
- | Shift + Spacebar | Select an entire row in a worksheet. |
- | Shift + F3 | Opens **Insert Function** dialog. |
- | Ctrl + Alt + N | Opens new workbook. |
- | Shift + Alt + K | Open the **List All Sheets** dropdown option. |
- | Alt + Down | Open the list data validation menu and filter menu. |
- | Alt + Up | Close the list data validation menu and filter menu. |
- | Delete | Remove the contents of the cell. |
- | Shift + F11 | Add a new sheet. |
- | Ctrl + Shift + 9 | Unhide the selected rows. |
- | Ctrl + Shift + 0 | Unhide the selected columns. |
- | Ctrl + D | Fill the cell down. |
- | Ctrl + R | Fill the cell right. |

#### Note

You can refer to our [React Spreadsheet](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Spreadsheet example](#) to know how to present and manipulate data.

See Also

- [Formatting](#)

## Ribbon in React Spreadsheet component

It helps to organize a spreadsheet's features into a series of tabs. By clicking the expand or collapse icon, you can expand or collapse the ribbon toolbar dynamically.

### Ribbon Customization

You can customize the ribbon using the following methods,

Method	Action
<a href="#">hideRibbonTabs</a>	Used to show or hide the existing ribbon tabs.
<a href="#">enableRibbonTabs</a>	Used to enable or disable the existing ribbon tabs.
<a href="#">addRibbonTabs</a>	Used to add custom ribbon tabs.
<a href="#">hideToolbarItems</a>	Used to show or hide the existing ribbon toolbar items.
<a href="#">enableToolbarItems</a>	Used to enable or disable the specified toolbar items.
<a href="#">addToolbarItems</a>	Used to add the custom items in ribbon toolbar.
<a href="#">hideFileMenuitems</a>	Used to show or hide the ribbon file menu items.
<a href="#">enableFileMenuitems</a>	Used to enable or disable file menu items.
<a href="#">addFileMenuitems</a>	Used to add custom file menu items.

The following code example shows the usage of ribbon customization.

### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { DropDownButton } from '@syncfusion/ej2-splitbuttons';
import { select, createElement } from '@syncfusion/ej2-base';
import { data } from './datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A1:F1');
      // Hiding the `Insert` from ribbon.
      spreadsheet.hideRibbonTabs(['Insert']);
      // Set disabled state to `View` ribbon tab.
      spreadsheet.enableRibbonTabs(['View'], false);
      // Adding the `Help` ribbon tab at the last index.
    }
  });
}
```

```

    // Specify the ribbon tab header text in last optional
    argument(`insertBefore`) for inserting new tab before any existing tab.
    spreadsheet.addRibbonTabs([
      header: { text: 'Help' }, content: [{
        text: 'Feedback', tooltipText: 'Feedback',
        click: () => { /* Any click action for this toolbar item
will come here. */ }
      }]
    ]]);
    // Hiding the unwanted toolbar items from `Home` by specifying
    its index.
    spreadsheet.hideToolbarItems('Home', [0, 1, 2, 4, 14, 15, 21,
24]);
    // Set disable state to `Underline`, `Wrap text` toolbar items
    from `Home` by specifying the item id.
    spreadsheet.enableToolbarItems('Home',
[`${spreadsheet.element.id}_underline`, `${spreadsheet.element.id}_wrap`],
false);
    // Set disable state to `Protect Sheet` toolbar item from `Data`
    by mentioning its index.
    spreadsheet.enableToolbarItems('Data', [0], false);
    // Adding the new `Custom Formulas` toolbar item under
    `Formulas` tab for adding custom formulas.
    spreadsheet.addToolbarItems('Formulas', [{ type: 'Separator' },
{
  text: 'Custom Formulas', tooltipText: 'Custom Formulas',
  // You can set click handler for each new custom toolbar
  item
  click: () => {
    // You can add custom formulas here.
  }
}]);
    // Adding new custom item `Import` after the existing `Open`
    item. By default, new item will add after the specified item.
    spreadsheet.addFileMenuItems([
      { text: 'Import', iconCss: 'e-open e-icons' }
    ], 'Open');
    // Adding new custom item `Export As` after the existing `Save
    As` item.
    // Set `insertAfter` optional argument as `false` for adding new
    item before the specified item.
    spreadsheet.addFileMenuItems([
      {
        text: 'Export As', iconCss: 'e-save e-icons', items: [
          { text: 'XLSX', iconCss: 'e-xlsx e-icons' },
          { text: 'XLS', iconCss: 'e-xls e-icons' },
          { text: 'CSV', iconCss: 'e-csv e-icons' }
        ]
      },
      { text: 'Save As', false }
    ], 'Save As', false);
    // Adding the new `custom dropdown button` in the ribbon toolbar
    item under the `Data` tab for adding a custom dropdown button using the
    addToolbarItems method in the spreadsheet ribbon.
    spreadsheet.addToolbarItems('Data', [
      { type: 'Separator' },
      {
        id: 'custombtn', tooltipText: 'Custom Btn', template:
        appendDropDownBtn('custombtn')
      }
    ], 7);
    const fileMenuBeforeOpen = () => {
      // Because the file menu items are created dynamically, you need to
      perform the hide or show and enable/disable operations

```

```

        // under filemenu before open event.
        // Hiding the `Save As` and `Open` item.
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            spreadsheet.hideFileMenuItems(['Save As', 'Open']);
            // Set disable state to `New` item. You can perform any file
            menu items customization by specifying the item id,
            // if it has more than one same item text. Set the last argument
            `isUniqueId` as true for using the item id.

        spreadsheet.enableFileMenuItems([`${spreadsheet.element.id}_New`], false,
        true);
        }
    };
    const fileMenuItemSelect = (args) => {
        let spreadsheet = spreadsheetRef.current;
        // Custom file menu items select handler
        switch (args.item.text) {
            case 'Import': select(`${spreadsheet.element.id}_fileUpload`,
            spreadsheet.element).click();
                break;
            case 'XLSX': if (spreadsheet) { spreadsheet.save({ saveType:
            'Xlsx' }) };
                break;
            case 'XLS': if (spreadsheet) { spreadsheet.save({ saveType:
            'Xls' }) };
                break;
            case 'CSV': if (spreadsheet) { spreadsheet.save({ saveType:
            'Csv' }) };
                break;
            default:
                break;
        }
    };
    const appendDropDownBtn = (id) => {
        let ddlItems = [
            {
                text: 'Download Excel',
            },
            {
                text: 'Download CSV',
            },
        ];
        let btnObj = new DropDownButton({
            items: ddlItems,
            content: 'Download',
            iconCss: 'e-icons e-download',
            select: (args) => {
                alert(args.item.text + ' clicked');
            },
        });
        btnObj.appendTo(createElement('button', { id: id }));
        return btnObj.element;
    }
    return (
        <div>

```

```

        <SpreadsheetComponent ref={spreadsheetRef}
fileMenuBeforeOpen={fileMenuBeforeOpen}
fileMenuItemSelect={fileMenuItemSelect}
        showFormulaBar={false} showSheetTabs={false}
allowInsert={false} allowDelete={false} allowMerge={false}

openUrl="https://services.syncfusion.com/react/production/api/spreadsheet/open"

saveUrl="https://services.syncfusion.com/react/production/api/spreadsheet/save">
        <SheetsDirective>
        <SheetDirective>
        <RangesDirective>
        <RangeDirective
dataSource={data}></RangeDirective>
        </RangesDirective>
        <ColumnsDirective>
        <ColumnDirective width={180}></ColumnDirective>
        <ColumnDirective width={130}></ColumnDirective>
        <ColumnDirective width={130}></ColumnDirective>
        <ColumnDirective width={180}></ColumnDirective>
        <ColumnDirective width={130}></ColumnDirective>
        <ColumnDirective width={120}></ColumnDirective>
        </ColumnsDirective>
        </SheetDirective>
        </SheetsDirective>
        </SpreadsheetComponent>
    </div>
    );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RangeDirective, ColumnsDirective, ColumnDirective,
MenuSelectEventArgs } from '@syncfusion/ej2-react-spreadsheet';
import { DropDownButton, ItemModel, MenuEventArgs } from '@syncfusion/ej2-splitbuttons';
import { select, createElement } from '@syncfusion/ej2-base';
import { data } from './datasource';
function App() {
    const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
    React.useEffect(() => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A1:F1');
        }
    });
    // Hiding the `Insert` from ribbon.

```

```

        spreadsheet.hideRibbonTabs(['Insert']);
        // Set disabled state to `View` ribbon tab.
        spreadsheet.enableRibbonTabs(['View'], false);
        // Adding the `Help` ribbon tab at the last index.
        // Specify the ribbon tab header text in last optional
        argument(`insertBefore`) for inserting new tab before any existing tab.
        spreadsheet.addRibbonTabs([
            {
                header: { text: 'Help' }, content: [
                    {
                        text: 'Feedback', tooltipText: 'Feedback',
                        click: (): void => { /* Any click action for this
toolbar item will come here. */ }
                    }
                ]
            }
        ]);
        // Hiding the unwanted toolbar items from `Home` by specifying
        its index.
        spreadsheet.hideToolbarItems('Home', [0, 1, 2, 4, 14, 15, 21,
24]);
        // Set disable state to `Underline`, `Wrap text` toolbar items
        from `Home` by specifying the item id.
        spreadsheet.enableToolbarItems('Home',
[`${spreadsheet.element.id}_underline`, `${spreadsheet.element.id}_wrap`],
false);
        // Set disable state to `Protect Sheet` toolbar item from `Data`
        by mentioning its index.
        spreadsheet.enableToolbarItems('Data', [0], false);
        // Adding the new `Custom Formulas` toolbar item under
        `Formulas` tab for adding custom formulas.
        spreadsheet.addToolbarItems('Formulas', [{ type: 'Separator' },
{
            text: 'Custom Formulas', tooltipText: 'Custom Formulas',
            // You can set click handler for each new custom toolbar
            item
            click: (): void => {
                // You can add custom formulas here.
            }
        }]);
        // Adding new custom item `Import` after the existing `Open`
        item. By default, new item will add after the specified item.
        spreadsheet.addFileMenuItems([
            { text: 'Import', iconCss: 'e-open e-icons' }
        ], 'Open');
        // Adding new custom item `Export As` after the existing `Save
        As` item.
        // Set `insertAfter` optional argument as `false` for adding new
        item before the specified item.
        spreadsheet.addFileMenuItems([
            {
                text: 'Export As', iconCss: 'e-save e-icons', items: [
                    {
                        text: 'XLSX', iconCss: 'e-xlsx e-icons' },
                    {
                        text: 'XLS', iconCss: 'e-xls e-icons' },
                    {
                        text: 'CSV',
                        iconCss: 'e-csv e-icons' }
                ]
            },
            { text: 'Save As', false }
        ], 'Save As', false);
        // Adding the new `custom dropdown button` in the ribbon toolbar
        item under the `Data` tab for adding a custom dropdown button using the
        addToolbarItems method in the spreadsheet ribbon.
        spreadsheet.addToolbarItems('Data', [
            { type: 'Separator' },
            {
                id: 'custombtn', tooltipText: 'Custom Btn', template:
                appendDropdownBtn('custombtn')
            }
        ], 7);
    }

```

```

    }, []);
    const fileMenuBeforeOpen = (): void => {
        // Because the file menu items are created dynamically, you need to
        // perform the hide or show and enable/disable operations
        // under filemenu before open event.
        // Hiding the `Save As` and `Open` item.
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            spreadsheet.hideFileMenuItems(['Save As', 'Open']);
            // Set disable state to `New` item. You can perform any file
            // menu items customization by specifying the item id,
            // if it has more than one same item text. Set the last argument
            // `isUniqueId` as true for using the item id.
            spreadsheet.enableFileMenuItems(['${spreadsheet.element.id}_New`, false,
            true);
        }
    };
    const fileMenuItemSelect = (args: MenuSelectEventArgs): void => {
        let spreadsheet = spreadsheetRef.current;
        // Custom file menu items select handler
        switch (args.item.text) {
            case 'Import': select(`${spreadsheet.element.id}_fileUpload`,
            spreadsheet.element).click();
                break;
            case 'XLSX': if (spreadsheet) { spreadsheet.save({ saveType:
            'Xlsx' }) };
                break;
            case 'XLS': if (spreadsheet) { spreadsheet.save({ saveType:
            'Xls' }) };
                break;
            case 'CSV': if (spreadsheet) { spreadsheet.save({ saveType:
            'Csv' }) };
                break;
            default:
                break;
        }
    };
    const appendDropDownBtn = (id: string): HTMLElement => {
        let ddlItems: ItemModel[] = [
            {
                text: 'Download Excel',
            },
            {
                text: 'Download CSV',
            },
        ];
        let btnObj: DropDownButton = new DropDownButton({
            items: ddlItems,
            content: 'Download',
            iconCss: 'e-icons e-download',
            select: (args: MenuEventArgs): void => {
                alert(args.item.text + ' clicked');
            },
        });
        btnObj.appendTo(createElement('button', { id: id }));
        return btnObj.element;
    };

```

```

    }
    return (
      <div>
        <SpreadsheetComponent ref={spreadsheetRef}
        fileMenuBeforeOpen={fileMenuBeforeOpen}
        fileMenuItemSelect={fileMenuItemSelect}
          showFormulaBar={false} showSheetTabs={false}
        allowInsert={false} allowDelete={false} allowMerge={false}

        openUrl="https://services.syncfusion.com/react/production/api/spreadsheet/open"

        saveUrl="https://services.syncfusion.com/react/production/api/spreadsheet/save">
          <SheetsDirective>
            <SheetDirective>
              <RangesDirective>
                <RangeDirective
dataSource={data}></RangeDirective>
              </RangesDirective>
              <ColumnsDirective>
                <ColumnDirective width={180}></ColumnDirective>
                <ColumnDirective width={130}></ColumnDirective>
                <ColumnDirective width={130}></ColumnDirective>
                <ColumnDirective width={180}></ColumnDirective>
                <ColumnDirective width={130}></ColumnDirective>
                <ColumnDirective width={120}></ColumnDirective>
              </ColumnsDirective>
            </SheetDirective>
          </SheetsDirective>
        </SpreadsheetComponent>
      </div>
    );
  };
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

## DATASOURCE.JSX

```

/**
 * Ribbon customization data source
 */
export let data = [
  {
    'Customer Name': 'Romona Heaslip',
    'Model': 'Taurus',
    'Color': 'Aquamarine',
    'Payment Mode': 'Debit Card',
    'Delivery Date': '07/11/2015',
    'Amount': '8529.22'
  },
  {
    'Customer Name': 'Clare Batterton',
    'Model': 'Sparrow',

```



```

    'Color': 'Pink',
    'Payment Mode': 'Cash On Delivery',
    'Delivery Date': '7/13/2016',
    'Amount': '17866.19'
  },
  {
    'Customer Name': 'Eamon Traise',
    'Model': 'Grand Cherokee',
    'Color': 'Blue',
    'Payment Mode': 'Net Banking',
    'Delivery Date': '09/04/2015',
    'Amount': '13853.09'
  },
  {
    'Customer Name': 'Julius Gorner',
    'Model': 'GTO',
    'Color': 'Aquamarine',
    'Payment Mode': 'Credit Card',
    'Delivery Date': '12/15/2017',
    'Amount': '2338.74'
  },
  {
    'Customer Name': 'Jenna Schoolfield',
    'Model': 'LX',
    'Color': 'Yellow',
    'Payment Mode': 'Credit Card',
    'Delivery Date': '10/08/2014',
    'Amount': '9578.45'
  },
  {
    'Customer Name': 'Marylynne Haring',
    'Model': 'Catera',
    'Color': 'Green',
    'Payment Mode': 'Cash On Delivery',
    'Delivery Date': '7/01/2017',
    'Amount': '19141.62'
  },
  {
    'Customer Name': 'Vilhelmina Leipelt',
    'Model': '7 Series',
    'Color': 'Goldenrod',
    'Payment Mode': 'Credit Card',
    'Delivery Date': '12/20/2015',
    'Amount': '6543.30'
  },
  {
    'Customer Name': 'Barby Heisler',
    'Model': 'Corvette',
    'Color': 'Red',
    'Payment Mode': 'Credit Card',
    'Delivery Date': '11/24/2014',
    'Amount': '13035.06'
  },
  {
    'Customer Name': 'Karyn Boik',
    'Model': 'Regal',
    'Color': 'Indigo',

```

```

    'Payment Mode': 'Debit Card',
    'Delivery Date': '05/12/2014',
    'Amount': '18488.80'
  },
  {
    'Customer Name': 'Jeanette Pamplin',
    'Model': 'S4',
    'Color': 'Fuscia',
    'Payment Mode': 'Net Banking',
    'Delivery Date': '12/30/2014',
    'Amount': '12317.04'
  },
  {
    'Customer Name': 'Cristi Espinos',
    'Model': 'TL',
    'Color': 'Aquamarine',
    'Payment Mode': 'Credit Card',
    'Delivery Date': '12/18/2013',
    'Amount': '6230.13'
  }
];

```

#### DATASOURCE.TSX

```

/**
 * Ribbon customization data source
 */
export let data: Object[] = [
  {
    'Customer Name': 'Romona Heaslip',
    'Model': 'Taurus',
    'Color': 'Aquamarine',
    'Payment Mode': 'Debit Card',
    'Delivery Date': '07/11/2015',
    'Amount': '8529.22'
  },
  {
    'Customer Name': 'Clare Batterton',
    'Model': 'Sparrow',
    'Color': 'Pink',
    'Payment Mode': 'Cash On Delivery',
    'Delivery Date': '7/13/2016',
    'Amount': '17866.19'
  },
  {
    'Customer Name': 'Eamon Traise',
    'Model': 'Grand Cherokee',
    'Color': 'Blue',
    'Payment Mode': 'Net Banking',
    'Delivery Date': '09/04/2015',
    'Amount': '13853.09'
  },
  {
    'Customer Name': 'Julius Gerner',
    'Model': 'GTO',
    'Color': 'Aquamarine',

```

```

    'Payment Mode': 'Credit Card',
    'Delivery Date': '12/15/2017',
    'Amount': '2338.74'
  },
  {
    'Customer Name': 'Jenna Schoolfield',
    'Model': 'LX',
    'Color': 'Yellow',
    'Payment Mode': 'Credit Card',
    'Delivery Date': '10/08/2014',
    'Amount': '9578.45'
  },
  {
    'Customer Name': 'Marylynne Harring',
    'Model': 'Catera',
    'Color': 'Green',
    'Payment Mode': 'Cash On Delivery',
    'Delivery Date': '7/01/2017',
    'Amount': '19141.62'
  },
  {
    'Customer Name': 'Vilhelmina Leipelt',
    'Model': '7 Series',
    'Color': 'Goldenrod',
    'Payment Mode': 'Credit Card',
    'Delivery Date': '12/20/2015',
    'Amount': '6543.30'
  },
  {
    'Customer Name': 'Barby Heisler',
    'Model': 'Corvette',
    'Color': 'Red',
    'Payment Mode': 'Credit Card',
    'Delivery Date': '11/24/2014',
    'Amount': '13035.06'
  },
  {
    'Customer Name': 'Karyn Boik',
    'Model': 'Regal',
    'Color': 'Indigo',
    'Payment Mode': 'Debit Card',
    'Delivery Date': '05/12/2014',
    'Amount': '18488.80'
  },
  {
    'Customer Name': 'Jeanette Pamplin',
    'Model': 'S4',
    'Color': 'Fuscia',
    'Payment Mode': 'Net Banking',
    'Delivery Date': '12/30/2014',
    'Amount': '12317.04'
  },
  {
    'Customer Name': 'Cristi Espinos',
    'Model': 'TL',
    'Color': 'Aquamarine',
    'Payment Mode': 'Credit Card',

```

```
'Delivery Date': '12/18/2013',  
'Amount': '6230.13'  
}  
];
```

### Note

You can refer to our [React Spreadsheet](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Spreadsheet example](#) to know how to present and manipulate data.

### See Also

- [Worksheet](#)
- [Rows and columns](#)

## Global local in React Spreadsheet component

### Localization

The [Localization](#) library allows you to localize the default text content of the Spreadsheet. The Spreadsheet has static text on some features (cell formatting, Merge, Data validation, etc.) that can be changed to other cultures (Arabic, Deutsch, French, etc.) by defining the

[locale](#) value and translation object.

The following list of properties and their values are used in the Spreadsheet.

Locale keywords | Text

Cut | Cut

Copy | Copy

Paste | Paste

PasteSpecial | Paste Special

All | All

Values | Values

Formats | Formats

Font | Font

FontSize | Font Size

Bold | Bold

Italic | Italic

Underline | Underline

Strikethrough | Strikethrough

TextColor | Text Color

FillColor | Fill Color

HorizontalAlignment | Horizontal Alignment

AlignLeft | Align Left

AlignCenter | Center

AlignRight | Align Right

VerticalAlignment | Vertical Alignment

AlignTop | Align Top

AlignMiddle | Align Middle

AlignBottom | Align Bottom

InsertFunction | Insert Function

Insert | Insert

Delete | Delete

Rename | Rename

Hide | Hide

Unhide | Unhide

NameBox | Name Box

ShowHeaders | Show Headers

HideHeaders | Hide Headers

ShowGridLines | Show Gridlines

HideGridLines | Hide Gridlines

AddSheet | Add Sheet

ListAllSheets | List All Sheets

FullScreen | Full Screen

CollapseToolbar | Collapse Toolbar

ExpandToolbar | Expand Toolbar

CollapseFormulaBar | Collapse Formula Bar

ExpandFormulaBar | Expand Formula Bar

File | File

Home | Home

Formulas | Formulas

View | View

New | New

Open | Open

SaveAs | Save As

ExcelXlsx | Microsoft Excel

ExcelXls | Microsoft Excel 97-2003

CSV | Comma-separated values

FormulaBar | Formula Bar

Ok | Ok

Close | Close

Cancel | Cancel

Apply | Apply

MoreColors | More Colors

StandardColors | Standard Colors

General | General

Number | Number

Currency | Currency

Accounting | Accounting

ShortDate | Short Date

LongDate | Long Date

Time | Time

Percentage | Percentage

Fraction | Fraction

Scientific | Scientific

Text | Text

NumberFormat | Number Format

MobileFormulaBarPlaceholder | Enter value or Formula

PasteAlert | You can't paste it here because the copy area and paste area aren't in the same size. Please try pasting in a different range.

DestroyAlert | Are you sure you want to destroy the current workbook without saving and create a new workbook?

SheetRenameInvalidAlert | Sheet name contains invalid character.

SheetRenameEmptyAlert | Sheet name cannot be empty.

SheetRenameAlreadyExistsAlert | Sheet name already exists. Please enter another name.

DeleteSheetAlert | Are you sure you want to delete this sheet?

DeleteSingleLastSheetAlert | A Workbook must contain at least one visible worksheet.

PickACategory | Pick a category

Description | Description

UnsupportedFile | Unsupported File

InvalidUrl | Invalid Url

SUM | Adds a series of numbers and/or cells.

SUMIF | Adds the cells based on the specified condition.

SUMIFS | Adds the cells based on the specified conditions.

ABS | Returns the value of a number without its sign.

RAND | Returns a random number between 0 and 1.

RANDBETWEEN | Returns a random integer based on the specified values.

FLOOR | Rounds a number down to the nearest multiple of a given factor.

CEILING | Rounds a number up to the nearest multiple of a given factor.

PRODUCT | Multiplies a series of numbers and/or cells.

AVERAGE | Calculates average for the series of numbers and/or cells excluding text.

AVERAGEIF | Calculates average for the cells based on the specified criterion.

AVERAGEIFS | Calculates average for the cells based on the specified conditions.

AVERAGEA | Calculates the average for the cells evaluating TRUE as 1 text and FALSE as 0.

COUNT | Counts the cells that contain numeric values in a range.

COUNTIF | Counts the cells based on the specified condition.

COUNTIFS | Counts the cells based on specified conditions.

COUNTA | Counts the cells that contain values in a range.

MIN | Returns the smallest number of the given arguments.

MAX | Returns the largest number of the given arguments.

DATE | Returns the date based on the given year, month, and day.

DAY | Returns the day from the given date.

DAYS | Returns the number of days between two dates.

IF | Returns value based on the given expression.

IFS | Returns value based on the given multiple expressions.

AND | Returns TRUE if all the arguments are TRUE otherwise returns FALSE.

OR | Returns TRUE if any of the arguments are TRUE otherwise returns FALSE.

IFERROR | Returns value if no error found else it will return specified value.

CHOOSE | Returns a value from the list of values based on the index number.

INDEX | Returns the value of the cell in a given range based on row and column number.

FIND | Returns the position of a string within another string which is case sensitive.

CONCATENATE | Combines two or more strings together.

CONCAT | Concatenates a list or a range of text strings.

SUBTOTAL | Returns subtotal for a range using the given function number.

RADIANS | Converts degrees into radians.

MATCH | Returns the relative position of a specified value in the given range.

DefineNameExists | This name already exists try a different name.

CircularReference | When a formula refers to one or more circular references this may result in an incorrect calculation.

ShowRowsWhere | Show rows where |

CustomFilterDatePlaceholder | Choose a date

CustomFilterPlaceholder | Enter the value

CustomFilter | Custom Filter

Between | Between

MatchCase | Match Case

DateTimeFilter | DateTime Filters

Undo | Undo

Redo | Redo

DateFilter | Date Filters

TextFilter | Text Filters

NumberFilter | Number Filters

ClearFilter | Clear Filter

NoResult | No Matches Found

FilterFalse | False

FilterTrue | True

Blanks | Blanks

SelectAll | Select All

GreaterThanOrEqual | Greater Than Or Equal

GreaterThan | Greater Than

LessThanOrEqual | Less Than Or Equal

LessThan | Less Than

NotEqual | Not Equal

Equal | Equal

Contains | Contains

EndsWith | Ends With

StartsWith | Starts With



ClearButton | Clear

FilterButton | Filter

CancelButton | Cancel

OKButton | OK

Search | Search

Link | Link

Hyperlink | Hyperlink

EditHyperlink | Edit Hyperlink

OpenHyperlink | Open Hyperlink

RemoveHyperlink | Remove Hyperlink

InsertLink | Insert Link

EditLink | Edit Link

WebPage | WEB PAGE

ThisDocument | THIS DOCUMENT

DisplayText | Display Text

Url | URL

CellReference | Cell Reference

Sheet | Sheet

DefinedNames | Defined Names

EnterTheTextToDisplay | Enter the text to display

EnterTheUrl | Enter the URL

ProtectSheet | Protect Sheet

UnprotectSheet | Unprotect Sheet

SelectCells | Select cells

FormatCells | Format cells

FormatRows | Format rows

Format Columns | Format columns

InsertLinks | Insert links

ProtectContent | Protect the contents of locked cells

ProtectAllowUser | Allow all users of this worksheet to |

EditAlert | The cell you're trying to change is protected. To make a change, unprotect the sheet.

FindReplaceTooltip | Find & Replace

InsertingEmptyValue | Reference value is not valid.

ByRow | By Row  
ByColumn | By Column  
MatchExactCellElements | Match Exact Cell Contents  
EnterCellAddress | Enter Cell Address  
FindAndReplace | Find and Replace  
ReplaceAllEnd | matches replaced with.  
FindNextBtn | Find Next  
FindPreviousBtn | Find Previous  
ReplaceBtn | Replace  
ReplaceAllBtn | Replace All  
GotoHeader | Go To  
GotoSpecialHeader | Go To Special  
SearchWithin | Search within  
SearchBy | Search by  
Reference | Reference  
Workbook | Workbook  
NoElements | We couldn't find what you were looking for.  
FindWhat | Find what  
ReplaceWith | Replace with  
EnterValue | Enter Value  
DefineNameInvalid | The name that you entered is not valid.  
FindValue | Find Value  
ReplaceValue | Replace Value  
DataValidation | Data Validation  
CLEARALL | CLEAR ALL  
APPLY | APPLY  
CellRange | Cell Range  
Allow | Allow  
Data | Data  
Minimum | Minimum  
Maximum | Maximum  
IgnoreBlank | Ignore blank  
WholeNumber | Whole Number

Decimal | Decimal

Date | Date

TextLength | Text Length

List | List

NotBetween | Not between

EqualTo | Equal to

NotEqualTo | Not equal to

Greaterthan | Greater than

Lessthan | Less than

GreaterThanOrEqaulTo | Greater than or eqaul to

LessThanOrEqualTo | Less than or equal to

InCellDropDown | In-cell-dropdown

Sources | Sources

Value | Value

Retry | Retry

DialogError | The list source must be a reference to a single row or column.

ValidationError | This value doesn't match the data validation restrictions defined for the cell.

HideRow | Hide Row

HideRows | Hide Rows

UnHideRows | UnHide Rows

HideColumn | Hide Column

HideColumns | Hide Columns

UnHideColumns | UnHide Columns

InsertRow | Insert Row

InsertRows | Insert Rows

InsertColumn | Insert Column

InsertColumns | Insert Columns

DeleteRow | Delete Row

DeleteRows | Delete Rows

DeleteColumn | Delete Column

DeleteColumns | Delete Columns

Borders | Borders

TopBorders | Top Borders

LeftBorders | Left Borders

RightBorders | Right Borders

BottomBorders | Bottom Borders

AllBorders | All Borders

HorizontalBorders | Horizontal Borders

VerticalBorders | Vertical Borders

OutsideBorders | Outside Borders

InsideBorders | Inside Borders

NoBorders | No Borders

BorderColor | Border Color

BorderStyle | Border Style

INTERCEPT | Calculates the point of the Y-intercept line via linear regression.

SLOPE | Returns the slope of the line from linear regression of the data points.

FreezePanes | Freeze Panes

FreezeRows | Freeze Rows

FreezeColumns | Freeze Columns

UnfreezePanes | Unfreeze Panes

UnfreezeRows | Unfreeze Rows

UnfreezeColumns | Unfreeze Columns

MergeCells | Merge Cells

MergeAll | Merge All

MergeHorizontally | Merge Horizontally

MergeVertically | Merge Vertically

Unmerge | Unmerge

UnmergeCells | Unmerge Cells

SortAscending | Ascending

SortDescending | Descending

CustomSort | Custom Sort

ClearAllFilter | Clear

ReapplyFilter | Reapply

Clear | Clear

ClearContents | Clear Contents

ClearAll | Clear All

ClearFormats | Clear Formats

ClearHyperlinks | Clear Hyperlinks

Image | Image

AddColumn | Add Column

SortBy | Sort by

ThenBy | Then by

Chart | Chart

Column | Column

Bar | Bar

Area | Area

Pie | Pie

Doughnut | Doughnut

PieAndDoughnut | Pie/Doughnut

Line | Line

Radar | Radar

Scatter | Scatter

ChartDesign | Chart Design

ClusteredColumn | Clustered Column

StackedColumn | Stacked Column

StackedColumn100 | 100% Stacked Column

ClusteredBar | Clustered Bar

StackedBar | Stacked Bar

StackedBar100 | 100% Stacked Bar

StackedArea | Stacked Area

StackedArea100 | 100% Stacked Area

StackedLine | Stacked Line

StackedLine100 | 100% Stacked Line

AddChartElement | Add Chart Element

Axes | Axes

AxisTitle | Axis Title

ChartTitle | Chart Title

DataLabels | Data Labels

Gridlines | Gridlines

Legends | Legends

PrimaryHorizontal | Primary Horizontal

PrimaryVertical | Primary Vertical

None | None

AboveChart | Above Chart

Center | Center

InsideEnd | Inside End

InsideBase | Inside Base

OutsideEnd | OutSide End

PrimaryMajorHorizontal | Primary Major Horizontal

PrimaryMajorVertical | Primary Major Vertical

PrimaryMinorHorizontal | Primary Minor Horizontal

PrimaryMinorVertical | Primary Minor Vertical

Right | Right

Left | Left

Bottom | Bottom

Top | Top

SwitchRowColumn | Switch Row/Column

ChartTheme | Chart Theme

ChartType | Chart Type

Material | Material

Fabric | Fabric

Bootstrap | Bootstrap

HighContrastLight | HighContrastLight

MaterialDark | MaterialDark

FabricDark | FabricDark

HighContrast | HighContrast

BootstrapDark | BootstrapDark

Bootstrap4 | Bootstrap4

VerticalAxisTitle | Vertical Axis Title

HorizontalAxisTitle | Horizontal Axis Title

EnterTitle | Enter Title

ProtectWorkbook | Protect Workbook

Password | Password (optional) |

unProtectPassword | Password

EnterThePassword | Enter the password

ConfirmPassword | Confirm Password

EnterTheConfirmPassword | Re-enter your password

PasswordAlert | Confirmation password is not identical

UnProtectWorkbook | Unprotect Workbook

UnProtectPasswordAlert | The password you supplied is not correct.

InCorrectPassword | Unable to open the file or worksheet with the given password.

PasswordAlertMsg | Please enter the password

ConfirmPasswordAlertMsg | Please enter the confirm password

IsProtected | is protected

### *Loading translations*

To load translation object in an application, use [load](#) function of the [L10n](#) class.

The following example demonstrates the Spreadsheet in **French** culture. In the below sample we have translated the ribbon tab names and Home tab content (clipboard, cell style).

### **APP.JSX**

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { L10n } from '@syncfusion/ej2-base';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
L10n.load({
  'fr-CH': {
    'spreadsheet': {
      'File': 'Fichier',
      'Home': 'Accueil',
      'Insert': 'Insérer',
      'Formulas': 'Formules',
      'Data': 'Les données',
      'View': 'Vue',
      'Cut': 'Coupe',
      'Copy': 'Copie',
      'Paste': 'Pâte',
      'PasteSpecial': 'Pâte spéciale',
      'All': 'Tous les',
      'Values': 'Valeurs',
      'Formats': 'Les formats',
      'Font': 'fonte',
      'FontSize': 'Taille de police',
      'Bold': 'Audacieux',
```

```

        'Italic': 'Italique',
        'Underline': 'Souligner',
        'Strikethrough': 'Barré',
        'TextColor': 'Couleur du texte',
        'FillColor': 'La couleur de remplissage',
        'HorizontalAlignment': 'Alignement horizontal',
        'AlignLeft': 'Alignez à gauche',
        'AlignCenter': 'centre',
        'AlignRight': 'Aligner à droite',
        'VerticalAlignment': 'Alignement vertical',
        'AlignTop': 'Aligner en haut',
        'AlignMiddle': 'Aligner le milieu',
        'AlignBottom': 'Aligner le bas',
        'InsertFunction': 'Insérer une fonction',
        'Delete': 'Effacer',
        'Rename': 'Rebaptiser',
        'Hide': 'Cacher',
        'Unhide': 'Démasquer',
        'NumberFormat': 'Nombre Format',
    }
}
});
function App() {
    return (
        <SpreadsheetComponent locale='fr-CH'>
            <SheetsDirective>
                <SheetDirective>
                    <RangesDirective>
                        <RangeDirective dataSource={data}></RangeDirective>
                    </RangesDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={100}></ColumnDirective>
                        <ColumnDirective width={110}></ColumnDirective>
                        <ColumnDirective width={100}></ColumnDirective>
                        <ColumnDirective width={180}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
            </SheetsDirective>
        </SpreadsheetComponent>
    );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { L10n } from '@syncfusion/ej2-base';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';

```



```

import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
L10n.load({
  'fr-CH': {
    'spreadsheet': {
      'File': 'Fichier',
      'Home': 'Accueil',
      'Insert': 'Insérer',
      'Formulas': 'Formules',
      'Data': 'Les données',
      'View': 'Vue',
      'Cut': 'Coupe',
      'Copy': 'Copie',
      'Paste': 'Pâte',
      'PasteSpecial': 'Pâte spéciale',
      'All': 'Tous les',
      'Values': 'Valeurs',
      'Formats': 'Les formats',
      'Font': 'fonte',
      'FontSize': 'Taille de police',
      'Bold': 'Audacieux',
      'Italic': 'Italique',
      'Underline': 'Souligner',
      'Strikethrough': 'Barré',
      'TextColor': 'Couleur du texte',
      'FillColor': 'La couleur de remplissage',
      'HorizontalAlignment': 'Alignement horizontal',
      'AlignLeft': 'Alignez à gauche',
      'AlignCenter': 'centre',
      'AlignRight': 'Aligner à droite',
      'VerticalAlignment': 'Alignement vertical',
      'AlignTop': 'Aligner en haut',
      'AlignMiddle': 'Aligner le milieu',
      'AlignBottom': 'Aligner le bas',
      'InsertFunction': 'Insérer une fonction',
      'Delete': 'Effacer',
      'Rename': 'Rebaptiser',
      'Hide': 'Cacher',
      'Unhide': 'Démasquer',
      'NumberFormat': 'Nombre Format',
    }
  }
});
function App() {
  return (
    <SpreadsheetComponent locale='fr-CH'>
      <SheetsDirective>
        <SheetDirective>
          <RangesDirective>
            <RangeDirective dataSource={data}></RangeDirective>
          </RangesDirective>
          <ColumnsDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={110}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={180}></ColumnDirective>
          </ColumnsDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  );
}

```

```

        <ColumnDirective width={130}></ColumnDirective>
        <ColumnDirective width={130}></ColumnDirective>
      </ColumnsDirective>
    </SheetDirective>
  </SheetsDirective>
</SpreadsheetComponent>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### DATASOURCE.JSX

```

/**
 * Default data source
 */
export let data = [{
  OrderID: 10248,
  CustomerID: 'VINET',
  EmployeeID: 5,
  ShipName: 'Vins et alcools Chevalier',
  ShipCity: 'Reims',
  ShipAddress: '59 rue de l'Abbaye'
},
{
  OrderID: 10249,
  CustomerID: 'TOMSP',
  EmployeeID: 6,
  ShipName: 'Toms Spezialitäten',
  ShipCity: 'Münster',
  ShipAddress: 'Luisenstr. 48'
},
{
  OrderID: 10250,
  CustomerID: 'HANAR',
  EmployeeID: 4,
  ShipName: 'Hanari Carnes',
  ShipCity: 'Rio de Janeiro',
  ShipAddress: 'Rua do Paço. 67'
},
{
  OrderID: 10251,
  CustomerID: 'VICTE',
  EmployeeID: 3,
  ShipName: 'Victuailles en stock',
  ShipCity: 'Lyon',
  ShipAddress: '2. rue du Commerce'
},
{
  OrderID: 10252,
  CustomerID: 'SUPRD',
  EmployeeID: 4,
  ShipName: 'Suprêmes délices',
  ShipCity: 'Charleroi',

```

```

      ShipAddress: 'Boulevard Tirou. 255'
    }
  ];

```

### DATASOURCE.TSX

```

/**
 * Default data source
 */
export let data: Object[] = [{
  OrderID: 10248,
  CustomerID: 'VINET',
  EmployeeID: 5,
  ShipName: 'Vins et alcools Chevalier',
  ShipCity: 'Reims',
  ShipAddress: '59 rue de l'Abbaye'
},
{
  OrderID: 10249,
  CustomerID: 'TOMSP',
  EmployeeID: 6,
  ShipName: 'Toms Spezialitäten',
  ShipCity: 'Münster',
  ShipAddress: 'Luisenstr. 48'
},
{
  OrderID: 10250,
  CustomerID: 'HANAR',
  EmployeeID: 4,
  ShipName: 'Hanari Carnes',
  ShipCity: 'Rio de Janeiro',
  ShipAddress: 'Rua do Paço. 67'
},
{
  OrderID: 10251,
  CustomerID: 'VICTE',
  EmployeeID: 3,
  ShipName: 'Victuailles en stock',
  ShipCity: 'Lyon',
  ShipAddress: '2. rue du Commerce'
},
{
  OrderID: 10252,
  CustomerID: 'SUPRD',
  EmployeeID: 4,
  ShipName: 'Suprêmes délices',
  ShipCity: 'Charleroi',
  ShipAddress: 'Boulevard Tirou. 255'
}
];

```

### Internationalization

The Internationalization library is used to globalize number, date, and time values in the spreadsheet component.

The following example demonstrates the Spreadsheet in French [ fr-CH] culture. In the below sample we have globalized the Date(Date column), Time(Time column), and Currency(Amount column) formats.

```
`ts
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { L10n, loadCldr, setCulture, setCurrencyCode } from '@syncfusion/ej2-base';
import { SpreadsheetComponent, SheetsDirective, SheetDirective, RangesDirective, RangeDirective,
ColumnsDirective, ColumnDirective } from '@syncfusion/ej2-react-spreadsheet';
import * as cagregorian from './ca-gregorian.json';
import * as currencies from './currencies.json';
import * as numberingSystems from './numberingSystems.json';
import * as numbers from './numbers.json';
import * as timeZoneNames from './timeZoneNames.json';
import { data } from './datasource';
loadCldr(currencies, cagregorian, numbers, timeZoneNames, numberingSystems);
setCulture('fr-CH');
setCurrencyCode('EUR');
L10n.load({
'fr-CH': {
'spreadsheet': {
'File': 'Fichier',
'Home': 'Accueil',
'Insert': 'Insérer',
'Formulas': 'Formules',
'Data': 'Les données',
'View': 'Vue',
'Cut': 'Coupe',
'Copy': 'Copie',
'Paste': 'Pte',
'PasteSpecial': 'Pte spéciale',
'All': 'Tous les',
'Values': 'Valeurs',
'Formats': 'Les formats',
'Font': 'fonte',
'FontSize': 'Taille de police',
```

```
'Bold': 'Audacieux',
'Italic': 'Italique',
'Underline': 'Souligner',
'Strikethrough': 'Barré',
'TextColor': 'Couleur du texte',
'FillColor': 'La couleur de remplissage',
'HorizontalAlignment': 'Alignement horizontal',
'AlignLeft': 'Alignez à gauche',
'AlignCenter': 'centre',
'AlignRight': 'Aligner à droite',
'VerticalAlignment': 'Alignement vertical',
'AlignTop': 'Aligner en haut',
'AlignMiddle': 'Aligner le milieu',
'AlignBottom': 'Aligner le bas',
'InsertFunction': 'Insérer une fonction',
'Delete': 'Effacer',
'Rename': 'Rebaptiser',
'Hide': 'Cacher',
'Unhide': 'Démasquer',
'NumberFormat': 'Nombre Format',
}
}
});
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      //Applies cell and number formatting to specified range of the active sheet
      spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center', verticalAlign: 'middle' }, 'A1:F1');
      spreadsheet.numberFormat('$#,##0.00', 'F2:F11');
    }
  }, []);
```

```

return (
  <SpreadsheetComponent ref={spreadsheetRef} locale='fr-CH' >
    <SheetsDirective>
      <SheetDirective>
        <RangesDirective>
          <RangeDirective dataSource={data}></RangeDirective>
        </RangesDirective>
        <ColumnsDirective>
          <ColumnDirective width={100}></ColumnDirective>
          <ColumnDirective width={110}></ColumnDirective>
          <ColumnDirective width={100}></ColumnDirective>
          <ColumnDirective width={180}></ColumnDirective>
          <ColumnDirective width={130}></ColumnDirective>
          <ColumnDirective width={130}></ColumnDirective>
        </ColumnsDirective>
      </SheetDirective>
    </SheetsDirective>
  </SpreadsheetComponent>
);
};

export default App;

const root = createRoot(document.getElementById('root')!);

root.render(<App />);
`

```

Internationalization [sample link](#)

### Right to left (RTL)

RTL provides an option to switch the text direction and layout of the Spreadsheet component from right to left. It improves the user experiences and accessibility for users who use right-to-left languages (Arabic, Farsi, Urdu, etc.). To enable RTL Spreadsheet, set the [enableRtl](#) to true.

### APP.JSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { L10n } from '@syncfusion/ej2-base';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';

```

```

import { data } from './datasource';
L10n.load({
  'ar-AE': {
    'spreadsheet': {
      "File": "ملف",
      "Home": "هم",
      "Insert": "إدراج",
      "Formulas": "الصيغ",
      "View": "معاينة",
      "Data": "البيانات",
      "Cut": "قطع",
      "Copy": "نسخ",
      "Paste": "معجون",
      "PasteSpecial": "لصق خاص",
      "All": "جميع",
      "Values": "القيم",
      "Formats": "شكل",
      "Font": "الخط",
      "FontSize": "حجم الخط",
      "Bold": "جريء",
      "Italic": "مائل",
      "Underline": "أكد",
      "Strikethrough": "يتوسطه",
      "TextColor": "لون الخط",
      "FillColor": "لون التعبئة",
      "HorizontalAlignment": "المحاذاة الأفقية",
      "AlignLeft": "محاذاة إلى اليسار",
      "AlignCenter": "مركز",
      "AlignRight": "محاذاة إلى اليمين",
      "VerticalAlignment": "محاذاة عمودية",
      "AlignTop": "محاذاة أعلى",
      "AlignMiddle": "محاذاة الأوسط",
      "AlignBottom": "أسفل محاذاة",
      "InsertFunction": "إدراج وظيفة",
      "Delete": "حذف",
      "Rename": "إعادة تسمية",
      "Hide": "إخفاء",
      "Unhide": "ظهار"
    }
  }
});
function App() {
  return (
    <SpreadsheetComponent locale='ar-AE' enableRtl={true}>
      <SheetsDirective>
        <SheetDirective>
          <RangesDirective>
            <RangeDirective dataSource={data}></RangeDirective>
          </RangesDirective>
          <ColumnsDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={110}></ColumnDirective>
            <ColumnDirective width={100}></ColumnDirective>
            <ColumnDirective width={180}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
          </ColumnsDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  );
}

```

```

        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

## APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { L10n } from '@syncfusion/ej2-base';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
L10n.load({
  'ar-AE': {
    'spreadsheet': {
      "File": "ملف",
      "Home": "هم",
      "Insert": "إدراج",
      "Formulas": "الصيغ",
      "View": "معاينة",
      "Data": "البيانات",
      "Cut": "قطع",
      "Copy": "نسخ",
      "Paste": "معجون",
      "PasteSpecial": "لصق خاص",
      "All": "جميع",
      "Values": "القيم",
      "Formats": "شكل",
      "Font": "الخط",
      "FontSize": "حجم الخط",
      "Bold": "جريء",
      "Italic": "مائل",
      "Underline": "أكد",
      "Strikethrough": "يتوسطه",
      "TextColor": "لون الخط",
      "FillColor": "لون التعبئة",
      "HorizontalAlignment": "المحاذاة الأفقية",
      "AlignLeft": "محاذاة إلى اليسار",
      "AlignCenter": "مركز",
      "AlignRight": "محاذاة إلى اليمين",
      "VerticalAlignment": "محاذاة عمودية",
      "AlignTop": "محاذاة أعلى",
      "AlignMiddle": "محاذاة الأوسط",
      "AlignBottom": "أسفل محاذاة",
      "InsertFunction": "إدراج وظيفة",
      "Delete": "حذف",
      "Rename": "إعادة تسمية",
      "Hide": "إخفاء",
    }
  }
});

```



```

        "Unhide": "ظهار"
    }
}
});
function App() {
    return (
        <SpreadsheetComponent locale='ar-AE' enableRtl={true}>
            <SheetsDirective>
                <SheetDirective>
                    <RangesDirective>
                        <RangeDirective dataSource={data}></RangeDirective>
                    </RangesDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={100}></ColumnDirective>
                        <ColumnDirective width={110}></ColumnDirective>
                        <ColumnDirective width={100}></ColumnDirective>
                        <ColumnDirective width={180}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
            </SheetsDirective>
        </SpreadsheetComponent>
    );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

## DATASOURCE.JSX

```

/**
 * Default data source
 */
export let data = [{
    OrderID: 10248,
    CustomerID: 'VINET',
    EmployeeID: 5,
    ShipName: 'Vins et alcools Chevalier',
    ShipCity: 'Reims',
    ShipAddress: '59 rue de l'Abbaye'
},
{
    OrderID: 10249,
    CustomerID: 'TOMSP',
    EmployeeID: 6,
    ShipName: 'Toms Spezialitäten',
    ShipCity: 'Münster',
    ShipAddress: 'Luisenstr. 48'
},
{
    OrderID: 10250,
    CustomerID: 'HANAR',
    EmployeeID: 4,
    ShipName: 'Hanari Carnes',

```

```

      ShipCity: 'Rio de Janeiro',
      ShipAddress: 'Rua do Paço, 67'
    },
    {
      OrderID: 10251,
      CustomerID: 'VICTE',
      EmployeeID: 3,
      ShipName: 'Victuailles en stock',
      ShipCity: 'Lyon',
      ShipAddress: '2, rue du Commerce'
    },
    {
      OrderID: 10252,
      CustomerID: 'SUPRD',
      EmployeeID: 4,
      ShipName: 'Suprêmes délices',
      ShipCity: 'Charleroi',
      ShipAddress: 'Boulevard Tirou, 255'
    }
  ]];

```

### DATASOURCE.TSX

```

/**
 * Default data source
 */
export let data: Object[] = [{
  OrderID: 10248,
  CustomerID: 'VINET',
  EmployeeID: 5,
  ShipName: 'Vins et alcools Chevalier',
  ShipCity: 'Reims',
  ShipAddress: '59 rue de l'Abbaye'
},
{
  OrderID: 10249,
  CustomerID: 'TOMSP',
  EmployeeID: 6,
  ShipName: 'Toms Spezialitäten',
  ShipCity: 'Münster',
  ShipAddress: 'Luisenstr. 48'
},
{
  OrderID: 10250,
  CustomerID: 'HANAR',
  EmployeeID: 4,
  ShipName: 'Hanari Carnes',
  ShipCity: 'Rio de Janeiro',
  ShipAddress: 'Rua do Paço, 67'
},
{
  OrderID: 10251,
  CustomerID: 'VICTE',
  EmployeeID: 3,
  ShipName: 'Victuailles en stock',
  ShipCity: 'Lyon',
  ShipAddress: '2, rue du Commerce'
}
];

```

```
},
{
  OrderID: 10252,
  CustomerID: 'SUPRD',
  EmployeeID: 4,
  ShipName: 'Suprêmes délices',
  ShipCity: 'Charleroi',
  ShipAddress: 'Boulevard Tirou, 255'
}];
```

### Note

You can refer to our [React Spreadsheet](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Spreadsheet example](#) to know how to present and manipulate data.

### See Also

- [Localization](#)

### Freeze pane in React Spreadsheet component

Freeze Panes helps you to keep particular rows or columns visible when scrolling the sheet content in the spreadsheet. You can specify the number of frozen rows and columns using the [frozenRows](#) and [frozenColumns](#) properties inside the [Link to the Video](#) property.

To get started quickly with Freeze Panes in React Spreadsheet, you can check on this video:

### Apply freeze panes on UI

#### User Interface:

In the active spreadsheet, click the cell where you want to create freeze panes. Freeze panes can be done in any of the following ways:

- Select the View tab in the Ribbon toolbar and choose the **Freeze Panes** item.
- Use the [freezePanels](#) method programmatically.

### FrozenRows

It allows you to keep a certain number of rows visible while scrolling vertically through the rest of the worksheet.

#### User Interface:

In the active spreadsheet, select the cell where you want to create frozen rows. Frozen rows can be done in any one of the following ways:

- Select the View tab in the Ribbon toolbar and choose the **Freeze Rows** item.
- You can specify the number of frozen rows using the **frozenRows** property inside the **Sheet** property.

### FrozenColumns

It allows you to keep a certain number of columns visible while scrolling horizontally through the rest of the worksheet.

**User Interface:**

In the active spreadsheet, select the cell where you want to create frozen columns. Frozen columns can be done in any one of the following ways:

- Select the View tab in the Ribbon toolbar and choose the Freeze Columns item.
- You can specify the number of frozen columns using the frozenColumns property inside the Sheet property.

In this demo, the frozenColumns is set as '2', and the frozenRows is set as '2'. Hence, the two columns on the left and the top two rows are frozen.

**APP.JSX**

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { tradeData } from './datasource';
function App() {
  return (
    <div>
      <SpreadsheetComponent>
        <SheetsDirective>
          <SheetDirective frozenRows={2} frozenColumns={2}
selectedRange='C1'>
            <RangesDirective>
              <RangeDirective
dataSource={tradeData}></RangeDirective>
            </RangesDirective>
            <ColumnsDirective>
              <ColumnDirective width={100}></ColumnDirective>
              <ColumnDirective width={130}></ColumnDirective>
              <ColumnDirective width={96}></ColumnDirective>
              <ColumnDirective width={130}></ColumnDirective>
              <ColumnDirective width={130}></ColumnDirective>
              <ColumnDirective width={96}></ColumnDirective>
              <ColumnDirective width={100}></ColumnDirective>
              <ColumnDirective width={100}></ColumnDirective>
              <ColumnDirective width={110}></ColumnDirective>
              <ColumnDirective width={100}></ColumnDirective>
              <ColumnDirective width={130}></ColumnDirective>
              <ColumnDirective width={150}></ColumnDirective>
            </ColumnsDirective>
          </SheetDirective>
        </SheetsDirective>
      </SpreadsheetComponent>
    </div>);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}
```

**APP.TSX**

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { tradeData } from '../datasource';
function App() {
    return (
        <div>
            <SpreadsheetComponent>
                <SheetsDirective>
                    <SheetDirective frozenRows={2} frozenColumns={2}
selectedRange='C1'>
                        <RangesDirective>
                            <RangeDirective
dataSource={tradeData}></RangeDirective>
                        </RangesDirective>
                        <ColumnsDirective>
                            <ColumnDirective width={100}></ColumnDirective>
                            <ColumnDirective width={130}></ColumnDirective>
                            <ColumnDirective width={96}></ColumnDirective>
                            <ColumnDirective width={130}></ColumnDirective>
                            <ColumnDirective width={130}></ColumnDirective>
                            <ColumnDirective width={96}></ColumnDirective>
                            <ColumnDirective width={100}></ColumnDirective>
                            <ColumnDirective width={100}></ColumnDirective>
                            <ColumnDirective width={110}></ColumnDirective>
                            <ColumnDirective width={100}></ColumnDirective>
                            <ColumnDirective width={130}></ColumnDirective>
                            <ColumnDirective width={150}></ColumnDirective>
                        </ColumnsDirective>
                    </SheetDirective>
                </SheetsDirective>
            </SpreadsheetComponent>
        </div>);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}
```

**DATASOURCE.JSX**

```
/**
 * Default data source
 */
export let defaultData = [
    { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
    { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
```

```

    { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
      Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
    { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
    { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
'12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
70 },
    { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
      Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
    { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
      Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
    { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
      Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
    { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
      Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
    { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
      Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
  ];
  /**
   * Grid datasource
   */
  export function getTradeData(dataCount) {
    let employees = [
      'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
      'Fuller', 'Leverling', 'Peacock',
      'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
      'Dodsworth', 'Anne',
      'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
      'Jack', 'Rose'
    ];
    let designation = ['Manager', 'CFO', 'Designer', 'Developer', 'Program
Directory', 'System Analyst', 'Project Lead'];
    let mail = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
'journrapide.com'];
    let location = ['UK', 'USA', 'Sweden', 'France', 'Canada', 'Argentina',
'Austria', 'Germany', 'Mexico'];
    let status = ['Active', 'Inactive'];
    let trustworthiness = ['Perfect', 'Sufficient', 'Insufficient'];
    let tradeData = [];
    let address = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do Paço, 67',
'2, rue du Commerce', 'Boulevard Tirou, 255',
'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
Mercado, 12',
'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
'Taucherstraße 10', 'Taucherstraße 10',
'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
Circus', 'Av. dos Lusíadas, 23',

```

```

    'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
123', '2817 Milton Dr.'];
    let employeeimg = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
        dataCount = parseInt(dataCount, 10);
    }
    for (let i = 1; i <= dataCount; i++) {
        let code = 10000;
        tradeData.push({
            'EmployeeID': code + i,
            'Employees': employees[Math.floor(Math.random() *
employees.length)] + ' ' + employees[Math.floor(Math.random() *
employees.length)],
            'Designation': designation[Math.floor(Math.random() *
designation.length)],
            'Location': location[Math.floor(Math.random() *
location.length)],
            'Status': status[Math.floor(Math.random() * status.length)],
            'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
            'Rating': Math.floor(Math.random() * 5),
            'Software': Math.floor(Math.random() * 100),
            'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
            'CurrentSalary': Math.floor((Math.random() * 100000)),
            'Address': address[Math.floor(Math.random() * address.length)],
        });
        let employee = 'Employees';
        let emp = tradeData[i - 1][employee];
        let sName = emp.substr(0, emp.indexOf(' ')).toLowerCase();
        let empmail = 'Mail';
        tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }
    return tradeData;
}
export let tradeData = [
    {
        "EmployeeID": 10001,
        "Employees": "Laura Nancy",
        "Designation": "Designer",
        "Location": "France",
        "Status": "Inactive",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
        "Software": 69,
        "EmployeeImg": "usermale",
        "CurrentSalary": 84194,
        "Address": "Taucherstraße 10",
        "Mail": "laura15@jourrapide.com"
    },
    {
        "EmployeeID": 10002,
        "Employees": "Zachery Van",
        "Designation": "CFO",
        "Location": "Canada",
        "Status": "Inactive",

```

```

    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 99,
    "EmployeeImg": "usermale",
    "CurrentSalary": 55349,
    "Address": "5ª Ave. Los Palos Grandes",
    "Mail": "zachery109@sample.com"
  },
  {
    "EmployeeID": 10003,
    "Employees": "Rose Fuller",
    "Designation": "CFO",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 1,
    "EmployeeImg": "usermale",
    "CurrentSalary": 16477,
    "Address": "2817 Milton Dr.",
    "Mail": "rose55@rpy.com"
  },
  {
    "EmployeeID": 10004,
    "Employees": "Jack Bergs",
    "Designation": "Manager",
    "Location": "Mexico",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 36,
    "EmployeeImg": "usermale",
    "CurrentSalary": 49040,
    "Address": "2, rue du Commerce",
    "Mail": "jack30@sample.com"
  },
  {
    "EmployeeID": 10005,
    "Employees": "Vinet Bergs",
    "Designation": "Program Directory",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 39,
    "EmployeeImg": "usermale",
    "CurrentSalary": 5495,
    "Address": "Rua da Panificadora, 12",
    "Mail": "vinet32@jourrapide.com"
  },
  {
    "EmployeeID": 10006,
    "Employees": "Buchanan Van",
    "Designation": "Designer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",

```



```
    "Rating": 4,  
    "Software": 78,  
    "EmployeeImg": "usermale",  
    "CurrentSalary": 42182,  
    "Address": "24, place Kléber",  
    "Mail": "buchanan18@mail.com"  
  },  
  {  
    "EmployeeID": 10007,  
    "Employees": "Dodsworth Nancy",  
    "Designation": "Project Lead",  
    "Location": "USA",  
    "Status": "Inactive",  
    "Trustworthiness": "Sufficient",  
    "Rating": 0,  
    "Software": 0,  
    "EmployeeImg": "userfemale",  
    "CurrentSalary": 35776,  
    "Address": "Rua do Paço, 67",  
    "Mail": "dodsworth84@mail.com"  
  },  
  {  
    "EmployeeID": 10008,  
    "Employees": "Laura Jack",  
    "Designation": "Developer",  
    "Location": "Austria",  
    "Status": "Inactive",  
    "Trustworthiness": "Perfect",  
    "Rating": 3,  
    "Software": 89,  
    "EmployeeImg": "usermale",  
    "CurrentSalary": 25108,  
    "Address": "Rua da Panificadora, 12",  
    "Mail": "laura82@mail.com"  
  },  
  {  
    "EmployeeID": 10009,  
    "Employees": "Anne Fuller",  
    "Designation": "Program Directory",  
    "Location": "Mexico",  
    "Status": "Active",  
    "Trustworthiness": "Insufficient",  
    "Rating": 0,  
    "Software": 19,  
    "EmployeeImg": "userfemale",  
    "CurrentSalary": 32568,  
    "Address": "Gran Vía, 1",  
    "Mail": "anne97@jourrapide.com"  
  },  
  {  
    "EmployeeID": 10010,  
    "Employees": "Buchanan Andrew",  
    "Designation": "Designer",  
    "Location": "Austria",  
    "Status": "Active",  
    "Trustworthiness": "Insufficient",  
    "Rating": 1,
```

```

    "Software": 62,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 12320,
    "Address": "P.O. Box 555",
    "Mail": "buchanan50@jourrapide.com"
  },
  {
    "EmployeeID": 10011,
    "Employees": "Andrew Janet",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 20890,
    "Address": "Starenweg 5",
    "Mail": "andrew63@mail.com"
  },
  {
    "EmployeeID": 10012,
    "Employees": "Margaret Tamer",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 7,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 22337,
    "Address": "Magazinweg 7",
    "Mail": "margaret26@mail.com"
  },
  {
    "EmployeeID": 10013,
    "Employees": "Tamer Fuller",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 89181,
    "Address": "Taucherstraße 10",
    "Mail": "tamer40@arpy.com"
  },
  {
    "EmployeeID": 10014,
    "Employees": "Tamer Anne",
    "Designation": "CFO",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 18,

```

```
"EmployeeImg": "usermale",
"CurrentSalary": 20998,
"Address": "Taucherstraße 10",
"Mail": "tamer68@arpy.com"
},
{
  "EmployeeID": 10015,
  "Employees": "Anton Davolio",
  "Designation": "Project Lead",
  "Location": "France",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 4,
  "Software": 8,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 48232,
  "Address": "Luisenstr. 48",
  "Mail": "anton46@mail.com"
},
{
  "EmployeeID": 10016,
  "Employees": "Buchanan Buchanan",
  "Designation": "System Analyst",
  "Location": "Austria",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 19,
  "EmployeeImg": "usermale",
  "CurrentSalary": 43041,
  "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
  "Mail": "buchanan68@mail.com"
},
{
  "EmployeeID": 10017,
  "Employees": "King Buchanan",
  "Designation": "Program Directory",
  "Location": "Sweden",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 44,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 25259,
  "Address": "Magazinweg 7",
  "Mail": "king80@jourrapide.com"
},
{
  "EmployeeID": 10018,
  "Employees": "Rose Michael",
  "Designation": "Project Lead",
  "Location": "Canada",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 4,
  "Software": 31,
  "EmployeeImg": "userfemale",
```

```
    "CurrentSalary": 91156,  
    "Address": "Fauntleroy Circus",  
    "Mail": "rose75@mail.com"  
  },  
  {  
    "EmployeeID": 10019,  
    "Employees": "King Bergs",  
    "Designation": "Developer",  
    "Location": "Germany",  
    "Status": "Active",  
    "Trustworthiness": "Sufficient",  
    "Rating": 2,  
    "Software": 29,  
    "EmployeeImg": "userfemale",  
    "CurrentSalary": 28826,  
    "Address": "2817 Milton Dr.",  
    "Mail": "king57@jourrapide.com"  
  },  
  {  
    "EmployeeID": 10020,  
    "Employees": "Davolio Fuller",  
    "Designation": "Designer",  
    "Location": "Canada",  
    "Status": "Inactive",  
    "Trustworthiness": "Sufficient",  
    "Rating": 3,  
    "Software": 35,  
    "EmployeeImg": "userfemale",  
    "CurrentSalary": 71035,  
    "Address": "Gran Vía, 1",  
    "Mail": "davolio29@arpy.com"  
  },  
  {  
    "EmployeeID": 10021,  
    "Employees": "Rose Rose",  
    "Designation": "CFO",  
    "Location": "Germany",  
    "Status": "Active",  
    "Trustworthiness": "Perfect",  
    "Rating": 3,  
    "Software": 38,  
    "EmployeeImg": "usermale",  
    "CurrentSalary": 68123,  
    "Address": "Rua do Mercado, 12",  
    "Mail": "rose54@arpy.com"  
  },  
  {  
    "EmployeeID": 10022,  
    "Employees": "Andrew Michael",  
    "Designation": "Program Directory",  
    "Location": "UK",  
    "Status": "Inactive",  
    "Trustworthiness": "Insufficient",  
    "Rating": 2,  
    "Software": 61,  
    "EmployeeImg": "userfemale",  
    "CurrentSalary": 75470,
```

```
      "Address": "2, rue du Commerce",
      "Mail": "andrew88@jourrapide.com"
    },
    {
      "EmployeeID": 10023,
      "Employees": "Davolio Kathryn",
      "Designation": "Manager",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Perfect",
      "Rating": 3,
      "Software": 25,
      "EmployeeImg": "usermale",
      "CurrentSalary": 25234,
      "Address": "Hauptstr. 31",
      "Mail": "davolio42@sample.com"
    },
    {
      "EmployeeID": 10024,
      "Employees": "Anne Fleet",
      "Designation": "System Analyst",
      "Location": "UK",
      "Status": "Active",
      "Trustworthiness": "Perfect",
      "Rating": 3,
      "Software": 0,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 8341,
      "Address": "59 rue de l'Abbaye",
      "Mail": "anne86@arpy.com"
    },
    {
      "EmployeeID": 10025,
      "Employees": "Margaret Andrew",
      "Designation": "System Analyst",
      "Location": "Germany",
      "Status": "Inactive",
      "Trustworthiness": "Insufficient",
      "Rating": 3,
      "Software": 51,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 84975,
      "Address": "P.O. Box 555",
      "Mail": "margaret41@arpy.com"
    },
    {
      "EmployeeID": 10026,
      "Employees": "Kathryn Laura",
      "Designation": "Project Lead",
      "Location": "Austria",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 3,
      "Software": 48,
      "EmployeeImg": "usermale",
      "CurrentSalary": 97282,
      "Address": "Avda. Azteca 123",
```

```
      "Mail": "kathryn82@rpy.com"
    },
    {
      "EmployeeID": 10027,
      "Employees": "Michael Michael",
      "Designation": "Developer",
      "Location": "UK",
      "Status": "Inactive",
      "Trustworthiness": "Perfect",
      "Rating": 4,
      "Software": 16,
      "EmployeeImg": "usermale",
      "CurrentSalary": 4184,
      "Address": "Rua do Paço, 67",
      "Mail": "michael58@jourrapide.com"
    },
    {
      "EmployeeID": 10028,
      "Employees": "Leverling Vinet",
      "Designation": "Project Lead",
      "Location": "Germany",
      "Status": "Inactive",
      "Trustworthiness": "Perfect",
      "Rating": 0,
      "Software": 57,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 38370,
      "Address": "59 rue de l'Abbaye",
      "Mail": "leverling102@sample.com"
    },
    {
      "EmployeeID": 10029,
      "Employees": "Rose Jack",
      "Designation": "Developer",
      "Location": "UK",
      "Status": "Active",
      "Trustworthiness": "Perfect",
      "Rating": 0,
      "Software": 46,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 84790,
      "Address": "Rua do Mercado, 12",
      "Mail": "rose108@jourrapide.com"
    },
    {
      "EmployeeID": 10030,
      "Employees": "Vinet Van",
      "Designation": "Developer",
      "Location": "USA",
      "Status": "Active",
      "Trustworthiness": "Sufficient",
      "Rating": 0,
      "Software": 40,
      "EmployeeImg": "usermale",
      "CurrentSalary": 71005,
      "Address": "Gran Vía, 1",
      "Mail": "vinet90@jourrapide.com"
    }
  ]
}
```

```
}
];
```

**DATASOURCE.TSX**

```
/**
 * Default data source
 */
export let defaultData: Object[] = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
    Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
  { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
    Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
  { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54
AM', Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
  { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time:
'12:43:59 AM', Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit:
70 },
  { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
    Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
  { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
    Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
  { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
    Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
  { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
    Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
  { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
    Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
/**
 * Grid datasource
 */
export function getTradeData(dataCount?: number): object {
  let employees: string[] = [
    'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
    'Fuller', 'Leverling', 'Peacock',
    'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
    'Dodsworth', 'Anne',
    'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
    'Jack', 'Rose'];
  let designation: string[] = ['Manager', 'CFO', 'Designer', 'Developer',
    'Program Directory', 'System Analyst', 'Project Lead'];
  let mail: string[] = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
    'jourrapide.com'];
  let location: string[] = ['UK', 'USA', 'Sweden', 'France', 'Canada',
    'Argentina', 'Austria', 'Germany', 'Mexico'];
  let status: string[] = ['Active', 'Inactive'];
  let trustworthiness: string[] = ['Perfect', 'Sufficient',
    'Insufficient'];
  let tradeData: Object[] = [];
  let address: string[] = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do
Paço, 67', '2, rue du Commerce', 'Boulevard Tirou, 255',
    'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
Mercado, 12',
```

```

    'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
    'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
    'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
    'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
    'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
    'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
    'Taucherstraße 10', 'Taucherstraße 10',
    'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
Circus', 'Av. dos Lusíadas, 23',
    'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
123', '2817 Milton Dr.'];
    let employeeimg: string[] = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
        dataCount = parseInt(dataCount, 10);
    }
    for (let i: number = 1; i <= dataCount; i++) {
        let code: any = 10000;
        tradeData.push({
            'EmployeeID': code + i,
            'Employees':
                employees[Math.floor(Math.random() * employees.length)] + '
' + employees[Math.floor(Math.random() * employees.length)],
            'Designation': designation[Math.floor(Math.random() *
designation.length)],
            'Location': location[Math.floor(Math.random() *
location.length)],
            'Status': status[Math.floor(Math.random() * status.length)],
            'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
            'Rating': Math.floor(Math.random() * 5),
            'Software': Math.floor(Math.random() * 100),
            'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
            'CurrentSalary': Math.floor((Math.random() * 100000)),
            'Address': address[Math.floor(Math.random() * address.length)],
        });
        let employee: string = 'Employees';
        let emp: string = tradeData[i - 1][employee];
        let sName: string = emp.substr(0, emp.indexOf(' ')).toLowerCase();
        let empmail: string = 'Mail';
        tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }
    return tradeData;
}
export let tradeData: Object[] = [
    {
        "EmployeeID": 10001,
        "Employees": "Laura Nancy",
        "Designation": "Designer",
        "Location": "France",
        "Status": "Inactive",
        "Trustworthiness": "Sufficient",

```



```

    "Rating": 0,
    "Software": 69,
    "EmployeeImg": "usermale",
    "CurrentSalary": 84194,
    "Address": "Taucherstraße 10",
    "Mail": "laura15@jourrapide.com"
  },
  {
    "EmployeeID": 10002,
    "Employees": "Zachery Van",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 99,
    "EmployeeImg": "usermale",
    "CurrentSalary": 55349,
    "Address": "5a Ave. Los Palos Grandes",
    "Mail": "zachery109@sample.com"
  },
  {
    "EmployeeID": 10003,
    "Employees": "Rose Fuller",
    "Designation": "CFO",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 1,
    "EmployeeImg": "usermale",
    "CurrentSalary": 16477,
    "Address": "2817 Milton Dr.",
    "Mail": "rose55@rpy.com"
  },
  {
    "EmployeeID": 10004,
    "Employees": "Jack Bergs",
    "Designation": "Manager",
    "Location": "Mexico",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 36,
    "EmployeeImg": "usermale",
    "CurrentSalary": 49040,
    "Address": "2, rue du Commerce",
    "Mail": "jack30@sample.com"
  },
  {
    "EmployeeID": 10005,
    "Employees": "Vinet Bergs",
    "Designation": "Program Directory",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,

```

```

    "Software": 39,
    "EmployeeImg": "usermale",
    "CurrentSalary": 5495,
    "Address": "Rua da Panificadora, 12",
    "Mail": "vinet32@jourrapide.com"
  },
  {
    "EmployeeID": 10006,
    "Employees": "Buchanan Van",
    "Designation": "Designer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 42182,
    "Address": "24, place Kléber",
    "Mail": "buchanan18@mail.com"
  },
  {
    "EmployeeID": 10007,
    "Employees": "Dodsworth Nancy",
    "Designation": "Project Lead",
    "Location": "USA",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 35776,
    "Address": "Rua do Paço, 67",
    "Mail": "dodsworth84@mail.com"
  },
  {
    "EmployeeID": 10008,
    "Employees": "Laura Jack",
    "Designation": "Developer",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 89,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25108,
    "Address": "Rua da Panificadora, 12",
    "Mail": "laura82@mail.com"
  },
  {
    "EmployeeID": 10009,
    "Employees": "Anne Fuller",
    "Designation": "Program Directory",
    "Location": "Mexico",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 0,
    "Software": 19,

```

```

      "EmployeeImg": "userfemale",
      "CurrentSalary": 32568,
      "Address": "Gran Vía, 1",
      "Mail": "anne97@jourrapide.com"
    },
    {
      "EmployeeID": 10010,
      "Employees": "Buchanan Andrew",
      "Designation": "Designer",
      "Location": "Austria",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 1,
      "Software": 62,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 12320,
      "Address": "P.O. Box 555",
      "Mail": "buchanan50@jourrapide.com"
    },
    {
      "EmployeeID": 10011,
      "Employees": "Andrew Janet",
      "Designation": "System Analyst",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 3,
      "Software": 8,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 20890,
      "Address": "Starenweg 5",
      "Mail": "andrew63@mail.com"
    },
    {
      "EmployeeID": 10012,
      "Employees": "Margaret Tamer",
      "Designation": "System Analyst",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 4,
      "Software": 7,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 22337,
      "Address": "Magazinweg 7",
      "Mail": "margaret26@mail.com"
    },
    {
      "EmployeeID": 10013,
      "Employees": "Tamer Fuller",
      "Designation": "CFO",
      "Location": "Canada",
      "Status": "Active",
      "Trustworthiness": "Insufficient",
      "Rating": 3,
      "Software": 78,
      "EmployeeImg": "usermale",

```

```
"CurrentSalary": 89181,
"Address": "Taucherstraße 10",
"Mail": "tamer40@arpy.com"
},
{
  "EmployeeID": 10014,
  "Employees": "Tamer Anne",
  "Designation": "CFO",
  "Location": "Sweden",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 18,
  "EmployeeImg": "usermale",
  "CurrentSalary": 20998,
  "Address": "Taucherstraße 10",
  "Mail": "tamer68@arpy.com"
},
{
  "EmployeeID": 10015,
  "Employees": "Anton Davolio",
  "Designation": "Project Lead",
  "Location": "France",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 4,
  "Software": 8,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 48232,
  "Address": "Luisenstr. 48",
  "Mail": "anton46@mail.com"
},
{
  "EmployeeID": 10016,
  "Employees": "Buchanan Buchanan",
  "Designation": "System Analyst",
  "Location": "Austria",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 19,
  "EmployeeImg": "usermale",
  "CurrentSalary": 43041,
  "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
  "Mail": "buchanan68@mail.com"
},
{
  "EmployeeID": 10017,
  "Employees": "King Buchanan",
  "Designation": "Program Directory",
  "Location": "Sweden",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 44,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 25259,
```

```
"Address": "Magazinweg 7",
"Mail": "king80@jourrapide.com"
},
{
  "EmployeeID": 10018,
  "Employees": "Rose Michael",
  "Designation": "Project Lead",
  "Location": "Canada",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 4,
  "Software": 31,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 91156,
  "Address": "Fauntleroy Circus",
  "Mail": "rose75@mail.com"
},
{
  "EmployeeID": 10019,
  "Employees": "King Bergs",
  "Designation": "Developer",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 2,
  "Software": 29,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 28826,
  "Address": "2817 Milton Dr.",
  "Mail": "king57@jourrapide.com"
},
{
  "EmployeeID": 10020,
  "Employees": "Davolio Fuller",
  "Designation": "Designer",
  "Location": "Canada",
  "Status": "Inactive",
  "Trustworthiness": "Sufficient",
  "Rating": 3,
  "Software": 35,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 71035,
  "Address": "Gran Vía, 1",
  "Mail": "davolio29@arpy.com"
},
{
  "EmployeeID": 10021,
  "Employees": "Rose Rose",
  "Designation": "CFO",
  "Location": "Germany",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 3,
  "Software": 38,
  "EmployeeImg": "usermale",
  "CurrentSalary": 68123,
  "Address": "Rua do Mercado, 12",
```

```

      "Mail": "rose54@arpy.com"
    },
    {
      "EmployeeID": 10022,
      "Employees": "Andrew Michael",
      "Designation": "Program Directory",
      "Location": "UK",
      "Status": "Inactive",
      "Trustworthiness": "Insufficient",
      "Rating": 2,
      "Software": 61,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 75470,
      "Address": "2, rue du Commerce",
      "Mail": "andrew88@jourrapide.com"
    },
    {
      "EmployeeID": 10023,
      "Employees": "Davolio Kathryn",
      "Designation": "Manager",
      "Location": "Germany",
      "Status": "Active",
      "Trustworthiness": "Perfect",
      "Rating": 3,
      "Software": 25,
      "EmployeeImg": "usermale",
      "CurrentSalary": 25234,
      "Address": "Hauptstr. 31",
      "Mail": "davolio42@sample.com"
    },
    {
      "EmployeeID": 10024,
      "Employees": "Anne Fleet",
      "Designation": "System Analyst",
      "Location": "UK",
      "Status": "Active",
      "Trustworthiness": "Perfect",
      "Rating": 3,
      "Software": 0,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 8341,
      "Address": "59 rue de l'Abbaye",
      "Mail": "anne86@arpy.com"
    },
    {
      "EmployeeID": 10025,
      "Employees": "Margaret Andrew",
      "Designation": "System Analyst",
      "Location": "Germany",
      "Status": "Inactive",
      "Trustworthiness": "Insufficient",
      "Rating": 3,
      "Software": 51,
      "EmployeeImg": "userfemale",
      "CurrentSalary": 84975,
      "Address": "P.O. Box 555",
      "Mail": "margaret41@arpy.com"
    }
  ]

```

```
},
{
  "EmployeeID": 10026,
  "Employees": "Kathryn Laura",
  "Designation": "Project Lead",
  "Location": "Austria",
  "Status": "Active",
  "Trustworthiness": "Insufficient",
  "Rating": 3,
  "Software": 48,
  "EmployeeImg": "usermale",
  "CurrentSalary": 97282,
  "Address": "Avda. Azteca 123",
  "Mail": "kathryn82@rpy.com"
},
{
  "EmployeeID": 10027,
  "Employees": "Michael Michael",
  "Designation": "Developer",
  "Location": "UK",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 4,
  "Software": 16,
  "EmployeeImg": "usermale",
  "CurrentSalary": 4184,
  "Address": "Rua do Paço, 67",
  "Mail": "michael58@jourrapide.com"
},
{
  "EmployeeID": 10028,
  "Employees": "Leverling Vinet",
  "Designation": "Project Lead",
  "Location": "Germany",
  "Status": "Inactive",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 57,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 38370,
  "Address": "59 rue de l'Abbaye",
  "Mail": "leverling102@sample.com"
},
{
  "EmployeeID": 10029,
  "Employees": "Rose Jack",
  "Designation": "Developer",
  "Location": "UK",
  "Status": "Active",
  "Trustworthiness": "Perfect",
  "Rating": 0,
  "Software": 46,
  "EmployeeImg": "userfemale",
  "CurrentSalary": 84790,
  "Address": "Rua do Mercado, 12",
  "Mail": "rosl08@jourrapide.com"
},
}
```

```
{
  "EmployeeID": 10030,
  "Employees": "Vinet Van",
  "Designation": "Developer",
  "Location": "USA",
  "Status": "Active",
  "Trustworthiness": "Sufficient",
  "Rating": 0,
  "Software": 40,
  "EmployeeImg": "usermale",
  "CurrentSalary": 71005,
  "Address": "Gran Vía, 1",
  "Mail": "vinet90@jourrapide.com"
}
```

### Limitations

Here, we have listed out the limitations with Freeze Panes feature.

- Merging the cells between freeze and unfreeze area.
- If images and charts are added inside the freeze area cells, their portion in the unfreeze area will not move when scrolling.

### Note

You can refer to our [React Spreadsheet](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Spreadsheet example](#) to know how to present and manipulate data.

### See Also

- [Sorting](#)
- [Filtering](#)
- [Undo Redo](#)

### Template in React Spreadsheet component

Cell Template is used for adding HTML elements into Spreadsheet. You can add the cell template in spreadsheet by using the `template` property and specify the address using the `address` property inside the `ranges` property. You can customize the HTML elements similar to Syncfusion components (TextBox, DropDownList, RadioButton, MultiSelect, DatePicker etc) by using the `beforeCellRender` event. In this demo, Cell template is applied to C2:C9 and instantiated with HTML input components like TextBox, RadioButton, TextArea. You need to bind the events to perform any operations through HTML elements or Syncfusion components. Here, we have added `change` event in to the MultiSelect control, and we have updated the selected data into the spreadsheet cell through that change event.

The following sample describes the above behavior.

Sample link: [Cell template](#)

### APP.JSX

```
{% raw %}
import * as React from 'react';
```



```

import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective, CellDirective, CellsDirective
} from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import { RadioButtonComponent, ButtonComponent } from '@syncfusion/ej2-
react-buttons';
import { DropDownListComponent, MultiSelectComponent } from
 '@syncfusion/ej2-react-dropdowns';
function App() {
    const spreadsheetRef = React.useRef(null);
    const scrollSettings = { isFinite: true };
    const selectionSettings = { mode: 'None' };
    const boldCenter = {
        fontWeight: 'bold', textAlign: 'center', fontSize: '12pt',
verticalAlign: 'middle',
        textDecoration: 'underline'
    };
    const nameTextbox = () => {
        return (<TextBoxComponent placeholder="Name"></TextBoxComponent>);
    };
    const dobTextbox = () => {
        return (<TextBoxComponent placeholder="DOB"></TextBoxComponent>);
    };
    const genderRadioButton = () => {
        return (<div>
            <RadioButtonComponent name="gender" value="male"
label="Male"></RadioButtonComponent>
            <RadioButtonComponent name="gender" value="female"
label="Female"></RadioButtonComponent>
        </div>);
    };
    const dropDownList = () => {
        let experience = ['0 - 1 year', '1 - 3 years', '3 - 5 years', '5 -
10 years'];
        return (<DropDownListComponent placeholder="Experience"
dataSource={experience}></DropDownListComponent>);
    };
    const multiSelect = () => {
        let languages = ['JAVA', 'C#', 'SQL'];
        return (<MultiSelectComponent placeholder="Areas of Interest"
showClearButton={false} dataSource={languages}></MultiSelectComponent>);
    };
    const mobileNoTextbox = () => {
        return (<TextBoxComponent placeholder="Mobile
Number"></TextBoxComponent>);
    };
    const emailTextbox = () => {
        return (<TextBoxComponent placeholder="Email"></TextBoxComponent>);
    };
    const textArea = () => {
        return (<TextBoxComponent multiline={true}></TextBoxComponent>);
    };
    const button = () => {

```

```

    return (<ButtonComponent content="Add" style={{ float: "right" }}
    cssClass="e-flat"></ButtonComponent>);
  };
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.cellFormat({ fontWeight: 'bold' }, 'B2:B9');
      // Merges B1 and C1 cells
      spreadsheet.merge('B1:C1');
    }
  }, []);
  return (
    <div className='control-pane'>
      <div className='control-section spreadsheet-control'>
        <SpreadsheetComponent ref={spreadsheetRef}
        showRibbon={false} showFormulaBar={false} allowOpen={false}
        allowSave={false} name='Candidates List' scrollSettings={scrollSettings}
        allowEditing={false} selectionSettings={selectionSettings}>
          <SheetsDirective>
            <SheetDirective name='Registration Form'
            rowCount={40} colCount={30} showGridLines={false}>
              <RowsDirective>
                <RowDirective height={55}>
                  <CellsDirective>
                    <CellDirective index={1}
                    value='Interview Registration Form' style={boldCenter}></CellDirective>
                  </CellsDirective>
                </RowDirective>
                <RowDirective height={45}>
                  <CellsDirective>
                    <CellDirective index={1}
                    value='Name:'></CellDirective>
                  </CellsDirective>
                </RowDirective>
                <RowDirective height={45}>
                  <CellsDirective>
                    <CellDirective index={1} value='Date
of Birth:'></CellDirective>
                  </CellsDirective>
                </RowDirective>
                <RowDirective height={45}>
                  <CellsDirective>
                    <CellDirective index={1}
                    value='Gender:'></CellDirective>
                  </CellsDirective>
                </RowDirective>
                <RowDirective height={45}>
                  <CellsDirective>
                    <CellDirective index={1} value='Year
of Experience:'></CellDirective>
                  </CellsDirective>
                </RowDirective>
                <RowDirective height={45}>
                  <CellsDirective>
                    <CellDirective index={1}
                    value='Areas of Interest:'></CellDirective>
                  </CellsDirective>

```

```

        </RowDirective>
        <RowDirective height={45}>
          <CellsDirective>
            <CellDirective index={1}
value='Mobile Number:'></CellDirective>
          </CellsDirective>
        </RowDirective>
        <RowDirective height={45}>
          <CellsDirective>
            <CellDirective index={1}
value='Email:'></CellDirective>
          </CellsDirective>
        </RowDirective>
        <RowDirective height={82}>
          <CellsDirective>
            <CellDirective index={1}
value='Address:'></CellDirective>
          </CellsDirective>
        </RowDirective>
      </RowsDirective>
      <ColumnsDirective>
        <ColumnDirective index={1}
width={190}></ColumnDirective>
        <ColumnDirective
width={350}></ColumnDirective>
      </ColumnsDirective>
      <RangesDirective>
        <RangeDirective template={nameTextbox}
address='C2'></RangeDirective>
        <RangeDirective template={dobTextbox}
address='C3'></RangeDirective>
        <RangeDirective template={genderRadioButton}
address='C4'></RangeDirective>
        <RangeDirective template={dropDownList}
address='C5'></RangeDirective>
        <RangeDirective template={multiSelect}
address='C6'></RangeDirective>
        <RangeDirective template={mobileNoTextbox}
address='C7'></RangeDirective>
        <RangeDirective template={emailTextbox}
address='C8'></RangeDirective>
        <RangeDirective template={textArea}
address='C9'></RangeDirective>
        <RangeDirective template={button}
address='C11'></RangeDirective>
      </RangesDirective>
    </SheetDirective>
  </SheetsDirective>
</SpreadsheetComponent>
</div>
</div>
);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

**APP.TSX**

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RowsDirective, RowDirective, CellDirective, CellsDirective,
ScrollSettingsModel, CellStyleModel } from '@syncfusion/ej2-react-
spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective,
SelectionSettingsModel } from '@syncfusion/ej2-react-spreadsheet';
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import { RadioButtonComponent, ButtonComponent } from '@syncfusion/ej2-
react-buttons';
import { DropDownListComponent, MultiSelectComponent } from
 '@syncfusion/ej2-react-dropdowns';
function App() {
    const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
    const scrollSettings: ScrollSettingsModel = { isFinite: true };
    const selectionSettings: SelectionSettingsModel = { mode: 'None' };
    const boldCenter: CellStyleModel = {
        fontWeight: 'bold', textAlign: 'center', fontSize: '12pt',
verticalAlign: 'middle',
        textDecoratation: 'underline'
    };
    const nameTextbox = (): JSX.Element => {
        return <TextBoxComponent placeholder="Name"></TextBoxComponent>;
    };
    const dobTextbox = (): JSX.Element => {
        return <TextBoxComponent placeholder="DOB"></TextBoxComponent>;
    };
    const genderRadioButton = (): JSX.Element => {
        return <div>
            <RadioButtonComponent name="gender" value="male"
label="Male"></RadioButtonComponent>
            <RadioButtonComponent name="gender" value="female"
label="Female"></RadioButtonComponent>
        </div>;
    };
    const dropDownList = (): JSX.Element => {
        let experience = ['0 - 1 year', '1 - 3 years', '3 - 5 years', '5 -
10 years'];
        return <DropDownListComponent placeholder="Experience"
dataSource={experience}></DropDownListComponent>;
    };
    const multiSelect = (): JSX.Element => {
        let languages = ['JAVA', 'C#', 'SQL'];
        return <MultiSelectComponent placeholder="Areas of Interest"
showClearButton={false} dataSource={languages}></MultiSelectComponent>;
    };
    const mobileNoTextbox = (): JSX.Element => {
        return <TextBoxComponent placeholder="Mobile
Number"></TextBoxComponent>;
    };
    const emailTextbox = (): JSX.Element => {

```

```

        return (<TextBoxComponent placeholder="Email"></TextBoxComponent>);
    };
    const textArea = (): JSX.Element => {
        return (<TextBoxComponent multiline={true}></TextBoxComponent>);
    };
    const button = (): JSX.Element => {
        return (<ButtonComponent content="Add" style={{ float: "right" }}
cssClass="e-flat"></ButtonComponent>);
    };
    React.useEffect(() => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            spreadsheet.cellFormat({ fontWeight: 'bold' }, 'B2:B9');
            // Merges B1 and C1 cells
            spreadsheet.merge('B1:C1');
        }
    }, []);
    return (
        <div className='control-pane'>
            <div className='control-section spreadsheet-control'>
                <SpreadsheetComponent ref={spreadsheetRef}
showRibbon={false} showFormulaBar={false} allowOpen={false}
allowSave={false} name='Candidates List' scrollSettings={scrollSettings}
allowEditing={false} selectionSettings={selectionSettings}>
                    <SheetsDirective>
                        <SheetDirective name='Registration Form'
rowCount={40} colCount={30} showGridLines={false}>
                            <RowsDirective>
                                <RowDirective height={55}>
                                    <CellsDirective>
                                        <CellDirective index={1}
value='Interview Registration Form' style={boldCenter}></CellDirective>
                                    </CellsDirective>
                                </RowDirective>
                                <RowDirective height={45}>
                                    <CellsDirective>
                                        <CellDirective index={1}
value='Name:'></CellDirective>
                                    </CellsDirective>
                                </RowDirective>
                                <RowDirective height={45}>
                                    <CellsDirective>
                                        <CellDirective index={1} value='Date
of Birth:'></CellDirective>
                                    </CellsDirective>
                                </RowDirective>
                                <RowDirective height={45}>
                                    <CellsDirective>
                                        <CellDirective index={1}
value='Gender:'></CellDirective>
                                    </CellsDirective>
                                </RowDirective>
                                <RowDirective height={45}>
                                    <CellsDirective>
                                        <CellDirective index={1} value='Year
of Experience:'></CellDirective>
                                    </CellsDirective>
                                </RowDirective>
                            </RowsDirective>
                        </SheetDirective>
                    </SheetsDirective>
                </div>
            </div>
        </div>
    );

```

```

        </RowDirective>
        <RowDirective height={45}>
          <CellsDirective>
            <CellDirective index={1}
value='Areas of Interest:'></CellDirective>
          </CellsDirective>
        </RowDirective>
        <RowDirective height={45}>
          <CellsDirective>
            <CellDirective index={1}
value='Mobile Number:'></CellDirective>
          </CellsDirective>
        </RowDirective>
        <RowDirective height={45}>
          <CellsDirective>
            <CellDirective index={1}
value='Email:'></CellDirective>
          </CellsDirective>
        </RowDirective>
        <RowDirective height={82}>
          <CellsDirective>
            <CellDirective index={1}
value='Address:'></CellDirective>
          </CellsDirective>
        </RowDirective>
      </RowsDirective>
    </ColumnsDirective>
    <ColumnDirective index={1}
width={190}></ColumnDirective>
    <ColumnDirective
width={350}></ColumnDirective>
  </ColumnsDirective>
  <RangesDirective>
    <RangeDirective template={nameTextbox}
address='C2'></RangeDirective>
    <RangeDirective template={dobTextbox}
address='C3'></RangeDirective>
    <RangeDirective template={genderRadioButton}
address='C4'></RangeDirective>
    <RangeDirective template={dropDownList}
address='C5'></RangeDirective>
    <RangeDirective template={multiSelect}
address='C6'></RangeDirective>
    <RangeDirective template={mobileNoTextbox}
address='C7'></RangeDirective>
    <RangeDirective template={emailTextbox}
address='C8'></RangeDirective>
    <RangeDirective template={textArea}
address='C9'></RangeDirective>
    <RangeDirective template={button}
address='C11'></RangeDirective>
  </RangesDirective>
</SheetDirective>
</SheetsDirective>
</SpreadsheetComponent>
</div>
</div>

```

```

    );
  };
  export default App;
  const root = createRoot(document.getElementById('root'));
  root.render(<App />);
  {% enddraw %}

```

### Note

You can refer to our [React Spreadsheet](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Spreadsheet example](#) to know how to present and manipulate data.

### See Also

- [Worksheet](#)
- [Rows and columns](#)

### Use Cases

#### Collaborative editing in React Spreadsheet component

The collaborative editing support allows you to work at a spreadsheet collaboratively with other users. Multiple users can access to the the same spreadsheet simultaneously.

#### Dependencies

The following list of dependencies are required to use the collaborative editing support in spreadsheet.

```

`ts
|-- @aspnet/signalr
|-- eventsource
|-- request
|-- ws
`

```

\* Run the command `npm i @aspnet/signalr` to install `@aspnet/signalr` packages in your application.

#### Client configuration

To broadcast the data for every action in the spreadsheet, you need to transfer the data to the server through `send` method in `actionComplete` event and receive the same data by using the `dataReceived` method. In the `dataReceived` method, you need to update the action to the connected clients through `updateAction` method.

The following code example shows `Collaborative Editing` support in the Spreadsheet control.

```

`ts
import * as React from 'react';
import { createRoot } from 'react-dom/client';

import { SpreadsheetComponent, SheetsDirective, SheetDirective, RangesDirective,
CollaborativeEditArgs } from '@syncfusion/ej2-react-spreadsheet';

```

```

import { RangeDirective, ColumnsDirective, ColumnDirective } from '@syncfusion/ej2-react-spreadsheet';
import * as signalR from '@aspnet/signalr';
import { defaultData } from './datasource';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  // For signalR Hub connection
  const connection: signalR.HubConnection = new
  signalR.HubConnectionBuilder().withUrl('https://localhost:44385/hubs/spreadsheethub', { // localhost
  from AspNetCore service
  skipNegotiation: true,
  transport: signalR.HttpTransportType.WebSockets
  }).build();
  const onActionComplete = (args: any) => {
    connection.send('BroadcastData', JSON.stringify(args)); // send the action data to the server
  }
  React.useEffect(() => {
    let spreadsheet = spreadsheetRef.current;
    connection.on('dataReceived', (data: string) => {
      let model: CollaborativeEditArgs = JSON.parse(data) as CollaborativeEditArgs;
      if (spreadsheet) {
        spreadsheet.updateAction(model); // update the action to the connected clients
      }
    });
    connection.start().then(() => { // to start the server
      console.log('server connected!!!');
    }).catch(err => console.log(err));
  }, [connection]); // passing an empty array as second argument triggers the callback in useEffect only
  after the initial render thus replicating componentDidMount lifecycle behaviour
  return (
    <SpreadsheetComponent ref={spreadsheetRef} actionComplete={onActionComplete}>
      <SheetsDirective>
        <SheetDirective>
          <RangesDirective>

```



```

<RangeDirective dataSource={defaultData}></RangeDirective>
</RangesDirective>
<ColumnsDirective>
  <ColumnDirective width={180}></ColumnDirective>
  <ColumnDirective width={130}></ColumnDirective>
  <ColumnDirective width={130}></ColumnDirective>
</ColumnsDirective>
</SheetDirective>
</SheetsDirective>
</SpreadsheetComponent>
);
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
`

```

#### *Server configuration*

To make the communication between the server to the connected clients and from clients to the server, you need to configure the signalR Hubs using the following code.

```

`ts
import * as signalR from '@aspnet/signalr';
// For signalR Hub connection
const connection: signalR.HubConnection = new
signalR.HubConnectionBuilder().withUrl('https://localhost:44385/hubs/spreadsheethub', { // localhost
from AspNetCore service
skipNegotiation: true,
transport: signalR.HttpTransportType.WebSockets
}).build();
`

```

#### *Hub configuration*

Initially create a ASP.NET Core project and add the hub file for sending and receiving the data between server and clients.

```

`ts
using Microsoft.AspNetCore.SignalR;
using System.Threading.Tasks;

```

```

namespace WebApplication.Hubs
{
    public class SpreadsheetHub : Hub
    {
        public async Task BroadcastData(string data)
        {
            await Clients.Others.SendAsync("dataReceived", data);
        }
    }
}

```

To configure the SignalR middleware by registering the following service in the `ConfigureServices` method of the `Startup` class.

```

`ts
services.AddSignalR(e =>
{
    e.MaximumReceiveMessageSize = int.MaxValue; // Option to increase message size for inserting image
    feature. By default, SignalR send messages up to 32 KB.
});

```

To set up the SignalR routes by calling `MapHub` in the `Configure` method of the `Startup` class.

```

`ts
app.UseEndpoints(endpoints =>
{
    endpoints.MapRazorPages();
    endpoints.MapHub<SpreadsheetHub>("/hubs/spreadsheethub");
});

```

For hosting the service, you may use the above code snippet or download and run the [local service](#).

*Prevent the particular action update for collaborative client*

Using the `action` argument from the `actionComplete` event, you can prevent the particular action update for collaborative client.

The following code example shows how to prevent collaborative client from updating the `format` action.

```

`ts

```

```

import * as React from 'react';
import { createRoot } from 'react-dom/client';

import { SpreadsheetComponent, SheetsDirective, SheetDirective, RangesDirective,
CollaborativeEditArgs } from '@syncfusion/ej2-react-spreadsheet';

import { RangeDirective, ColumnsDirective, ColumnDirective } from '@syncfusion/ej2-react-spreadsheet';

import * as signalR from '@aspnet/signalr';
import { defaultData } from './datasource';

function App() {

const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);

// For signalR Hub connection

const connection: signalR.HubConnection = new
signalR.HubConnectionBuilder().withUrl('https://localhost:44385/hubs/spreadsheethub', { // localhost
from AspNetCore service

skipNegotiation: true,

transport: signalR.HttpTransportType.WebSockets

}).build();

const onActionComplete = (args: any) => {
if (args.action !== 'format') { // prevent the format action
connection.send('BroadcastData', JSON.stringify(args)); // send the action data to the server
}
}

React.useEffect(() => {
let spreadsheet = spreadsheetRef.current;
connection.on('dataReceived', (data: string) => {
if (spreadsheet) {
let model: CollaborativeEditArgs = JSON.parse(data) as CollaborativeEditArgs;
spreadsheet.updateAction(model); // update the action to the connected clients
}
});

connection.start().then(() => { // to start the server
console.log('server connected!!!');
}).catch(err => console.log(err));

}, [connection]); // passing an empty array as second argument triggers the callback in useEffect only
after the initial render thus replicating componentDidMount lifecycle behaviour

```

```

return (
  <SpreadsheetComponent ref={spreadsheetRef} actionComplete={onActionComplete}>
    <SheetsDirective>
      <SheetDirective>
        <RangesDirective>
          <RangeDirective dataSource={defaultData}></RangeDirective>
        </RangesDirective>
        <ColumnsDirective>
          <ColumnDirective width={180}></ColumnDirective>
          <ColumnDirective width={130}></ColumnDirective>
          <ColumnDirective width={130}></ColumnDirective>
        </ColumnsDirective>
      </SheetDirective>
    </SheetsDirective>
  </SpreadsheetComponent>
);
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
`

```

#### *Perform import action for collaborative clients*

Using the `action` argument from the `actionComplete` event, you can identify whether the import action is performed or not. If the action is `import`, then you need to send the `response data` to the server and also update the same to the collaborative clients.

The following code example shows how to perform the import functionality for collaborative clients.

```

`ts
import * as React from 'react';
import { isNullOrUndefined } from "@syncfusion/ej2-base";
import { SpreadsheetComponent, CollaborativeEditArgs } from '@syncfusion/ej2-react-spreadsheet';
import * as signalR from "@microsoft/signalr";
import { useEffect, useRef } from "react";
import "./App.css";
function App() {

```

```
const height: string = "400px";
const connection: signalR.HubConnection = new
signalR.HubConnectionBuilder().withUrl(https://localhost:44385/hubs/spreadsheethub, {
skipNegotiation: true,
transport: signalR.HttpTransportType.WebSockets
}).build();
const onActionComplete = (args: {action: string, response: any}): void => {
if (args.action === 'import') {
// Send the action data to the server in args.response at the time of importing an excel file.
connection.send("BroadcastData", JSON.stringify(args.response.data));
}
else {
// Send the action data to the server for other than import actions.
connection.send("BroadcastData", JSON.stringify(args));
}
};
const editor = useRef<SpreadsheetComponent>(null);
useEffect(() => {
connection.on("dataReceived", (data: string) => {
const model: CollaborativeEditArgs = JSON.parse(
data
) as CollaborativeEditArgs;
// Condition to check whether action performed is import.
if (isNullOrUndefined(model['action'])) {
// Load the imported excel file data as JSON to the connected clients.
const jsonData: object = { Workbook: model };
editor.current!.openFromJson({ file: jsonData });
}
else {
// Update the action details to the connected clients.
editor.current!.updateAction(model);
}
});
});
```

```

connection
.start()
.then(() => {
console.log("server connected!!!");
})
.catch((err) => console.log(err));
}, [connection]);
return (
<SpreadsheetComponent
ref={editor}
height={height}
openUrl="https://services.syncfusion.com/react/production/api/spreadsheet/open"
actionComplete={onActionComplete}
/>
);
};
export default App;
`

```

*See Also*

- [Filtering](#)
- [Sorting](#)
- [Hyperlink](#)

## How To

Sort a range by custom list in React Spreadsheet component

You can also define the sorting of cell values based on your own customized personal list. In this article, custom list is achieved using **custom sort comparer**.

In the following demo, the **Trustworthiness** column is sorted based on the custom lists **Perfect**, **Sufficient**, and **Insufficient**.

### APP.JSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { DataUtil } from '@syncfusion/ej2-data';

```

```

import { tradeData } from './datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  const mySortComparer = (x, y) => {
    // custom sort comparer to sort based on the custom list.
    let customList = ['Perfect', 'Sufficient', 'Insufficient'];
    let comparer = DataUtil.fnSort('Ascending');
    return comparer(x ? customList.indexOf(x.value) : x, y ?
customList.indexOf(y.value) : y);
  };
  const onDataBound = () => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet && spreadsheet.activeSheetIndex === 0) {
      spreadsheet.sort({ sortDescriptors: { field: 'F', sortComparer:
mySortComparer }, containsHeader: true }, 'A1:H20');
    }
  };
  const onSortComplete = (args) => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.selectRange(args.range);
    }
  };
  return (
    <SpreadsheetComponent ref={spreadsheetRef} dataBound={onDataBound}
sortComplete={onSortComplete}>
      <SheetsDirective>
        <SheetDirective>
          <RangesDirective>
            <RangeDirective
dataSource={tradeData}></RangeDirective>
          </RangesDirective>
          <ColumnsDirective>
            <ColumnDirective width={90}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
          </ColumnsDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, CellModel, SortEventArgs } from '@syncfusion/ej2-react-
spreadsheet';

```

```

import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { DataUtil } from '@syncfusion/ej2-data';
import { tradeData } from './datasource';
function App() {
    const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
    const mySortComparer = (x: CellModel, y: CellModel): number => {
        // custom sort comparer to sort based on the custom list.
        let customList: string[] = ['Perfect', 'Sufficient',
'Insufficient'];
        let comparer: Function = DataUtil.fnSort('Ascending');
        return comparer(x ? customList.indexOf(x.value as string) : x, y ?
customList.indexOf(y.value as string) : y);
    };
    const onDataBound = (): void => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet && spreadsheet.activeSheetIndex === 0) {
            spreadsheet.sort({ sortDescriptors: { field: 'F', sortComparer:
mySortComparer }, containsHeader: true }, 'A1:H20');
        }
    };
    const onSortComplete = (args: SortEventArgs): void => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            spreadsheet.selectRange(args.range as string);
        }
    };
    return (
        <SpreadsheetComponent ref={spreadsheetRef} dataBound={onDataBound}
sortComplete={onSortComplete}>
            <SheetsDirective>
                <SheetDirective>
                    <RangesDirective>
                        <RangeDirective
dataSource={tradeData}></RangeDirective>
                    </RangesDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={90}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
            </SheetsDirective>
        </SpreadsheetComponent>
    );
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

### DATASOURCE.JSX

```

/**
 * Default data source
 */

```



```

export let defaultData = [
  { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
    Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
  { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
    Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
  { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
    Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
  { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54 AM',
    Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
  { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time: '12:43:59 AM',
    Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit: 70 },
  { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
    Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
  { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
    Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
  { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
    Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
  { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
    Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
  { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
    Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
/**
 * Grid datasource
 */
export function getTradeData(dataCount) {
  let employees = [
    'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
    'Fuller', 'Leverling', 'Peacock',
    'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
    'Dodsworth', 'Anne',
    'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
    'Jack', 'Rose'
  ];
  let designation = ['Manager', 'CFO', 'Designer', 'Developer', 'Program
Directory', 'System Analyst', 'Project Lead'];
  let mail = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
'jourrapide.com'];
  let location = ['UK', 'USA', 'Sweden', 'France', 'Canada', 'Argentina',
'Austria', 'Germany', 'Mexico'];
  let status = ['Active', 'Inactive'];
  let trustworthiness = ['Perfect', 'Sufficient', 'Insufficient'];
  let tradeData = [];
  let address = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do Paço, 67',
'2, rue du Commerce', 'Boulevard Tirou, 255',
'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
Mercado, 12',
'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',

```

```

    'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
    'Taucherstraße 10', 'Taucherstraße 10',
    'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
    Circus', 'Av. dos Lusíadas, 23',
    'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
    123', '2817 Milton Dr.'];
    let employeeimg = ['usermale', 'userfemale'];
    if (typeof dataCount === 'string') {
        dataCount = parseInt(dataCount, 10);
    }
    for (let i = 1; i <= dataCount; i++) {
        let code = 10000;
        tradeData.push({
            'EmployeeID': code + i,
            'Employees': employees[Math.floor(Math.random() *
employees.length)] + ' ' + employees[Math.floor(Math.random() *
employees.length)],
            'Designation': designation[Math.floor(Math.random() *
designation.length)],
            'Location': location[Math.floor(Math.random() *
location.length)],
            'Status': status[Math.floor(Math.random() * status.length)],
            'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
            'Rating': Math.floor(Math.random() * 5),
            'Software': Math.floor(Math.random() * 100),
            'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
            'CurrentSalary': Math.floor((Math.random() * 100000)),
            'Address': address[Math.floor(Math.random() * address.length)],
        });
        let employee = 'Employees';
        let emp = tradeData[i - 1][employee];
        let sName = emp.substr(0, emp.indexOf(' ')).toLowerCase();
        let empmail = 'Mail';
        tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
    }
    return tradeData;
}
export let tradeData = [
    {
        "EmployeeID": 10001,
        "Employees": "Laura Nancy",
        "Designation": "Designer",
        "Location": "France",
        "Status": "Inactive",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
        "Software": 69,
        "EmployeeImg": "usermale",
        "CurrentSalary": 84194,
        "Address": "Taucherstraße 10",
        "Mail": "laura15@jourrapide.com"
    },
    {
        "EmployeeID": 10002,

```

```

    "Employees": "Zachery Van",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 99,
    "EmployeeImg": "usermale",
    "CurrentSalary": 55349,
    "Address": "5ª Ave. Los Palos Grandes",
    "Mail": "zachery109@sample.com"
  },
  {
    "EmployeeID": 10003,
    "Employees": "Rose Fuller",
    "Designation": "CFO",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 1,
    "EmployeeImg": "usermale",
    "CurrentSalary": 16477,
    "Address": "2817 Milton Dr.",
    "Mail": "rose55@rpy.com"
  },
  {
    "EmployeeID": 10004,
    "Employees": "Jack Bergs",
    "Designation": "Manager",
    "Location": "Mexico",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 36,
    "EmployeeImg": "usermale",
    "CurrentSalary": 49040,
    "Address": "2, rue du Commerce",
    "Mail": "jack30@sample.com"
  },
  {
    "EmployeeID": 10005,
    "Employees": "Vinet Bergs",
    "Designation": "Program Directory",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 39,
    "EmployeeImg": "usermale",
    "CurrentSalary": 5495,
    "Address": "Rua da Panificadora, 12",
    "Mail": "vinet32@jourrapide.com"
  },
  {
    "EmployeeID": 10006,
    "Employees": "Buchanan Van",

```

```

    "Designation": "Designer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 42182,
    "Address": "24, place Kléber",
    "Mail": "buchanan18@mail.com"
  },
  {
    "EmployeeID": 10007,
    "Employees": "Dodsworth Nancy",
    "Designation": "Project Lead",
    "Location": "USA",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 35776,
    "Address": "Rua do Paço, 67",
    "Mail": "dodsworth84@mail.com"
  },
  {
    "EmployeeID": 10008,
    "Employees": "Laura Jack",
    "Designation": "Developer",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 89,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25108,
    "Address": "Rua da Panificadora, 12",
    "Mail": "laura82@mail.com"
  },
  {
    "EmployeeID": 10009,
    "Employees": "Anne Fuller",
    "Designation": "Program Directory",
    "Location": "Mexico",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 32568,
    "Address": "Gran Vía, 1",
    "Mail": "anne97@jourrapide.com"
  },
  {
    "EmployeeID": 10010,
    "Employees": "Buchanan Andrew",
    "Designation": "Designer",

```

```

        "Location": "Austria",
        "Status": "Active",
        "Trustworthiness": "Insufficient",
        "Rating": 1,
        "Software": 62,
        "EmployeeImg": "userfemale",
        "CurrentSalary": 12320,
        "Address": "P.O. Box 555",
        "Mail": "buchanan50@jourrapide.com"
    },
    {
        "EmployeeID": 10011,
        "Employees": "Andrew Janet",
        "Designation": "System Analyst",
        "Location": "Germany",
        "Status": "Active",
        "Trustworthiness": "Insufficient",
        "Rating": 3,
        "Software": 8,
        "EmployeeImg": "userfemale",
        "CurrentSalary": 20890,
        "Address": "Starenweg 5",
        "Mail": "andrew63@mail.com"
    },
    {
        "EmployeeID": 10012,
        "Employees": "Margaret Tamer",
        "Designation": "System Analyst",
        "Location": "Germany",
        "Status": "Active",
        "Trustworthiness": "Insufficient",
        "Rating": 4,
        "Software": 7,
        "EmployeeImg": "userfemale",
        "CurrentSalary": 22337,
        "Address": "Magazinweg 7",
        "Mail": "margaret26@mail.com"
    },
    {
        "EmployeeID": 10013,
        "Employees": "Tamer Fuller",
        "Designation": "CFO",
        "Location": "Canada",
        "Status": "Active",
        "Trustworthiness": "Insufficient",
        "Rating": 3,
        "Software": 78,
        "EmployeeImg": "usermale",
        "CurrentSalary": 89181,
        "Address": "Taucherstraße 10",
        "Mail": "tamer40@arpy.com"
    },
    {
        "EmployeeID": 10014,
        "Employees": "Tamer Anne",
        "Designation": "CFO",
        "Location": "Sweden",

```

```

        "Status": "Active",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
        "Software": 18,
        "EmployeeImg": "usermale",
        "CurrentSalary": 20998,
        "Address": "Taucherstraße 10",
        "Mail": "tamer68@arpy.com"
    },
    {
        "EmployeeID": 10015,
        "Employees": "Anton Davolio",
        "Designation": "Project Lead",
        "Location": "France",
        "Status": "Active",
        "Trustworthiness": "Sufficient",
        "Rating": 4,
        "Software": 8,
        "EmployeeImg": "userfemale",
        "CurrentSalary": 48232,
        "Address": "Luisenstr. 48",
        "Mail": "anton46@mail.com"
    },
    {
        "EmployeeID": 10016,
        "Employees": "Buchanan Buchanan",
        "Designation": "System Analyst",
        "Location": "Austria",
        "Status": "Inactive",
        "Trustworthiness": "Perfect",
        "Rating": 0,
        "Software": 19,
        "EmployeeImg": "usermale",
        "CurrentSalary": 43041,
        "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
        "Mail": "buchanan68@mail.com"
    },
    {
        "EmployeeID": 10017,
        "Employees": "King Buchanan",
        "Designation": "Program Directory",
        "Location": "Sweden",
        "Status": "Active",
        "Trustworthiness": "Sufficient",
        "Rating": 0,
        "Software": 44,
        "EmployeeImg": "userfemale",
        "CurrentSalary": 25259,
        "Address": "Magazinweg 7",
        "Mail": "king80@jourrapide.com"
    },
    {
        "EmployeeID": 10018,
        "Employees": "Rose Michael",
        "Designation": "Project Lead",
        "Location": "Canada",
        "Status": "Active",

```

```

    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 31,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 91156,
    "Address": "Fauntleroy Circus",
    "Mail": "rose75@mail.com"
  },
  {
    "EmployeeID": 10019,
    "Employees": "King Bergs",
    "Designation": "Developer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 2,
    "Software": 29,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 28826,
    "Address": "2817 Milton Dr.",
    "Mail": "king57@jourrapide.com"
  },
  {
    "EmployeeID": 10020,
    "Employees": "Davolio Fuller",
    "Designation": "Designer",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 3,
    "Software": 35,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 71035,
    "Address": "Gran Vía, 1",
    "Mail": "davolio29@arpy.com"
  },
  {
    "EmployeeID": 10021,
    "Employees": "Rose Rose",
    "Designation": "CFO",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 38,
    "EmployeeImg": "usermale",
    "CurrentSalary": 68123,
    "Address": "Rua do Mercado, 12",
    "Mail": "rose54@arpy.com"
  },
  {
    "EmployeeID": 10022,
    "Employees": "Andrew Michael",
    "Designation": "Program Directory",
    "Location": "UK",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",

```

```

    "Rating": 2,
    "Software": 61,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 75470,
    "Address": "2, rue du Commerce",
    "Mail": "andrew88@jourrapide.com"
  },
  {
    "EmployeeID": 10023,
    "Employees": "Davolio Kathryn",
    "Designation": "Manager",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 25,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25234,
    "Address": "Hauptstr. 31",
    "Mail": "davolio42@sample.com"
  },
  {
    "EmployeeID": 10024,
    "Employees": "Anne Fleet",
    "Designation": "System Analyst",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 8341,
    "Address": "59 rue de l'Abbaye",
    "Mail": "anne86@arpy.com"
  },
  {
    "EmployeeID": 10025,
    "Employees": "Margaret Andrew",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 51,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 84975,
    "Address": "P.O. Box 555",
    "Mail": "margaret41@arpy.com"
  },
  {
    "EmployeeID": 10026,
    "Employees": "Kathryn Laura",
    "Designation": "Project Lead",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,

```



```
    "Software": 48,  
    "EmployeeImg": "usermale",  
    "CurrentSalary": 97282,  
    "Address": "Avda. Azteca 123",  
    "Mail": "kathryn82@rpy.com"  
  },  
  {  
    "EmployeeID": 10027,  
    "Employees": "Michael Michael",  
    "Designation": "Developer",  
    "Location": "UK",  
    "Status": "Inactive",  
    "Trustworthiness": "Perfect",  
    "Rating": 4,  
    "Software": 16,  
    "EmployeeImg": "usermale",  
    "CurrentSalary": 4184,  
    "Address": "Rua do Paço, 67",  
    "Mail": "michael58@jourrapide.com"  
  },  
  {  
    "EmployeeID": 10028,  
    "Employees": "Leverling Vinet",  
    "Designation": "Project Lead",  
    "Location": "Germany",  
    "Status": "Inactive",  
    "Trustworthiness": "Perfect",  
    "Rating": 0,  
    "Software": 57,  
    "EmployeeImg": "userfemale",  
    "CurrentSalary": 38370,  
    "Address": "59 rue de l'Abbaye",  
    "Mail": "leverling102@sample.com"  
  },  
  {  
    "EmployeeID": 10029,  
    "Employees": "Rose Jack",  
    "Designation": "Developer",  
    "Location": "UK",  
    "Status": "Active",  
    "Trustworthiness": "Perfect",  
    "Rating": 0,  
    "Software": 46,  
    "EmployeeImg": "userfemale",  
    "CurrentSalary": 84790,  
    "Address": "Rua do Mercado, 12",  
    "Mail": "rosl08@jourrapide.com"  
  },  
  {  
    "EmployeeID": 10030,  
    "Employees": "Vinet Van",  
    "Designation": "Developer",  
    "Location": "USA",  
    "Status": "Active",  
    "Trustworthiness": "Sufficient",  
    "Rating": 0,  
    "Software": 40,
```

```

        "EmployeeImg": "usermale",
        "CurrentSalary": 71005,
        "Address": "Gran Vía, 1",
        "Mail": "vinet90@jourrapide.com"
    }
};

```

### DATASOURCE.TSX

```

/**
 * Default data source
 */
export let defaultData: Object[] = [
    { 'Item Name': 'Casual Shoes', Date: '02/14/2014', Time: '11:34:32 AM',
      Quantity: 10, Price: 20, Amount: 200, Discount: 1, Profit: 10 },
    { 'Item Name': 'Sports Shoes', Date: '06/11/2014', Time: '05:56:32 AM',
      Quantity: 20, Price: 30, Amount: 600, Discount: 5, Profit: 50 },
    { 'Item Name': 'Formal Shoes', Date: '07/27/2014', Time: '03:32:44 AM',
      Quantity: 20, Price: 15, Amount: 300, Discount: 7, Profit: 27 },
    { 'Item Name': 'Sandals & Floaters', Date: '11/21/2014', Time: '06:23:54 AM',
      Quantity: 15, Price: 20, Amount: 300, Discount: 11, Profit: 67 },
    { 'Item Name': 'Flip- Flops & Slippers', Date: '06/23/2014', Time: '12:43:59 AM',
      Quantity: 30, Price: 10, Amount: 300, Discount: 10, Profit: 70 },
    { 'Item Name': 'Sneakers', Date: '07/22/2014', Time: '10:55:53 AM',
      Quantity: 40, Price: 20, Amount: 800, Discount: 13, Profit: 66 },
    { 'Item Name': 'Running Shoes', Date: '02/04/2014', Time: '03:44:34 AM',
      Quantity: 20, Price: 10, Amount: 200, Discount: 3, Profit: 14 },
    { 'Item Name': 'Loafers', Date: '11/30/2014', Time: '03:12:52 AM',
      Quantity: 31, Price: 10, Amount: 310, Discount: 6, Profit: 29 },
    { 'Item Name': 'Cricket Shoes', Date: '07/09/2014', Time: '11:32:14 AM',
      Quantity: 41, Price: 30, Amount: 1210, Discount: 12, Profit: 166 },
    { 'Item Name': 'T-Shirts', Date: '10/31/2014', Time: '12:01:44 AM',
      Quantity: 50, Price: 10, Amount: 500, Discount: 9, Profit: 55 },
];
/**
 * Grid datasource
 */
export function getTradeData(dataCount?: number): object {
    let employees: string[] = [
        'Michael', 'Kathryn', 'Tamer', 'Martin', 'Davolio', 'Nancy',
        'Fuller', 'Leverling', 'Peacock',
        'Margaret', 'Buchanan', 'Janet', 'Andrew', 'Callahan', 'Laura',
        'Dodsworth', 'Anne',
        'Bergs', 'Vinet', 'Anton', 'Fleet', 'Zachery', 'Van', 'King',
        'Jack', 'Rose'];
    let designation: string[] = ['Manager', 'CFO', 'Designer', 'Developer',
        'Program Directory', 'System Analyst', 'Project Lead'];
    let mail: string[] = ['sample.com', 'arpy.com', 'rpy.com', 'mail.com',
        'jourrapide.com'];
    let location: string[] = ['UK', 'USA', 'Sweden', 'France', 'Canada',
        'Argentina', 'Austria', 'Germany', 'Mexico'];
    let status: string[] = ['Active', 'Inactive'];
    let trustworthiness: string[] = ['Perfect', 'Sufficient',
        'Insufficient'];
    let tradeData: Object[] = [];

```

```

let address: string[] = ['59 rue de l'Abbaye', 'Luisenstr. 48', 'Rua do
Paço, 67', '2, rue du Commerce', 'Boulevard Tirou, 255',
'Rua do mailPaço, 67', 'Hauptstr. 31', 'Starenweg 5', 'Rua do
Mercado, 12',
'Carrera 22 con Ave. Carlos Soublette #8-35', 'Kirchgasse 6',
'Sierras de Granada 9993', 'Mehrheimerstr. 369', 'Rua da
Panificadora, 12', '2817 Milton Dr.', 'Kirchgasse 6',
'Åkergatan 24', '24, place Kléber', 'Torikatu 38', 'Berliner Platz
43', '5ª Ave. Los Palos Grandes', '1029 - 12th Ave. S.',
'Torikatu 38', 'P.O. Box 555', '2817 Milton Dr.', 'Taucherstraße
10', '59 rue de l'Abbaye', 'Via Ludovico il Moro 22',
'Avda. Azteca 123', 'Heerstr. 22', 'Berguvsvägen 8', 'Magazinweg
7', 'Berguvsvägen 8', 'Gran Vía, 1', 'Gran Vía, 1',
'Carrera 52 con Ave. Bolívar #65-98 Llano Largo', 'Magazinweg 7',
'Taucherstraße 10', 'Taucherstraße 10',
'Av. Copacabana, 267', 'Strada Provinciale 124', 'Fauntleroy
Circus', 'Av. dos Lusíadas, 23',
'Rua da Panificadora, 12', 'Av. Inês de Castro, 414', 'Avda. Azteca
123', '2817 Milton Dr.'];
let employeeimg: string[] = ['usermale', 'userfemale'];
if (typeof dataCount === 'string') {
    dataCount = parseInt(dataCount, 10);
}
for (let i: number = 1; i <= dataCount; i++) {
    let code: any = 10000;
    tradeData.push({
        'EmployeeID': code + i,
        'Employees':
            employees[Math.floor(Math.random() * employees.length)] + '
' + employees[Math.floor(Math.random() * employees.length)],
        'Designation': designation[Math.floor(Math.random() *
designation.length)],
        'Location': location[Math.floor(Math.random() *
location.length)],
        'Status': status[Math.floor(Math.random() * status.length)],
        'Trustworthiness': trustworthiness[Math.floor(Math.random() *
trustworthiness.length)],
        'Rating': Math.floor(Math.random() * 5),
        'Software': Math.floor(Math.random() * 100),
        'EmployeeImg': employeeimg[Math.floor(Math.random() *
employeeimg.length)],
        'CurrentSalary': Math.floor((Math.random() * 100000)),
        'Address': address[Math.floor(Math.random() * address.length)],
    });
    let employee: string = 'Employees';
    let emp: string = tradeData[i - 1][employee];
    let sName: string = emp.substr(0, emp.indexOf(' ')).toLowerCase();
    let empmail: string = 'Mail';
    tradeData[i - 1][empmail] = sName + (Math.floor(Math.random() * 100)
+ 10) + '@' + mail[Math.floor(Math.random() * mail.length)];
}
return tradeData;
}
export let tradeData: Object[] = [
    {
        "EmployeeID": 10001,
        "Employees": "Laura Nancy",
    }
]

```

```

    "Designation": "Designer",
    "Location": "France",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 69,
    "EmployeeImg": "usermale",
    "CurrentSalary": 84194,
    "Address": "Taucherstraße 10",
    "Mail": "laura15@jourrapide.com"
  },
  {
    "EmployeeID": 10002,
    "Employees": "Zachery Van",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 99,
    "EmployeeImg": "usermale",
    "CurrentSalary": 55349,
    "Address": "5a Ave. Los Palos Grandes",
    "Mail": "zachery109@sample.com"
  },
  {
    "EmployeeID": 10003,
    "Employees": "Rose Fuller",
    "Designation": "CFO",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 1,
    "EmployeeImg": "usermale",
    "CurrentSalary": 16477,
    "Address": "2817 Milton Dr.",
    "Mail": "rose55@rpy.com"
  },
  {
    "EmployeeID": 10004,
    "Employees": "Jack Bergs",
    "Designation": "Manager",
    "Location": "Mexico",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 36,
    "EmployeeImg": "usermale",
    "CurrentSalary": 49040,
    "Address": "2, rue du Commerce",
    "Mail": "jack30@sample.com"
  },
  {
    "EmployeeID": 10005,
    "Employees": "Vinet Bergs",
    "Designation": "Program Directory",

```

```

    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 39,
    "EmployeeImg": "usermale",
    "CurrentSalary": 5495,
    "Address": "Rua da Panificadora, 12",
    "Mail": "vinet32@jourrapide.com"
  },
  {
    "EmployeeID": 10006,
    "Employees": "Buchanan Van",
    "Designation": "Designer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 42182,
    "Address": "24, place Kléber",
    "Mail": "buchanan18@mail.com"
  },
  {
    "EmployeeID": 10007,
    "Employees": "Dodsworth Nancy",
    "Designation": "Project Lead",
    "Location": "USA",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 35776,
    "Address": "Rua do Paço, 67",
    "Mail": "dodsworth84@mail.com"
  },
  {
    "EmployeeID": 10008,
    "Employees": "Laura Jack",
    "Designation": "Developer",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 89,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25108,
    "Address": "Rua da Panificadora, 12",
    "Mail": "laura82@mail.com"
  },
  {
    "EmployeeID": 10009,
    "Employees": "Anne Fuller",
    "Designation": "Program Directory",
    "Location": "Mexico",

```

```

    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 32568,
    "Address": "Gran Vía, 1",
    "Mail": "anne97@jourrapide.com"
  },
  {
    "EmployeeID": 10010,
    "Employees": "Buchanan Andrew",
    "Designation": "Designer",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 1,
    "Software": 62,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 12320,
    "Address": "P.O. Box 555",
    "Mail": "buchanan50@jourrapide.com"
  },
  {
    "EmployeeID": 10011,
    "Employees": "Andrew Janet",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 20890,
    "Address": "Starenweg 5",
    "Mail": "andrew63@mail.com"
  },
  {
    "EmployeeID": 10012,
    "Employees": "Margaret Tamer",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 4,
    "Software": 7,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 22337,
    "Address": "Magazinweg 7",
    "Mail": "margaret26@mail.com"
  },
  {
    "EmployeeID": 10013,
    "Employees": "Tamer Fuller",
    "Designation": "CFO",
    "Location": "Canada",
    "Status": "Active",

```

```

    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 78,
    "EmployeeImg": "usermale",
    "CurrentSalary": 89181,
    "Address": "Taucherstraße 10",
    "Mail": "tamer40@arpy.com"
  },
  {
    "EmployeeID": 10014,
    "Employees": "Tamer Anne",
    "Designation": "CFO",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 18,
    "EmployeeImg": "usermale",
    "CurrentSalary": 20998,
    "Address": "Taucherstraße 10",
    "Mail": "tamer68@arpy.com"
  },
  {
    "EmployeeID": 10015,
    "Employees": "Anton Davolio",
    "Designation": "Project Lead",
    "Location": "France",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 4,
    "Software": 8,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 48232,
    "Address": "Luisenstr. 48",
    "Mail": "anton46@mail.com"
  },
  {
    "EmployeeID": 10016,
    "Employees": "Buchanan Buchanan",
    "Designation": "System Analyst",
    "Location": "Austria",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 19,
    "EmployeeImg": "usermale",
    "CurrentSalary": 43041,
    "Address": "Carrera 52 con Ave. Bolívar #65-98 Llano Largo",
    "Mail": "buchanan68@mail.com"
  },
  {
    "EmployeeID": 10017,
    "Employees": "King Buchanan",
    "Designation": "Program Directory",
    "Location": "Sweden",
    "Status": "Active",
    "Trustworthiness": "Sufficient",

```

```

    "Rating": 0,
    "Software": 44,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 25259,
    "Address": "Magazinweg 7",
    "Mail": "king80@jourrapide.com"
  },
  {
    "EmployeeID": 10018,
    "Employees": "Rose Michael",
    "Designation": "Project Lead",
    "Location": "Canada",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 31,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 91156,
    "Address": "Fauntleroy Circus",
    "Mail": "rose75@mail.com"
  },
  {
    "EmployeeID": 10019,
    "Employees": "King Bergs",
    "Designation": "Developer",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 2,
    "Software": 29,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 28826,
    "Address": "2817 Milton Dr.",
    "Mail": "king57@jourrapide.com"
  },
  {
    "EmployeeID": 10020,
    "Employees": "Davolio Fuller",
    "Designation": "Designer",
    "Location": "Canada",
    "Status": "Inactive",
    "Trustworthiness": "Sufficient",
    "Rating": 3,
    "Software": 35,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 71035,
    "Address": "Gran Vía, 1",
    "Mail": "davolio29@arpy.com"
  },
  {
    "EmployeeID": 10021,
    "Employees": "Rose Rose",
    "Designation": "CFO",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,

```



```

    "Software": 38,
    "EmployeeImg": "usermale",
    "CurrentSalary": 68123,
    "Address": "Rua do Mercado, 12",
    "Mail": "rose54@arpy.com"
  },
  {
    "EmployeeID": 10022,
    "Employees": "Andrew Michael",
    "Designation": "Program Directory",
    "Location": "UK",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 2,
    "Software": 61,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 75470,
    "Address": "2, rue du Commerce",
    "Mail": "andrew88@jourrapide.com"
  },
  {
    "EmployeeID": 10023,
    "Employees": "Davolio Kathryn",
    "Designation": "Manager",
    "Location": "Germany",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 25,
    "EmployeeImg": "usermale",
    "CurrentSalary": 25234,
    "Address": "Hauptstr. 31",
    "Mail": "davolio42@sample.com"
  },
  {
    "EmployeeID": 10024,
    "Employees": "Anne Fleet",
    "Designation": "System Analyst",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 3,
    "Software": 0,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 8341,
    "Address": "59 rue de l'Abbaye",
    "Mail": "anne86@arpy.com"
  },
  {
    "EmployeeID": 10025,
    "Employees": "Margaret Andrew",
    "Designation": "System Analyst",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 51,

```

```

    "EmployeeImg": "userfemale",
    "CurrentSalary": 84975,
    "Address": "P.O. Box 555",
    "Mail": "margaret41@arpy.com"
  },
  {
    "EmployeeID": 10026,
    "Employees": "Kathryn Laura",
    "Designation": "Project Lead",
    "Location": "Austria",
    "Status": "Active",
    "Trustworthiness": "Insufficient",
    "Rating": 3,
    "Software": 48,
    "EmployeeImg": "usermale",
    "CurrentSalary": 97282,
    "Address": "Avda. Azteca 123",
    "Mail": "kathryn82@rpy.com"
  },
  {
    "EmployeeID": 10027,
    "Employees": "Michael Michael",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 4,
    "Software": 16,
    "EmployeeImg": "usermale",
    "CurrentSalary": 4184,
    "Address": "Rua do Paço, 67",
    "Mail": "michael58@jourrapide.com"
  },
  {
    "EmployeeID": 10028,
    "Employees": "Leverling Vinet",
    "Designation": "Project Lead",
    "Location": "Germany",
    "Status": "Inactive",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 57,
    "EmployeeImg": "userfemale",
    "CurrentSalary": 38370,
    "Address": "59 rue de l'Abbaye",
    "Mail": "leverling102@sample.com"
  },
  {
    "EmployeeID": 10029,
    "Employees": "Rose Jack",
    "Designation": "Developer",
    "Location": "UK",
    "Status": "Active",
    "Trustworthiness": "Perfect",
    "Rating": 0,
    "Software": 46,
    "EmployeeImg": "userfemale",

```

```

    "CurrentSalary": 84790,
    "Address": "Rua do Mercado, 12",
    "Mail": "rose108@jourrapide.com"
  },
  {
    "EmployeeID": 10030,
    "Employees": "Vinet Van",
    "Designation": "Developer",
    "Location": "USA",
    "Status": "Active",
    "Trustworthiness": "Sufficient",
    "Rating": 0,
    "Software": 40,
    "EmployeeImg": "usermale",
    "CurrentSalary": 71005,
    "Address": "Gran Vía, 1",
    "Mail": "vinet90@jourrapide.com"
  }
]

```

*See Also*

- [Filtering](#)
- [Sorting](#)
- [Hyperlink](#)

### Print in React Spreadsheet component

You can use the `print` method by importing from `ej2-base` package. Here, the `Select` event in the dropdown and the `dataBound` event are used to print the single/multiple sheets of data. To print the single/multiple sheets, use the dropdown button and select the `Print` (or) `Print All` option. In the following sample, you can be able to print the single/multiple sheets.

#### APP.JSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { DropDownButtonComponent } from '@syncfusion/ej2-react-
splitbuttons';
import { print } from '@syncfusion/ej2-base';
import { budgetData, salaryData, printElement } from './datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  const items = [
    { text: "Print" },
    { text: "Print All" }
  ];
  let isPrint = false;
  const dataBound = () => {
    let spreadsheet = spreadsheetRef.current;

```

```

        if (spreadsheet && isPrint) {
            printElement.querySelector(".e-sheet-content").innerHTML +=
document.querySelector(".e-sheet-content").outerHTML;
            let usedRange = spreadsheet.getActiveSheet().usedRange;
            let tbody = printElement.querySelector('.e-sheet-
content').children[spreadsheet.activeSheetIndex].querySelector('tbody');
            for (let i = tbody.getElementsByClassName('e-row').length; i >=
0; i--) {
                if (tbody.getElementsByClassName('e-row')[i] &&
parseInt(tbody.getElementsByClassName('e-row')[i].getAttribute('aria-
rowindex')) > usedRange.rowIndex + 1) {
                    tbody.getElementsByClassName('e-row')[i].remove();
                }
            }
            let sheets = spreadsheet.sheets;
            if (sheets.length - 1 === spreadsheet.activeSheetIndex) {
                let count = printElement.querySelector(".e-sheet-
content").childElementCount;
                if (count > 1) {
                    for (let i = 0; i < count; i++) {
                        (printElement.querySelector('.e-sheet-
content').children[i].getElementsByClassName('e-
virtualtrack')[0]).style.height = 'auto';
                        printElement.querySelector('.e-sheet-
content').children[i].setAttribute('style', 'page-break-after: always;')
                    }
                }
                print(printElement);
                isPrint = false;
                printElement.querySelector(".e-sheet-content").innerHTML =
'';
            } else {
                if (sheets[spreadsheet.activeSheetIndex + 1]) {
                    spreadsheet.goTo(sheets[spreadsheet.activeSheetIndex +
1].name + "!A1");
                }
            }
        }
    };
    React.useEffect(() => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A1:D1');
            spreadsheet.cellFormat({ fontWeight: 'bold' }, 'A11:D11');
            spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'Salary!A1:D1');
        }
    }, []);
    const itemSelect = (args) => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            if (args.item.text === 'Print') {
                printElement.querySelector(".e-sheet-content").innerHTML =
document.querySelector(".e-sheet-content").outerHTML; // To add the
spreadsheet table
                let usedRange = spreadsheet.getActiveSheet().usedRange;

```

```

        let tbody = printElement.querySelector('tbody');
        for (let i = tbody.getElementsByClassName('e-row').length; i
    >= 0; i--) {
            if (tbody.getElementsByClassName('e-row')[i] &&
    parseInt(tbody.getElementsByClassName('e-row')[i].getAttribute('aria-
    rowindex')) > usedRange.rowIndex + 1) {
                tbody.getElementsByClassName('e-row')[i].remove();
            }
        }
        (printElement.querySelector('.e-sheet-
    content').children[0].getElementsByClassName('e-
    virtualtrack')[0]).style.height = 'auto';
        print(printElement);
        printElement.querySelector(".e-sheet-content").innerHTML =
    '';
    }
    if (args.item.text === 'Print All') {
        let sheets = spreadsheet.sheets;
        if (spreadsheet.activeSheetIndex === 0) {
            printElement.querySelector(".e-sheet-content").innerHTML
    = document.querySelector(".e-sheet-content").outerHTML; // To add the
    spreadsheet table
            let usedRange = spreadsheet.getActiveSheet().usedRange;
            let tbody = printElement.querySelector('tbody');
            for (let i = tbody.getElementsByClassName('e-
    row').length; i >= 0; i--) {
                if (tbody.getElementsByClassName('e-row')[i] &&
    parseInt(tbody.getElementsByClassName('e-row')[i].getAttribute('aria-
    rowindex')) > usedRange.rowIndex + 1) {
                    tbody.getElementsByClassName('e-
    row')[i].remove();
                }
            }
            if (sheets[spreadsheet.activeSheetIndex + 1]) {
                spreadsheet.goTo(sheets[spreadsheet.activeSheetIndex
    + 1].name + "!A1");
                isPrint = true;
            } else {
                print(printElement);
                printElement.querySelector(".e-sheet-
    content").innerHTML = '';
            }
        } else {
            if (sheets[0]) {
                spreadsheet.goTo(sheets[0].name + "!A1");
                isPrint = true;
            }
        }
    }
};
return (
    <div>
        <DropDownButtonComponent items={items} select={itemSelect}>
    Print </DropDownButtonComponent>
        <SpreadsheetComponent ref={spreadsheetRef} dataBound={dataBound}
    >

```

```

        <SheetsDirective>
          <SheetDirective name={"Budget"}>
            <RangesDirective>
              <RangeDirective
dataSource={budgetData}></RangeDirective>
            </RangesDirective>
            <ColumnsDirective>
              <ColumnDirective width={100}></ColumnDirective>
              <ColumnDirective width={100}></ColumnDirective>
              <ColumnDirective width={100}></ColumnDirective>
              <ColumnDirective width={100}></ColumnDirective>
            </ColumnsDirective>
          </SheetDirective>
          <SheetDirective name={"Salary"}>
            <RangesDirective>
              <RangeDirective
dataSource={salaryData}></RangeDirective>
            </RangesDirective>
            <ColumnsDirective>
              <ColumnDirective width={100}></ColumnDirective>
              <ColumnDirective width={100}></ColumnDirective>
              <ColumnDirective width={100}></ColumnDirective>
              <ColumnDirective width={100}></ColumnDirective>
            </ColumnsDirective>
          </SheetDirective>
        </SheetsDirective>
      </SpreadsheetComponent>
    </div>
  );
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, UsedRangeModel, SheetModel, MenuSelectEventArgs } from
'@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { DropDownButtonComponent, ItemModel } from '@syncfusion/ej2-react-
splitbuttons';
import { print } from '@syncfusion/ej2-base';
import { budgetData, salaryData, printElement } from '../datasource';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  const items: ItemModel[] = [
    { text: "Print" },
    { text: "Print All" }
  ];
  let isPrint: boolean = false;

```

```

const dataBound = (): void => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet && isPrint) {
        printElement.querySelector(".e-sheet-content").innerHTML +=
document.querySelector(".e-sheet-content").outerHTML;
        let usedRange: UsedRangeModel =
spreadsheet.getActiveSheet().usedRange as UsedRangeModel;
        let tbody: HTMLTableSectionElement =
printElement.querySelector('.e-sheet-
content').children[spreadsheet.activeSheetIndex].querySelector('tbody') as
HTMLTableSectionElement;
        for (let i: number = tbody.getElementsByClassName('e-
row').length; i >= 0; i--) {
            if (tbody.getElementsByClassName('e-row')[i] &&
parseInt(tbody.getElementsByClassName('e-row')[i].getAttribute('aria-
rowindex')) > usedRange.rowIndex + 1) {
                tbody.getElementsByClassName('e-row')[i].remove();
            }
        }
        let sheets: SheetModel[] = spreadsheet.sheets;
        if (sheets.length - 1 === spreadsheet.activeSheetIndex) {
            let count: number = printElement.querySelector(".e-sheet-
content").childElementCount;
            if (count > 1) {
                for (let i: number = 0; i < count; i++) {
                    (printElement.querySelector('.e-sheet-
content').children[i].getElementsByClassName('e-virtualtrack')[0] as
HTMLElement).style.height = 'auto';
                    printElement.querySelector('.e-sheet-
content').children[i].setAttribute('style', 'page-break-after: always;')
                }
                print(printElement);
                isPrint = false;
                printElement.querySelector(".e-sheet-content").innerHTML =
'';
            } else {
                if (sheets[spreadsheet.activeSheetIndex + 1]) {
                    spreadsheet.goTo(sheets[spreadsheet.activeSheetIndex +
1].name + "!A1");
                }
            }
        }
    };
    React.useEffect(() => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'A1:D1');
            spreadsheet.cellFormat({ fontWeight: 'bold' }, 'A11:D11');
            spreadsheet.cellFormat({ fontWeight: 'bold', textAlign: 'center'
}, 'Salary!A1:D1');
        }
    }, []);
    const itemSelect = (args: MenuSelectEventArgs): void => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {

```

```

        if (args.item.text === 'Print') {
            printElement.querySelector(".e-sheet-content").innerHTML =
document.querySelector(".e-sheet-content").outerHTML; // To add the
spreadsheet table
            let usedRange: UsedRangeModel =
spreadsheet.getActiveSheet().usedRange as UsedRangeModel;
            let tbody: HTMLTableSectionElement =
printElement.querySelector('tbody') as HTMLTableSectionElement;
            for (let i: number = tbody.getElementsByClassName('e-
row').length; i >= 0; i--) {
                if (tbody.getElementsByClassName('e-row')[i] &&
parseInt(tbody.getElementsByClassName('e-row')[i].getAttribute('aria-
rowindex')) > usedRange.rowIndex + 1) {
                    tbody.getElementsByClassName('e-row')[i].remove();
                }
            }
            (printElement.querySelector('.e-sheet-
content').children[0].getElementsByClassName('e-virtualtrack')[0] as
HTMLElement).style.height = 'auto';
            print(printElement);
            printElement.querySelector(".e-sheet-content").innerHTML =
'';
        }
        if (args.item.text === 'Print All') {
            let sheets: SheetModel[] = spreadsheet.sheets;
            if (spreadsheet.activeSheetIndex === 0) {
                printElement.querySelector(".e-sheet-content").innerHTML
= document.querySelector(".e-sheet-content").outerHTML; // To add the
spreadsheet table
                let usedRange: UsedRangeModel =
spreadsheet.getActiveSheet().usedRange as UsedRangeModel;
                let tbody: HTMLTableSectionElement =
printElement.querySelector('tbody') as HTMLTableSectionElement;
                for (let i: number = tbody.getElementsByClassName('e-
row').length; i >= 0; i--) {
                    if (tbody.getElementsByClassName('e-row')[i] &&
parseInt(tbody.getElementsByClassName('e-row')[i].getAttribute('aria-
rowindex')) > usedRange.rowIndex + 1) {
                        tbody.getElementsByClassName('e-
row')[i].remove();
                    }
                }
                if (sheets[spreadsheet.activeSheetIndex + 1]) {
                    spreadsheet.goTo(sheets[spreadsheet.activeSheetIndex
+ 1].name + "!A1");
                    isPrint = true;
                } else {
                    print(printElement);
                    printElement.querySelector(".e-sheet-
content").innerHTML = '';
                }
            } else {
                if (sheets[0]) {
                    spreadsheet.goTo(sheets[0].name + "!A1");
                    isPrint = true;
                }
            }
        }
    }
}

```



```

    }
  };
  return (
    <div>
      <DropDownButtonComponent items={items} select={itemSelect}>
Print </DropDownButtonComponent>
      <SpreadsheetComponent ref={spreadsheetRef} dataBound={dataBound}
>
        <SheetsDirective>
          <SheetDirective name={"Budget"}>
            <RangesDirective>
              <RangeDirective
dataSource={budgetData}></RangeDirective>
            </RangesDirective>
            <ColumnsDirective>
              <ColumnDirective width={100}></ColumnDirective>
              <ColumnDirective width={100}></ColumnDirective>
              <ColumnDirective width={100}></ColumnDirective>
              <ColumnDirective width={100}></ColumnDirective>
            </ColumnsDirective>
          </SheetDirective>
          <SheetDirective name={"Salary"}>
            <RangesDirective>
              <RangeDirective
dataSource={salaryData}></RangeDirective>
            </RangesDirective>
            <ColumnsDirective>
              <ColumnDirective width={100}></ColumnDirective>
              <ColumnDirective width={100}></ColumnDirective>
              <ColumnDirective width={100}></ColumnDirective>
              <ColumnDirective width={100}></ColumnDirective>
            </ColumnsDirective>
          </SheetDirective>
        </SheetsDirective>
      </SpreadsheetComponent>
    </div>
  );
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

### DATASOURCE.JSX

```

import { createElement } from '@syncfusion/ej2-base';
export let budgetData = [
  {
    "Expense Type": "Housing",
    "Projected Cost": 7000,
    "Actual Cost": 7500,
    "Difference": -500,
  },
  {
    "Expense Type": "Transportation",

```

```

        "Projected Cost": 500,
        "Actual Cost": 500,
        "Difference": 0,
    },
    {
        "Expense Type": "Insurance",
        "Projected Cost": 1000,
        "Actual Cost": 1000,
        "Difference": 0,
    },
    {
        "Expense Type": "Food",
        "Projected Cost": 2000,
        "Actual Cost": 1800,
        "Difference": 200,
    },
    {
        "Expense Type": "Pets",
        "Projected Cost": 300,
        "Actual Cost": 200,
        "Difference": 100,
    },
    {
        "Expense Type": "Personel Care",
        "Projected Cost": 500,
        "Actual Cost": 500,
        "Difference": 0,
    }, {
        "Expense Type": "Loan",
        "Projected Cost": 1000,
        "Actual Cost": 1000,
        "Difference": 0,
    }, {
        "Expense Type": "Tax",
        "Projected Cost": 200,
        "Actual Cost": 200,
        "Difference": 0,
    }, {
        "Expense Type": "Savings",
        "Projected Cost": 1000,
        "Actual Cost": 900,
        "Difference": 100,
    },
    {
        "Expense Type": "Total",
        "Projected Cost": 13500,
        "Actual Cost": 13600,
        "Difference": -100,
    }
];
export let salaryData = [
    {
        "Earnings": "Basic",
        "Credit Amount": 20000,
        "Deductions": "Provident Fund",
        "Debit Amount": 2400,
    },

```

```

    {
      "Earnings": "HRA",
      "Credit Amount": 8000,
      "Deductions": "ESI",
      "Debit Amount": 0,
    },
    {
      "Earnings": "Special Allowance",
      "Credit Amount": 25000,
      "Deductions": "Professional Tax",
      "Debit Amount": 200,
    },
    {
      "Earnings": "Incentives",
      "Credit Amount": 2000,
      "Deductions": "TDS",
      "Debit Amount": 2750,
    },
    {
      "Earnings": "Bonus",
      "Credit Amount": 1500,
      "Deductions": "Other Deduction",
      "Debit Amount": 0,
    },
    {
      "Earnings": "Total Earnings",
      "Credit Amount": 56500,
      "Deductions": "Total Deductions",
      "Debit Amount": 5350,
    }
  ];
  export let printElement = createElement("div", {
    className: "e-sheet-panel",
    innerHTML: '<div class="e-spreadsheet-print"></div><div class="e-sheet"><div class="e-main-panel style="height:100%" style="overflow:unset"><div class="e-sheet-content" ></div></div></div>'
  }); // creating same hierarchy of element as DOM

```

## DATASOURCE.TSX

```

import { createElement, getComponent } from '@syncfusion/ej2-base';
import { SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
export let budgetData: Object[] = [
  {
    "Expense Type": "Housing",
    "Projected Cost": 7000,
    "Actual Cost": 7500,
    "Difference": -500,
  },
  {
    "Expense Type": "Transportation",
    "Projected Cost": 500,
    "Actual Cost": 500,
    "Difference": 0,
  },
  {

```

```

    "Expense Type": "Insurance",
    "Projected Cost": 1000,
    "Actual Cost": 1000,
    "Difference": 0,
  },
  {
    "Expense Type": "Food",
    "Projected Cost": 2000,
    "Actual Cost": 1800,
    "Difference": 200,
  },
  {
    "Expense Type": "Pets",
    "Projected Cost": 300,
    "Actual Cost": 200,
    "Difference": 100,
  },
  {
    "Expense Type": "Personel Care",
    "Projected Cost": 500,
    "Actual Cost": 500,
    "Difference": 0,
  },
  {
    "Expense Type": "Loan",
    "Projected Cost": 1000,
    "Actual Cost": 1000,
    "Difference": 0,
  },
  {
    "Expense Type": "Tax",
    "Projected Cost": 200,
    "Actual Cost": 200,
    "Difference": 0,
  },
  {
    "Expense Type": "Savings",
    "Projected Cost": 1000,
    "Actual Cost": 900,
    "Difference": 100,
  },
  {
    "Expense Type": "Total",
    "Projected Cost": 13500,
    "Actual Cost": 13600,
    "Difference": -100,
  }
]
export let salaryData: Object[] = [
  {
    "Earnings": "Basic",
    "Credit Amount": 20000,
    "Deductions": "Provident Fund",
    "Debit Amount": 2400,
  },
  {
    "Earnings": "HRA",
    "Credit Amount": 8000,
    "Deductions": "ESI",
    "Debit Amount": 0,
  }
]

```

```

    },
    {
      "Earnings": "Special Allowance",
      "Credit Amount": 25000,
      "Deductions": "Professional Tax",
      "Debit Amount": 200,
    },
    {
      "Earnings": "Incentives",
      "Credit Amount": 2000,
      "Deductions": "TDS",
      "Debit Amount": 2750,
    },
    {
      "Earnings": "Bonus",
      "Credit Amount": 1500,
      "Deductions": "Other Deduction",
      "Debit Amount": 0,
    },
    {
      "Earnings": "Total Earnings",
      "Credit Amount": 56500,
      "Deductions": "Total Deductions",
      "Debit Amount": 5350,
    }
  ]
  export let printElement: HTMLElement = createElement("div", {
    className: "e-sheet-panel",
    innerHTML:
      '<div class="e-spreadsheet-print"></div><div class="e-sheet"><div class="e-main-panel style="height:100%" style="overflow: unset"><div class="e-sheet-content" ></div></div></div>'
  }); // creating same hierarchy of element as DOM

```

### Create a object structure in React Spreadsheet component

This topic guides you to construct a JSON structure that can be passed to the [openFromJson](#) method to render the spreadsheet. The JSON structure is an object with the key as **Workbook** and the [properties](#) of the spreadsheet as value.

```

`js
{ Workbook: {} }
`

```

The following properties are the root level properties of the **Workbook** object.

Property	Type	Description
activeSheetIndex	number	Specifies active sheet index in the workbook.
sheets	<b>Sheet[]</b>	Contains a list of sheet properties.
definedNames	<b>DefineName[]</b>	Specifies the name for a range and uses it in the formula for calculation.

The following table defines each property of the **Sheet**.

Property	Type	Description
name	string	Specifies the name of the sheet.
selectedRange	string	Specifies selected range in the sheet.
activeCell	string	Specifies active cell within <b>selectedRange</b> in the sheet.
topLeftCell	string	Specified cell will be positioned at the upper-left corner of the sheet.
showHeaders	boolean	Specifies to show or hide column and row headers in the sheet.
showGridLines	boolean	Specifies to show or hide gridlines in the sheet.
isProtected	boolean	Specifies to protect the cells in the sheet.
state	<a href="#">SheetState</a>	Specifies the sheet visibility state. There must be at least one visible sheet in Spreadsheet.
columns	<b>Column[]</b>	Contains a list of column properties
rows	<b>Row[]</b>	Contains a list of row properties
protectSettings	<a href="#">ProtectSettings</a>	Configures protect and its options.
conditionalFormats	<b>ConditionalFormat[]</b>	Specifies the conditional formatting for the sheet.

The following table defines each property of the **Column**.

Property	Type	Description
width	number	Specifies the width of the column.
customWidth	boolean	Specifies custom width of the column.
hidden	boolean	To hide or show the column in the sheet.

The following table defines each property of the **Row**.

Property	Type	Description
height	number	Specifies the height of the row.
customHeight	boolean	Specifies the custom height of the row.
hidden	boolean	To hide or show the row in the sheet.
cells	<b>Cell[]</b>	Contains a list of cell properties

The following table defines each property of the **Cell**.

Property	Type	Description
value	string	Defines the value of the cell which can be text or number.

| formula | string | Defines the formula or expression of the cell. |

| format | string | Specifies the number format code to display the value in specified number format. |

| hyperlink | string | Specifies the hyperlink of the cell. |

| wrap | boolean | Wraps the cell text to the next line, if the text width exceeds the column width. |

| isLocked | boolean | Specifies the cell whether it is locked or not, for allowing edit range in the spreadsheet protect option. |

| colSpan | number | Specifies the column-wise cell merge count. |

| rowSpan | number | Specifies the row-wise cell merge count. |

| style | **CellStyle** | Specifies the cell style options. |

| validation | **Validation** | Specifies the validation of the cell. |

| image | **Image[]** | Specifies the image of the cell. |

The following table defines each property of the **CellStyle**.

Property	Type	Description
-----	-----	-----
fontFamily	<b>FontFamily</b>	Specifies font family of the cell.
verticalAlign	<b>VerticalAlign</b>	Specifies vertical align of the cell.
textAlign	<b>TextAlign</b>	Specifies text align style of the cell.
textIndent	string	Specifies text indent style of the cell.
color	string	Specifies font color of the cell.
backgroundColor	string	Specifies the background color of the cell.
fontWeight	<b>FontWeight</b>	Specifies font weight of the cell.
fontStyle	<b>FontStyle</b>	Specifies font style of the cell.
fontSize	string	Specifies font size of the cell.
textDecoration	<b>TextDecoration</b>	Specifies text decoration of the cell.
border	string	Specifies border of the cell.
borderTop	string	Specifies top border of the cell.
borderBottom	string	Specifies bottom border of the cell.
borderLeft	string	Specifies left border of the cell.
borderRight	string	Specifies right border of the cell.

`js

```
type FontFamily = 'Arial' | 'Arial Black' | 'Axettac Demo' | 'Batang' | 'Book Antiqua' | 'Calibri' | 'Courier' |
'Courier New' | 'Din Condensed' | 'Georgia' | 'Helvetica' | 'Helvetica New' | 'Roboto' | 'Tahoma' | 'Times
New Roman' | 'Verdana';
```

```
type VerticalAlign = 'bottom' | 'middle' | 'top';
type TextAlign = 'left' | 'center' | 'right';
type FontWeight = 'bold' | 'normal';
type FontStyle = 'italic' | 'normal';
type TextDecoration = 'underline' | 'line-through' | 'underline line-through' | 'none';
`js
```

The following table defines each property of the Validation.

Property	Type	Description
type	ValidationType	Specifies Validation Type.
operator	ValidationOperator	Specifies Validation Operator.
value1	string	Specifies Validation Minimum Value.
value2	string	Specifies Validation Maximum Value.
ignoreBlank	boolean	Specifies IgnoreBlank option in Data Validation.
inCellDropDown	boolean	Specifies InCellDropDown option in Data Validation.
isHighlighted	boolean	Specifies to allow Highlight Invalid Data.

```
`js
type ValidationType = 'WholeNumber' | 'Decimal' | 'Date' | 'TextLength' | 'List' | 'Time';
type ValidationOperator = 'Between' | 'NotBetween' | 'EqualTo' | 'NotEqualTo' | 'LessThan' | 'GreaterThan' | 'GreaterThanOrEqualTo' | 'LessThanOrEqualTo';
`js
```

The following table defines each property of the Image.

Property	Type	Description
src	string	Specifies the image source.
id	string	Specifies image element id.
height	number	Specifies the height of the image.
width	number	Specifies the width of the image.
top	number	Specifies the top position of the image.
left	number	Specifies the left position of the image.

The following table defines each property of the ConditionalFormat.

Property	Type	Description
----------	------	-------------



```
|-----|-----|-----|
| type | HighlightCell or TopBottom or DataBar or ColorScale or IconSet | Specifies Conditional
formatting Type. |
| format | Format | Specifies format. |
| cFColor | CFColor | Specifies Conditional formatting Highlight Color. |
| value | string | Specifies Conditional formatting value. |
| range | string | Specifies Conditional formatting range. |
`js
type HighlightCell = 'GreaterThan' | 'LessThan' | 'Between' | 'EqualTo' | 'ContainsText' | 'DateOccur' |
'Duplicate' | 'Unique';
type TopBottom = 'Top10Items' | 'Bottom10Items' | 'Top10Percentage' | 'Bottom10Percentage' |
'BelowAverage' | 'AboveAverage';
type DataBar = 'BlueDataBar' | 'GreenDataBar' | 'RedDataBar' | 'OrangeDataBar' | 'LightBlueDataBar' |
'PurpleDataBar';
type ColorScale = 'GYRColorScale' | 'RYGColorScale' | 'GWRColorScale' | 'RWGColorScale' |
'BWRColorScale' | 'RWBColorScale' | 'WRCColorScale' | 'RWColorScale' | 'GWColorScale' |
'WGColorScale' | 'GYColorScale' | 'YGColorScale';
type IconSet = 'ThreeArrows' | 'ThreeArrowsGray' | 'FourArrowsGray' | 'FourArrows' | 'FiveArrowsGray'
| 'FiveArrows' | 'ThreeTrafficLights1' | 'ThreeTrafficLights2' | 'ThreeSigns' | 'FourTrafficLights' |
'FourRedToBlack' | 'ThreeSymbols' | 'ThreeSymbols2' | 'ThreeFlags' | 'FourRating' | 'FiveQuarters' |
'FiveRating' | 'ThreeTriangles' | 'ThreeStars' | 'FiveBoxes';
type CFColor = 'RedFT' | 'YellowFT' | 'GreenFT' | 'RedF' | 'RedT';
`
```

The following table defines each property of the **Format**.

Property	Type	Description
format	string	Specifies the number format code to display the value in specified number format.
style	CellStyle	Specifies the cell style options.

The following table defines each property of the **DefinedName**.

Property	Type	Description
name	string	Specifies a name for the defined name, which can be used in the formula.
scope	string	Specifies scope for the defined name.
comment	string	Specifies comment for the defined name.
refersTo	string	Specifies reference for the defined name.

In the following demo, the JSON structure is passed to the `openFromJson` method to render the spreadsheet in the `created` event.

### APP.JSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
import { jsonData } from '../datasource';
function App() {
    const spreadsheetRef = React.useRef(null);
    const onCreate = () => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            spreadsheet.openFromJson({ file: jsonData });
        }
    };
    return (<SpreadsheetComponent ref={spreadsheetRef}
created={onCreate}></SpreadsheetComponent>);
};
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}
```

### APP.TSX

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
import { jsonData } from '../datasource';
function App() {
    const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
    const onCreate = (): void => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            spreadsheet.openFromJson({ file: jsonData });
        }
    };
    return (<SpreadsheetComponent ref={spreadsheetRef}
created={onCreate}></SpreadsheetComponent>);
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}
```

### DATASOURCE.JSX

```
/**
 * json structure
 */
export let jsonData = {
    "Workbook": {
```

```

    "definedNames": [],
    "sheets": [
      {
        "columns": [
          {
            "width": 100
          },
          {
            "width": 158
          },
          {
            "width": 72
          },
          {
            "width": 113
          },
          {
            "width": 113
          },
          {
            "width": 77
          },
          {
            "width": 97
          },
          {
            "width": 73
          }
        ],
        "conditionalFormats": [
          {
            "type": "GYRColorScale",
            "range": "C3:C10"
          },
          {
            "type": "LessThan",
            "cFColor": "RedF",
            "range": "G3:G10",
            "value": "43707,"
          },
          {
            "type": "BlueDataBar",
            "range": "D3:D10"
          },
          {
            "type": "GreenDataBar",
            "range": "E3:E10"
          },
          {
            "type": "Top10Items",
            "format": {
              "style": {
                "backgroundColor": "#009999",
                "fontWeight": "Bold"
              }
            },
            "range": "F3:F10",

```

```

        "value": "1,"
      },
      {
        "type": "Bottom10Items",
        "format": {
          "style": {
            "backgroundColor": "#C68D53",
            "fontWeight": "Bold"
          }
        },
        "range": "F3:F10",
        "value": "1,"
      },
      {
        "type": "ThreeStars",
        "range": "H3:H10"
      }
    ],
    "name": "Inventory List",
    "rows": [
      {
        "cells": [
          {
            "colSpan": 8,
            "style": {
              "fontSize": "13pt",
              "fontWeight": "Bold",
              "textAlign": "Center",
              "verticalAlign": "Middle"
            },
            "value": "Inventory List"
          },
          {
            "style": {
              "fontSize": "13pt",
              "fontWeight": "Bold",
              "textAlign": "Center",
              "verticalAlign": "Middle"
            }
          },
          {
            "style": {
              "fontSize": "13pt",
              "fontWeight": "Bold",
              "textAlign": "Center",
              "verticalAlign": "Middle"
            }
          }
        ]
      }
    ]
  }

```

```

        "style": {
            "fontSize": "13pt",
            "fontWeight": "Bold",
            "textAlign": "Center",
            "verticalAlign": "Middle"
        },
    },
    {
        "style": {
            "fontSize": "13pt",
            "fontWeight": "Bold",
            "textAlign": "Center",
            "verticalAlign": "Middle"
        },
    },
    {
        "style": {
            "fontSize": "13pt",
            "fontWeight": "Bold",
            "textAlign": "Center",
            "verticalAlign": "Middle"
        },
    },
    {
        "style": {
            "fontSize": "13pt",
            "fontWeight": "Bold",
            "textAlign": "Center",
            "verticalAlign": "Middle"
        },
    },
    ],
    "height": 30
},
{
    "cells": [
        {
            "style": {
                "fontWeight": "Bold",
                "textAlign": "Center"
            },
            "value": "Item Code"
        },
        {
            "style": {
                "fontWeight": "Bold",
                "textAlign": "Center"
            },
            "value": "Item Name"
        },
        {
            "style": {
                "fontWeight": "Bold",
                "textAlign": "Center"
            },
            "value": "Quantity"
        },
    ],

```

```

        {
            "style": {
                "fontWeight": "Bold",
                "textAlign": "Center"
            },
            "value": "Purchase Price"
        },
        {
            "style": {
                "fontWeight": "Bold",
                "textAlign": "Center"
            },
            "value": "Selling Price"
        },
        {
            "style": {
                "fontWeight": "Bold",
                "textAlign": "Center"
            },
            "value": "Profit"
        },
        {
            "style": {
                "fontWeight": "Bold",
                "textAlign": "Center"
            },
            "value": "Last Updated"
        },
        {
            "style": {
                "fontWeight": "Bold",
                "textAlign": "Center"
            },
            "value": "Rating"
        }
    ],
    },
    {
        "cells": [
            {
                "value": "AG940Z"
            },
            {
                "value": "Laser Printer"
            },
            {
                "value": "144"
            },
            {
                "format": "$#,##0.00",
                "value": "169.5"
            },
            {
                "format": "$#,##0.00",
                "value": "172"
            },
        ],
    }

```

```

        "format": "$#,##0.00",
        "formula": "=E3-D3"
      },
      {
        "format": "mm-dd-yyyy",
        "value": "43610"
      },
      {
        "value": "4.5"
      }
    ]
  },
  {
    "cells": [
      {
        "value": "BJ120K"
      },
      {
        "value": "Scientific Calculator"
      },
      {
        "value": "116"
      },
      {
        "format": "$#,##0.00",
        "value": "21.8"
      },
      {
        "format": "$#,##0.00",
        "value": "23"
      },
      {
        "format": "$#,##0.00",
        "formula": "=E4-D4"
      },
      {
        "format": "mm-dd-yyyy",
        "value": "43674"
      },
      {
        "value": "4"
      }
    ]
  },
  {
    "cells": [
      {
        "value": "BC120M"
      },
      {
        "value": "Wired Keyboard"
      },
      {
        "value": "438"
      },
      {
        "format": "$#,##0.00",

```

```

        "value": "26.8"
      },
      {
        "format": "$#,##0.00",
        "value": "29"
      },
      {
        "format": "$#,##0.00",
        "formula": "=E5-D5"
      },
      {
        "format": "mm-dd-yyyy",
        "value": "43920"
      },
      {
        "value": "4.25"
      }
    ]
  },
  {
    "cells": [
      {
        "value": "BS121L"
      },
      {
        "value": "Memory Card"
      },
      {
        "value": "486"
      },
      {
        "format": "$#,##0.00",
        "value": "23.5"
      },
      {
        "format": "$#,##0.00",
        "value": "25"
      },
      {
        "format": "$#,##0.00",
        "formula": "=E6-D6"
      },
      {
        "format": "mm-dd-yyyy",
        "value": "43697"
      },
      {
        "value": "3.5"
      }
    ]
  },
  {
    "cells": [
      {
        "value": "BU121K"
      },

```



```

        "value": "Coffee Maker"
      },
      {
        "value": "176"
      },
      {
        "format": "$#,##0.00",
        "value": "56.5"
      },
      {
        "format": "$#,##0.00",
        "value": "59"
      },
      {
        "format": "$#,##0.00",
        "formula": "=E7-D7"
      },
      {
        "format": "mm-dd-yyyy",
        "value": "43863"
      },
      {
        "value": "4.5"
      }
    ]
  },
  {
    "cells": [
      {
        "value": "BD121M"
      },
      {
        "value": "Table Lamp"
      },
      {
        "value": "0"
      },
      {
        "format": "$#,##0.00",
        "value": "22.5"
      },
      {
        "format": "$#,##0.00",
        "value": "25"
      },
      {
        "format": "$#,##0.00",
        "formula": "=E8-D8"
      },
      {
        "format": "mm-dd-yyyy",
        "value": "43780"
      },
      {
        "value": "5"
      }
    ]
  }
]

```

```

    },
    {
      "cells": [
        {
          "value": "AT992X"
        },
        {
          "value": "Document Scanner"
        },
        {
          "value": "116"
        },
        {
          "format": "$#,##0.00",
          "value": "175"
        },
        {
          "format": "$#,##0.00",
          "value": "177"
        },
        {
          "format": "$#,##0.00",
          "formula": "=E9-D9"
        },
        {
          "format": "mm-dd-yyyy",
          "value": "43568"
        },
        {
          "value": "4.75"
        }
      ]
    },
    {
      "cells": [
        {
          "value": "AP992Z"
        },
        {
          "value": "Gaming Headset"
        },
        {
          "value": "58"
        },
        {
          "format": "$#,##0.00",
          "value": "32"
        },
        {
          "format": "$#,##0.00",
          "value": "35"
        },
        {
          "format": "$#,##0.00",
          "formula": "=E10-D10"
        }
      ]
    }
  ]
}

```

```

        "format": "mm-dd-yyyy",
        "value": "43875"
      },
      {
        "value": "4.4"
      }
    ]
  },
  "selectedRange": "A1:H1"
}
];
};

```

### **DATASOURCE.TSX**

```

/**
 * json structure
 */
export let jsonData: Object = {
  "Workbook": {
    "definedNames": [],
    "sheets": [
      {
        "columns": [
          {
            "width": 100
          },
          {
            "width": 158
          },
          {
            "width": 72
          },
          {
            "width": 113
          },
          {
            "width": 113
          },
          {
            "width": 77
          },
          {
            "width": 97
          },
          {
            "width": 73
          }
        ],
        "conditionalFormats": [
          {
            "type": "GYRColorScale",
            "range": "C3:C10"
          },

```

```

    {
      "type": "LessThan",
      "cFColor": "RedF",
      "range": "G3:G10",
      "value": "43707,"
    },
    {
      "type": "BlueDataBar",
      "range": "D3:D10"
    },
    {
      "type": "GreenDataBar",
      "range": "E3:E10"
    },
    {
      "type": "Top10Items",
      "format": {
        "style": {
          "backgroundColor": "#009999",
          "fontWeight": "Bold"
        }
      },
      "range": "F3:F10",
      "value": "1,"
    },
    {
      "type": "Bottom10Items",
      "format": {
        "style": {
          "backgroundColor": "#C68D53",
          "fontWeight": "Bold"
        }
      },
      "range": "F3:F10",
      "value": "1,"
    },
    {
      "type": "ThreeStars",
      "range": "H3:H10"
    }
  ],
  "name": "Inventory List",
  "rows": [
    {
      "cells": [
        {
          "colSpan": 8,
          "style": {
            "fontSize": "13pt",
            "fontWeight": "Bold",
            "textAlign": "Center",
            "verticalAlign": "Middle"
          },
          "value": "Inventory List"
        },
        {
          "style": {

```

[illegible]

```

{
  "cells": [
    {
      "style": {
        "fontWeight": "Bold",
        "textAlign": "Center"
      },
      "value": "Item Code"
    },
    {
      "style": {
        "fontWeight": "Bold",
        "textAlign": "Center"
      },
      "value": "Item Name"
    },
    {
      "style": {
        "fontWeight": "Bold",
        "textAlign": "Center"
      },
      "value": "Quantity"
    },
    {
      "style": {
        "fontWeight": "Bold",
        "textAlign": "Center"
      },
      "value": "Purchase Price"
    },
    {
      "style": {
        "fontWeight": "Bold",
        "textAlign": "Center"
      },
      "value": "Selling Price"
    },
    {
      "style": {
        "fontWeight": "Bold",
        "textAlign": "Center"
      },
      "value": "Profit"
    },
    {
      "style": {
        "fontWeight": "Bold",
        "textAlign": "Center"
      },
      "value": "Last Updated"
    },
    {
      "style": {
        "fontWeight": "Bold",
        "textAlign": "Center"
      },
      "value": "Rating"
    }
  ]
}

```

```

    }
  ]
},
{
  "cells": [
    {
      "value": "AG940Z"
    },
    {
      "value": "Laser Printer"
    },
    {
      "value": "144"
    },
    {
      "format": "$#,##0.00",
      "value": "169.5"
    },
    {
      "format": "$#,##0.00",
      "value": "172"
    },
    {
      "format": "$#,##0.00",
      "formula": "=E3-D3"
    },
    {
      "format": "mm-dd-yyyy",
      "value": "43610"
    },
    {
      "value": "4.5"
    }
  ]
},
{
  "cells": [
    {
      "value": "BJ120K"
    },
    {
      "value": "Scientific Calculator"
    },
    {
      "value": "116"
    },
    {
      "format": "$#,##0.00",
      "value": "21.8"
    },
    {
      "format": "$#,##0.00",
      "value": "23"
    },
    {
      "format": "$#,##0.00",
      "formula": "=E4-D4"
    }
  ]
}

```

```

    },
    {
      "format": "mm-dd-yyyy",
      "value": "43674"
    },
    {
      "value": "4"
    }
  ]
},
{
  "cells": [
    {
      "value": "BC120M"
    },
    {
      "value": "Wired Keyboard"
    },
    {
      "value": "438"
    },
    {
      "format": "$#,##0.00",
      "value": "26.8"
    },
    {
      "format": "$#,##0.00",
      "value": "29"
    },
    {
      "format": "$#,##0.00",
      "formula": "=E5-D5"
    },
    {
      "format": "mm-dd-yyyy",
      "value": "43920"
    },
    {
      "value": "4.25"
    }
  ]
},
{
  "cells": [
    {
      "value": "BS121L"
    },
    {
      "value": "Memory Card"
    },
    {
      "value": "486"
    },
    {
      "format": "$#,##0.00",
      "value": "23.5"
    }
  ]
}

```



```

        {
            "format": "$#,##0.00",
            "value": "25"
        },
        {
            "format": "$#,##0.00",
            "formula": "=E6-D6"
        },
        {
            "format": "mm-dd-yyyy",
            "value": "43697"
        },
        {
            "value": "3.5"
        }
    ]
},
{
    "cells": [
        {
            "value": "BU121K"
        },
        {
            "value": "Coffee Maker"
        },
        {
            "value": "176"
        },
        {
            "format": "$#,##0.00",
            "value": "56.5"
        },
        {
            "format": "$#,##0.00",
            "value": "59"
        },
        {
            "format": "$#,##0.00",
            "formula": "=E7-D7"
        },
        {
            "format": "mm-dd-yyyy",
            "value": "43863"
        },
        {
            "value": "4.5"
        }
    ]
},
{
    "cells": [
        {
            "value": "BD121M"
        },
        {
            "value": "Table Lamp"
        }
    ]
}

```

```

        {
            "value": "0"
        },
        {
            "format": "$#,##0.00",
            "value": "22.5"
        },
        {
            "format": "$#,##0.00",
            "value": "25"
        },
        {
            "format": "$#,##0.00",
            "formula": "=E8-D8"
        },
        {
            "format": "mm-dd-yyyy",
            "value": "43780"
        },
        {
            "value": "5"
        }
    ]
},
{
    "cells": [
        {
            "value": "AT992X"
        },
        {
            "value": "Document Scanner"
        },
        {
            "value": "116"
        },
        {
            "format": "$#,##0.00",
            "value": "175"
        },
        {
            "format": "$#,##0.00",
            "value": "177"
        },
        {
            "format": "$#,##0.00",
            "formula": "=E9-D9"
        },
        {
            "format": "mm-dd-yyyy",
            "value": "43568"
        },
        {
            "value": "4.75"
        }
    ]
},
{

```

```

        "cells": [
          {
            "value": "AP992Z"
          },
          {
            "value": "Gaming Headset"
          },
          {
            "value": "58"
          },
          {
            "format": "$#,##0.00",
            "value": "32"
          },
          {
            "format": "$#,##0.00",
            "value": "35"
          },
          {
            "format": "$#,##0.00",
            "formula": "=E10-D10"
          },
          {
            "format": "mm-dd-yyyy",
            "value": "43875"
          },
          {
            "value": "4.4"
          }
        ],
        "selectedRange": "A1:H1"
      }
    ]
  }
}

```

### *Changing the active sheet while importing a file in React Spreadsheet component*

You can change the active sheet of imported file by updating [activeSheetIndex](#) property on the [openComplete](#) event.

The following code example shows how to set the active sheet when importing an Excel file.

### **APP.JSX**

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
function App() {
  const spreadsheetRef = React.useRef(null);
  const openCompleteHandler = () => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.activeSheetIndex = 2;
    }
  };
}

```

```

    }
  }
  return (
    <SpreadsheetComponent
      openUrl='https://services.syncfusion.com/react/production/api/spreadsheet/open'
      saveUrl='https://services.syncfusion.com/react/production/api/spreadsheet/save'
      ref={spreadsheetRef}
      openComplete={openCompleteHandler}
    >
    </SpreadsheetComponent>
  );
}
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  const openCompleteHandler = (): void => {
    let spreadsheet = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.activeSheetIndex = 2;
    }
  }
  return (
    <SpreadsheetComponent
      openUrl='https://services.syncfusion.com/react/production/api/spreadsheet/open'
      saveUrl='https://services.syncfusion.com/react/production/api/spreadsheet/save'
      ref={spreadsheetRef}
      openComplete={openCompleteHandler}
    >
    </SpreadsheetComponent>
  );
}
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

*Identify the context menu opened in React Spreadsheet component*

The Spreadsheet includes several context menus that will open and display depending on the action. When you right-click on a cell, for example, a context menu with options related to the cell element appears.

The class name returned by the [contextMenuBeforeOpen](#) event can be used to identify the context menu that is opened. The context menus and their class names are tabulated below.

Class name	Context menu name
----- -----	
.e-sheet-content	Cell context menu
.e-toolbar-item	Footer context menu
.e-rowhdr-table	Row header context menu
.e-colhdr-table	Column header context menu

The following code example shows how to identify the context menu opened.

**APP.JSX**

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
import { closest } from '@syncfusion/ej2-react-base';
function App() {
  const contextMenuBeforeOpen = (args) => {
    if (closest(args.event.target, '.e-sheet-content')) {
      console.log('Cell Context Menu');
    } else if (closest(args.event.target, '.e-colhdr-table')) {
      console.log('Column Header Context Menu');
    } else if (closest(args.event.target, '.e-rowhdr-table')) {
      console.log('Row Header Context Menu');
    } else if (closest(args.event.target, '.e-toolbar-item')) {
      console.log('Footer Context Menu');
    }
  }
  return (
    <SpreadsheetComponent
      contextMenuBeforeOpen={contextMenuBeforeOpen}
    > </SpreadsheetComponent>
  );
}
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}
```

**APP.TSX**

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
```

```
import { closest } from '@syncfusion/ej2-react-base';
import { BeforeOpenCloseMenuEventArgs } from '@syncfusion/ej2-react-navigations';
function App() {
    const contextMenuBeforeOpen = (args: BeforeOpenCloseMenuEventArgs): void
=> {
        if (closest(args.event.target, '.e-sheet-content')) {
            console.log('Cell Context Menu');
        } else if (closest(args.event.target, '.e-colhdr-table')) {
            console.log('Column Header Context Menu');
        } else if (closest(args.event.target, '.e-rowhdr-table')) {
            console.log('Row Header Context Menu');
        } else if (closest(args.event.target, '.e-toolbar-item')) {
            console.log('Footer Context Menu');
        }
    }
    return (
        <SpreadsheetComponent
            contextMenuBeforeOpen={contextMenuBeforeOpen}
        > </SpreadsheetComponent>
    );
}
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}
```

#### *Import an excel document using file uploader in ##Platform\_Name## Spreadsheet component*

If you explore your machine to select and upload an excel document using the file uploader, you will receive the uploaded document as a raw file in the [success](#) event of the file uploader. In this [success](#) event, you should pass the received raw file as an argument to the Spreadsheet's [open](#) method to see the appropriate output.

The following code example shows how to import an excel document using file uploader in spreadsheet.

#### **APP.JSX**

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
import { UploaderComponent } from '@syncfusion/ej2-react-inputs';
function App() {
    const spreadsheetRef = React.useRef(null);
    const uploaderRef = React.useRef(null);
    const asyncSettings = {
        saveUrl:
            'https://services.syncfusion.com/react/production/api/FileUploader/Save',
        removeUrl:
            'https://services.syncfusion.com/react/production/api/FileUploader/Remove',
    };
    const allowedExtensions = '.xlsx, .xls, .csv';
    const onSuccess = (args) => {
        if (args.operation == 'upload')
```

```

        spreadsheetRef.current.open({ file: args.file.rawFile });
    };
    return (
        <div>
            <SpreadsheetComponent
                ref={spreadsheetRef}
            />
        </div>
    );
}
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% enddraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
import { UploaderComponent } from '@syncfusion/ej2-react-inputs';
function App() {
    const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
    const uploaderRef = React.useRef<UploaderComponent>(null);
    const asyncSettings = {
        saveUrl:
            'https://services.syncfusion.com/react/production/api/FileUploader/Save',
        removeUrl:
            'https://services.syncfusion.com/react/production/api/FileUploader/Remove',
    };
    const allowedExtensions: string = '.xlsx, .xls, .csv';
    const onSuccess = (args) => {
        if (args.operation == 'upload')
            spreadsheetRef.current.open({ file: args.file.rawFile });
    };
    return (
        <div>
            <SpreadsheetComponent
                ref={spreadsheetRef}
            />
        </div>
    );
}
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

```

        ref={uploaderRef}
        asyncSettings={asyncSettings}
        success={onSuccess}
        allowedExtensions={allowedExtensions}
      ></UploaderComponent>
    </div>
  );
}
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

### Save and open Spreadsheet data as a Base64 string in React Spreadsheet component

In the Spreadsheet component, there is currently no direct option to save and open data as a **Base64** string. You can achieve this by saving the Spreadsheet data as blob data and then converting that saved blob data to a **Base64** string using **FileReader**.

You can get the Spreadsheet data as blob in the [saveComplete](#) event when you set the **needBlobData** as **true** and **isFullPost** as **false** in the [beforeSave](#) event.

The following code example shows how to save and open the spreadsheet data as base64 string.

#### APP.JSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RangeDirective, ColumnsDirective, ColumnDirective } from
'@syncfusion/ej2-react-spreadsheet';
import { data } from './datasource';
function App() {
  const spreadsheetRef = React.useRef(null);
  const [base64String, setBase64String] = React.useState('');
  const beforeSave = (args) => {
    args.needBlobData = true; // To trigger the saveComplete event.
    args.isFullPost = false; // Get the spreadsheet data as blob data in
the saveComplete event.
  }

  const saveComplete = (args) => {
    // Convert blob data to base64 string.
    let reader = new FileReader();
    reader.readAsDataURL(args.blobData);
    reader.onloadend = function () {
      setBase64String(reader.result);
    };
  }

  const importBtn = () => {
    let spreadsheet = spreadsheetRef.current;
    fetch(base64String)
      .then((response) => response.blob())
      .then((fileBlob) => {
        let file = new File([fileBlob], 'Sample.xlsx');
        spreadsheet.open({ file: file });
      });
  }
}

```



```

    });
  }
  const exportBtn = () => {
    let spreadsheet = spreadsheetRef.current;
    spreadsheet.save({
      url:
        'https://services.syncfusion.com/react/production/api/spreadsheet/save',
      fileName: 'Worksheet',
      saveType: 'Xlsx',
    }); // Specifies the save URL, file name, file type need to be
    saved.
    // Logs base64 string into the console.
    console.log('Base64 String - ', base64String);
  }
  return (
    <div className='control-section spreadsheet-control'>
      <button className="e-btn custom-btn" onClick={importBtn}>Import
      Base64</button>
      <button className="e-btn custom-btn" onClick={exportBtn}>Export
      as Base64</button>
      <SpreadsheetComponent
        openUrl='https://services.syncfusion.com/react/production/api/spreadsheet/op
        en' ref={spreadsheetRef} beforeSave={beforeSave}
        saveComplete={saveComplete} >
        <SheetsDirective>
          <SheetDirective name="Car Sales Report">
            <RangesDirective>
              <RangeDirective
                dataSource={data}></RangeDirective>
            </RangesDirective>
            <ColumnsDirective>
              <ColumnDirective
                width={180}></ColumnDirective>
              <ColumnDirective
                width={130}></ColumnDirective>
              <ColumnDirective
                width={130}></ColumnDirective>
              <ColumnDirective
                width={180}></ColumnDirective>
              <ColumnDirective
                width={130}></ColumnDirective>
              <ColumnDirective
                width={120}></ColumnDirective>
            </ColumnsDirective>
          </SheetDirective>
        </SheetsDirective>
      </SpreadsheetComponent>
    </div>
  );
}
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

**APP.TSX**

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, RangeDirective, ColumnsDirective, ColumnDirective,
SaveCompleteEventArgs, BeforeSaveEventArgs } from '@syncfusion/ej2-react-
spreadsheet';
import { data } from './datasource';
function App() {
    const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
    const [base64String, setBase64String] = React.useState<string>('')
    const beforeSave = (args: BeforeSaveEventArgs): void => {
        args.needBlobData = true; // To trigger the saveComplete event.
        args.isFullPost = false; // Get the spreadsheet data as blob data in
the saveComplete event.
    }

    const saveComplete = (args: SaveCompleteEventArgs): void => {
        // Convert blob data to base64 string.
        let reader: FileReader = new FileReader();
        reader.readAsDataURL(args.blobData);
        reader.onloadend = function () {
            setBase64String(reader.result);
        };
    }

    const importBtn = (): void => {
        let spreadsheet: SpreadsheetComponent = spreadsheetRef.current;
        fetch(base64String)
            .then((response) => response.blob())
            .then((fileBlob) => {
                let file: File = new File([fileBlob], 'Sample.xlsx');
                spreadsheet.open({ file: file });
            });
    }

    const exportBtn = (): void => {
        let spreadsheet: SpreadsheetComponent = spreadsheetRef.current;
        spreadsheet.save({
            url:
'https://services.syncfusion.com/react/production/api/spreadsheet/save',
            fileName: 'Worksheet',
            saveType: 'Xlsx',
        }); // Specifies the save URL, file name, file type need to be
saved.

        // Logs base64 string into the console.
        console.log('Base64 String - ', base64String);
    }

    return (
        <div className='control-section spreadsheet-control'>
            <button className="e-btn custom-btn" onClick={importBtn}>Import
Base64</button>
            <button className="e-btn custom-btn" onClick={exportBtn}>Export
as Base64</button>
            <SpreadsheetComponent
openUrl='<a href="https://services.syncfusion.com/react/production/api/spreadsheet/op
en">https://services.syncfusion.com/react/production/api/spreadsheet/op
en' ref={spreadsheetRef} beforeSave={beforeSave}
saveComplete={saveComplete} >
                <SheetsDirective>

```

```

        <SheetDirective name="Car Sales Report">
            <RangesDirective>
                <RangeDirective
dataSource={data}></RangeDirective>
            </RangesDirective>
            <ColumnsDirective>
                <ColumnDirective
width={180}></ColumnDirective>
                <ColumnDirective
width={130}></ColumnDirective>
                <ColumnDirective
width={130}></ColumnDirective>
                <ColumnDirective
width={180}></ColumnDirective>
                <ColumnDirective
width={130}></ColumnDirective>
                <ColumnDirective
width={120}></ColumnDirective>
            </ColumnsDirective>
        </SheetDirective>
    </SheetsDirective>
</SpreadsheetComponent>
</div>
);
}
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

### *Insert a sheet programmatically and make it the active sheet in React Spreadsheet component*

A sheet is a collection of cells organized in the form of rows and columns that allows you to store, format, and manipulate the data. Using [insertSheet](#) method, you can insert one or more sheets at the desired index. Then, you can make the inserted sheet as active sheet by focusing the start cell of that sheet using the [goTo](#) method.

The following code example shows how to insert a sheet programmatically and make it the active sheet.

#### **APP.JSX**

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
import { SheetsDirective, SheetDirective, RangesDirective, RangeDirective,
ColumnsDirective, ColumnDirective } from '@syncfusion/ej2-react-
spreadsheet';
import { data, employeeData } from './datasource';
function App() {
    const spreadsheetRef = React.useRef(null);
    const btnClick = () => {
        let spreadsheet = spreadsheetRef.current;
        if (spreadsheet) {
            spreadsheet.insertSheet(
                [
                    {

```

```

        index: 1,
        name: 'new_sheet',
        ranges: [
          {
            dataSource: employeeData,
            startCell: 'A1'
          },
        ]
      },
    ]
  );
  // Use the timeout function to wait until the sheet is inserted.
  setTimeout(() => {
    // Method for switching to a new sheet.
    spreadsheet.goTo('new_sheet!A1');
  })
}
};
return (
  <div>
    <button className='e-btn custom-btn' onClick={btnClick}>Insert
Sheet</button>
    <SpreadsheetComponent ref={spreadsheetRef}>
      <SheetsDirective>
        <SheetDirective name="Car Sales Report">
          <RangesDirective>
            <RangeDirective dataSource={data}></RangeDirective>
          </RangesDirective>
          <ColumnsDirective>
            <ColumnDirective width={180}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
            <ColumnDirective width={180}></ColumnDirective>
            <ColumnDirective width={130}></ColumnDirective>
            <ColumnDirective width={120}></ColumnDirective>
          </ColumnsDirective>
        </SheetDirective>
      </SheetsDirective>
    </SpreadsheetComponent>
  </div>
);
}
export default App;
const root = createRoot(document.getElementById('root'));
root.render(<App />);
{% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent } from '@syncfusion/ej2-react-spreadsheet';
import { SheetsDirective, SheetDirective, RangesDirective, RangeDirective,
ColumnsDirective, ColumnDirective } from '@syncfusion/ej2-react-
spreadsheet';

```

```

import { data, employeeData } from './datasource';
function App() {
  const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
  const btnClick = (): void => {
    let spreadsheet: SpreadsheetComponent = spreadsheetRef.current;
    if (spreadsheet) {
      spreadsheet.insertSheet(
        [
          {
            index: 1,
            name: 'new_sheet',
            ranges: [
              {
                dataSource: employeeData,
                startCell: 'A1'
              }
            ],
          }
        ],
      );
      // Use the timeout function to wait until the sheet is inserted.
      setTimeout(() => {
        // Method for switching to a new sheet.
        spreadsheet.goTo('new_sheet!A1');
      })
    }
  };
  return (
    <div>
      <button className='e-btn custom-btn' onClick={btnClick}>Insert
Sheet</button>
      <SpreadsheetComponent ref={spreadsheetRef}>
        <SheetsDirective>
          <SheetDirective name="Car Sales Report">
            <RangesDirective>
              <RangeDirective dataSource={data}></RangeDirective>
            </RangesDirective>
            <ColumnsDirective>
              <ColumnDirective width={180}></ColumnDirective>
              <ColumnDirective width={130}></ColumnDirective>
              <ColumnDirective width={130}></ColumnDirective>
              <ColumnDirective width={180}></ColumnDirective>
              <ColumnDirective width={130}></ColumnDirective>
              <ColumnDirective width={120}></ColumnDirective>
            </ColumnsDirective>
          </SheetDirective>
        </SheetsDirective>
      </SpreadsheetComponent>
    </div>
  );
}
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

*Get the filtered rows in React Spreadsheet component*

Filtering allows you to view specific rows in a spreadsheet while hiding the others. The [allowFiltering](#) property allows you to enable or disable filtering functionality through the UI. You can also use the [allowFiltering](#) property and [applyFilter](#) method combination to filter data via code behind. The filtered rows can be identified by iterating through the row collection on the sheet and using the `isFiltered` property available in each row object.

The following code example shows how to get the filtered rows.

**APP.JSX**

```
{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective } from '@syncfusion/ej2-react-spreadsheet';
import { RangeDirective, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-spreadsheet';
import { defaultData } from './datasource';
function App() {
    const spreadsheetRef = React.useRef(null);
    const created = () => {
        spreadsheetRef.current.cellFormat({ fontWeight: 'bold', textAlign:
'center', verticalAlign: 'middle' }, 'A1:F1');
        // Construct the predicate model to be updated to the data.
        let predicates = [{
            field: 'C',
            operator: 'equal',
            value: 'Pink',
            matchCase: false
        }];
        // Apply filter to the specified range.
        spreadsheetRef.current.applyFilter(predicates, 'A1:C7');
    };
    const onClick = () => {
        let spreadsheet = spreadsheetRef.current;
        let activeSheet = spreadsheet.getActiveSheet();
        let usedRange = activeSheet.usedRange;
        for (let i = 0; i <= usedRange.rowIndex; i++) {
            // Get the filtered row using isFiltered property.
            let filteredRow = (activeSheet.rows[i]).isFiltered;
            if (!filteredRow) {
                let rowData = spreadsheet.getRowData(i);
                console.log("Row:", i + 1, "Cells", rowData);
            }
        }
    }
    return (
        <div className='control-section spreadsheet-control'>
            <button className="e-btn custom-btn" onClick={onClick}> Get
Filtered Data </button>
            <SpreadsheetComponent ref={spreadsheetRef} created={created}>
                <SheetsDirective>
                    <SheetDirective>
                        <RangesDirective>
```

```

                                <RangeDirective
dataSource={defaultData}></RangeDirective>
                                </RangesDirective>
                                <ColumnsDirective>
                                    <ColumnDirective width={180}></ColumnDirective>
                                    <ColumnDirective width={130}></ColumnDirective>
                                    <ColumnDirective width={130}></ColumnDirective>
                                    <ColumnDirective width={180}></ColumnDirective>
                                    <ColumnDirective width={130}></ColumnDirective>
                                    <ColumnDirective width={120}></ColumnDirective>
                                </ColumnsDirective>
                            </SheetDirective>
                        </SheetsDirective>
                    </SpreadsheetComponent>
                </div>
            );
        };
    export default App;
    const root = createRoot(document.getElementById('root'));
    root.render(<App />);
    {% endraw %}

```

### APP.TSX

```

{% raw %}
import * as React from 'react';
import { createRoot } from 'react-dom/client';
import { SpreadsheetComponent, SheetsDirective, SheetDirective,
RangesDirective, ExtendedRowModel, UsedRangeModel, SheetModel,
RangeDirective, ColumnsDirective, ColumnDirective } from '@syncfusion/ej2-
react-spreadsheet';
import { PredicateModel } from '@syncfusion/ej2-react-grids';
import { defaultData } from '../datasource';
function App() {
    const spreadsheetRef = React.useRef<SpreadsheetComponent>(null);
    const created = (): void => {
        spreadsheetRef.current.cellFormat({ fontWeight: 'bold', textAlign:
'center', verticalAlign: 'middle' }, 'A1:F1');
        // Construct the predicate model to be updated to the data.
        let predicates: PredicateModel[] = [{
            field: 'C',
            operator: 'equal',
            value: 'Pink',
            matchCase: false
        }];
        // Apply filter to the specified range.
        spreadsheetRef.current.applyFilter(predicates, 'A1:C7');
    };
    const onClick = () => {
        let spreadsheet = spreadsheetRef.current;
        let activeSheet: SheetModel = spreadsheet.getActiveSheet();
        let usedRange: UsedRangeModel = activeSheet.usedRange;
        for (let i: number = 0; i <= usedRange.rowIndex; i++) {
            // Get the filtered row using isFiltered property.
            let filteredRow: Object = (activeSheet.rows[i] as
ExtendedRowModel).isFiltered;

```

```

        if (!filteredRow) {
            let rowData: Object = spreadsheet.getRowData(i);
            console.log("Row:", i + 1, "Cells", rowData);
        }
    }
}
return (
    <div className='control-section spreadsheet-control'>
        <button className="e-btn custom-btn" onClick={onClick}> Get
Filtered Data </button>
        <SpreadsheetComponent ref={spreadsheetRef} created={created}>
            <SheetsDirective>
                <SheetDirective>
                    <RangesDirective>
                        <RangeDirective
dataSource={defaultData}></RangeDirective>
                    </RangesDirective>
                    <ColumnsDirective>
                        <ColumnDirective width={180}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                        <ColumnDirective width={180}></ColumnDirective>
                        <ColumnDirective width={130}></ColumnDirective>
                        <ColumnDirective width={120}></ColumnDirective>
                    </ColumnsDirective>
                </SheetDirective>
            </SheetsDirective>
        </SpreadsheetComponent>
    </div>
);
};
export default App;
const root = createRoot(document.getElementById('root')!);
root.render(<App />);
{% endraw %}

```

### DATASOURCE.JSX

```

/**
 * Default data source
 */
export let defaultData = [
    {
        "Customer Name": "Romona Heaslip",
        "Model": "Taurus",
        "Color": "Aquamarine",
        "Payment Mode": "Debit Card",
        "Delivery Date": "07/11/2015",
        "Amount": "8529.22"
    },
    {
        "Customer Name": "Clare Batterton",
        "Model": "Sparrow",
        "Color": "Pink",
        "Payment Mode": "Cash On Delivery",
        "Delivery Date": "7/13/2016",
    }
]

```



```

    "Amount": "17866.19"
  },
  {
    "Customer Name": "Eamon Traise",
    "Model": "Grand Cherokee",
    "Color": "Blue",
    "Payment Mode": "Net Banking",
    "Delivery Date": "09/04/2015",
    "Amount": "13853.09"
  },
  {
    "Customer Name": "Julius Gorner",
    "Model": "GTO",
    "Color": "Aquamarine",
    "Payment Mode": "Credit Card",
    "Delivery Date": "12/15/2017",
    "Amount": "2338.74"
  },
  {
    "Customer Name": "Jenna Schoolfield",
    "Model": "LX",
    "Color": "Yellow",
    "Payment Mode": "Credit Card",
    "Delivery Date": "10/08/2014",
    "Amount": "9578.45"
  },
  {
    "Customer Name": "Marylynne Haring",
    "Model": "Catera",
    "Color": "Green",
    "Payment Mode": "Cash On Delivery",
    "Delivery Date": "7/01/2017",
    "Amount": "19141.62"
  },
  {
    "Customer Name": "Vilhelmina Leipelt",
    "Model": "7 Series",
    "Color": "Goldenrod",
    "Payment Mode": "Credit Card",
    "Delivery Date": "12/20/2015",
    "Amount": "6543.30"
  },
  {
    "Customer Name": "Barby Heisler",
    "Model": "Corvette",
    "Color": "Red",
    "Payment Mode": "Credit Card",
    "Delivery Date": "11/24/2014",
    "Amount": "13035.06"
  },
  {
    "Customer Name": "Karyn Boik",
    "Model": "Regal",
    "Color": "Indigo",
    "Payment Mode": "Debit Card",
    "Delivery Date": "05/12/2014",
    "Amount": "18488.80"
  }

```

```

    },
    {
        "Customer Name": "Jeanette Pamplin",
        "Model": "S4",
        "Color": "Fuscia",
        "Payment Mode": "Net Banking",
        "Delivery Date": "12/30/2014",
        "Amount": "12317.04"
    },
    {
        "Customer Name": "Cristi Espinos",
        "Model": "TL",
        "Color": "Aquamarine",
        "Payment Mode": "Credit Card",
        "Delivery Date": "12/18/2013",
        "Amount": "6230.13"
    },
    {
        "Customer Name": "Issy Humm",
        "Model": "Club Wagon",
        "Color": "Pink",
        "Payment Mode": "Cash On Delivery",
        "Delivery Date": "02/02/2015",
        "Amount": "9709.49"
    },
    {
        "Customer Name": "Tuesday Fautly",
        "Model": "V8 Vantage",
        "Color": "Crimson",
        "Payment Mode": "Debit Card",
        "Delivery Date": "11/19/2014",
        "Amount": "9766.10"
    },
    {
        "Customer Name": "Rosemaria Thomann",
        "Model": "Caravan",
        "Color": "Violet",
        "Payment Mode": "Net Banking",
        "Delivery Date": "02/08/2014",
        "Amount": "7685.49"
    },
    {
        "Customer Name": "Lyell Fuentes",
        "Model": "Bravada",
        "Color": "Violet",
        "Payment Mode": "Debit Card",
        "Delivery Date": "08/05/2016",
        "Amount": "18012.45"
    },
    {
        "Customer Name": "Raynell Layne",
        "Model": "Colorado",
        "Color": "Pink",
        "Payment Mode": "Credit Card",
        "Delivery Date": "05/30/2016",
        "Amount": "2785.49"
    },
    },

```

```

{
  "Customer Name": "Raye Whines",
  "Model": "4Runner",
  "Color": "Red",
  "Payment Mode": "Debit Card",
  "Delivery Date": "12/10/2016",
  "Amount": "9967.74"
},
{
  "Customer Name": "Virgina Aharoni",
  "Model": "TSX",
  "Color": "Pink",
  "Payment Mode": "Cash On Delivery",
  "Delivery Date": "10/23/2014",
  "Amount": "5584.33"
},
{
  "Customer Name": "Peta Cheshir",
  "Model": "Pathfinder",
  "Color": "Red",
  "Payment Mode": "Net Banking",
  "Delivery Date": "12/24/2015",
  "Amount": "5286.53"
},
{
  "Customer Name": "Jule Urion",
  "Model": "Charger",
  "Color": "Violet",
  "Payment Mode": "Debit Card",
  "Delivery Date": "11/20/2013",
  "Amount": "13511.91"
},
{
  "Customer Name": "Lew Gilyatt",
  "Model": "Bonneville",
  "Color": "Crimson",
  "Payment Mode": "Credit Card",
  "Delivery Date": "11/19/2013",
  "Amount": "6498.19"
},
{
  "Customer Name": "Jobey Fortun",
  "Model": "B-Series",
  "Color": "Blue",
  "Payment Mode": "Net Banking",
  "Delivery Date": "10/30/2014",
  "Amount": "10359.67"
},
{
  "Customer Name": "Blondie Crump",
  "Model": "Voyager",
  "Color": "Turquoise",
  "Payment Mode": "Credit Card",
  "Delivery Date": "04/06/2018",
  "Amount": "8118.39"
},
{

```

```

        "Customer Name": "Florentia Binns",
        "Model": "Grand Prix",
        "Color": "Orange",
        "Payment Mode": "Cash On Delivery",
        "Delivery Date": "10/13/2016",
        "Amount": "10204.37"
    },
    {
        "Customer Name": "Jaquelin Galtone",
        "Model": "Sunbird",
        "Color": "Red",
        "Payment Mode": "Net Banking",
        "Delivery Date": "10/22/2013",
        "Amount": "6528.06"
    },
    {
        "Customer Name": "Hakeem Easseby",
        "Model": "Mirage",
        "Color": "Crimson",
        "Payment Mode": "Debit Card",
        "Delivery Date": "9/12/2014",
        "Amount": "5619.25"
    },
    {
        "Customer Name": "Nickolaus Gidman",
        "Model": "XK",
        "Color": "Orange",
        "Payment Mode": "Debit Card",
        "Delivery Date": "05/12/2016",
        "Amount": "5091.43"
    },
    {
        "Customer Name": "Jenine Iglesia",
        "Model": "Accord",
        "Color": "Orange",
        "Payment Mode": "Debit Card",
        "Delivery Date": "09/03/2018",
        "Amount": "14566.08"
    },
    {
        "Customer Name": "Fax Witherspoon",
        "Model": "Range Rover Sport",
        "Color": "Orange",
        "Payment Mode": "Credit Card",
        "Delivery Date": "2/22/2018",
        "Amount": "5284.87"
    }
];

```

**DATASOURCE.TSX**

```

/**
 * Default data source
 */
export let defaultData: Object[] = [
    {

```

```

    "Customer Name": "Romona Heaslip",
    "Model": "Taurus",
    "Color": "Aquamarine",
    "Payment Mode": "Debit Card",
    "Delivery Date": "07/11/2015",
    "Amount": "8529.22"
  },
  {
    "Customer Name": "Clare Batterton",
    "Model": "Sparrow",
    "Color": "Pink",
    "Payment Mode": "Cash On Delivery",
    "Delivery Date": "7/13/2016",
    "Amount": "17866.19"
  },
  {
    "Customer Name": "Eamon Traise",
    "Model": "Grand Cherokee",
    "Color": "Blue",
    "Payment Mode": "Net Banking",
    "Delivery Date": "09/04/2015",
    "Amount": "13853.09"
  },
  {
    "Customer Name": "Julius Gorner",
    "Model": "GTO",
    "Color": "Aquamarine",
    "Payment Mode": "Credit Card",
    "Delivery Date": "12/15/2017",
    "Amount": "2338.74"
  },
  {
    "Customer Name": "Jenna Schoolfield",
    "Model": "LX",
    "Color": "Yellow",
    "Payment Mode": "Credit Card",
    "Delivery Date": "10/08/2014",
    "Amount": "9578.45"
  },
  {
    "Customer Name": "Marylynne Harring",
    "Model": "Catera",
    "Color": "Green",
    "Payment Mode": "Cash On Delivery",
    "Delivery Date": "7/01/2017",
    "Amount": "19141.62"
  },
  {
    "Customer Name": "Vilhelmina Leipelt",
    "Model": "7 Series",
    "Color": "Goldenrod",
    "Payment Mode": "Credit Card",
    "Delivery Date": "12/20/2015",
    "Amount": "6543.30"
  },
  {
    "Customer Name": "Barby Heisler",

```

```
"Model": "Corvette",
"Color": "Red",
"Payment Mode": "Credit Card",
"Delivery Date": "11/24/2014",
"Amount": "13035.06"
},
{
  "Customer Name": "Karyn Boik",
  "Model": "Regal",
  "Color": "Indigo",
  "Payment Mode": "Debit Card",
  "Delivery Date": "05/12/2014",
  "Amount": "18488.80"
},
{
  "Customer Name": "Jeanette Pamplin",
  "Model": "S4",
  "Color": "Fuscia",
  "Payment Mode": "Net Banking",
  "Delivery Date": "12/30/2014",
  "Amount": "12317.04"
},
{
  "Customer Name": "Cristi Espinos",
  "Model": "TL",
  "Color": "Aquamarine",
  "Payment Mode": "Credit Card",
  "Delivery Date": "12/18/2013",
  "Amount": "6230.13"
},
{
  "Customer Name": "Issy Humm",
  "Model": "Club Wagon",
  "Color": "Pink",
  "Payment Mode": "Cash On Delivery",
  "Delivery Date": "02/02/2015",
  "Amount": "9709.49"
},
{
  "Customer Name": "Tuesday Fautly",
  "Model": "V8 Vantage",
  "Color": "Crimson",
  "Payment Mode": "Debit Card",
  "Delivery Date": "11/19/2014",
  "Amount": "9766.10"
},
{
  "Customer Name": "Rosemaria Thomann",
  "Model": "Caravan",
  "Color": "Violet",
  "Payment Mode": "Net Banking",
  "Delivery Date": "02/08/2014",
  "Amount": "7685.49"
},
{
  "Customer Name": "Lyell Fuentes",
  "Model": "Bravada",
```

```
"Color": "Violet",
"Payment Mode": "Debit Card",
"Delivery Date": "08/05/2016",
"Amount": "18012.45"
},
{
  "Customer Name": "Raynell Layne",
  "Model": "Colorado",
  "Color": "Pink",
  "Payment Mode": "Credit Card",
  "Delivery Date": "05/30/2016",
  "Amount": "2785.49"
},
{
  "Customer Name": "Raye Whines",
  "Model": "4Runner",
  "Color": "Red",
  "Payment Mode": "Debit Card",
  "Delivery Date": "12/10/2016",
  "Amount": "9967.74"
},
{
  "Customer Name": "Virgina Aharoni",
  "Model": "TSX",
  "Color": "Pink",
  "Payment Mode": "Cash On Delivery",
  "Delivery Date": "10/23/2014",
  "Amount": "5584.33"
},
{
  "Customer Name": "Peta Cheshir",
  "Model": "Pathfinder",
  "Color": "Red",
  "Payment Mode": "Net Banking",
  "Delivery Date": "12/24/2015",
  "Amount": "5286.53"
},
{
  "Customer Name": "Jule Urion",
  "Model": "Charger",
  "Color": "Violet",
  "Payment Mode": "Debit Card",
  "Delivery Date": "11/20/2013",
  "Amount": "13511.91"
},
{
  "Customer Name": "Lew Gilyatt",
  "Model": "Bonneville",
  "Color": "Crimson",
  "Payment Mode": "Credit Card",
  "Delivery Date": "11/19/2013",
  "Amount": "6498.19"
},
{
  "Customer Name": "Jobey Fortun",
  "Model": "B-Series",
  "Color": "Blue",
```

```

    "Payment Mode": "Net Banking",
    "Delivery Date": "10/30/2014",
    "Amount": "10359.67"
  },
  {
    "Customer Name": "Blondie Crump",
    "Model": "Voyager",
    "Color": "Turquoise",
    "Payment Mode": "Credit Card",
    "Delivery Date": "04/06/2018",
    "Amount": "8118.39"
  },
  {
    "Customer Name": "Florentia Binns",
    "Model": "Grand Prix",
    "Color": "Orange",
    "Payment Mode": "Cash On Delivery",
    "Delivery Date": "10/13/2016",
    "Amount": "10204.37"
  },
  {
    "Customer Name": "Jaquelin Galtone",
    "Model": "Sunbird",
    "Color": "Red",
    "Payment Mode": "Net Banking",
    "Delivery Date": "10/22/2013",
    "Amount": "6528.06"
  },
  {
    "Customer Name": "Hakeem Easseby",
    "Model": "Mirage",
    "Color": "Crimson",
    "Payment Mode": "Debit Card",
    "Delivery Date": "9/12/2014",
    "Amount": "5619.25"
  },
  {
    "Customer Name": "Nickolaus Gidman",
    "Model": "XK",
    "Color": "Orange",
    "Payment Mode": "Debit Card",
    "Delivery Date": "05/12/2016",
    "Amount": "5091.43"
  },
  {
    "Customer Name": "Jenine Iglesia",
    "Model": "Accord",
    "Color": "Orange",
    "Payment Mode": "Debit Card",
    "Delivery Date": "09/03/2018",
    "Amount": "14566.08"
  },
  {
    "Customer Name": "Fax Witherspoon",
    "Model": "Range Rover Sport",
    "Color": "Orange",
    "Payment Mode": "Credit Card",

```



```

    "Delivery Date": "2/22/2018",
    "Amount": "5284.87"
  }
];

```

### Mobile responsiveness in React Spreadsheet component

The Spreadsheet control rendered in desktop mode will be adaptive in all mobile devices where the layout gets adjusted based on their parent element's dimensions to accommodate any resolution.

You can see the overflowed items of ribbon header, ribbon content, and sheet tab using touch and swipe action. The right navigation arrow is added at the end of the ribbon content through which the user can navigate towards overflowed items. Once you reached the rightmost end of the ribbon content, the right navigation arrow will change to left navigation arrow through which you can navigate to the left of the ribbon content.

	A	B
1	Customer Name	Model
2	Romona Heaslip	Taurus
3	Clare Batterton	Sparrow
4	Eamon Traise	Grand Cherokee
5	Julius Gorner	GTO
6	Jenna Schoolfield	LX
7	Marylynne Harring	Catera
8	Vilhelmina Leipelt	7 Series
9	Barby Heisler	Corvette
10	Karyn Boik	Regal
11	Jeanette Pamplin	S4
12	Cristi Espinos	TL
13	Issy Humm	Club Wagon
14	Tuesday Fautly	V8 Vantage
15	Rosemaria Thomann	Caravan
16	Lyell Fuentez	Bravada
17	Raynell Layne	Colorado
18	Raye Whines	4Runner
19	Virgina Aharoni	TSX

### Feature list in React Spreadsheet component

The following table compares Excel functionality with the availability of EJ1 and EJ2 Spreadsheet features.

Features	Available in EJ1 Spreadsheet	Available in EJ2 Spreadsheet	Comments
---	---	---	---
Ribbon	Yes	Yes	-
Formula bar	Yes	Yes	-
Sheet tab	Yes	Yes	-
Show / Hide gridlines and header	Yes	Yes	-
Scrolling	Partially	Yes	-
Selection	Yes	Yes	-
Editing	Yes	Yes	-
Formulae	Yes	Partially	EJ2 supports limited number of <a href="#">most used formulas</a>
Named range	Yes	Partially	EJ2 Spreadsheet Named range supports only in workbook scope
Data Binding	Yes	Yes	-
Formatting	Yes	Yes	-
Context menu	Yes	Yes	-
Keyboard navigation	Yes	Yes	-
Keyboard shortcuts	Yes	Yes	-
Sorting	Yes	Yes	-
Filtering	Yes	Yes	-
Hyperlink	Yes	Yes	-
Undo & redo	Yes	Yes	-
Open and Save	Yes	Yes	-
Resize / Autofit	Yes	Yes	-
Clipboard	Yes	Yes	-
Collaborative editing	No	Yes	-
Wrap text	Yes	Yes	-
Template	No	Yes	-
Merge cells	Yes	Yes	-
Show / Hide rows and columns	Yes	Yes	-
Sheet customizations	Yes	Partially	Move or copy sheet is not supported in EJ2 spreadsheet.
Data Validation	Yes	Yes	-
Table	Yes	No	-

| Chart | Yes | Yes | - |

| Image | Yes | Yes | - |

| Conditional formatting | Yes | Yes | - |

| Freeze Pane | Yes | Yes | - |

| Scaling | No | No | - |

| Print | Yes | No | - |

| Grouping | No | No | - |

| Autofill | Yes | No | - |

| Auto Sum | Yes | Yes | - |

| Format painter | Yes | No | - |

| Cell Style | Yes | Partially | We can only customize the cell style in EJ2 Spreadsheet through API. |

| Protection | Yes | Partially | Custom encryption is not supported in EJ2 Spreadsheet's protect workbook. |

| Find and replace | Yes | Yes | - |

| Drag and Drop | Yes | No | - |

| Notes | Yes | No | - |

| Comments | No | No | - |

| Pivot table | Yes | No | - |

| Sparklines | Yes | No | - |

| Form controls | Yes | No | - |

| Shapes | No | No | - |

| Clear | Yes | Yes | - |

#### See Also

- [Cell range](#)
- [Formatting](#)
- [Hyperlink](#)
- [Undo and Redo](#)
- [Unlock the particular cells in the protected sheet](#)

## Stepper

### Getting Started

This section explains how to create a simple Stepper and configure its available functionalities in the React environment.

### Dependencies

The following list of dependencies is required to use the Stepper component in your application.

```
`js
|-- @syncfusion/ej2-react-navigations
|-- @syncfusion/ej2-base
|-- @syncfusion/ej2-popups
|-- @syncfusion/ej2-navigations
|-- @syncfusion/ej2-react-base
`,`
```

### Installation and Configuration

You can use [Create-react-app](#) to setup the applications. To install `create-react-app` run the following command.

```
`bash
npm install -g create-react-app
`,`
```

Start a new project using create-react-app command as follows

```
<div class='tsx'>
```

```
`bash
create-react-app quickstart --scripts-version=react-scripts-ts
cd quickstart
`,`
```

```
</div>
```

```
<div class='jsx'>
```

```
`bash
create-react-app quickstart
cd quickstart
`,`
```

```
</div>
```

'react-scripts-ts' is used for creating React app with typescript.

### Adding Syncfusion packages

All the available Essential JS 2 packages are published in [npmjs.com](https://www.npmjs.com) public registry.

To install `Stepper` component, use the following command

```
`bash
npm install @syncfusion/ej2-react-navigations --save
`,`
```

### Adding CSS Reference

Import the Stepper component's required CSS references as follows in `src/App.css`.

```
`css
@import "../node_modules/@syncfusion/ej2-base/styles/material.css";
@import "../node_modules/@syncfusion/ej2-popups/styles/material.css";
@import "../node_modules/@syncfusion/ej2-navigations/styles/material.css";
`
```

### Adding Stepper component to the application

To include the Stepper component in your application, import the `StepperComponent` from `ej2-react-navigations` package in the `App.tsx`.

To define each step, you can use the `StepDirective` tag inside the `StepsDirective` tag as shown in below code example.

```
{% raw %}
`ts
import { StepperComponent, StepsDirective, StepDirective } from '@syncfusion/ej2-react-navigations';
import * as React from "react";
import * as ReactDOM from "react-dom";
import "../App.css";
function App() {
  return (
    <StepperComponent>
    <StepsDirective>
    <StepDirective/>
    <StepDirective/>
    <StepDirective/>
    <StepDirective/>
    <StepDirective/>
    </StepsDirective>
    </StepperComponent>
  );
}
const root = ReactDOM.createRoot(document.getElementById("element"));
root.render(<App />);
`
```

```
{% endraw %}
```

### Running the application

Run the application in the browser using the following command:

```
,
```

```
npm start
```

```
,
```

The following example shows a basic Stepper component.

#### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
    return (
        <div className="default-stepper">
            <StepperComponent>
                <StepsDirective>
                    <StepDirective/>
                    <StepDirective/>
                    <StepDirective/>
                    <StepDirective/>
                    <StepDirective/>
                </StepsDirective>
            </StepperComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

#### INDEX.TSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
    return (
        <div className="default-stepper">
            <StepperComponent>
                <StepsDirective>
                    <StepDirective/>
                    <StepDirective/>
                    <StepDirective/>
                    <StepDirective/>
                    <StepDirective/>
                </StepsDirective>
            </StepperComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

```

        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

### Configure icon and label

You can define the step icon and label by setting the `iconCss` and `label` properties using the `StepDirective` tag.

#### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
"@syncfusion/ej2-react-navigations";
function App() {
    return (
        <div className="stepper-icon-label">
            <StepperComponent>
                <StepsDirective>
                    <StepDirective iconCss= {'sf-icon-cart'} label=
{'Cart'}/>
                    <StepDirective iconCss= {'sf-icon-user'} label=
{'Address'}/>
                    <StepDirective iconCss= {'sf-icon-transport'} label=
{'Delivery'}/>
                    <StepDirective iconCss= {'sf-icon-payment'} label=
{'Payment'}/>
                    <StepDirective iconCss= {'sf-icon-success'} label=
{'Ordered'}/>
                </StepsDirective>
            </StepperComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

#### INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
"@syncfusion/ej2-react-navigations";
function App() {
    return (
        <div className="stepper-icon-label">
            <StepperComponent>
                <StepsDirective>

```

```

        <StepDirective iconCss= {'sf-icon-cart'} label=
{'Cart'}/>
        <StepDirective iconCss= {'sf-icon-user'} label=
{'Address'}/>
        <StepDirective iconCss= {'sf-icon-transport'} label=
{'Delivery'}/>
        <StepDirective iconCss= {'sf-icon-payment'} label=
{'Payment'}/>
        <StepDirective iconCss= {'sf-icon-success'} label=
{'Ordered'}/>
    </StepsDirective>
  </StepperComponent>
</div>
);
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

## INDEX.CSS

```

/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
@font-face {
  font-family: 'Default';
  src:
    url(data:application/x-font-ttf;charset=utf-
8;base64,AAEAAAAKAIAAAwAgTlMvMjlvSgcAAAEoAAAAVmNtYXCDelPaAAABmAAAAF5nbHlmEwr+
pwAAAggAAAJQaGVhZCYp2+EAADQAAAAANmhoZWEIUQQHAAAArAAAACRobXR4GAAAAAAAYAAAAAYb
G9jYQhUBlAAAAH4AAAAADm1heHABFgErAAABCAAAACBuYW1lUf5THQAACTgAAAIlcG9zdJ8LuoMAAA
0AAAAAbwABAAAEAAAAAFwEAAAAAAD9AABAAAAAAAAAAAAAAAAAABgABAAAAAQAArXT6wV8PPPU
ACwQAAAAAOGly6UAAAAA4YvLpQAAAAAD9AOaAAAAACAACAAAAAAAAAAAAEAAAAGAR8ABgAAAAAAgAA
AAoACgAAAP8AAAAAAAAAAQAAZAABQAAAokCzAAAAI8CiQLMAAAB6wAyAQgAAAIABQMAAAAAAAAA
AAAAAAAAAAAAAAAAAUGZFZABA5wLnFQQAAAAAXAQAAAAAAAAABAAAAAAAAABAAAAAQAAAAEAAAABA
AAAAQAAAAEAAAAAAAAAgAAAAAMAAUAAMAAQAAABQABABKAAAADAAIAAIAB0cC5wbnC0cQ5xX//wA
A5wLnBucI5xDnFf//AAAAAAAAAAAAAAAAABAawADAAMAawADAAAAAEABAACAAMABQAAAAAAAAEQaiwC
3AQkBGgAAAAFAAAAAAP0A18APwB/AIkAxgDrAAABHw8/Dy8OKwEPDQUfDz8PLw4rAQ8NAR8FFSM1J
xEfBz80OwEfDjM/BzUnIw8GATM/Dx8PMxEhAq8BAQIEBAUFBwYICAgJCQoKCgkKCAkIBwcHBQUEAw
MBAQEBAwMEBQUHBwcICQgKCQoKCgkJCAgIBgcFBQOEAgH+CwEBAgQEBQUHBggICakJCgoKCQoICQg
HBwcFBQQDAwEBAQEDAwQFBQcHBwgJCAoJCgoKCQkICAgGBwUFBAQCAQJ8AwUIWAwD3n0BAwMGBgYI
CAMEBQYHBwkJCgsLDA0NDQ4ODQ4MDAwLCgkJCAYGBQMDKAgIBwYFBAECvLsICAYHBQMD/beAAwQFB
QcHCAkKCGsLDA0MDg0NDQwLCwsJCQkHBwUFawKE/eMBAQoJCQkJCAcHBgYFBAMDAQEBAQMDBAUGBg
cHCAkJCQkKCGoJCQgICAcGBgQFAwICAgIDBAUFBgCHCAkJCQoLCgkJCQkIBwcGBgUEAwMBAQEBAwM
EBQYGBwcICQkJCQoKCgkJCAgIBwYGBAUDAgICAgMEBQUGBwcICQkJCgGuAQIGehYJBKYp/10ICAcG
BQQCAQ0NDQwLCgoJCAgGBQUdAgIDBQUGCAGJCgoLDA0NDQECBAUGBwQI1foBAGQFBGcH/iwNDawLC
woJCQgHBgUEAwEBAQEDBAUGBwgJCQoLCwwMDQJJAABAgAAAAAD8wOWAAYAqgBaAGwArQDuAAABZ
cfAwUhLwIHlY8PNS8CKwIPHQEHLwEjDwE1LwMjNz0BJzcFbTcfAg8BLwY30wEfaQcVHw8/Dy8PDw4
BFR8PPw8vDw8OAxEWBgEDAgb8/wNuBAUEDQsVFBQTEhEPDw0GCwoIBgQCFhITE+wQDw8PDg4ODg0N
DQwNCwwKCwoKCQgJBwcHBgYEBQMEA5FrBAQDBAMBAwMDBgIDagIEBgYGBxwCAwIBFQYGBAgFBgIWA

```



```

gQHCPcBAgQGBggKCgsMDQ4PDxAQEBAPDw4NDAsLCQgGBgQCAQECBAYGCAkLCwwNDg8PEBAQEA8PDg
0MCwoKCAYGBAL+KgEEBQgKCw0PEBETFBQWFxcXFhYUFBMREQ4NDakIBgMBAQMGCakMDQ4RERMUFBY
WFxcXFhQUEExEQDw0LCggFBAEXBhcFBAMDrxYWDQEBawUHCAsmDQ4IERESFBQUFQDQAgECAGMEBAUG
BgYIBWgJCQoKCwsLDAwMDQ0ODQ4PDgEZawIBAQIGBQMCAQQDBgZqBggoHBQMDMAMHBwMWAQICBQYKC
hYCB1wICBAPDw4NDAsLCQgGBgQCAQECBAYGCAkLCwwNDg8PEBAQEA8PDg0MCwoKCAYGAwMBAQMDBg
YICgoLDA0ODw8QATMLDBYVFRQSERAPDQsKCAUEAQEEBQgKCw0PEBESFBUVFhcXFxYVFBISEA8NCwo
IBQQBAQQFCAoLDQ8QEHuIFRYXAAAAAQAaaaaa/QDRwA/AH8AhwCRAAABFR8OPw49AS8NKwEPDQUV
Hw4/Dj0BLw0rAQ8NEwcTIRMnMSMhMxMhNSEDBzUha0YBAgMEBAQGBQcGBWgICAgICAgIBWYHBQYEB
AQDagEBAGMEBAQGBQcGBWgICAgICAgIBWYHBQYEBAQDagH+aAICAgQEBAYFBWYIBWgICAgICAgHBg
cFBgQEBAMCAQECawQEBAYFBWYHCAGICAgICAcIBgcFBgQEBaICAsH6jAFKjPpu/Z3NwgJZ/dzDAf8
AAQkICAgHBwcGBgUFBAQCAgEBAQECAGQEBQUGBgCHBwGICaKIBWgHBWYGBQUEAwMCAQECawMEBQUG
BgCHCAcICQgICAcHBWYGBQUEBAICAQEBAQICBAQFBQYGBWCHCAgICQgHCAcHBgYFBQDQAwIBAQIDA
wQFBQYGBWcIBWgB+wH+vQFABP5dOgGkAQEAAAADAAAAANkA5oAnQDxAR4AAAEzHwEdAR8HFQ8DIY
8HDwYdAR8WDw0dAQ8BKwIvAT0BLwc9AT8COWEfBj8HLxc/DTU/AwEfDjsBPxEvFiMPFR8BEw8CFR8
HMz8HNS8GIw8ELwQrAQ8BAGoCAGENDAwKCggHBQEBaIkCAGIEAwQFDA0SBwcGAGIBAQICBgCHBxYK
CQkJCAcHBgUFBAMCAQEBAQIDAQWQFBQYGBWcPEQECaHUCAQINDAsLCQgHBQICKQICAgQDBAULdhIHB
wYCAQEBAQEBAgYHBwcWCgkKCAgHBWYFBQDQAgEBAQECawMEBAYFBgcHEBABAQED/qwUFRUVFRYWFh
YWFxYXFhcXFxcWFxYXFhYWFhYVFRUVFAQCAQICBAUGCAGJCgsLDA0MDQ0NDBk2EQYgqgYGCEsZDQ0
NDA0MCwsKCQgIBgUEAgIBaQCAQEBAwKRNRIHBqADChI1DQoFAGEBAGMEBAOMew8eTw4IVxkXCwkJ
BwYCOAIBAIIDAuUGBwgJCgICAQENAEFAWIDAQECAGMFAWMEBAUDBAMFAWIBAQECawMEBAUGBgYHC
AgICQgHBwcGBgYFBQQEBAQYDgICAQECAiICBAUGBwgJCQMBAGEMAQUDAwIDAQICBAQDBAQEBAQEAW
QEAgEBAQICBAMFBQUGBwcICAgJBWgHBgcGBgUFBAQEBQqIAgEBAf6RDAsLCQkICAYGBQUdAwIBAQI
DAwUFBgYICaKJCwsMKsckIiAeGxoYFhQTERAPDQwLCgkIDxsJBQUFBQqNEAkKCwwNDxARExQWGBob
HiAiJCcCoAMDawQECA8XPRcKCgUPFz0REAkIBAMDAwMCAQICAwCYAwEaBwQBAGIAAAEAAAAAAAA/MDN
AA0AAABDwQvAw8EHwQ/ETUnIw8LAYsEJwwGAGIwXmMXFBIIcCsqKaEqRUclJSYnJykpKiosLC4GFg
sCAWMhISiIiIjIkJAPRwB8AQmCQMBARQuNgsMDgcIJCYnmyZOTycmJiYlJSQjIiIgHwULCAMCAQ4
RERITFBUVKy0tFgAAABIA3gABAAAAAAAAAAAAEAAAABAAAAAABAAcAAQABAAAAAACAACACAABAAAA
AAADAACADwABAAAAAAAEAAcAFgABAAAAAAFAAsAHQABAAAAAAGAAcAKAABAAAAAAKACwALwABA
AAAAALABIAWwADAAEECQAAAAIAbQADAAEECQABAA4AbwADAAEECQACAA4AfQADAAEECQADAA4AiW
ADAAEECQAEAA4AmQADAAEECQAFABYApwADAAEECQAGAA4AvQADAAEECQAKAFgAywADAAEECQALACQ
BIyBEZWZhdWx0UmVndWxhcRlZmFlbHREZWZhdWx0VmVyc2lubiAxLjBEZWZhdWx0Rm9udCBnZW5l
cmF0ZWQgdXNpbmcgU3luY2Zlc2lubiBNZXRYbyBTdHVkaW93d3cuc3luY2Zlc2lubi5jb20AIABEA
GUAZgBhAHUAbAB0AFIAZQBnAHUAbABhAHIArABlAGYAYQB1AGwAdABEAGUAZgBhAHUAbAB0AFYAZQ
ByAHMAaQBvAG4AIAAxAx4AMABEAGUAZgBhAHUAbAB0AEYAbwBuAHQAIABnAGUAbgBIAHIAYQB0AGU
AZAAgAHUAcwBpAG4AZwAgAFMAeQBvAGMAZgB1AHMAaQBvAG4AIAABNAGUAdABYAG8AIAABTAHQAdQBk
AGkAbwB3AHcAdwAuAHMAeQBvAGMAZgB1AHMAaQBvAG4ALgBjAG8AbQAAAAACAAAAAaAAAAA
AAAAAAAAAAAAAAAAAAAAAAYBAgEDAQBBQEGAQCADXRYW5zcG9ydC12YW4LdXNlcil1tb2RpZn
kRc2hvcHBpbmctY2FydF8wMS0Lc3BlbmQtbW9uZXkFY2h1Y2sAAAA=) format('trueType');
font-weight: normal;
font-style: normal;
}
.stepper-icon-label [class^="sf-icon-"], .stepper-icon-label [class*=" sf-
icon-"] {
font-family: 'Default' !important;
speak: none;
font-style: normal;
font-weight: normal;
font-variant: normal;
text-transform: none;
line-height: inherit;
-webkit-font-smoothing: antialiased;
-moz-osx-font-smoothing: grayscale;
}
.stepper-icon-label {
margin-top: 30px;
padding: 30px;
}

```

```
.stepper-icon-label .sf-icon-cart:before { content: "\e710"; }
.stepper-icon-label .sf-icon-user:before { content: "\e708"; }
.stepper-icon-label .sf-icon-transport:before { content: "\e702"; }
.stepper-icon-label .sf-icon-payment:before { content: "\e706"; }
.stepper-icon-label .sf-icon-success:before { content: "\e715"; }
```

## Steps in React Stepper component

The React Stepper component allows you to add steps using `StepDirective` tag. Each step can be configured with options such as `iconCss`, `text`, `label`, `cssClass` and more.

### Adding steps

You can define the icon and text content for each step using the `iconCss`, `text` and `label` properties.

### Defining icon CSS

You can define the CSS class to show the icon for each step using the `iconCss` property.

### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
"@syncfusion/ej2-react-navigations";
function App() {
  return (
    <div className="stepper-icon">
      <StepperComponent>
        <StepsDirective>
          <StepDirective iconCss= {'sf-icon-cart'}/>
          <StepDirective iconCss= {'sf-icon-transport'} />
          <StepDirective iconCss= {'sf-icon-payment'} />
          <StepDirective iconCss= {'sf-icon-success'} />
        </StepsDirective>
      </StepperComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

### INDEX.TSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
"@syncfusion/ej2-react-navigations";
function App() {
  return (
    <div className="stepper-icon">
      <StepperComponent>
        <StepsDirective>
          <StepDirective iconCss= {'sf-icon-cart'}/>
          <StepDirective iconCss= {'sf-icon-transport'} />
        </StepsDirective>
      </StepperComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

```

        <StepDirective iconCss= {'sf-icon-payment'} />
        <StepDirective iconCss= {'sf-icon-success'} />
      </StepsDirective>
    </StepperComponent>
  </div>
);
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

## INDEX.CSS

```

/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
@font-face {
  font-family: 'Default';
  src:
    url(data:application/x-font-ttf;charset=utf-
8;base64,AAEAAAAKAIAAAwAgTlMvMjlvSgcAAAEoAAAAVmNtYXCDelPaAAABmAAAAF5nbHlmEwr+
pwAAAggAAAJQaGVhZCYp2+EAADQAAAAANmhoZWEIUQQHAAAArAAAACRobXR4GAAAAAAAYAAAAAYb
G9jYQhUBlAAAAH4AAAAADmlheHABFgErAAABCAAAACBuYWllluF5THQAACtgAAAIlcG9zdJ8LuoMAAA
0AAAAAbwABAAAEAAAAAFwEAAAAAAD9AABAAAAAABAAAAAABAAAAAQAARxT6wV8PPPU
ACwQAAAAAAGLy6UAAAAA4YvLpQAAAAAD9AOaAAAAACAACAAAAAABAAAAAAGAR8ABgAAAAAAgAA
AAoACgAAAP8AAAAAQAQAAZAABQAAAokCzAAAAI8CiQLMAAAB6wAyAQgAAAIABQMAAAAAAABAAAA
AAAAAAAAAAAAAAAAAUGZFZABA5wLnFQQAAAAAXAQAAAAAABAAAAAABAAAAAQAQAAAAEAAAAABA
AAAAQAQAAAEAAAAAAGAAAAAAMAAUAAMAAQAABQABABKAAAAADAAIAAIABoC5wbnCOcQ5xX//wA
A5wLnBucI5xDnFf//AAAAAAAAAAAAAAAAABAawADAAMAawADAAAAAEABAACAAMABQAAAAAAAEQAiwC
3AQkBGgAAAAFAAAAAAP0A18APwB/AIkAxgDrAAABHw8/Dy8OKwEPDQUfdz8PLw4rAQ8NAR8FFSM1J
xEfBz80OwEfDjM/BzUnIw8GATM/Dx8PMxEhAq8BAQIEBAUFbWYICAgJCQoKCgkKCAkIBwCHBQUEAw
MBAQEBAwMEBQUHBwCICQgKCQoKCgkKCAgIBgcFBQQEAgH+CwEBAgQEBQUHBggICakJCgoKCQoICQg
HBwCQBQDAwEBAQEDAwQFBQcHBwgJCAoJCgoKCQkICAgGBwUFBAQCAQJ8AwUIWAwD3n0BAwMGBgYI
CAMEBQYHBwkJCgsLDA0MDQ4ODQ4MDAwLCgkJCAYGBQMDKAgIBwYFBAECvLsICAYHBQMD/beAAwQFB
QcHCAkKCgsLDA0MDg0NDQwLCwsJCQkHBwUFawKE/eMBAQoJCQkJCAcHBgYFBAMDAQEBAQMDBAUGBg
cHCAkJCQkKCgoJCQgICAcGBgQFAwICAgIDBAUFbGcHCAkJCQoLCgkJCQkIBwCGBgUEAwMBAQEBAwM
EBQYGBwCICQkJCQoKCgkKCAgIBwYGBAUDAQICAgMEBQUGBwCICQkJCgGuAQIGehYJBKYp/10ICAcG
BQCAQ0NDQwLCgoJCAgGBQUdAgIDBQUGCAgJCgoLDA0NDQECBAUGBwQI1foBAGQFBgcH/iwNDawLC
woJCQgHBgUEAwEBAQEDBAUGBwgJCQoLCwwMDQJJAAAABgAAAAAD8wOWAAYAQgBaAGwArQDuAAABBz
cfAwUhLwIHlY8PNS8CKwIPHQEHLwEjDwE1LwMjNz0BJzcfBTcfAg8BLwY3OwEfaQcVHw8/Dy8PDw4
BFR8PPw8vDw8OAxEWBgEDAg8/wNuBAUEDQsVFBQTEhEPDw0GCwoIBgQCFhITE+wQDw8PDg4ODg0N
DQwNCwwKCwoKCQgJBwCHBgYEBQMEA5FrBAQDBAMBawMDBgIDagIEBgYGBxwCAwIBFQYGBAgFBgIWA
gQHCPcBAGQGBggKCgsMDQ4PDxAQEBAPDw4NDAsLCQgGBgQCAQECBAYGCAkLCwwNDg8PEBAQEAE8PDg
0MCwoKCAYGBAL+KgEEBQgKCw0PEBETFBQWFxcXFhYUFBMREQ4NDakIBgMBAQMGCakMDQ4RERMUFBY
WFxcXFhQUExEQDw0LCggFBAEXBhcFBAMDrxYWDQEBawUHCAsmDQ4IERESFBQUFQQDAGECAGMEBAUG
BgYIBwgJCQoKCwsLDAwMDQ0ODQ4PDgEZawIBAQIGBQMCAQQDBgZqBgoHBQMDMAMHBwMWAQICBQYKC
hYCB1wICBAPDw4NDAsLCQgGBgQCAQECBAYGCAkLCwwNDg8PEBAQEAE8PDg0MCwoKCAYGawMBAQMDBg
YICgoLDA0ODw8QATMLDBYVFRQSERAPDQsKCAUEAQEEBQgKCw0PEBESFBUVfHcXFxYVFBISEA8NCwo
IBQQAQQAQCAoLDQ8QEHlUFRYXAAAAAQAAAAA/QDRwA/AH8AhwCRAAABFR8OPw49AS8NKwEPDQUV
Hw4/Dj0BLw0rAQ8NEwcTIRMnMSMhMxMhNSEDBzUha0YBAGMEBAQGBQcGBwgICAgICAgIBwYHBQYEB

```

```

AQDAGEBAGMEBAQGBQCGBWgICAgICAgIBwYHBQYEBAQDAGH+aAICAgQEBAYFBwYIBWgICAgICAgHBg
cFBgQEBAMCAQECAwQEBAYFBwYHCAGICAgICAcIBgcFBgQEBaICAsH6jAFKjPpu/Z3NwgJZ/dzDAf8
AAQkICAgHBwcGBgUFBAQCAgEBAQECAgQEBQUGBgCHBwgICAkIBwgHBwYGBQUEAwMCAQECAwMEBQUG
BgCHCAcICQgICAcHBwYGBQUEBAICAQEBAQICBAQFBQYGBwcHCAgICQgHCAcHBgYFBQQDAwIBAQIDA
wQFBQYGBwcIBwgB+wH+vQFABP5dOgGkAQEAAAADAAAAANkA5oAnQDxAR4AAAEzHwEdAR8HFQ8DIy
8HDWydAR8WDW0dAQ8BKwIvAT0BLwc9AT8COWEfBj8HLxc/DTU/AwEfDjsBPxEvFiMPFR8BEw8CFR8
HMz8HNS8GIw8ELwQrAQ8BAGoCAGENDAwKCgGHBQEBAikCAGIEAwQFDA0SBwcGAGIBAQICBgCHBxYK
CQkJCAcHBgUFBAQCAQEBAQIDAQFBQYGBwcPEQECAhUCAQINDAsLCQgHBQICKQICAgQDBAULdHIBH
wYCAQEBAQEBAgYHBwcWCgkKCAGHBwYFBQQDAGEBQECAwMEBAYFBgcHEBABAQED/qwUFRUVFRYWFh
YWFxYXFhcXFxcWFxYXFhYWFhYVFRUVFAQCAQICBAUGCAGJCgsLDA0MDQ0NDBk2EQYGqgYGCEsZDQ0
NDA0MCwsKCQgIBgUEAgIBaQCAQEBAwKRNRIHBqADChI1DQoFAGEBAGMEBAoMEw8eTw4IVxkXCwkJ
BwYCOAIBAIIDAUGBwgJCgICAQENAEFAwIDAEGCAGMFAwMEBAUDBAMFAwIBAQECAwMEBAUGBgYHC
AgICQgHBwcGBgYFBQEBAYDIgICAQECAiICBAUGBwgJCQMBAGEMAQUDAwIDAQICBAQDBAQEBAQEAw
QEAgEBAQICBAMFBQUGBwcICAgJBwgHBgcGBgUFBAQEBCQiAgEBaf6RDAsLCQkICAYGBQUdAwIBAQI
DAwUFBgYICakJCwsMKSckIiAeGxoYFhQTERAPDQwLCgkIDxsJBQUBBQnEakKCwwNDxARExQWGBob
HiAiJCcCoAMDawQECA8XPRcKCgUPFz0REakIBAMDawMCAQICAwcYAwEaBwQBAGIAAAEAAAAAA/MDN
AA0AAABDwQvAw8EHwQ/ETUnIw8LAYsEJwwGAGIwXmMXFBIICCsqKaEqRUclJSYnJykpKiosLC4GFg
sCAWMhISiIiIjIkJAPRwB8AQmQMBARQuNgsMDgcIJCYnmyZOTycmJiYlJSQjIiIgHwULCAMCAQ4
RERITFBUVKy0tFgAAABIA3gABAAAAAEECAAAABAAAAAABAAcAAQABAAAAAACAACAABAAAA
AAADAACADWABAAAAAEECAAFgABAAAAAFAAsAHQABAAAAAAGAAcAKAABAAAAAAKACwALWABA
AAAAAALABIAWwADAAEECQAAAAIAbQADAAEECQABAA4AbwADAAEECQACAA4AfQADAAEECQADAA4Aiw
ADAAEECQAEAA4AmQADAAEECQAFABYApwADAAEECQAGAA4AvQADAAEECQAKAFgAywADAAEECQALACQ
BIyBEZWZhdWx0UmVndWxhckRlZmFlbHREZWZhdWx0VmVyc2lvbiAxLjBEZWZhdWx0Rm9udCBnZW5l
cmF0ZWQgdXNpbmcgU3luY2Z1c2lvbiBNZXRYbyBTdHVkaW93d3cuc3luY2Z1c2lvbi5jb20AIABEA
GUAZgBhAHUAbAB0AFIAZQBnAHUAbABhAHIArABLAGYAYQBLAGwAdABEAGUAZgBhAHUAbAB0AFYAZQ
ByAHMAaQBvAG4AIAAxAAC4AMABEAGUAZgBhAHUAbAB0AEYAbwBuAHQAIABnAGUAbgBIAHIAYQB0AGU
AZAAgAHUAcwBpAG4AZwAgAFMAeQBwAGMAZgB1AHMAaQBvAG4AIAABNAGUAdABYAG8AIABTAHQAdQBk
AGkAbwB3AHcAdwAuAHMAeQBwAGMAZgB1AHMAaQBvAG4ALgBjAG8AbQAAAAACAAAAAaAAAAA
AAAAAAAAAAAAAAAAAAAAAYBAGEDAQBBQEGAQcADXRyYW5zcG9ydC12YW4LdXNlcil1b2R2Rn
kRc2hvcHBpbmctY2FydF8wMS0Lc3BlbmQtbW9uZXkFY2hlyY2sAAAA=) format('trueType');
    font-weight: normal;
    font-style: normal;
}
.stepper-icon [class^="sf-icon-"], .stepper-icon [class*=" sf-icon-"] {
    font-family: 'Default' !important;
    speak: none;
    font-style: normal;
    font-weight: normal;
    font-variant: normal;
    text-transform: none;
    line-height: inherit;
    -webkit-font-smoothing: antialiased;
    -moz-osx-font-smoothing: grayscale;
}
.stepper-icon {
    margin-top: 30px;
    padding: 30px;
}
.stepper-icon .sf-icon-cart:before { content: "\e710"; }
.stepper-icon .sf-icon-transport:before { content: "\e702"; }
.stepper-icon .sf-icon-payment:before { content: "\e706"; }
.stepper-icon .sf-icon-success:before { content: "\e715"; }

```

### Defining text content

You can define text instead of an icon by setting the `text` property and display label content for a step using the `label` property.

When both label and text are defined, the label takes priority for display based on the `stepType`.

#### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
  return (
    <div className="stepper-text">
      <StepperComponent>
        <StepsDirective>
          <StepDirective text= {'Cart'}/>
          <StepDirective text= {'Delivery Address'}/>
          <StepDirective text= {'Payment'}/>
          <StepDirective text= {'Confirmation'}/>
        </StepsDirective>
      </StepperComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

#### INDEX.TSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
  return (
    <div className="stepper-text">
      <StepperComponent>
        <StepsDirective>
          <StepDirective text= {'Cart'}/>
          <StepDirective text= {'Delivery Address'}/>
          <StepDirective text= {'Payment'}/>
          <StepDirective text= {'Confirmation'}/>
        </StepsDirective>
      </StepperComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

**INDEX.CSS**

```

/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
@font-face {
  font-family: 'Default';
  src:
    url(data:application/x-font-ttf;charset=utf-
8;base64,AAEAAAAKAIAAAwAgTlMvMjlvSgcAAAEoAAAAVmNtYXCDeIPaAAABmAAAAF5nbHlmEwr+
pwAAAggAAAJQaGVhZCYp2+EAADQAAAAANmhoZWEIUQQHAAAArAAAACRobXR4GAAAAAAAAAYAAAAyb
G9jYQhUBlAAAAH4AAAAADm1heHABFgErAAABCAAAACBuYW1luF5THQAACtgAAAIlcG9zdJ8LuoMAAA
0AAAAAbwABAAAEAAAAAFwEAAAAAAD9AABAAAAAAAAAAAAAAAAAAAAABgABAAAAAQAArxT6wV8PPPU
ACwQAAAAAAOGLy6UAAAAA4YvLpQAAAAAD9A0aAAAAACAACAAAAAAAAAAAAEAAAAGAR8ABgAAAAAAgAA
AAoACgAAAP8AAAAAAAAAAQAAZAABQAAAokCzAAAAI8CiQLMAAAB6wAyAQgAAAIABQMAAAAAAAAAAA
AAAAAAAAAAAAAAAAAUGZFZABA5wLnFQQAAAAAXAQAAAAAAAABAAAAAABAAAAAQAAAAEAAAABA
AAAAQAAAAEAAAAAAAAGAAAAAMAAAUAMAAQAAABQABABKAAAAADAAIAAIABOcC5wbnCOcQ5xX//wA
A5wLnBucI5xDnFf//AAAAAAAAAAAAAAAAABAawADAAMAawADAAAAAEABAACAAMABQAAAAAAAQAIwC
3AQkBGgAAAAFAAAAAAP0A18APwB/AIkAxgDrAAABHw8/Dy8OKwEPDQUfDz8PLw4rAQ8NAR8FFSM1J
xEfBz8OOwEfDjM/BzUnIw8GATM/Dx8PMxEhAq8BAQIEBAUFbWYICAgJCQoKCgkKCAkIBwCHBQIEAw
MBAQEBAwMEBQUHBwCICQgKCQoKCgkKCAgIBgcFBQQEAgH+CwEBAGQEBQUHBggICAKJCgoKCQoICQg
HBwCFBQQDAwEBAQEDAwQFBQcHBwgJCAoJCgoKCQkICAgGBwUFBAQCAQJ8AwUIWAwD3n0BAwMGBgYI
CAMEBQYHBwkJCgsLDA0NDQ4ODQ4MDAwLCgkJCAYGBQMDKAgIBwYFBAECvLsICAYHBQMD/beAAwQFB
QcHCAkKCgsLDA0MDg0NDQwLCwsJCQkHBwUFawKE/eMBAQoJCQkKJCAcHBgYFBAMDAQEBAQMDBAUGBg
cHCAkKJCQkKCgoJCQgICAcGBgQFAwICAgIDBAUFBgHCAkKJCQoLCgkKJCQkIBwCGBgUEAwMBAQEBAwM
EBQYGBwCICQkKJCQoKCgkKJCAgIBwYGBAUDAgICAgMEBQUGBwCICQkKJCgGuAQIGehYJBKYp/10ICAcG
BQQCAQ0NDQwLCgoJCAGGBQUDAgIDBQUGCAGJCgoLDA0NDQECBAUGBwQI1foBAGQFBgCH/iwNDAwLC
woJCQgHBgUEAwEBAQEDBAUGBwgJCQoLCwwMDQJJAAABgAAAAAD8wOWAAYAqgBaAgWArQDuAAABBz
cfAwUhLwIHlY8PNS8CKwIPHQEHLwEjDwE1LwMjNz0BJzcFbTcfAg8BLwY3OwEfAQcVHw8/Dy8PDw4
BFR8PPw8vDw8OAxEWBgEDAgb8/wNuBAUEDQsVFBQTEhEPDw0GCwoIBgQCfHITE+wQDw8PDg4ODg0N
DQwNCwwKCwoKCQgJBwCHBgyEBQMEAF5FrBAQDBAMBawMDBgIDagIEBgYGBxwCAwIBFYQYGBAgFBgIWA
gQHCPcBAGQGBgKCGsMDQ4PDxAQEBApDw4NDAsLCQgGBgQCAQECEBAYGCAkLCwwNDg8PEBAQEA8PDg
0MCwoKCAYGBAL+KgEEBQgKCw0PEBETFBQWFxcXFhYUFBMREQ4NDakIBgMBAQMGCakMDQ4RERMUFBY
WFxcXFhQUExEQDw0LCgGFBAEXBhcFBAMDrxYWDQEBawUHCAsmDQ4IERESFBQUFQQDAgECAGMEBAUG
BgYIBwgJCQoKCwsLDAwMDQ0ODQ4PDgEZawIBAQIGBQMCAQQDBgZqBgOHBQMDMAMHBwMWAQICBQYKC
hYCB1wICBAPDw4NDAsLCQgGBgQCAQECEBAYGCAkLCwwNDg8PEBAQEA8PDg0MCwoKCAYGawMBAQMDBg
YICgoLDA0ODw8QATMLDBYVFRQSERAPDQsKCAUEAQEEBQgKCw0PEBESFBUVFhcXFxYVFBISEA8NCwo
IBQQAQQFCAoLDQ8QEhIUFRYXAAAAAQAIAAAQ/QRwA/AH8AhwCRAAABFR8OPw49AS8NKwEPDQUV
Hw4/Dj0BLw0rAQ8NEwCTIRMnMSMhMxMhNSEDBzUHA0YBAGMEBAQGBQcGBwgICAgICAgIBwYHBQYEB
AQDAgEBAgMEBAQGBQcGBwgICAgICAgIBwYHBQYEBAQDAgH+aaICAgQEBAYFBwYIBwgICAgICAgHBg
cFBgQEBAQCAQECAwQEBAYFBwYHCAgICAgICAcIBgcFBgQEBAICAsH6jAFKjPpu/Z3NwgJZ/dzDAf8
AAQkICAgHBwCGBgUFBAQCAgEBAQECAgQEBQUGBgCHBwgICAKIBwgHBwYGBQIEAwMCAQECAwMEBQUG
BgCHCAcICQgICAcHBwYGBQIEBAICAQEBAQICBAQFBQYGBwCHCAgICQgHCAcHBgYFBQQDAwIBAQIDA
wQFBQYGBwCIBwgB+wH+vQFABP5dOgGkAQEAAAADAAAAANkA5oAnQDxAR4AAAEzHwEdAR8HFQ8DIy
8HDwYdAR8WDw0dAQ8BKwIvAT0BLwC9AT8COWEfBj8HLxc/DTU/AwEfDjsBPxEvFiMPFR8BEw8CFR8
HMz8HNS8GIw8ELwQrAQ8BAGoCAGENDAwKCgHQBQEBaIkCAGIEAwQFDA0SBwCAGIBAQICBgCHBxYK
CQkKJCAcHBgYFBAMCAQEBAQIDAQFBQYGBwCPEQECAhUCAQINDAsLCQgHBQICKQICAgQDBAULdhIHB
wYCAQEBAQEBAgYHBwCWCgkKCAgHBwYFBQQDAgEBAQECAwMEBAYFBgCHEBABAQED/qwUFRUVFRYWFh
YWFxYXFhCwFxcWFxYXFhYWFhYVFRUVFAQCAQICBAUGCAGJCgsLDA0MDQ0NDBk2EQYgGgYGCEsZDQ0
NDA0MCwsKCQgIBgUEAgIBAgQCAQEBAwKRNRIHBqADChI1DQoFAGEBAGMEBAoMEw8eTw4IVxkXCwkJ
BwYCOAIBaIIdAwUGBwgJCgICAQENAQEFawIDAgECAGMFAwMEBAUDBAMFAwIBAQECAwMEBAUGBgYHC

```

```

ICQgHbwcGBgYFBQQEBAYDIgICAQECAiICBAUGBwgJCQMBAgEMAQUDawIDAQICBAQDBAQEBAQEAw
QEAgEBAQICBAMFBQUGBwcICAgJBwgHBgcGBgUFBAQEBOQIaAgEBAf6RDAsLCQkICAYGBQUDawIBAQI
DAwUFBgYICakJCwsMKSckIiAeGxoYFhQTERAPDQwLCgkIDxsJBQUFBQQnEakKCwwNDxARExQWGBob
HiAiJCCoAMDawQECA8XPRcKCGUPFz0REAKIBAMDawMCAQICAwcYAwEaBwQBAGIAAAEAAAAAAAA/MDN
AA0AAABDwQvAw8EHwQ/ETUnIw8LAYsEJwwGAgIwXmMXFBIICCSqKaEqRUclJSYnJykpKiosLC4GFg
sCAWMhISIiIiIjIkJAPRwB8AQmCQMBARQuNgsMDgcIJCYnmyZOTycmJiYlJSQjIiIgHwULCAMCAQ4
RERITFBUVKy0tFgAAABIA3gABAAAAAAAAAAAAEAAAAABAAAAAAAAABAAcAAQABAAAAAAAAACAACAAABAAAA
AAADAACADwABAAAAAAAAEAACAFgABAAAAAAAAFAAsAHQABAAAAAAAAAGAAcAKAABAAAAAAAAAKACwALwABA
AAAAAAAAALABIAWwADAAEECQAAAAIAbQADAAEECQABAA4AbwADAAEECQACAA4AfQADAAEECQADAA4AiW
ADAAEECQAEAA4AmQADAAEECQAFABYApwADAAEECQAGAA4AvQADAAEECQAKAFgAywADAAEECQALACQ
BIyBEZwZhdWx0UmVndWxhcRlZmFlbHREZWZhdWx0VmVyc2lvdBiBNZXRybyBTdHVkaW93d3cuc3luY2Z1c2lvdBi5jb20AIABEA
GUAZgBhAHUAbAB0AFIAZQBnAHUAbABhAHIAARABLAGYAYQB1AGwAdABEAGUAZgBhAHUAbAB0AFYAZQ
ByAHMAaQBvAG4AIAAxAAC4AMABEAGUAZgBhAHUAbAB0AEYAbwBuAHQAIAABnAGUAbgBIAHIAYQB0AGU
AZAAgAHUAcwBpAG4AZwAgAFMAeQBvAGMAZgB1AHMAaQBvAG4AIAABNAGUAdABYAG8AIABTAHQAdQBk
AGkAbwB3AHcAdwAuAHMAeQBvAGMAZgB1AHMAaQBvAG4ALgBjAG8AbQAAAAACAAAAAAAAAAAAoAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAYBAGEDAQQBBQEGAQcADXRyYW5zcG9ydC12YW4LdXNlcil1b2RpZn
kRc2hvcHBpbmctY2FydF8wMS0Lc3BlbmQtbW9uZXkFY2h1Y2sAAAA=) format('trueType');
font-weight: normal;
font-style: normal;
}
 stepper-text [class^="sf-icon-"], .stepper-text [class*=" sf-icon-"] {
font-family: 'Default' !important;
speak: none;
font-style: normal;
font-weight: normal;
font-variant: normal;
text-transform: none;
line-height: inherit;
-webkit-font-smoothing: antialiased;
-moz-osx-font-smoothing: grayscale;
}
 stepper-text {
margin-top: 30px;
padding: 30px;
}
 stepper-text .sf-icon-cart:before { content: "\e710"; }
 stepper-text .sf-icon-user:before { content: "\e708"; }
 stepper-text .sf-icon-transport:before { content: "\e702"; }
 stepper-text .sf-icon-payment:before { content: "\e706"; }
 stepper-text .sf-icon-success:before { content: "\e715"; }

```

## Optional steps

You can show whether the step is optional or not by using `optional` property. By default, the `optional` property is `false`.

## INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
  return (
    <div className="stepper-optional-step">
```

```

        <StepperComponent>
          <StepsDirective>
            <StepDirective iconCss= {'sf-icon-cart'}/>
            <StepDirective iconCss= {'sf-icon-transport'} />
            <StepDirective iconCss= {'sf-icon-payment'}
optional={true}/>
            <StepDirective iconCss= {'sf-icon-success'} />
          </StepsDirective>
        </StepperComponent>
      </div>
    );
  }
  export default App;
  ReactDOM.render(<App />, document.getElementById("element"));
  {% endraw %}

```

**INDEX.TSX**

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
  return (
    <div className="stepper-optional-step">
      <StepperComponent>
        <StepsDirective>
          <StepDirective iconCss= {'sf-icon-cart'}/>
          <StepDirective iconCss= {'sf-icon-transport'} />
          <StepDirective iconCss= {'sf-icon-payment'}
optional={true}/>
          <StepDirective iconCss= {'sf-icon-success'} />
        </StepsDirective>
      </StepperComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

**INDEX.CSS**

```

/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
@font-face {
  font-family: 'Default';
  src:

```



```
url(data:application/x-font-ttf;charset=utf-
8;base64,AAEAAAAKAIAAAwAgTlMvMjlvSgcAAAEoAAAAVmNtYXCDeIPaAAABmAAAAF5nbHlmEwr+
pwAAAggAAAJQaGVhZCYp2+EAADQAAAAANmhoZWEIUQQHAAAArAAAACRobXR4GAAAAAAAYAAAAAYb
G9jYQhUBlAAAAH4AAAAADm1heHABFgErAAABCAAAACBuYw1luF5THQAACtgAAAIlcG9zdJ8LuoMAAA
0AAAAAbwABAAAEAAAAAFwEAAAAAAD9AABAAAAAABAAAAAABAAAAAQAARxT6wV8PPPU
ACwQAAAAAAOGLy6UAAAAA4YvLpQAAAAAD9A0aAAAAACAACAAAAAABAAAAAAGAR8ABgAAAAAAAgAA
AAoACgAAAP8AAAAAQAQAAZAABQAAAokCzAAAAI8CiQLMAAAB6wAyAQgAAAIABQMAAAAAAABAAAA
AAAAAAAAAAAAAAAAAUGZFZABA5wLnFQQAAAAAXAQAAAAAABAAAAAABAAAAAQAABAAAAEAAAAABA
AAAAQAAAAEAAAAAAGgAAAAAUAAMAAQAAABQABABKAAAAADAAIAAIABoCc5wbnCOcQ5xX//wA
A5wLnBucI5xDnFf//AAAAAAAAAAAAAAAAABAawADAAMAawADAAAAAEABAACAAMABQAAAAAAAEQAIwC
3AQkBGgAAAAFAAAAAAP0A18APwB/AIkAxgDrAAABHw8/Dy8OKwEPDQUfDz8PLw4rAQ8NAR8FFSM1J
xEfBz80OwEfDjM/BzUnIw8GATM/Dx8PMxEhAq8BAQIEBAUFbWYICAgJCQoKCgkKCAkIBwcHBQIEAw
MBAQEBAwMEBQUHBwCICQgKCQoKCgkKCAgIBgcFBQqEAgH+CwEBAGQEBQUHBwGICAgJCgoKCQoICQg
HBwcFBQqDAwEBAQEDAwQFBQcHBwgJCAoJCgoKCQkICAgGBwUFBAQCAQJ8AwUIWawD3n0BAwMGBgYI
CAMEBQYHBwkJCgsLDA0NDQ4ODQ4MDAwLCgkJCAYGBQMDKAgIBwYFBAECvLsICAYHBQMD/beAAwQFB
QcHCAkKCgsLDA0MDg0NDQwLCwsJCQkHBwUFawKE/eMBAQoJCQkJCACHBgYFBAMDAQEBAQMDBAUGBg
cHCAkJCQkKCgoJCQgICAcGBgQFAwICAgIDBAUFbGcHCAkJCQoLCgkJCQkIBwcGBgUEAwMBAQEBAwM
EBQYGBwCICQkJCQoKCgkJCAGIBwYGBAUDAgICAgMEBQUGBwCICQkJCgGuAQIGehYJBKYp/10ICAcG
BQqCAQ0NDQwLCgoJCAgGBQUDAgIDBQUGCAGJCgoLDA0NDQECBAUGBwQI1foBAGQFBGcH/iwNDawLC
woJCQgHBgUEAwEBAQEDBAUGBwgJCQoLCwwMDQJJAABAgAAAAAD8wOWAAYAqgBaAgwArQDuAAABZ
cfAwUhLwIHlY8PNS8CKwIPHQEHLwEjDwE1LwMjNz0BJzcFBTcfAg8BLwY3OwEfAQcVHw8/Dy8PDw4
BFR8PPw8vDw8OAxEWBgEDAg8/wNuBAUEDQsVFBQTEhEPDw0GCwoIBgQCfHITE+wQDw8PDg4ODg0N
DQwNCwwKCwoKCQgJBwcHBgYEBQMEA5FrBAQDBAMBawMDBgIDagIEBgYGBxwCAwIBFQYGBAgFBgIWA
gQHCPcBAgQGBgKCGsMDQ4PDxAQEBApDw4NDAsLCQgGBgQCAQECAwYGCakLCwwNDg8PEBAQEAE8PDg
0MCwoKCAyGBAL+KgEEBQgKCw0PEBETFBQWFxcXFhYUFBMREQ4NDakIBgMBAQMGCakMDQ4RERMUFBY
WFxcXFhQUExEQDw0LCgGFBAEXBhcFBAMDrxYWDQEBawUHCAsMDQ4IERESFBQUFQQDAgECAGMEBAUG
BgYIBwgJCQoKCwsLDAwMDQ0ODQ4PDgEZawIBAQIGBQMCAQQDBgZqBgOHBQMDMAMHBwMWAQICBQYKC
hYCB1wICBAPDw4NDAsLCQgGBgQCAQECAwYGCakLCwwNDg8PEBAQEAE8PDg0MCwoKCAyGawMBAQMDBg
YICgoLDA0ODw8QATMLDBYVFRQSERAPDQsKCAUEAQEEBQgKCw0PEBESFBUVFhcXFxYVFBISEA8NCwo
IBQQAQgQFCa0LDQ8QEHlUFRYXAAAAAQAQAAAAA/QDRwA/AH8AhwCRAAABFR8OPw49AS8NKwEPDQUV
Hw4/Dj0BLw0rAQ8NEwCTIRMnMSMhMxMhNSEDBzUha0YBAgMEBAQGBQcGBwgICAgICAgIBwYHBgYEB
AQDagEBAGMEBAQGBQcGBwgICAgICAgIBwYHBgYEBAQDagH+aaICAgQEBAYFBwYIBwgICAgICAgHBg
cFBgQEBAMCAQECAwQEBAYFBwYHCAgICAgICAcIBgcFBgQEBaICAsH6jAFKjPpu/Z3NwgJZ/dzDAf8
AAQkICAgHBwCGBgUFBAQCAgEBAQECAgQEBQUGBgcHBwgICAkIBwgHBwYGBQUEAwMCAQECAwMEBQUG
BgCHCAcICQgICAcHBwYGBQUEBAICAQEBAQICBAQFBQYGBwCHCAgICQgHCAcHBgYFBQqDAwIBAQIDA
wQFBQYGBwCIBwgB+wH+vQFABP5dOgGkAQEAAAADAAAAANkA5oAnQDxAR4AAAEzHwEdAR8HFQ8DIy
8HDwYdAR8WDw0dAQ8BKwIvAT0BLwc9AT8COWEfBj8HLxc/DTU/AwEfDjsBPxEvFiMPFR8BEw8CFR8
HMz8HNS8GIw8ELwQrAQ8BAGoCAGENDAwKCgGHBQEBaIkCAGIEAwQFDA0SBwCAGIBAQICBgcHBxYK
CQkJCAcHBgUFBAQCAQEBAQIDAQFBQYGBwCPEQECAhUCAQINDAsLCQgHBQICKQICAgQDBAULdhIHB
wYCAQEBAQEBAgYHBwCWCgkKCAgHBwYFBQqDAgEBAQECAwMEBAYFBgCHEBABAQED/qwUFRUVFRYWFh
YWFxYXFhcXFxcWFxYXFhYWFhYVFRUVFAQCAQICBAUGCAGJCgsLDA0MDQ0NDBk2EQYQGgYGCEsZDQ0
NDA0MCwsKCQgIBgUEAgIBaQCAQEBAwKRNRIHBqADChI1DQoFAgEBAgMEBAoMEw8eTw4IVxkXCwkJ
BwYCOAIBAIDAwUGBwgJCgICAQENAQEFawIDAQECAGMFAwMEBAUDBAwMFAwIBAQECAwMEBAUGBgYHC
AgICQgHBwCGBgYFBQqEBAYDIgICAQECAICBAUGBwgJCQMBAGEMAQUDAwIDAQICBAQDBAQEBQEAw
QEAgEBAQICBAMFBQUGBwCICAgJBwgHBgCGBgUFBAQEBOQIAGeBAf6RDAsLCQkICAYGBQUDAwIBAQI
DAwUFBgYICAKJCwsMKSckIiAeGxoYFhQTERAPDQwLCgkIDxsJBQUBQqNEakKCwwNDxARExQWGBob
HiAiJCcCwQMDAwQECA8XPRcKCgUPFz0REAKIBAMDawMCAQICAwCYawEaBwQBAgIAAAEAAAAA/MDN
AA0AAABDQAwAw8EHwQ/ETUnIw8LAYsEJwwGAGIwXmMXFBIIcCsqKaEqRUclJSYnJykpKiosLC4GFg
sCAWMhISiIiIjIkJAPRwB8AQmCQMBARQuNgsMDgcIJCYnmyZOTycmJiYlJSQjIiIgHwULCAMCAQ4
RERITFBUVKy0tFgAAABIA3gABAAAAAABAAAAEAAAAAABAAAAAABAAAAAQAABAAAAAACAABAAAA
AAADAACADwABAAAAAEEAaAFgABAAAAAFAAAsAHQABAAAAAAGAAcAKAABAAAAAABKACwALwABA
AAAAAALABIAWwADAAEECQAAAAIAbQADAAEECQABAA4AbwADAAEECQACAA4AfQADAAEECQADAA4AiW
ADAAEECQAEAA4AmQADAAEECQAFABYApwADAAEECQAGAA4AvQADAAEECQAKAFgAywADAAEECQALACQ
BIyBEZWZhdWx0UmVndWxhcRlZmF1bHREZWZhdWx0VmVyc2lvbiAxLjBEZWZhdWx0Rm9udCBnZW5l
cmF0ZWQgdXNpbmcgU3luY2Z1c2lvbiBNZXRYbyBTdHVkaW93d3cuc3luY2Z1c2lvbi5jb20AIABEA
GUAZgBhAHUAbAB0AFIAZQBnAHUAbABhAHIArABLAGYAYQB1AGwAdABEAGUAZgBhAHUAbAB0AFYAZQ
ByAHMAaQBvAG4AIAAxAAC4AMABEAGUAZgBhAHUAbAB0AEYAbwBuAHQAIABnAGUAbgBlAHIAAYQB0AGU
```

```

AZAAGAHUAcwBpAG4AZwAgAFMAeQBvAGMAZgB1AHMAaQBvAG4AIABNAGUAdABYAG8AIABTAHQAdQBk
AGkAbwB3AHcAdwAuAHMAeQBvAGMAZgB1AHMAaQBvAG4ALgBjAG8AbQAAAAACAAAAAaAAAAA
AAAAAAAAAAAAAAAAAAAAAAYBAGEDAQQBBQEGAQcADXRyYW5zcG9ydC12YW4LdXNlci1tb2RpZn
kRc2hvcHBpbmctY2FydF8wMS0Lc3B1bmQtbW9uZXkFY2hlY2sAAAA=) format('true');
  font-weight: normal;
  font-style: normal;
}
.stepper-optional-step [class^="sf-icon-"], .stepper-optional-step [class*="
sf-icon-"] {
  font-family: 'Default' !important;
  speak: none;
  font-style: normal;
  font-weight: normal;
  font-variant: normal;
  text-transform: none;
  line-height: inherit;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}
.stepper-optional-step {
  margin-top: 30px;
  padding: 30px;
}
.stepper-optional-step .sf-icon-cart:before { content: "\e710"; }
.stepper-optional-step .sf-icon-transport:before { content: "\e702"; }
.stepper-optional-step .sf-icon-payment:before { content: "\e706"; }
.stepper-optional-step .sf-icon-success:before { content: "\e715"; }

```

### Disabling steps

You can use the `disabled` property to disable a step, preventing user interaction when set to `true`. By default, the value is `false`.

### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
"@syncfusion/ej2-react-navigations";
function App() {
  return (
    <div className="stepper-disabled-step">
      <StepperComponent>
        <StepsDirective>
          <StepDirective iconCss= {'sf-icon-cart'} />
          <StepDirective iconCss= {'sf-icon-transport'} />
          <StepDirective iconCss= {'sf-icon-payment'} />
          <StepDirective iconCss= {'sf-icon-success'} />
        </StepsDirective>
      </StepperComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));

```

```
{% endraw %}
```

## INDEX.TSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
 '@syncfusion/ej2-react-navigations';
function App() {
  return (
    <div className="stepper-disabled-step">
      <StepperComponent>
        <StepsDirective>
          <StepDirective iconCss= {'sf-icon-cart'} />
          <StepDirective iconCss= {'sf-icon-transport'} />
          <StepDirective iconCss= {'sf-icon-payment'}
disabled={true}/>
          <StepDirective iconCss= {'sf-icon-success'} />
        </StepsDirective>
      </StepperComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

## INDEX.CSS

```
/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
@font-face {
  font-family: 'Default';
  src:
    url(data:application/x-font-ttf;charset=utf-
8;base64,AAEAAAIAIAAAAwAgT1MvMjlvSgcAAAEoAAAAVmNtYXCDelPaAAABmAAAAF5nbHlmEwr+
pwAAAggAAAJQaGVhZCYp2+EAADQAAAAANmhoZWEIUQQHAAARAAAACRobXR4GAAAAAAAYAAAAAYb
G9jYQhUBlAAAAH4AAADm1heHABFgErAAABCAAAACBuYW1luF5THQAACTgAAAILcG9zdJ8LuoMAAA
0AAAAAbwABAAAEAAAAAFwEAAAAAAD9AABAAAAAIAAAAAAABgABAAAAAQAArxT6wV8PPPU
ACwQAAAAAOGly6UAAAAA4YvLpQAAAAAD9AOaAAAAACAACAAAAAIAAAAAAAGAR8ABgAAAAAAgAA
AAoACgAAAP8AAAAAIAAAQAAZAABQAAAokCzAAAAI8CiQLMAAB6wAyAQgAAAIABQMAAAAAAIAAA
AAAAAAAAAAAAAAAAAUGZFZABA5wLnFQQAAAAAXAQAAAAAABAAAAAABAAAAAQAAAAEAAAABA
AAAAQAAAAEAAAAAAGAAAAAIAAAUAAQAABQABABKAAAADAAIAAIABoC5wbnCOcQ5xX//wA
A5wLnBucI5xDnFf//AAAAAAAAAAAAAAAAABAawADAAMAawADAAAAEABAACAAMABQAAAAAAAEQAiwC
3AQkBGGAAAAFAAAAAAP0A18APwB/AIkAxgDrAAABHw8/Dy8OKwEPDQUFDz8PLw4rAQ8NAR8FFSM1J
xEfBz80OwEfDjM/BzUnIw8GATM/Dx8PMxEhAq8BAQIEBAUFBwYICAgJCQoKCgkKCAkIBwCHBQUEAw
MBAQEBawMEBQUHBwCICQgKCQoKCgkKCAgIBgcFBQQEAgH+CwEBAgQEBQUHBggICakJCgoKCQoICQg
HBwCFBQQDAwEBAQEDAwQFBQcHBwgJCAoJCgoKCQkICAgGBwUFBAQCAQJ8AwUIWAwD3n0BAwMGBgYI
```

```

CAMEBQYHBwkJCgsLDA0NDQ4ODQ4MDAwLCgkJCAYGBQMDKAgIBwYFBAECvLsICAYHBQMD/beAAwQFB
QcHCAkKCgsLDA0MDg0NDQwLCwsJCQkHBwUFawKE/eMBAQoJCQkJCAcHBgYFBAMDAQEBAQMDBAUGBg
cHCAkJCQkKCgoJCQgICAcGBgQFAwICAgIDBAUFBgcHCAkJCQoLCgkJCQkIBwCGBgUEAwMBAQEBAwM
EBQYGBwCICQkJCQoKCgkJCAgIBwYGBAUDAgICAgMEBQUGBwCICQkJCgGuAQIGehYJBKYp/10ICAcG
BQCAQ0NDQwLCgoJCAgGBQUdAgIDBQUGCAgJCgoLDA0NDQECEBAUGBwQI1foBAGQFBgcH/iwNDawLC
woJCQgHBgUEAwEBAQEDBAUGBwgJCQoLCwwMDQJJAABAgAAAAAD8wOWAAYAqgBaAGwArQDuAAABZ
cfAwUhLwIHIY8PNS8CKwIPHQEHLwEjDwE1LwMjNz0BJzcFbTcfAg8BLwY30wEfaQcVHw8/Dy8PDw4
BFR8PPw8vDw8OAxEWBgEDAgb8/wNuBAUEDQsVFBQTEhEPDw0GCwoIBgQCFhITE+wQDw8PDg4ODg0N
DQwNCwwKCwoKCQgJBwCHBgYEBQMEA5FrBAQDBAMBAwMDBgIDagIEBgYGBxwCAwIBFYGBAgFBgIWA
gQHCPCBAgQGBgKcgsMDQ4PDxAQEBApDw4NDAsLCQgGBgQCAQECEBAYGCAkLCwwNDg8PEBAQE8PDg
0MCwoKCAyGBAL+KgEEBQgKCw0PEBETFBQWFxcXFhYUFBMREQ4NDakIBgMBAQMGCakMDQ4RERMUFBY
WFxcXFhJUEExEQDw0LCggFBAEXBhcFBAMDrxYWDQEBawUHCAsmDQ4IERESFBQUDQDagECAGMEBAUG
BgYIBwgJCQoKCwsLDAwMDQ0ODQ4PDgEZawIBAQIGBQMCAQQDBgZqBgOHBQMDFMAMHwMWAQICBQYKC
hYCB1wICBAPDw4NDAsLCQgGBgQCAQECEBAYGCAkLCwwNDg8PEBAQE8PDg0MCwoKCAyGAwMBAQMDBg
YICgoLDA0ODw8QATMLDBYVFRQSERAPDQsKCAUEAQEEBQgKCw0PEBESFBUVFhcXFxYVFBISEA8NCwo
IBQQBAQQFCAoLDQ8QEhIUFRYXAAAAAQAQAAAAA/QDRwA/AH8AhwCRAAABFR8OPw49AS8NKwEPDQUV
Hw4/Dj0BLw0rAQ8NEwcTIRMnMSMhMxMhNSEDBzUha0YBAGMEBAQGBQcGBwgICAgICAgIBwYHBQYEB
AQDagEBAgMEBAQGBQcGBwgICAgICAgIBwYHBQYEBAQDagH+aAICAgQEBAYFBwYIBwgICAgICAgHBg
cFBgQEBAMCAQECAwQEBAYFBwYHCAGICAgICAcIBgcFBgQEBaICAsh6jAFKjPpu/Z3NwgJZ/dzDAf8
AAQkICAgHBwCGBgUFBAQCAgEBAQECAgQEBQUGBgCHBwgICakIBwgHBwYGBQUEAwMCAQECAwMEBQUG
BgCHCAcICQgICAcHBwYGBQUEBAICAQEBAQICBAQFBQYGBwCHCAgICQgHCAcHBgYFBQDQAwIBAQIDA
wQFBQYGBwCIBwgB+wH+vQFABP5dOgGkAQEAAAADAAAAANkA5oAnQDxAR4AAAEzHwEdAR8HFQ8DIY
8HDwYdAR8WDw0dAQ8BKwIvAT0BLwc9AT8COWEfBj8HLxc/DTU/AwEfDjsBPxEvFiMPFR8BEw8CFR8
HMz8HNS8GIw8ELwQrAQ8BAGoCAgENDAwKCgGHBQEBaikaAgIEAwQFDA0SBwCGAgIBAQICBgCHBxYK
CQkJCAcHBgUFBAQCAQEBAQIDAQwQFBQYGBwCPEQECAhUCAQINDAsLCQgHBQICKQICAgQDBAULdhIHB
wYCAQEBAQEBAgYHBwCWCgkKCAgHBwYFBQDQAgEBAQECAwMEBAYFBgcHEBABAQED/qwUFRUVFRYWFh
YWFxYXFhcXFxcWFxYXFhYWFhYVFRUVFAQCAQICBAUGCAgJCgsLDA0MDQ0NDBk2EQYgqgYGCEsZDQ0
NDA0MCwsKCQgIBgUEAgIBaQCAQEBAwKRNRIHBqADChI1DQoFagEBAgMEBAoMEw8eTw4IVxkXCwkJ
BwYCOAIBaIIDAUGBwgJCgICAQENAEFAwIDAQECAGMFAwMEBAUDBAMFAwIBAQECAwMEBAUGBgYHC
AgICQgHBwCGBgYFBQUEBAYDIgICAQECAiICBAUGBwgJCQMBAGEMAQUDAwIDAQICBAQDBAQEBAQEA
QEAgEBAQICBAMFBQUGBwCICAgJBwgHBgcGBgUFBAQEBAQIqAgEBAf6RDAsLCQkICAYGBQUDAwIBAQI
DAwUFBgYICakJCwsMKsckIiAeGxoYFhQTERAPDQwLCgkIDxsJBQUBQqNEakKCwwNDxARExQWGBob
HiAiJCcCoAMDawQECA8XPRcKCgUPFz0REakIBAMDawMCAQICAwcYAwEaBwQBAGIAAAEAAAAAAAA/MDN
AA0AAABDwQvAw8EHwQ/ETUnIw8LAYsEJwwGAgIwXmMXFBIIcCsKqKaEqRUclJSYnJykpKiosLC4GFg
sCAWMhISiIiIjIkJAPRwB8AQmCQMBARQuNgsMDgcIJCYnmyZOTycmJiYlJSQjIiIgHwULCAMCAQ4
RERITFBUVKy0tFgAAABIA3gABAAAAAAAAAAAAAABAAAAAABAAcAAQABAAAAAACAACAABAAAA
AAADAACADwABAAAAAAAEAAcAFgABAAAAAAFAAsAHQABAAAAAAGAACAABAAAAAAKACwALwABA
AAAAAALABIAWwADAAEECQAAAAIAbQADAAEECQABAA4AbwADAAEECQACAA4AfQADAAEECQADAA4AiW
ADAAEECQAEAA4AmQADAAEECQAFABYApwADAAEECQAGAA4AvQADAAEECQAKAFgAywADAAEECQALACQ
BIyBEZWZhdWx0UmVndWxhcRlZmFlbHREZWZhdWx0VmVyc2lvbiAxLjBEZWZhdWx0Rm9udCBnZW5l
cmF0ZWQgdXNpbmcgU3luY2Z1c2lvbiBNZXRYbyBTdHVkaW93d3cuc3luY2Z1c2lvbi5jb20AIABEA
GUAZgBhAHUAbAB0AFIAZQBnAHUAbABhAHIArABlAGYAYQB1AGwAdABEAGUAZgBhAHUAbAB0AFYAZQ
ByAHMAaQBvAG4AIAAxAx4AMABEAGUAZgBhAHUAbAB0AEYAbwBuAHQAIAABnAGUAbgBIAHIAAYQB0AGU
AZAAgAHUAcwBpAG4AZwAgAFMAeQBwAGMAZgB1AHMAaQBvAG4AIAABNAGUAdABYAG8AIAABTAHQAdQBk
AGkAbwB3AHcAdwAuAHMAeQBwAGMAZgB1AHMAaQBvAG4ALgBjAG8AbQAAAAACAAAAAoooooAAAAA
AAAAAAAAAAAAAAAAAAAAAYBAGEDAQBBQEGAQcADXRyYW5zcG9ydC12YW4LdXNlcil1tb2RpZn
kRc2hvcHBpbmctY2FydF8wMS0Lc3BlbmQtbW9uZXkFY2hly2sAAAA=) format('trueype');
font-weight: normal;
font-style: normal;
}
.stepper-disabled-step [class^="sf-icon-"], .stepper-disabled-step [class*="
sf-icon-"] {
font-family: 'Default' !important;
speak: none;
font-style: normal;
font-weight: normal;
font-variant: normal;

```

```

text-transform: none;
line-height: inherit;
-webkit-font-smoothing: antialiased;
-moz-osx-font-smoothing: grayscale;
}
.stepper-disabled-step {
margin-top: 30px;
padding: 30px;
}
.stepper-disabled-step .sf-icon-cart:before { content: "\e710"; }
.stepper-disabled-step .sf-icon-transport:before { content: "\e702"; }
.stepper-disabled-step .sf-icon-payment:before { content: "\e706"; }
.stepper-disabled-step .sf-icon-success:before { content: "\e715"; }

```

### Setting active step

You can set the active step by specifying its index using the [activeStep](#) property. The default value is 0.

#### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
  return (
    <div className="stepper-activestep">
      <StepperComponent activeStep={1}>
        <StepsDirective>
          <StepDirective iconCss= {'sf-icon-cart'} />
          <StepDirective iconCss= {'sf-icon-transport'} />
          <StepDirective iconCss= {'sf-icon-payment'} />
          <StepDirective iconCss= {'sf-icon-success'} />
        </StepsDirective>
      </StepperComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

#### INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
  return (
    <div className="stepper-activestep">
      <StepperComponent activeStep={1}>
        <StepsDirective>
          <StepDirective iconCss= {'sf-icon-cart'} />

```

```

        <StepDirective iconCss= {'sf-icon-transport'} />
        <StepDirective iconCss= {'sf-icon-payment'} />
        <StepDirective iconCss= {'sf-icon-success'} />
      </StepsDirective>
    </StepperComponent>
  </div>
);
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

## INDEX.CSS

```

/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
@font-face {
  font-family: 'Default';
  src:
    url(data:application/x-font-ttf;charset=utf-
8;base64,AAEAAAAKAIAAAwAgTlMvMjlvSgcAAAEoAAAAVmNtYXCDeIPaAAABmAAAAF5nbHlmEwr+
pwAAAggAAAJQaGVhZCYp2+EAADQAAAAANmhoZWEIUQQHAAAArAAAACRobXR4GAAAAAAAYAAAAAYb
G9jYQhUBlAAAAH4AAAAADm1heHABFgErAAABCAAAACBuYW11luF5THQAACtgAAAIlcG9zdJ8LuoMAAA
0AAAAAbwABAAAAEAAAAAFwEAAAAAAD9AABAAAAAABAAAAAABAAAAAQAARxT6wV8PPPU
ACwQAAAAAOGly6UAAAAA4YvLpQAAAAAD9AOaAAAAACAACAAAAAABAAAAAAGAR8ABgAAAAAAAgAA
AAoACgAAAP8AAAAAQAQAAZAABQAAAokCzAAAAI8CiQLMAAAB6wAyAQgAAAIABQMAAAAAAABAAAA
AAAAAAAAAAAAAAAAAUGZFZABA5wLnFQQAAAAAXAQAAAAAABAAAAAABAAAAAQAQAAAAEAAAAABA
AAAAQAQAAAAEAAAAAAGAAAAAAMAAUAAMAAQAABQABABKAAAAADAAIAAIABoC5wbnCOcQ5xX//wA
A5wLnBucI5xDnFf//AAAAAAAAAAAAAAAAAABAawADAAMAawADAAAAAEABAACAAMABQAAAAAABEQaiwC
3AQkBGgAAAAFAAAAAAP0A18APwB/AIkAxgDrAAABHw8/Dy8OKwEPDQUfDz8PLw4rAQ8NAR8FFSM1J
xEfBz80OwEfDjM/BzUnIw8GATM/Dx8PMxEhAq8BAQIEBAUFBwYICAgJCQoKCgkKCAkIBwcHBQUEAw
MBAQEBAwMEBQUHBwCICQgKCQoKCgkKCAgIBgcFBQQAeAgH+CwEBAgQEBQUHBggICakJCgoKCQoICQg
HBwcFBQQAeBAQEDAwQFBQcHBwgJCAoJCgoKCQkICAgGBwUFBAQCAQJ8AwUIWAwD3n0BAwMGBgYI
CAMEBQYHBwkJCgsLDA0NDQ4ODQ4MDAwLCgkJCAYGBQMDKAgIBwYFBAECvLsICAYHBQMD/beAAwQFB
QcHCAkKCgsLDA0MDg0NDQwLCwsJCQkHBwUFawKE/eMBAQoJCQkKJCAcHBgYFBAMDAQEBAQMDBAUGBg
cHCAkKJCQkKCgoJCQgICAcGBgQFAwICAgIDBAUFBgHCAkKJCQoLCgkKJCkIBwcGBgUEAwMBAQEBAwM
EBQYGBwCICQkKJCQoKCgkKCAgIBwYGBAUDAgICAgMEBQUGBwCICQkKJCgGuAQIGehYJBKYp/10ICAcG
BQCAQ0NDQwLCgoJCAGGBQUdAgIDBQUGCAgJCgoLDA0NDQECBAUGBwQI1foBAGQFBgcH/iwNDawLC
woJCQgHBgUEAwEBAQEDBAUGBwgJCQoLCwWMDQJJAAAABgAAAAAD8wOWAAYAQgBaAGwArQDuAAABZz
cfAwUhLwIHlY8PNS8CKwIPHQEHLwEjDwE1LwMjNz0BJzcFBTcfAg8BLwY3OwEfAQcVHw8/Dy8PDw4
BFR8PPw8vDw8OAxEWBgEDAg8/wNuBAUEDQsVFBQTEhEPDw0GCwoIBgQCFhITE+wQDw8PDg4ODg0N
DQwNCwwKCwoKCQgJBwcHBgYEBQMEA5FrBAQDBAMBawMDBgIDagIEBgYGBxwCAwIBFQYGBAgFBgIWA
gQHCPcBAGQGBgKCGsMDQ4PDxAQEBAPDw4NDAsLCQgGBgQCAQECBAYGCAkLCwwNDg8PEBAQEAE8PDg
0MCwoKCAYGBAL+KgEEBQgKCw0PEBETFBQWFxcXFhYUFBMREQ4NDakIBgMBAQMGCakMDQ4RERMUFBY
WFxcXFhQUExEQDw0LCgGFBAEXBhcFBAMDrxYWDQEBawUHCAsMDQ4IERESFBQUFQQDagECAGMEBAUG
BgYIBwgJCQoKCwsLDAwMDQ0ODQ4PDgEZawIBAQIGBQMCAQQDBgZqBgoHBQMDMAMHBwMWAQICBQYKC
hYCB1wICBAPDw4NDAsLCQgGBgQCAQECBAYGCAkLCwwNDg8PEBAQEAE8PDg0MCwoKCAYGawMBAQMDBg
YICgoLDA0ODw8QATMLDBYVFRQSERAPDQsKCAUEAQEEBQgKCw0PEBESFBUVFhcXFxYVFBISEA8NCwo
IBQQAQQAQFCAoLDQ8QEhIUFRYXAAAAAQAQAAAAA/QDRwA/AH8AhwCRAAABFR8OPw49AS8NKwEPDQUV

```

```

Hw4/Dj0BLw0rAQ8NEwcTIRMnMSMhMxMhNSEDBzUhA0YBAgMEBAQGBQcGBwgICAgICAgIBwYHBQYEB
AQDAgEBAGMEBAQGBQcGBwgICAgICAgIBwYHBQYEB AQDAgH+aAICAgQEBAYFBwYIBwgICAgICAgHBg
cFBgQEBAMCAQECAwQEBAYFBwYHCAGICAgICAcIBgcFBgQEBAICAsH6jAFKjPpu/Z3NwgJZ/dzDAf8
AAQkICAgHBwcGBgUFBAQCAgEBAQECAgQEBQUGBgCHBwgICAkIBwgHBwYGBQUEAwMCAQECAwMEBQUG
BgCHCACICQgICAcHBwYGBQUEBAICAQEBAQICBAQFBQYGBwCHCAGICQgHCACHBgYFBQQDAwIBAQIDA
wQFBQYGBwCIBwgB+wH+vQFABP5dOgGkAQEAAAADAAAAANkA5oAnQDxAR4AAAEzHwEdAR8HFQ8DIY
8HDwYdAR8WDw0dAQ8BKwIvAT0BLwc9AT8COWEfBj8HLxc/DTU/AwEfDjsBPxEvFiMPFR8BEw8CFR8
HMz8HNS8GIw8ELwQrAQ8BAGoCAGENDAwKCgGHBQEBAikCAGIEAwQFDA0SBwcGAGIBAQICBgCHBxYK
CQkJCACHBgUFBAMCAQEBAQIDA wQFBQYGBwCPEQECAhUCAQINDAsLCQgHBQICKQICAgQDBAULdhIHB
wYCAQEBAQEBAgYHBwCWCgkKCAGHBwYFBQQDAgEBAQECAwMEBAYFBgcHEBABAQED/qwUFRUVFRYWFh
YWFxYXFhCxFxcWFxYXFhYWFhYVFRUVFAQCAQICBAUGCAgJCgsLDA0MDQ0NDBk2EQYGGgYGCEsZDQ0
NDA0MCwsKCQgIBgUEAgIBAgQCAQEBAwKRNRIHBqADChI1DQoFAgEBAgMEBAoMEw8eTw4IVxkXCwkJ
BwYCOAIBAiIDA wUGBwgJCgICAQENAQEFawIDA gECAGMFAwMEBAUDBAMFAwIBAQECAwMEBAUGBgYHC
AgICQgHBwCGBgYFBQQEBAYDIgICAQECAiICBAUGBwgJCQMBAGEMAQUDAwIDAQICBAQDBAQEBAGQEAw
QEAgEBAQICBAMFBQUGBwCICAgJBwgHBgcGBgUFBAQEBOQiAgEBAf6RDAsLCQkICAYGBQUdAwIBAQI
DAwUFBgYICAKJCwsMKsckIiAeGxoYFhQTERAPDQwLCgkIDxsJBQUBQqNEAkKCwwNDxARExQWGBob
HiAiJCcCoAMDawQECA8XPRcKCgUPFz0REAkIBAMDawMCAQICAwCYAwEaBwQBAGIAAAEAAAAAA/MDN
AA0AAABDwQvAw8EHwQ/ETUnIw8LAYsEJwwGagIwXmMXFBIICCsqKaEqRUclJSYnJykpKiosLC4GFg
sCAWMhISiIiIjIkJAPRwB8AQmCQMBARQuNgsMDgcIJCYnmyZOTycmJiYlJSQjIiIgHwULCAMCAQ4
RERITFBUVKy0tFgAAABIA3gABAAAAAEEEEAAAAAABAAAAAABAAcAAQABAAAAAACAACAABAAAA
AAADAACADwABAAAAAEEAAcAFgABAAAAAFAAsAHQABAAAAAAGAAcAKAABAAAAAAKACwALwABA
AAAAAALABIAWwADAAEEECQAAAAIAbQADAAEEECQABAA4AbwADAAEEECQACAA4AfQADAAEEECQADAA4Aiw
ADAAEEECQAEAA4AmQADAAEEECQAFABYApwADAAEEECQAGAA4AvQADAAEEECQAKAFgAywADAAEEECQALACQ
BIyBEZWZhdWx0UmVndWxhckRlZmFlbHREZWZhdWx0VmVyc2lvbiAxLjBEZWZhdWx0Rm9udCBnZW5l
cmF0ZWQgdXNpbmcgU3luY2Zlc2lvbiBNZXRYbyBTdHVkaW93d3cuc3luY2Zlc2lvbi5jb20AIABEA
GUAZgBhAHUAbAB0AFIAZQBnAHUAbABhAHIArABlAGYAYQB1AGwAdABEAGUAZgBhAHUAbAB0AFYAZQ
ByAHMAaQBvAG4AIAAxA4AMABEAGUAZgBhAHUAbAB0AEYAbwBuAHQAIABnAGUAbgB1AHIAIAYQB0AGU
AZAAgAHUAcwBpAG4AZwAgAFMAeQBwAGMAZgB1AHMAaQBvAG4AIAIBNAGUAdABYAG8AIABTAHQAdQBk
AGkAbwB3AHcAdwAuAHMAeQBwAGMAZgB1AHMAaQBvAG4ALgBjAG8AbQAAAAACAAAAAaAAAAAaAAAAA
AAAAAAAAAAAAAAAAAAAAAYBAgEDAQQBBQEQAQcADXRyYW5zcG9ydC12YW4LdXNlcil1tb2RpZn
kRc2hvcHBpbmctY2FydF8wMS0Lc3BlbmQtbW9uZXkFY2h1Y2sAAAA=) format('truetype');
    font-weight: normal;
    font-style: normal;
}
.stepper-activestep [class^="sf-icon-"], .stepper-activestep [class*=" sf-
icon-"] {
    font-family: 'Default' !important;
    speak: none;
    font-style: normal;
    font-weight: normal;
    font-variant: normal;
    text-transform: none;
    line-height: inherit;
    -webkit-font-smoothing: antialiased;
    -moz-osx-font-smoothing: grayscale;
}
.stepper-activestep {
    margin-top: 30px;
    padding: 30px;
}
.stepper-activestep .sf-icon-cart:before { content: "\e710"; }
.stepper-activestep .sf-icon-transport:before { content: "\e702"; }
.stepper-activestep .sf-icon-payment:before { content: "\e706"; }
.stepper-activestep .sf-icon-success:before { content: "\e715"; }

```

### Step status

Each step's progress state can be specified using the `status` property. The possible values are `NotStarted`, `InProgress` and `Completed`. By default, the value is `NotStarted`.

#### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
    let stepperObj = React.useRef(null);
    const [color, setColor] = React.useState("#e74d4d");
    const [text, setText] = React.useState("Your payment has not started
yet");
    const statusMap = {
        'NotStarted': { text: 'Your payment has not started yet', color:
'#e74d4d' },
        'InProgress': { text: 'Processing your payment', color: 'orange' },
        'Completed': { text: 'Payment successful', color: '#4CAF50' }
    };
    const handleStepChange = () => {
        let status = stepperObj.current.steps[1].status;
        setColor(statusMap[status].color);
        setText(statusMap[status].text);
    }
    return (
        <div className="stepper-status">
            <StepperComponent ref={stepperObj}
stepChanged={handleStepChange}>
                <StepsDirective>
                    <StepDirective iconCss= {'sf-icon-cart'} label=
{'Cart'}/>
                    <StepDirective iconCss= {'sf-icon-payment'} label=
{'Payment'}/>
                    <StepDirective iconCss= {'sf-icon-success'} label=
{'Ordered'}/>
                </StepsDirective>
            </StepperComponent>
            <div id="paymentStatus" style={{ marginTop: '50px',
backgroundColor: color }}> {text} </div>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

#### INDEX.TSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
```



```
function App() {
  let stepperObj = React.useRef<StepperComponent>(null);
  const [color, setColor] = React.useState("#e74d4d");
  const [text, setText] = React.useState("Your payment has not started yet");
  const statusMap = {
    'NotStarted': { text: 'Your payment has not started yet', color: '#e74d4d' },
    'InProgress': { text: 'Processing your payment', color: 'orange' },
    'Completed': { text: 'Payment successful', color: '#4CAF50' }
  };
  const handleStepChange = () => {
    const { status } = stepperObj.current.steps[1];
    setColor(statusMap[status].color);
    setText(statusMap[status].text);
  }
  return (
    <div className="stepper-status">
      <StepperComponent ref={stepperObj}
stepChanged={handleStepChange}>
        <StepsDirective>
          <StepDirective iconCss= {'sf-icon-cart'} label=
{'Cart'}/>
          <StepDirective iconCss= {'sf-icon-payment'} label=
{'Payment'}/>
          <StepDirective iconCss= {'sf-icon-success'} label=
{'Ordered'}/>
        </StepsDirective>
      </StepperComponent>
      <div id="paymentStatus" style={{ marginTop: '50px',
backgroundColor: color }}> {text} </div>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

## INDEX.CSS

```
/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
@font-face {
  font-family: 'Default';
  src:
url(data:application/x-font-ttf;charset=utf-
8;base64,AAEAAAAKIAAAwAgTlMvMjlvSgcAAAEoAAAAVmNtYXCDelPaAAABmAAAAF5nbHlmEwr+
pwAAAggAAAJQaGVhZCYp2+EAADQAAAAANmhoZWEIUQQHAAAArAAAACRobXR4GAAAAAAAAYAAAAAYb
G9jYQhUBlAAAAH4AAAADmlheHABFgErAAABCAAAACBuYWllluF5THQAACTgAAAIlcG9zdJ8LuoMAAA
```

Copyright © 2001 -2024 Syncfusion Inc. 1222

```

    font-weight: normal;
    font-style: normal;
  }
  .stepper-status [class^="sf-icon-"], .stepper-status [class*=" sf-icon-"] {
    font-family: 'Default' !important;
    speak: none;
    font-style: normal;
    font-weight: normal;
    font-variant: normal;
    text-transform: none;
    line-height: inherit;
    -webkit-font-smoothing: antialiased;
    -moz-osx-font-smoothing: grayscale;
  }
  .stepper-status {
    width: 75%;
    margin: 0px auto;
    min-width: 85px;
    padding: 25px 0;
  }
  .stepper-status #paymentStatus {
    padding: 10px;
    background-color: #e74d4d;
    color: white;
    border-radius: 5px;
    margin: 10px auto;
    text-align: center;
    font-weight: bold;
    max-width: 350px;
  }
  .stepper-status .sf-icon-cart:before { content: "\e710"; }
  .stepper-status .sf-icon-user:before { content: "\e708"; }
  .stepper-status .sf-icon-transport:before { content: "\e702"; }
  .stepper-status .sf-icon-payment:before { content: "\e706"; }
  .stepper-status .sf-icon-success:before { content: "\e715"; }

```

### Step styling

You can use the `cssClass` property to customize the appearance of each step.

#### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
"@syncfusion/ej2-react-navigations";
function App() {
  return (
    <div className="stepper-custom-style">
      <StepperComponent>
        <StepsDirective>
          <StepDirective label={"Cart"} iconCss= {'sf-icon-cart'}
        />
          <StepDirective label={"Delivery Address"} iconCss= {'sf-
icon-transport'} />

```

```

        <StepDirective label={"Payment"} iconCss= {'sf-icon-
payment'} cssClass="custom-step" optional={true}/>
        <StepDirective label={"Confirmation"} iconCss= {'sf-icon-
success'} />
      </StepsDirective>
    </StepperComponent>
  </div>
);
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

### INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
  return (
    <div className="stepper-custom-style">
      <StepperComponent>
        <StepsDirective>
          <StepDirective label={"Cart"} iconCss= {'sf-icon-cart'}
/>
          <StepDirective label={"Delivery Address"} iconCss= {'sf-
icon-transport'} />
          <StepDirective label={"Payment"} iconCss= {'sf-icon-
payment'} cssClass="custom-step" optional={true}/>
          <StepDirective label={"Confirmation"} iconCss= {'sf-icon-
success'} />
        </StepsDirective>
      </StepperComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

### INDEX.CSS

```

/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
@font-face {
  font-family: 'Default';
  src:

```

```
url(data:application/x-font-ttf;charset=utf-
8;base64,AAEAAAAKIAIAAwAgTlMvMjlvSgcAAAEoAAAAVmNtYXCDeIPaAAABmAAAAF5nbHlmEwr+
pwAAAggAAAJQaGVhZCYp2+EAADQAAAAANmhoZWEIUQQHAAAArAAAACRobXR4GAAAAAAAYAAAAAYb
G9jYQhUBlAAAAH4AAAAADm1heHABFgErAAABCAAAACBuYW11uF5THQAACtgAAAIlcG9zdJ8LuoMAAA
0AAAAAbwABAAAEAAAAAFwEAAAAAAD9AABAAAAAABAAAAAABAAAAAQAARxT6wV8PPPU
ACwQAAAAAAGLy6UAAAAA4YvLpQAAAAAD9A0aAAAAACAACAAAAAABAAAAAAGAR8ABgAAAAAAGAA
AAoACgAAAP8AAAAAQAQAAZAABQAAAokCzAAAAI8CiQLMAAAB6wAyAQgAAAIABQMAAAAAA
AAAAAUGZFZABA5wLnFQQAAAAAXAQAAAAAABAAAAAABAAAAAQAQAAEAAAABA
AAAAQAAAAEAAAAAAGAAAAAUAAMAAQAAABQABABKAAAAADAAIAAIABoC5wbnCOcQ5xX//wA
A5wLnBucI5xDnFf//AAAAAABAAwADAAMAawADAAAAEABAACAAMABQAAAAAAAEQAIwC
3AQkBGgAAAAFAAAAAAP0A18APwB/AIkAxgDrAAABHw8/Dy8OKwEPDQUfDz8PLw4rAQ8NAR8FFSM1J
xEfBz800wEfDjM/BzUnIw8GATM/Dx8PMxEhAq8BAQIEBAUFBwYICAgJCQoKCgkKCAkIBwcHBQEAw
MBAQEBAwMEBQUHBwCICQgKCQoKCgkICAgIBgcFBQQEAgH+CwEBAGQEBQUHBwCICAgJCgoKCQoICQg
HBwcFBQQDAwEBAQEDAwQFBQcHBwgJCAoJCgoKCQkICAgGBwUFBAQCAQJ8AwUIWawD3n0BAwMGBgYI
CAMEBQYHBwkJCgsLDA0NDQ4ODQ4MDAwLCgkJCAYGBQMDKAgIBwYFBAECvLsICAYHBQMD/beAAwQFB
QcHCAkKCgsLDA0MDg0NDQwLCwsJCQkHBwUFawKE/eMBAQoJCQkJCACHBgYFBAMDAQEBAQMDBAUGBg
cHCAkJCQkKCgoJCQgICAcGBgQFAwICAgIDBAUFBgHCAkJCQoLCgkJCQkIBwcGBgUEAwMBAQEBAwM
EBQYGBwCICQkJCQoKCgkJCAGIBwYGBAUDAgICAgMEBQUGBwCICQkJCgGuAQIGehYJBKYp/10ICAcG
BQQAQ0NDQwLCgoJCAgGBQUDAgIDBQUGCAGJCgoLDA0NDQCEBAUGBwQI1foBAGQFBgCH/iwNDawLC
woJCQgHBgUEAwEBAQEDBAUGBwgJCQoLCwwMDQJJAABAgAAAAAD8wOWAAYAqgBaAGwArQDuAAABZ
cfAwUhLwIHlY8PNS8CKwIPHQEHLwEjDwE1LwMjNz0BJzcFBTcfAg8BLwY3OwEfAQcVHw8/Dy8PDw4
BFR8PPw8vDw8OAxEWBgEDAg8/wNuBAUEDQsVFBQTEhEPDw0GCwoIBgQCfHITE+wQDw8PDg4ODg0N
DQwNCwwKCwoKCQgJBwcHBgYEBQMEA5FrBAQDBAMBawMDBgIDagIEBgYGBxwCAwIBFYQYGBAGFBgIWA
gQHCPcBAGQGBgKCGsMDQ4PDxAQEBApDw4NDAsLCQgGBgQCAQECAwYGCakLCwwNDg8PEBAQEAE8PDg
0MCwoKCAyGBAL+KgEEBQgKCw0PEBETFBQWFxcXFhYUFBMREQ4NDakIBgMBAQMGCakMDQ4RERMUFBY
WFxcXFhQUExEQDw0LCggFBAEXBhcFBAMDrxYWDQEBawUHCAsmDQ4IERESFBQUFQQDAgECAGMEBAUG
BgYIBwgJCQoKCwsLDAwMDQ0ODQ4PDgEZawIBAQIGBQMCAQQDBgZqBgOHBQMDMAMHBwMWAQICBQYKC
hYCB1wICBAPDw4NDAsLCQgGBgQCAQECAwYGCakLCwwNDg8PEBAQEAE8PDg0MCwoKCAyGawMBAQMDBg
YICgoLDA0ODw8QATMLDBYVFRQSERAPDQsKCAUEAQEEBQgKCw0PEBESFBUVFhcXFxYVFBISEA8NCwo
IBQQAQCFCAoLDQ8QEHlUFRYXAAAAAQAQAAAAA/QDRwA/AH8AhwCRAAABFR8OPw49AS8NKwEPDQUV
Hw4/Dj0BLw0rAQ8NEwCTIRMnMSMhMxMhNSEDBZUha0YBAQMEBAQGBQcGBwgICAgICAgIBwYHBQYEB
AQDagEBAGMEBAQGBQcGBwgICAgICAgIBwYHBQYEBAQDagH+aaICAgQEBAYFBwYIBwgICAgICAgHBg
cFBgQEBAMCAQECAwQEBAYFBwYHCAgICAgICAcIBgcFBgQEBaICAsH6jAFKjPpu/Z3NwgJZ/dzDAf8
AAQkICAgHBwCGBgUFBAQCAgEBAQECAgQEBQUGBgHbWgICAkIBwgHBwYGBQUEAwMCAQECAwMEBQUG
BgHCAcICQgICAcHBwYGBQUEBAICAQEBAQICBAQFBQYGBwCAGICQgHCAcHBgYFBQQDAwIBAQIDA
wQFBQYGBwCIBwGB+wH+vQFABP5dOgGkAQEAAAADAAAAANkA5oAnQDxAR4AAAEzHwEdAR8HFQ8DIY
8HDwYdAR8WDw0dAQ8BKwIvAT0BLwc9AT8COWEfBj8HLxc/DTU/AwEfDjsBPxEvFiMPFR8BEw8CFR8
HMz8HNS8GIw8ELwQrAQ8BAGoCAGENDAwKCgHQBQEBaIkCAGIEAwQFDA0SBwCAGIBAQICBgCHBxYK
CQkJCAcHBgUFBAQCAQEBAQIDAQFBQYGBwCPEQECAhUCAQINDAsLCQgHBQICKQICAgQDBAULdhIHB
wYCAQEBAQEBAgYHBwCWCgkKCAgHBwYFBQQDAgEBAQECAwMEBAYFBgCHEBABAQED/qwUFRUVFRYWFh
YWFxYXFhcXFxcWFxYXFhYWFhYVFRUVFAQCAQICBAUGCAGJCgsLDA0MDQ0NDBk2EQYQGgYGCEsZDQ0
NDA0MCwsKCQgIBgUEAgIBaQQAQEBAwKRNRIHBqADChI1DQoFAgEBAQMEBAoMEw8eTw4IVxkXCwkJ
BwYCOAIBAIDAwUGBwgJCgICAQENAQEFawIDAQECAGMFAwMEBAUDBAwMFAwIBAQECAwMEBAUGBgYHC
AgICQgHBwCGBgYFBQQEBAyDgICAQECAiICBAUGBwgJCQMBAGEMAQUDAwIDAQICBAQDBAQEBAQEAW
QEAgEBAQICBAMFBQUGBwCICAgJBwgHBgCGBgUFBAQEBOQIAGeBAf6RDAsLCQkICAYGBQUDAwIBAQI
DAwUFBgYICAKJCwsMKSckIiAeGxoYFhQTERAPDQwLCgkIDxsJBQUBQOQnEakKCwNDxARExQWGBob
HiAiJCcCwQMDAwQECA8XPRcKCgUPFz0REAKIBAMDawMCAQICAwCYawEaBwQBAgIAAAEAAAAA/MDN
AA0AAABDQAwAw8EHwQ/ETUnIw8LAYsEJwwGAGIwXmMXFBIIcCsqKaEqRUclJSYnJykpKiosLC4GFg
sCAWMhISiIiIjIkJAPRwB8AQmCQMBARQuNgsMDgcIJCYnmyZOTycmJiYlJSQjIiIgHwULCAMCAQ4
RERITFBUVKy0tFgAAABIA3gABAAAAAABAAAAAABAAAAAABAAAAAABAAAAAABAAAAAABAAAAA
AAADAACADwABAAAAAABAAAAAFgABAAAAAABAAAAAFAAsAHQABAAAAAABAAAAAAGAAcAKAABAAAAAABKACwALwABA
AAAAAALABIAWwADAAEECQAAAAIAbQADAAEECQABAA4AbwADAAEECQACAA4AfQADAAEECQADAA4AiW
ADAAEECQAEAA4AmQADAAEECQAFABYApwADAAEECQAGAA4AvQADAAEECQAKAFgAywADAAEECQALACQ
BIyBEZwZhdWx0UmVndWxhcRlZmF1bHREZwZhdWx0VmVyc2lvbiAxLjBEZwZhdWx0Rm9udCBnZW5l
cmF0ZWQgdXNpbmcgU3luY2Z1c2lvbiBNZXRybyBTdHVkaW93d3cuc3luY2Z1c2lvbi5jb20AIABEA
GUAZgBhAHUAbAB0AFIAZQBnAHUAbABhAHIArABLAGYAYQB1AGwAdABEAGUAZgBhAHUAbAB0AFYAZQ
ByAHMAaQBvAG4AIAAxAAC4AMABEAGUAZgBhAHUAbAB0AEYAbwBuAHQAIABnAGUAbgBlAHIAIAYQB0AGU
```

```

AZAAgAHUAcwBpAG4AZwAgAFMAeQBvAGMAZgB1AHMAaQBvAG4AIABNAGUAdABYAG8AIABTAHQAdQBk
AGkAbwB3AHcAdwAuAHMAeQBvAGMAZgB1AHMAaQBvAG4ALgBjAG8AbQAAAAACAAAAAaAAAAA
AAAAAAAAAAAAAAAAAAAAAAYBAGEDAQQBBQEGAQcADXRyYW5zcG9ydC12YW4LdXNlci1tb2RpZn
kRc2hvcHBpbmctY2FydF8wMS0Lc3B1bmQtbW9uZXkFY2hlY2sAAAA=) format('true');
  font-weight: normal;
  font-style: normal;
}
.stepper-custom-style [class^="sf-icon-"], .stepper-custom-style [class*="
sf-icon-"] {
  font-family: 'Default' !important;
  speak: none;
  font-style: normal;
  font-weight: normal;
  font-variant: normal;
  text-transform: none;
  line-height: inherit;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}
.stepper-custom-style {
  margin-top: 30px;
  padding: 30px;
}
.stepper-custom-style .sf-icon-cart:before { content: "\e710"; }
.stepper-custom-style .sf-icon-transport:before { content: "\e702"; }
.stepper-custom-style .sf-icon-payment:before { content: "\e706"; }
.stepper-custom-style .sf-icon-success:before { content: "\e715"; }
/* Custom styles */
.stepper-custom-style .custom-step .e-step-label-optional {
  font-style: italic;
  font-weight: 900;
  color: #299100;
}

```

## Step validation

You can set the validation state for each step to displaying a success or error icon by using `isValid` property.

To know more about Stepper validation, please refer to the [Validation](#) section.

## Step types in React Stepper component

The Stepper component provides support for displaying steps with the following step types.

### Default type

In default type, the Stepper displays steps with a combination of both indicators and labels by setting the `stepType` property as `Default`. By default, the Stepper displays steps in the `Default` type.

### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
"@syncfusion/ej2-react-navigations";
function App() {
  return (

```

```

        <div className="stepper-steptype-default">
            <StepperComponent stepType="Default">
                <StepsDirective>
                    <StepDirective iconCss={'sf-icon-cart'} label={'Cart'} />
                    <StepDirective iconCss={'sf-icon-transport'}
label={'Delivery Address'} />
                    <StepDirective iconCss={'sf-icon-payment'}
label={'Payment'} />
                    <StepDirective iconCss={'sf-icon-success'}
label={'Confirmation'} />
                </StepsDirective>
            </StepperComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

**INDEX.TSX**

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
    return (
        <div className="stepper-steptype-default">
            <StepperComponent stepType="Default">
                <StepsDirective>
                    <StepDirective iconCss={'sf-icon-cart'} label={'Cart'} />
                    <StepDirective iconCss={'sf-icon-transport'}
label={'Delivery Address'} />
                    <StepDirective iconCss={'sf-icon-payment'}
label={'Payment'} />
                    <StepDirective iconCss={'sf-icon-success'}
label={'Confirmation'} />
                </StepsDirective>
            </StepperComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

**INDEX.CSS**

```

/* Represents the styles for loader */
#loader {
    color: #008cff;
    height: 40px;
    left: 45%;
    position: absolute;
    top: 45%;
}

```

```
width: 30%;
}
@font-face {
  font-family: 'Default';
  src:
    url(data:application/x-font-ttf;charset=utf-
8;base64,AAEAAAAKAIAAAwAgTlMvMjlvSgcAAAEoAAAAVmNtYXCDeIPaAAABmAAAAF5nbHlmEwr+
pwAAAggAAAJQaGVhZCYp2+EAADQAAAAANmhoZWEIUQQHAAAArAAAACRobXR4GAAAAAAAYAAAAAYb
G9jYQhUBlAAAAH4AAAAADm1heHABFgErAAABCAAAACBuYW1lUf5THQAACTgAAAIcG9zdJ8LuoMAAA
0AAAAAbwABAAAEAAAAAFwEAAAAAAD9AABAAAAAAAAAAAAAAAAABgABAAAAAQAArxT6wV8PPPU
ACwQAAAAAAOGLy6UAAAAA4YvLpQAAAAAD9A0aAAAAACAACAAAAAAAAAAEAAAAGAR8ABgAAAAAAgAA
AAoACgAAAP8AAAAAAQAQAAZAABQAAAokCzAAAAI8CiQLMAAAB6wAyAQgAAAIABQMAAAAAAAAAA
AAAAAAAAAAAAAAAAAUGZFZABA5wLnFQQAAAAAXAQAAAAAAAAABAAAAAAAAABAAAAQAAAAEAAAABA
AAAAQAAAAEAAAAAAAAAgAAAAAAAUAAAAQAABQABABKAAAADAAIAAIABOC5wbnCocQ5xX//wA
A5wLnBucI5xDnFf//AAAAAAAAAAAAAAAAABAawADAAMAawADAAAAAEABAACAAMABQAAAAAAAAEQaiwC
3AQkBGgAAAAFAAAAAAP0A18APwB/AIkAxgDrAAABHw8/Dy8OKwEPDQUfDz8PLw4raQ8NAR8FFSM1J
xEfBz800wEfDjM/BzUnIw8GATM/Dx8PMxEhAq8BAQIEBAUFwYICAgJCQoKCgkKCAkIBwCHBQUEAw
MBAQEBAwMEBQUHBwCICQgKCQoKCgkJCAGIBGcFBQQEAgH+CwEBAGQEBQUHBggICakJCgoKCQoICQg
HBwCFBQQDAwEBAQEDAwQFBQcHBwgJCAoJCgoKCQkICAGGBwUFBAQCAQJ8AwUIWAwD3n0BAwMGBgYI
CAMEBQYHBwkJCgsLDA0NDQ4ODQ4MDAwLCgkJCAYGBQMDKAgIBwYFBAECvLsICAYHBQMD/beAAwQFB
QcHCAkKCgsLDA0MDg0NDQwLCwsJCQkHBwUFAwKE/eMBAQoJCQkJCACHBgYFBAMDAQEBAQMDBAUGBg
cHCAkJCQkKCgoJCQgICAcGBgQFAwICAgIDBAUFBgHCAkJCQoLCgkJCQkIBwCGBgUEAwMBAQEBAwM
EBQYGBwCICQkJCQoKCgkJCAGIBwYGBAUDAgICAgMEBQUGBwCICQkJCgGuAQIGehYJBKYp/10ICAcG
BQQCAQ0NDQwLCgoJCAGGBQUDAgIDBQUGCAGJCgoLDA0NDQECBAUGBwQI1foBAGQFBGcH/iwNDawLC
woJCQgHBgUEAwEBAQEDBAUGBwgJCQoLCwMDQJJAAAABgAAAAAD8wOWAAYAqgBaAGwArQDuAAABBz
cfAwUhLwIHlY8PNS8CKwIPHQEHLwEjDwE1LwMjNz0BJzcFBTcfAg8BLwY30wEfaQcVHw8/Dy8PDw4
BFR8PPw8vDw8OAxEWBgEDAgb8/wNuBAUEDQsVFBQTEhEPDw0GCwoIBgQCFhITE+wQDw8PDg4ODg0N
DQwNCwwKCwoKCQgJBwCHBgyEBQMEa5FrBAQDBAMBAwMDBgIDagIEBgYGBxwCAwIBFYQGBAgFBgIWA
gQHCPCBAgQGBBgKCgsMDQ4PDxQEBApDw4NDAsLCQgGBgQCAQECAwYGCakLCwWNDg8PEBAQEa8PDg
0MCwoKCAYGBAL+KgEEBQgKCw0PEBETFBQWFxcXfHYUFBMREQ4NDakIBgMBAQMGCakMDQ4RERMUFBY
WFxcXfHQUEXEQDw0LCgGfBAEXBhcFBAMDrxYWDQEBawUHCAcMDQ4IERESFBQUFQDQAgECAGMEBAUG
BgYIBwgJCQoKCwsLDawMDQ0ODQ4PDgEzawIBAQIGBQMCAQQDBgZqBgOHBQMDMAMHBwMWAQICBQYKC
hYCB1wICBAPDw4NDAsLCQgGBgQCAQECAwYGCakLCwWNDg8PEBAQEa8PDg0MCwoKCAYGAwMBAQMDBg
YICgoLDA0ODw8QATMLDBYVFRQSERAPDQsKCAUEAQEEBQgKCw0PEBESFBUVFhcXfYyVFBISEA8NCwo
IBQQBAQQFCAoLDQ8QEHlUFRYXAAAAAQAAAAA/QDRwA/AH8AhwCRAAABFR8OPw49AS8NKwEPDQUV
Hw4/Dj0BLw0raQ8NEwcTIRMnMSMhMxMhNSEDBzUha0YBAgMEBAQGBQcGBwgICAgICAgIBwYHBQYEB
AQDAGEBAGMEBAQGBQcGBwgICAgICAgIBwYHBQYEBADAGH+aAICAgQEBAYFBwYIBwgICAgICAgHBg
cFBgQEBAMCAQECAwQEBAYFBwYHCAgICAgICAcIBGcFBgQEBaICAsh6jAFKjPpu/Z3NwgJZ/dzDAf8
AAQkICAgHBwCGBgUFBAQCAgEBAQECAgQEBQUGBGcHBwgICakIBwgHBwYGBQUEAwMCAQECAwMEBQUG
BgCHCAcICQgICAcHBwYGBQUEBAICAQEBAQICBAQFBQYGBwCHCAgICQgHCAcHBgYFBQQDAwIBAQIDA
wQFBQYGBwCIBwgB+wH+vQFABP5dOgGkAQEAAAADAAAAANkA5oAnQDxAR4AAAEzHwEdAR8HFQ8DIy
8HDWydAR8WDw0dAQ8BKwIvAT0BLwc9AT8COWEfBj8HLxc/DTU/AwEfDjsBPxEvFiMPFR8BEw8CFR8
HMz8HNS8GIw8ELwQrAQ8BAgoCAGENDAwKCgGHBQEBAikCAGIEAwQFDA0SBwCAGIBAQICBgcHBxYK
CQkJCACHBgUFBAQCAQEBAQIDAQwQFBQYGBwCPEQECAhUCAQINDAsLCQgHBQICKQICAgQDBAULdhIHB
wYCAQEBAQEBAgYHBwCWCgkKCAgHBwYFBQQDAGEBQECAwMEBAYFBgcHEBABAQED/qwUFRUVFRYWFh
YWFxYXfHcXfxcWFxYXfHYWFhYVFRUVFAQCAQICBAUGCAGJCgsLDA0MDQ0NDBk2EQYgqgYGCEsZDQ0
NDA0MCwsKCQgIBgUEAgIBAgQCAQEBAwKRNRIHBqADChI1DQoFAGEBAGMEBAoMEw8eTw4IVxkXCwkJ
BwYCOAIBAIIDAuUGBwgJCgICAQENAQEFawIDAgECAGMFAwMEBAUDBAMFAwIBAQECAwMEBAUGBgYHC
AgICQgHBwCGBgYFBQQEBAyDIgICAQECAICBAUGBwgJCQMBAGEMAQUDAwIDAQICBAQDBAQEBAQEAw
QEAgEBAQICBAMFBQUGBwCICAgJBwGHBgcGBgUFBAQEBAQAgEBAf6RDAsLCQkICAYGBQUDAwIBAQI
DAwUFBgYICakJCwsMKSckIiAeGxoYfHqTERAPDQwLCgkIDxsJBQUBFQQnEakKCwWNDxARExQWGBob
HiAiJCcCoAMDawQECA8XPRcKCgUPFz0REakIBAMDawMCAQICAwCYAwEaBwQBAGIAAAEAAAAAA/MDN
AA0AAABDwQvAw8EHwQ/ETUnIw8LAYsEJwwGAgIwXmMXFBIIcCsKqKaEqRUclJSYnJykpKiosLC4GFg
sCAWMhISiIiIjIkJAPRwB8AQmCQMBARQuNgsMDgcIJCYnmyZOTycmJiYlJSQjIiIgHwULCAMCAQ4
RERITFBUVKy0tFgAAABIA3gABAAAAAAAAAAEAAAABAAAAAAAAABAAcAAQABAAAAAAAAACAACAABAAAA
AAADAACADwABAAAAAAAAAAEAAAFgABAAAAAAAAFAAsAHQABAAAAAAAAAGAAcAKAABAAAAAAAAAKACwALwABA
AAAAALABIAWwADAAEECQAAAAIAbQADAAEECQABAA4AbwADAAEECQACAA4AfQADAAEECQADAA4AiW
```



```
ADAAEEECQAEAA4AmQADAAEEECQAFABYApwADAAEEECQAGAA4AvQADAAEEECQAKAFgAywADAAEEECQALACQ
BIyBEZWZhdWx0UmVndWxhcKRlZmF1bHREZWZhdWx0VmVyc2lubiAxLjBEZWZhdWx0Rm9udCBnZW5l
cmF0ZWQgdXNpbmcgU3luY2Z1c2lubiBNZXRYbyBTdHVkaW93d3cuc3luY2Z1c2lubi5jb20AIABEA
GUAZgBhAHUAbAB0AFIAZQBnAHUAbABhAHIAARABlAGYAYQB1AGwAdABEAGUAZgBhAHUAbAB0AFYAZQ
ByAHMAaQBvAG4AIAAxAAC4AMABEAGUAZgBhAHUAbAB0AEYAAbwBuAHQAIAABnAGUAbgBlAHIAIAYQB0AGU
AZAAgAHUAacwBpAG4AZwAgAFMAeQBuAGMAZgB1AHMAaQBvAG4AIAABNAGUAdABYAG8AIAABTAHQAdQBk
AGkAbwB3AHcAdwAuAHMAeQBuAGMAZgB1AHMAaQBvAG4ALgBjAG8AbQAAAAACAAAAAAAAAAoAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAYBAGEDAQQBBQEGAQcADXRyYW5zcG9ydC12YW4LdXNlci1tb2RpZn
kRc2hvcHBpbmctY2FydF8wMS0Lc3BlbmQtbw9uZXXkFY2hlY2sAAAA=) format('trueype');
font-weight: normal;
font-style: normal;
}

.stepper-steptype-default [class^="sf-icon-"], .stepper-steptype-default
[class*=" sf-icon-"] {
font-family: 'Default' !important;
speak: none;
font-style: normal;
font-weight: normal;
font-variant: normal;
text-transform: none;
line-height: inherit;
-webkit-font-smoothing: antialiased;
-moz-osx-font-smoothing: grayscale;
}

.stepper-steptype-default {
margin-top: 30px;
padding: 30px;
}

.stepper-steptype-default .sf-icon-cart:before { content: "\e710"; }
.stepper-steptype-default .sf-icon-transport:before { content: "\e702"; }
.stepper-steptype-default .sf-icon-payment:before { content: "\e706"; }
.stepper-steptype-default .sf-icon-success:before { content: "\e715"; }
```

Label type

In label type, the Stepper displays the steps with only the step labels by setting the `stepType` property as `Label`.

When both label and text are defined, the label takes priority in displaying the steps.

## INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
  return (
    <div className="stepper-steptype-label">
      <StepperComponent stepType="Label">
        <StepsDirective>
          <StepDirective iconCss={'sf-icon-cart'} label={'Cart'} />
          <StepDirective iconCss={'sf-icon-transport'}
label={'Delivery Address'} />
          <StepDirective iconCss={'sf-icon-payment'}
label={'Payment'} />
        </StepsDirective>
      </StepperComponent>
    </div>
  );
}
```

```

        <StepDirective iconCss={'sf-icon-success'}
label={'Confirmation'} />
      </StepsDirective>
    </StepperComponent>
  </div>
);
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

## INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
 '@syncfusion/ej2-react-navigations';
function App() {
  return (
    <div className="stepper-steptype-label">
      <StepperComponent stepType="Label">
        <StepsDirective>
          <StepDirective iconCss={'sf-icon-cart'} label={'Cart'} />
          <StepDirective iconCss={'sf-icon-transport'}
label={'Delivery Address'} />
          <StepDirective iconCss={'sf-icon-payment'}
label={'Payment'} />
          <StepDirective iconCss={'sf-icon-success'}
label={'Confirmation'} />
        </StepsDirective>
      </StepperComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

## INDEX.CSS

```

/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
@font-face {
  font-family: 'Default';
  src:
    url(data:application/x-font-ttf;charset=utf-
8;base64,AAEAAAIAIAAwAgTlMvMjlvSgcAAAEoAAAAVmNtYXCDeIPaAAABmAAAAF5nbHlmEwr+
pwAAAggAAAJQaGVhZCYp2+EAADQAAAAANmhoZWEIUQQHAAAArAAAACRobXR4GAAAAAAAYAAAAAYb

```

G9jYQHUB1AAAAH4AAAAADm1heHABFgErAAABCAAAACBuYW1luF5THQAACTgAAAIlcG9zdJ8LuoMAAA  
0AAAAAbwABAAAAEAAAAAFwEAAAAAAD9AABAAAAAABAAAAAABgABAAAAAQAArxT6wV8PPPU  
ACwQAAAAAOGly6UAAAAA4YvLpQAAAAAD9AOaAAAAACAACAAAAAABAAAAAAGAR8ABgAAAAAAAgAA  
AAoACgAAAP8AAAAAQAQAAZAABQAAAokCzAAAAI8CiQLMAAAB6wAyAQgAAAIABQMAAAAAA  
AAAAAUGZFZABA5wLnFQAAAAAXAQAAAAAABAAAAAABAAAAAQAQAAAAEAAAAABA  
AAAAQAQAAAAEAAAAAAGAAAAAUAAMAAQAABQABABKAAAAADAAIAIABOCC5wbnCOCQ5xX//wA  
A5wLnBucI5xDnFf//AAAAAABAAwADAAMAawADAAAAAEABAACAAMABQAAAAAAAEQAiwC  
3AQkBGgAAAAFAAAAAAP0A18APwB/AIkAxgDrAAABHw8/Dy8OKwEPDQUfDz8PLw4rAQ8NAR8FFSM1J  
xEfBz80OwEfDjM/BzUnIw8GATM/Dx8PMxEhAq8BAQIEBAUFbWYICAgJCQoKCgkKCAkIBwCHBQUEAw  
MBAQEBAwMEBQUHBwCICQgKCQoKCgkJCAGIBGcFBQOEAgH+CwEBAGQEBQUHBggICakJCgoKCQoICQg  
HBwCFBQQDAwEBAQEDAwQFBQcHBwgJCAoJCgoKCQkICAGGBwUFBAQCAQJ8AwUIAwD3n0BAwMGBgYI  
CAMEBAKYHBwkJCGSLDA0NDQ4ODQ4MDAwLCgkJCAYGBQMDKAgIBwYFBAECvLsICAYHBQMD/beAAwQFB  
QCCHCAkKCgSLDA0MDQ4ODQ4MDAwLCgkJCQkHBwUFawKE/eMBAQoJCQkJCACHBgYFBAMDAQEBAQMDBAUGB  
cHCAkJCQkKCgoJCQgICACGBgQFAwICAgIDBAUFBgCHCAkJCQoLCgkJCQkIBwCHBUEAwMBAQEBAwM  
EBQYGBwCICQkJCQoKCgkJCAGIBwYGBAUDAgICAgMEBQUGBwCICQkJCgGuAQIGehYJBKYP/10ICACG  
BQCAQ0NDQwLCgoJCAGGBQUdAgIDBQUGCAGJCgoLDA0NDQECBAUGBwQI1foBAGQFBGcH/iwNDAwLC  
woJCQgHBgUEAwEBAQEDBAUGBwGJCQoLCwWMDQJJAABAgAAAAAD8wOWAAYAqgBaAGwArQDuAAABZ  
cfAwUhLwIH1y8PNS8CKwIPHQHLWEdJdWElLwMjNz0BJzcfBTcfAg8BLWY3OwEfAQcVHw8/Dy8PDw4  
BFR8PPw8vDw8OAxEWBgEDAg8/wNuBAUEDQsVFBQTEhEPDw0GCwoIBgQCFhITE+wQDw8PDg4ODg0N  
DQwNCwWKCwoKCQgJBwCHBgYEBQMEA5FrBAQDBAMBAwMDBgIDagIEBgYGBxwCAwIBFQYGBAGFBgIWA  
gQHCPCBAgQGBgKCGsMDQ4PDxAQEBAPDw4NDAsLCQgGBgQCAQECBAYGCAkLCwWNg8PEBAQE8PDg  
0MCwoKCAyGBAL+KgEEBQgKCw0PEBETFBQWFxcXFhYUFBMREQ4NDakIBgMBAQMGCakMDQ4RERMUFBY  
WFxcXFhQUExEQDw0LCggFBAEXBhcFBAMDrxYWDQEBawUHCAsMDQ4IERESFBQUFQDAGECAGMEBAUG  
BgYIBwGJCQoKCwsLDAwMDQ0ODQ4PDgEZawIBAQIGBQMCAQQDBgZqBgOHBQMDMAMHBwMWAQICBQYKC  
hYCB1wICBAPDw4NDAsLCQgGBgQCAQECBAYGCAkLCwWNg8PEBAQE8PDg0MCwoKCAyGawMBAQMDBg  
YICgoLDA0ODw8QATMLDBYVFRQSERAPDQsKCAUEAQEEBQgKCw0PEBESFBUVFhcXFxYVFBISEA8NCwo  
IBQQAQOQFCAoLDQ8QEH1UFRYXAAAAAQAAAAA/QDRwA/AH8AhwCRAAABFR8OPw49AS8NKwEPDQUV  
Hw4/Dj0BLw0rAQ8NEwcTIRMnMSMhMxMhNSEDBzUha0YBAGMEBAQGBQcGBwGICAgICAgIBwYHBQYEB  
AQDAGEBAGMEBAQGBQcGBwGICAgICAgIBwYHBQYEBAQDAGH+aAICAgQEBAYFBwYIBwGICAgICAgHBg  
cFBgQEBAMCAQECAwQEBAYFBwYHCAGICAgICAgIBGcFBgQEBaICAsh6jAFKjPpu/Z3NwgJZ/dzDAf8  
AAQkICAgHBwCGBgUFBAQCAgEBAQECAGQEBQUGBgCHBgICakIBwGHBwYGBQUEAwMCAQECAwMEBAQUG  
BgCHCAkICQgICACHBwYGBQUEBAICAQEBAQICBAQFBQYGBwCHCAgICQgHCACHBgYFBQDQDAwIBAQIDA  
wQFBQYGBwCIBwGB+wH+vQFABP5dOgGkAQEAAAADAAAAANkA5oAnQDxAR4AAAEzHwEdAR8HFQ8DIy  
8HDWydAR8WDw0dAQ8BKwIvAT0BLwc9AT8COWEfBj8HLxc/DTU/AwEfDjsBPxEvFiMPFR8BEw8CFR8  
HMz8HNS8GIw8ELwQrAQ8BAGoCAGENDAwKCgGHBQEBaIkAgIEAwQFDA0SBwCAGIBAQICBgCHBxYK  
CQkJCACHBgUFBAQCAQEBAQIDAQwQFBQYGBwCPEQECAhUCAQINDAsLCQgHBQICKQICAgQDBAULdhIHB  
wYCAQEBAQEBAgYHBwCWCgkKCAgHBwYFBQDAGEBQECawMEBAYFBGcHEBABAQED/qwUFRUVFRYWFh  
YWFxYXFhcXFxcWFxYXFhYWFhYVFRUVFAQCAQICBAUGCAGJCgsLDA0MDQ0NDBk2EQYgGqYGCEsZDQ0  
NDA0MCwsKCQgIBgUEAgIBaQCAQEBAwKRNRIHBqADChI1DQoFAGEBAGMEBAoMEw8eTw4IVxkXCwkJ  
BwYCOAIBAIDAwUGBwGJCgICAQENAEFAwIDAAGCAgMFAwMEBAUDBAMFAwIBAQECAwMEBAUGBgYHC  
AgICQgHBwCGBgYFBQQEBAyDIAgICAQECAiICBAUGBwGJCQMBAGEMAQUDAwIDAQICBAQDBAQEBAQEAW  
QEAgEBAQICBAMFBQUGBwCICAgJBwGHBGcGBgUFBAQEBQqIAGeBAf6RDAsLCQkICAYGBQUdAwIBAQI  
DAwUFBgYICakJCwsMKsckIiAeGxoYFhQTERAPDQwLCgkIDxsJBQUBQqNEakKCwWNDxARExQWGBob  
HiAiJCcCoAMDawQECA8XPRcKCgUPFz0REakIBAMDawMCAQICAwYAwEaBwQBAGIAAAEAAAAA/MDN  
AA0AAABDwQvAw8EHwQ/ETUnIw8LAYsEJwWGAglwXmMXFBIIcCsQKaEqRUclJSYnJykpKiosLC4GFg  
sCAWMhISiIiIjIkJAPRwB8AQmCQMBARQuNGsMDgcIJCYnmyZOTycmJiYlJSQjIiIgHwULCAMCAQ4  
RERITFBUVKy0tFgAAABIA3gABAAAAAABAAAAAABAAcAAQABAAAAAACAACAABAAAA  
AAADAACAdwABAAAAAAEAACAFgABAAAAAFAAsAHQABAAAAAAGAACAKAABAAAAAAKACwALwABA  
AAAAALABIAWwADAAEECQAAAAIAbQADAAEECQABAA4AbwADAAEECQACAA4AfQADAAEECQADAA4Ai  
wADAAEECQAEAA4AmQADAAEECQAFABYApwADAAEECQAGAA4AvQADAAEECQAKAFgAYwADAAEECQALACQ  
BIyBEZWZhdWx0UmVndWxhckRlZmFlbHREZWZhdWx0VmVyc2lvbiAxLjBEZWZhdWx0Rm9udCBnZW5l  
cmF0ZWQgdXNpbmcgU3luY2Z1c2lvbiBNZXRybyBTdHVkaW93d3cuc3luY2Z1c2lvbi5jb20AIBEA  
GUAZgBhAHUAbAB0AFIAZQBnAHUAbABhAHIArABLAGYAYQB1AGwAdABEAGUAZgBhAHUAbAB0AFYAZQ  
ByAHMAaQBvAG4AIAAxAAC4AMABEAGUAZgBhAHUAbAB0AEYAbwBuAHQAIAIBnAGUAbgBlAHIAIAYQB0AGU  
AZAAgAHUAcwBpAG4AZwAgAFMAeQBwAGMAZgB1AHMAaQBvAG4AIAIBNAGUAdABYAG8AIAIBTAHQAdQBk  
AGkAbwB3AHcAdwAuAHMAeQBwAGMAZgB1AHMAaQBvAG4ALgBjAG8AbQAAAAACAAAAAoooooAAAAA

```

AAAAAAAAAAAAAAAAAAAAAAAAAAYBAGEDAQQBBQEGAQcADXRyYW5zcG9ydC12YW4LdXNlci1tb2RpZn
kRc2hvcHBpbmctY2FydF8wMS0Lc3BlbmQtbW9uZXkFY2hlY2sAAAA=) format('trueType');
  font-weight: normal;
  font-style: normal;
}
.stepper-steptype-label [class^="sf-icon-"], .stepper-steptype-label
[class*=" sf-icon-"] {
  font-family: 'Default' !important;
  speak: none;
  font-style: normal;
  font-weight: normal;
  font-variant: normal;
  text-transform: none;
  line-height: inherit;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}
.stepper-steptype-label {
  margin-top: 30px;
  padding: 30px;
}
.stepper-steptype-label .sf-icon-cart:before { content: "\e710"; }
.stepper-steptype-label .sf-icon-transport:before { content: "\e702"; }
.stepper-steptype-label .sf-icon-payment:before { content: "\e706"; }
.stepper-steptype-label .sf-icon-success:before { content: "\e715"; }

```

### Label positions

You can display the label on the top, bottom, left, or right side of the steps using the [labelPosition](#) property.

The following label positions are supported in Stepper:

- | Value       | Description   |
|-------------|---|
| ----- ----- |   |
| Top         | Positions the label at the top of each step.        |
| Bottom      | Positions the label at the bottom of each step.     |
| Start       | Positions the label to the left side of each step.  |
| End         | Positions the label to the right side of each step. |

### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
  let stepperObj = React.useRef(null);
  const updateLabelPosition = (args) => {
    stepperObj.current.labelPosition = args.currentTarget.value;
  };
  return (
    <div id="stepper-label-position">

```

```

        <div className="e-btn-group">
            <input type="radio" id="start" name="position" value="start"
onClick={updateLabelPosition} />
            <label className="e-btn" htmlFor="start">Start</label>
            <input type="radio" id="end" name="position" value="end"
onClick={updateLabelPosition} />
            <label className="e-btn" htmlFor="end">End</label>
            <input type="radio" id="top" name="position" value="top"
onClick={updateLabelPosition} />
            <label className="e-btn" htmlFor="top">Top</label>
            <input type="radio" id="bottom" name="position"
value="bottom" onClick={updateLabelPosition} defaultChecked />
            <label className="e-btn" htmlFor="bottom">Bottom</label>
        </div>
        <StepperComponent id="label-stepper-control" ref={stepperObj}>
            <StepsDirective>
                <StepDirective iconCss={'sf-icon-cart'} label={'Cart'} />
                <StepDirective iconCss={'sf-icon-transport'}
label={'Delivery Address'} />
                <StepDirective iconCss={'sf-icon-payment'}
label={'Payment'} />
                <StepDirective iconCss={'sf-icon-success'}
label={'Confirmation'} />
            </StepsDirective>
        </StepperComponent>
    </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

## INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
    let stepperObj = React.useRef<StepperComponent>(null);
    const updateLabelPosition = (args) => {
        stepperObj.current.labelPosition = args.currentTarget.value;
    };
    return (
        <div id="stepper-label-position">
            <div className="e-btn-group">
                <input type="radio" id="start" name="position" value="start"
onClick={updateLabelPosition} />
                <label className="e-btn" htmlFor="start">Start</label>
                <input type="radio" id="end" name="position" value="end"
onClick={updateLabelPosition} />
                <label className="e-btn" htmlFor="end">End</label>
                <input type="radio" id="top" name="position" value="top"
onClick={updateLabelPosition} />
                <label className="e-btn" htmlFor="top">Top</label>
            </div>
        </div>
    );
}

```

```

        <input type="radio" id="bottom" name="position"
value="bottom" onClick={updateLabelPosition} defaultChecked />
        <label className="e-btn" htmlFor="bottom">Bottom</label>
      </div>
      <StepperComponent id="label-stepper-control" ref={stepperObj}>
        <StepsDirective>
          <StepDirective iconCss={'sf-icon-cart'} label={'Cart'} />
          <StepDirective iconCss={'sf-icon-transport'}
label={'Delivery Address'} />
          <StepDirective iconCss={'sf-icon-payment'}
label={'Payment'} />
          <StepDirective iconCss={'sf-icon-success'}
label={'Confirmation'} />
        </StepsDirective>
      </StepperComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
[% endraw %]

```

## INDEX.CSS

```

/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
@font-face {
  font-family: 'Default';
  src:
    url(data:application/x-font-ttf;charset=utf-
8;base64,AAEAAAIAIAAAwAgTlMvMjlvSgcAAAEoAAAAVmNtYXCDelPaAAABmAAAAF5nbHlmEwr+
pwAAAggAAAJQaGVhZCYp2+EAADQAAAAANmhoZWEIuQQHAAAArAAAAACRobXR4GAAAAAAAYAAAAyb
G9jYQhUBlAAAAH4AAAAADm1heHABFgErAAABCAAAACBuYw1luF5THQAACtgAAAIlcG9zdJ8LuoMAAA
0AAAAAbwABAAAEAAAAAFwEAAAAAAD9AABAAAAAAAAAAAAAAAAAAAAABgABAAAAQAARxT6wV8PPPU
ACwQAAAAAOGLy6UAAAAA4YvLpQAAAAAD9AOaAAAAACAACAAAAAAAAAAAAEAAAAAGAR8ABgAAAAAAgAA
AAoACgAAAP8AAAAAAAAAAQAAZAABQAAAokCzAAAAI8CiQLMAAAB6wAyAQgAAAIABQMAAAAAAAAAAA
AAAAAAAAAAAAAAAAAUGZFZABA5wLnFQQAAAAAXAQAAAAAAAABAAAAAAAAABAAAAAQAAAAEAAAAABA
AAAAQAAAAEAAAAAAAAGAAAAAMAAAAUAMAAQAAABQABABKAAAADAAIAAIABOCc5wbnCOcQ5xX//wA
A5wLnBucI5xDnFf//AAAAAAAAAAAAAAAAABAAwADAAMAawADAAAAAEABAACAAMABQAAAAAAAAAEQAIwC
3AQkBGgAAAAFAAAAAAP0A18APwB/AIkAxgDrAAABHw8/Dy8OKwEPDQUfDz8PLw4rAQ8NAR8FFSM1J
xEfBz80OwEfDjM/BzUnIw8GATM/Dx8PMxEhAq8BAQIEBAUFBwYICAgJCQoKCgkKCAkIBwcHBQUEAw
MBAQEBAwMEBQUHBwcICQgKCQoKCgkJCAgIBgcFBQQEAgH+CwEBAgQEBQUHBggICakJCgoKCQoICQg
HBwcFBQQDAwEBAQEADAwQFBQcHBwgJCAoJCgoKCQkICAgGBwUFBAQCAQJ8AwUIWAwD3n0BAwMGBgYI
CAMEBQYHBwkJCgsLDA0NDQ4ODQ4MDAwLCgkJCAYGBQMDKAgIBwYFBAECvLsICAYHBQMD/beAAwQFB
QcHCAkKCgsLDA0MDg0NDQwLCwsJCQkHBwUFAwKE/eMBAQoJCQkJCAcHBgyYFBAMDAQEBAQMDBAUGBg
cHCAkJCQkKCgoJCQgICAcGBgQFAwICAgIDBAUFBgHCAkJCQoLCgkJCQkIBwcGBgUEAwMBAQEBAwM
EBQYGBwcICQkJCQoKCgkJCagIBwYGBAUDAgICAgMEBQUGBwcICQkJCgGuAQIGehYJBKYp/10ICAcG
BQQCAQ0NDQwLCgoJCAGGBQUdAgIDBQUGCagJCgoLDA0NDQCEBAUGBwQI1foBAGQFBgcH/iwNDawLC
woJCQgHBgUEAwEBAQEDBAUGBwgJCQoLCwwMDQJJAABABgAAAAAD8wOWAAYAQgBaAGwArQDuAAABZ

```

```

cfAwUhLwIHlY8PNS8CKwIPHQEHLwEjDwE1LwMjNz0BJzcFbTcfAg8BLwY3OwEfAQcVHw8/Dy8PDw4
BFR8PPw8vDw8OAxEWBgEDAgb8/wNuBAUEDQsVFBQTEhEPDw0GCwoIBgQCFhITE+wQDw8PDg4ODg0N
DQwNCwwKCwoKCQgJBwcHBgYEBQMEA5FrBAQDBAMBawMDBgIDagIEBgYGBxwCAwIBFQYGBAgFBgIWA
gQHCPCBAgQGBgKCgMDQ4PDxAQEBAPDw4NDAsLCQgGBgQCAQECCBAYGCAkLCwwNDg8PEBAQEAE8PDg
0MCwoKCAYGBAL+KgEEBQgKCw0PEBETFBQWFxcXfHYUFBMREQ4NDakIBgMBAQMGCakMDQ4RERMUFBY
WFxcXfHQUExEQDw0LCggFBAEXBhcFBAMDrxYWDQEBAwUHCAsmDQ4IERESFBQUFQQDagECAGMEBAUG
BgYIBwgJCQoKCwsLDAwMDQ0ODQ4PDgEZawIBAQIGBQMCAQQDBgZqBgOHBQMDMAMHBwMWAQICBQYKC
hYCB1wICBAPDw4NDAsLCQgGBgQCAQECCBAYGCAkLCwwNDg8PEBAQEAE8PDg0MCwoKCAYGAwMBAQMDBg
YICgoLDA0ODw8QATMLDBYVFRQSERAPDQsKCAUEAQEEBQgKCw0PEBESFBUVFhcXfYVFBISEA8NCwo
IBQQAQQFCAoLDQ8QEHlUFYRYAAAAAQAQAAAAA/QDRwA/AH8AhwCRAAABFR8OPw49AS8NKwEPDQUV
Hw4/Dj0BLw0rAQ8NEwcTIRMnMSMhMxMhNSEDBzUha0YBAgMEBAQGBQcGBwgICAgICAgIBwYHBQYEB
AQDagEBAGMEBAQGBQcGBwgICAgICAgIBwYHBQYEBAQDagH+aAICAgQEBAYFBwYIBwgICAgICAgHBg
cFBgQEBAMCAQECAwQEBAYFBwYHCAGICAgICAcIBgcFBgQEBACAsH6jAFKjPpu/Z3NwgJZ/dzDAf8
AAQkICAgHBwcGBgUFBAQCAgEBAQECAgQEBQUGBgchBwgICAkIBwgHBwYGBQUEAwMCAQECAwMEBQUG
BgCHCAcICQgICAcHBwYGBQUEBAICAQEBAQICBAQFBQYGBwCHCAgICQgHCAcHBgYFBQDDAwIBAQIDA
wQFBQYGBwCIBwgB+wH+vQFABP5dOgGkAQEAAAADAAAAAANKa5oAnQDxAR4AAAEzHwEdAR8HFQ8DIY
8HDwYdAR8WDw0dAQ8BKwIvAT0BLwc9AT8COWefBj8HLxc/DTU/AwEfDjsBPxEvFiMPFR8BEw8CFR8
HMz8HNS8GIw8ELwQrAQ8BAGoCAGENDAwKCgGHBQEBAikCAGIEAwQFDA0SBwcGAGIBAQICBgcHBxYK
CQkJCAcHBgUFBAQCAQEBAQIDAQWFBQYGBwCPEQECAhUCAQINDAsLCQgHBQICKQICAgQDBAULdhIHB
wYCAQEBAQEBAgYHBwCWCgkKCAgHBwYFBQDDagEBAQECAwMEBAYFBgcHEBABAQED/qwUFRUVFRYWFh
YWFxYXfHcXfxcWFxYXfHYWFHYVFRUVFAQCAQICBAUGCAGJCgsLDA0MDQ0NDBk2EQYgqgYGCEsZDQ0
NDA0MCwsKCQgIBgUEAgIBAgQCAQEBAwKRNRIHBqADChI1DQoFAgEBAgMEBAoMEw8eTw4IVxkXCwkJ
BwYCOAIBAiIDAwUGBwgJCgICAQENAQEFawIDAgECAGMFAwMEBAUDBAMFAwIBAQECAwMEBAUGBgYHC
AgICQgHBwCGBgYFBQQEBAyDIGICAQECAiICBAUGBwgJCQMBAGEMAQUdAwIDAQICBAQDBAQEBAQEAW
QEAgEBAQICBAMFBQUGBwCICAgJBwgHBgcGBgUFBAQEBOQiAgEBAf6RDAsLCQkICAYGBQUdAwIBAQI
DAwUFBgYICakJCwsMKsckIiAeGxoYfHqTERAPDQwLCgkIDxsJBQUBQqNEakKCwwNDxARExQWGBob
HiAiJCcCoAMDawQECA8XPRcKCgUPFz0REakIBAMDawMCAQICAwCYAwEaBwQBAGIAAAEAAAAAA/MDN
AA0AAABDwQvAw8EHwQ/ETUnIw8LAYsEJwGAGIwXmMXFBIICCsqKaEqRUclJSYnJykpKiosLC4GFg
sCAWMhISiIiIjIkJAPRwB8AQmCQMBARQuNgsMDgcIJCYNmyZOTycmJiYlJSQjIiIgHwULCAMCAQ4
RERITFBUVKy0tFgAAABIA3gABAAAAAEEEEAAAAAABAAAAAABAAcAAQABAAAAAACAACAABAAAA
AAADAACADwABAAAAAEEAAcAFgABAAAAAFAAsAHQABAAAAAAGAAcAKAABAAAAAACAALwABA
AAAAALABIAWwADAAEEECQAAAAIAbQADAAEEECQABAA4AbwADAAEEECQACAA4AfQADAAEEECQADAA4AiW
ADAAEEECQAEAA4AmQADAAEEECQAFABYApwADAAEEECQAGAA4AvQADAAEEECQAKAFgAywADAAEEECQALACQ
BIyBEZWZhdWx0UmVndWxhckRlZmFlbHREZWZhdWx0VmVyc2lvbiAxLjBEZWZhdWx0Rm9udCBnZW5l
cmF0ZWQgdXNpbmcgU3luY2Z1c2lvbiBNZXRybyBTdHVkaW93d3cuc3luY2Z1c2lvbi5jb20AIBEA
GUAZgBhAHUAbAB0AFIAZQBnAHUAbABhAHIArABLAGYAYQB1AGwAdABEAGUAZgBhAHUAbAB0AFYAZQ
ByAHMAaQBvAG4AIAAxAAC4AMABEAGUAZgBhAHUAbAB0AEYAbwBuAHQAIAABnAGUAbgBIAHIAAYQB0AGU
AZAAgAHUAcwBpAG4AZwAgAFMAeQBvAGMAZgB1AHMAaQBvAG4AIAABNAGUAdABYAG8AIAABTAHQAdQBk
AGkAbwB3AHcAdwAuAHMAeQBvAGMAZgB1AHMAaQBvAG4ALgBjAG8AbQAAAAACAAAAAaAaAaAaAaAaA
AAAAAAYBAgEDAQBBQEGAQcADXRyYW5zcG9ydC12YW4LdXNlci1tb2RpZn
kRc2hvcHBpbmctY2FydF8wMS0Lc3BlbmQtbW9uZXXkFY2hly2sAAAA=) format('trueType');
font-weight: normal;
font-style: normal;
}
#stepper-label-position [class^="sf-icon-"], #stepper-label-position
[class*=" sf-icon-"] {
font-family: 'Default' !important;
speak: none;
font-style: normal;
font-weight: normal;
font-variant: normal;
text-transform: none;
line-height: inherit;
-webkit-font-smoothing: antialiased;
-moz-osx-font-smoothing: grayscale;
}
#stepper-label-position {

```

```

text-align: center;
margin-top: 30px;
padding: 30px;
}
#label-stepper-control {
margin: 50px 100px;
}
#stepper-label-position .sf-icon-cart:before { content: "\e710"; }
#stepper-label-position .sf-icon-user:before { content: "\e708"; }
#stepper-label-position .sf-icon-transport:before { content: "\e702"; }
#stepper-label-position .sf-icon-payment:before { content: "\e706"; }
#stepper-label-position .sf-icon-success:before { content: "\e715"; }

```

### Indicator type

In indicator type, the Stepper displays steps with only the step indicators by setting the [stepType](#) property as **Indicator**.

#### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
 '@syncfusion/ej2-react-navigations';
function App() {
  return (
    <div className="stepper-steptype-container">
      <div className="default-stepper">
        <StepperComponent stepType="Indicator">
          <StepsDirective>
            <StepDirective />
            <StepDirective />
            <StepDirective />
            <StepDirective />
          </StepsDirective>
        </StepperComponent>
      </div>
      <div className="text-stepper">
        <StepperComponent stepType="Indicator">
          <StepsDirective>
            <StepDirective text={'1'} />
            <StepDirective text={'2'} />
            <StepDirective text={'3'} />
            <StepDirective text={'4'} />
          </StepsDirective>
        </StepperComponent>
      </div>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

#### INDEX.TSX



```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
  return (
    <div className="stepper-steptype-container">
      <div className="default-stepper">
        <StepperComponent stepType="Indicator">
          <StepsDirective>
            <StepDirective />
            <StepDirective />
            <StepDirective />
            <StepDirective />
          </StepsDirective>
        </StepperComponent>
      </div>
      <div className="text-stepper">
        <StepperComponent stepType="Indicator">
          <StepsDirective>
            <StepDirective text={ '1' } />
            <StepDirective text={ '2' } />
            <StepDirective text={ '3' } />
            <StepDirective text={ '4' } />
          </StepsDirective>
        </StepperComponent>
      </div>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

## INDEX.CSS

```
/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
@font-face {
  font-family: 'Default';
  src:
    url(data:application/x-font-ttf;charset=utf-
8;base64,AAEAAAKAIAAAwAgTlMvMjlvSgcAAAEoAAAAVmNtYXCDelPaAAABmAAAAF5nbHlmEwr+
pwAAAggAAAJQaGVhZCYp2+EAADQAAAAANmhoZWEIUQQHAAArAAAACRobXR4GAAAAAAAAAYAAAAAYb
G9jYQhUBlAAAAH4AAAAADm1heHABFgErAAABCAAAACBuYW1luF5THQAACtgAAAIlcG9zdJ8LuoMAAA
0AAAAAbwABAAAEAAAAAFwEAAAAAAD9AABAAAAAAAAAAAAAAAAAABgABAAAAAQAArxT6wV8PPPU
ACwQAAAAAOGly6UAAAAA4YvLpQAAAAAD9AOaAAAAACAACAAAAAAAAAAAAEAAAAGAR8ABgAAAAAAgAA
AAoACgAAAP8AAAAAAAAAQAAZAABQAAAokCzAAAAI8CiQLMAAAB6wAyAQgAAAIABQMAAAAAAAAAAA
```

```

AAAAAAAAAAAAAAAAAUGZFZABA5wLnFQQAAAAAXAQAAAAAAAAABAAAAAAAAABAAAAAQAAAAEAAAABA
AAAAQAAAAEAAAAAAAAAgAAAAAMAAUAAMAAQAAABQABABKAAAADAAIAAIBOCc5wbnCocQ5xX//wA
A5wLnBucI5xDnFf//AAAAAAAAAAAAAAAAABAAwADAAMAawADAAAAEABAACAAMABQAAAAAAAAEQaiwC
3AQkBGgAAAAFAAAAAAP0A18APwB/AIkAxgDrAAABHw8/Dy8OKwEPDQUfDz8PLw4rAQ8NAR8FFSM1J
xEfBz800wEfDjM/BzUnIw8GATM/Dx8PMxEhAq8BAQIEBAUFbWYICAgJCQoKCgkKCAkIBwcHBQUEAw
MBAQEBAwMEBQUHBwCICQgKCQoKCgkJCAgIBgcFBQQEAgH+CwEBAgQEBQUHBggICakJCgoKCQoICQg
HBwcFBQQDAwEBAQEDAwQFBQcHBwgJCAoJCgoKCQkICAgGBwUFBAQCAQJ8AwUIAwD3n0BAwMGBgYI
CAMEBQYHBwkJCgsLDA0NDQ4ODQ4MDAwLCgkJCAYGBQMDKAgIBwYFBAECvLsICAYHBQMD/beAAwQFB
QcHCAkKCgsLDA0MDg0NDQwLCwsJCQkHBwUFAwKE/eMBAQoJCQkJCACHBgYFBAMDAQEBAQMDBAUGBg
cHCAkJCQkKCgoJCQgICAcGBgQFAwICAgIDBAUFbGcHCAkJCQoLCgkJCQkIBwcGBgUEAwMBAQEBAwM
EBQYGBwCICQkJCQoKCgkJCAgIBwYGBAUDAgICAgMEBQUGBwCICQkJCgGuAQIGehYJBKYp/10ICAcG
BQQCAQ0NDQwLCgoJCAgGBQUDAgIDBQUCAgJCgoLDA0NDQECBAUGBwQI1foBAGQFBgCH/iwNDAwLC
woJCQgHBgUEAwEBAQEDBAUGBwgJCQoLCwwMDQJJAABABgAAAAAD8wOWAAYAqgBaAGwArQDuAAABBz
cfAwUhLwIHlY8PNS8CKwIPHQEHLwEjDwE1LwMjNz0BJzcfBTcfAg8BLwY30wEfAQcVHw8/Dy8PDw4
BFR8PPw8vDw8OAxEWBgEDAg8/wNuBAUEDQsVFBQTEhEPDw0GCwoIBgQCFhITE+wQDw8PDg4ODg0N
DQwNCwwKCwoKCQgJBwcHBgYEBQMEA5FrBAQDBAMBAwMDBgIDagIEBgYGBxwCAwIBFQYGBAgFBgIWA
gQHCPcBAGQGBgKCGsMDQ4PDxAQEBAPDw4NDAsLCQgGBgQCAQECBAYGCAkLCwwNDg8PEBAQEA8PDg
0MCwoKCAyGBAL+KgEEBQgKCw0PEBETFBQWFxcXFhYUFBMREQ4NDakIBgMBAQMGCakMDQ4RERMUFBY
WFxcXFhQUEExEQDw0LCggFBAEXBhcFBAMDrxYWDQEBawUHCAsmDQ4IERESFBQUFQQDAgECAGMEBAUG
BgYIBwgJCQoKCwsLDAwMDQ0ODQ4PDgEZawIBAQIGBQMCAQQDBgZqBgoHBQMDMAMHBwMWAQICBQYKC
hYCB1wICBAPDw4NDAsLCQgGBgQCAQECBAYGCAkLCwwNDg8PEBAQEA8PDg0MCwoKCAyGAwMBAQMDBg
YICgoLDA0ODw8QATMLDBYVFRQSERAPDQsKCAUEAQEEBQgKCw0PEBESFBUVFhcXFxYVFBISEA8NCwo
IBQQBAQQFCAoLDQ8QEhIUFRYXAAAAAQAAAAA/QDRwA/AH8AhwCRAAABFR8OPw49AS8NKwEPDQUV
Hw4/Dj0BLw0rAQ8NEwcTIRMnMSMhMxMhNSEDBzUha0YBAGMEBAQGBQcGBwgICAgICAgIBwYHBQYEB
AQDAgEBAgMEBAQGBQcGBwgICAgICAgIBwYHBQYEBAQDAgH+aaICAgQEBAyFBwYIBwgICAgICAgHBg
cFBgQEBAQCAQECawQEBAyFBwYHCAgICAgICAcIBgcFBgQEBAICAsH6jAFKjPpu/Z3NwgJZ/dzDAf8
AAQkICAgHBwcGBgUFBAQCAgEBAQECAGQEBOUGBgHBwgICakIBwgHBwYGBQUEAwMCAQECawMEBQUG
BgCHCAcICQgICAcHBwYGBQUEBAICAEBAQICBAQFBQYGBwCHCAgICQgHCAcHBgYFBQQDAwIBAQIDA
wQFBQYGBwCIBwgB+wH+vQFABP5dOgGkAQEAAAADAAAAANkA5oAnQDxAR4AAAEzHwEdAR8HFQ8DIy
8HDwYdAR8WDw0dAQ8BKwIvAT0BLwc9AT8COWEfBj8HLxc/DTU/AwEfDjsBPxEvFiMPFR8BEw8CFR8
HMz8HNS8GIw8ELwCAQ8BAgoCAGENDAwKCgHBQEBAikCAGIEAwQFDA0SBwCAGIBAQICBgcHBxYK
CQkJCACHBgYFBAMCAQEBAQIDAwQFBQYGBwCPEQECAhUCAQINDAsLCQgHBQICKQICAgQDBAULdhIHB
wYCAQEBAQEBAgYHBwcWCgkKCAgHBwYFBQQDAgEBAQECawMEBAyFBgcHEBABAQED/qwUFRUVFRYWFh
YWFxYXFhcXFxcWFxYXFhYWFhYVFRUVFAQCAQICBAUGCAGJCgsLDA0MDQ0NDBk2EQYgqgYGCEsZDQ0
NDA0MCwsKCQgIBgUEAgIBAgQCAQEBAwKRNRIHBqADChI1DQoFAGEBAGMEBAOMew8eTw4IVxkXCwkJ
BwYCOAIBAIIDAwUGBwgJCgICAQENAQEFawIDAgECAGMFAwMEBAUDBAMFAwIBAQECawMEBAUGBgYHC
AgICQgHBwcGBgYFBQQEBAYDIgICAQECAICBAUGBwgJCQMBAGEMAQUDAwIDAQICBAQDBAQEBQEAw
QEAgEBAQICBAMFBQUGBwCICAgJBwgHBgcGBgUFBAQEBOQIAgEBAf6RDAsLCQkICAYGBQUDAwIBAQI
DAwUFBgYICakJCwsMKSckIiAeGxoYFhQTERAPDQwLCgkIDxsJBQUBQqNEakKCwwNDxARExQWGBob
HiAiJCcCoAMDawQECA8XPRcKCgUPFz0REakIBAMDawMCAQICAwCYAwEaBwQBAgIAAAEAAAAAA/MDN
AA0AAABDwQvAw8EHwQ/ETUnIw8LAYsEJwwGagIwXmMXFBIIcCsQKaEqRUclJSYnJykpKiosLC4GFg
sCAWMhISiIiIjIkJAPRwB8AQmCQMBARQuNgsMDgcIJCYnmyZOTycmJiYlJSQJiiIgHwULCAMCAQ4
RERITFBUVKy0tFgAAABIA3gABAAAAAAAAAAAAEAAAABAAAAAABAAcAAQABAAAAAAACAAcACAABAAAA
AAADAACADwABAAAAAAAEAAcAFgABAAAAAAAFaAsAHQABAAAAAAAGAAcAKAABAAAAAAAKACwALwABA
AAAAAALABIAWwADAAEECQAAAAIAbQADAAEECQABAA4AbwADAAEECQACAA4AfQADAAEECQADAA4AiW
ADAAEECQAEAA4AmQADAAEECQAFABYApwADAAEECQAGAA4AvQADAAEECQAKAFgAywADAAEECQALACQ
BIyBEZwZhdWx0UmVndWxhcRlZmF1bHREZwZhdWx0VmVyc2lvbiAxLjBEZwZhdWx0Rm9udCBnZW5l
cmF0ZWQgdXNpbmcgU3luY2Z1c2lvbiBNZXRybyBTdHVkaW93d3cuc3luY2Z1c2lvbi5jb20AIABEA
GUAZgBhAHUAbAB0AFIAZQBnAHUAbABhAHIArABlAGYAYQB1AGwAdABEAGUAZgBhAHUAbAB0AFYAZQ
ByAHMAAQBVAG4AIAAxAc4AMABEAGUAZgBhAHUAbAB0AEYABwBuAHQAIAbnAGUAbgBlAHIAyQB0AGU
AZAAgAHUAcwBpAG4AZwAgAFMAeQBvAGMAZgB1AHMAAQBVAG4AIAbnAGUAdABYAG8AIAbTAHQAdQBk
AGkAbwB3AHcAdwAuAHMAeQBvAGMAZgB1AHMAAQBVAG4ALgBjAG8AbQAAAAACAAAAAoooooAAAAA
AAAAAAAAAAAAAAAAAAAAAYBAGEDAQBBQEGAQcADXRyYW5zcG9ydC12YW4ldXNlcil1b2R2R2pZn
kRc2hvcHBpbmctY2FydF8wMS0Lc3BlbmQtbW9uZXkFY2h1Y2sAAAA=) format('trueType');
    font-weight: normal;
    font-style: normal;
}

```

```
.stepper-steptype-container [class^="sf-icon-"], .stepper-steptype-container
[class*=" sf-icon-"] {
  font-family: 'Default' !important;
  speak: none;
  font-style: normal;
  font-weight: normal;
  font-variant: normal;
  text-transform: none;
  line-height: inherit;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}
.stepper-steptype-container {
  margin-top: 30px;
  padding: 30px;
}
.stepper-steptype-container .text-stepper {
  margin-top: 50px;
}
.stepper-steptype-container .sf-icon-cart:before { content: "\e710"; }
.stepper-steptype-container .sf-icon-transport:before { content: "\e702"; }
.stepper-steptype-container .sf-icon-payment:before { content: "\e706"; }
.stepper-steptype-container .sf-icon-success:before { content: "\e715"; }
```

## Orientations in React Stepper component

The Stepper component supports the display of steps in both horizontal and vertical orientations by using the [orientation](#) property.

### Horizontal

In horizontal orientation, the steps are displayed in a side-by-side manner by setting the [orientation](#) property to **Horizontal**. By default, the steps are displayed in the horizontal orientation.

### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
  return (
    <div className="stepper-horizontal">
      <StepperComponent orientation="horizontal">
        <StepsDirective>
          <StepDirective iconCss= {'sf-icon-cart'} />
          <StepDirective iconCss= {'sf-icon-transport'} />
          <StepDirective iconCss= {'sf-icon-payment'} />
          <StepDirective iconCss= {'sf-icon-success'} />
        </StepsDirective>
      </StepperComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
```

```
{% endraw %}
```

## INDEX.TSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
 '@syncfusion/ej2-react-navigations';
function App() {
  return (
    <div className="stepper-horizontal">
      <StepperComponent orientation="horizontal">
        <StepsDirective>
          <StepDirective iconCss= {'sf-icon-cart'} />
          <StepDirective iconCss= {'sf-icon-transport'} />
          <StepDirective iconCss= {'sf-icon-payment'} />
          <StepDirective iconCss= {'sf-icon-success'} />
        </StepsDirective>
      </StepperComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

## INDEX.CSS

```
/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
@font-face {
  font-family: 'Default';
  src:
    url(data:application/x-font-ttf;charset=utf-
8;base64,AAEAAAKAIAAAwAgTlMvMjlvSgcAAAEoAAAAVmNtYXCDelPaAAABmAAAAF5nbHlmEwr+
pwAAAggAAAJQaGVhZCYp2+EAADQAAAAANmhoZWEIUQQHAAAArAAAACRobXR4GAAAAAAAYAAAAAYb
G9jYQhUBlAAAAH4AAAAADm1heHABFgErAAABCAAAACBuYW1luf5THQAACTgAAAIlcG9zdJ8LuoMAAA
0AAAAAbwABAAAEAAAAAFwEAAAAAAD9AABAAAAAAAAAAAAAAAAAAAAABgABAAAAAQAArxT6wV8PPPU
ACwQAAAAAOGly6UAAAAA4YvLpQAAAAAD9AOaAAAAACAACAAAAAAAAAAAAEAAAAGAR8ABgAAAAAAgAA
AAoACgAAAP8AAAAAAAAAAQAAZAABQAAAokCzAAAAI8CiQLMAAAB6wAyAQgAAAIABQMAAAAAAAAAAA
AAAAAAAAAAAAAAAAAUGZFZABA5wLnFQQAAAAAXAQAAAAAAAAABAAAAAAAAABAAAAAQAAAAEAAAABA
AAAAQAAAAEAAAAAAAAAgAAAAAMAAUAAQAABQABABKAAAADAIAIAB0cC5wbnC0cQ5xX//wA
A5wLnBucI5xDnFf//AAAAAAAAAAAAAAAAABAawADAAMAawADAAAAAEABAACAAMABQAAAAAAAAAEQAiwC
3AQkBGgAAAAFAAAAAAP0A18APwB/AIkAxgDrAAABHw8/Dy8OKwEPDQUFDz8PLw4rAQ8NAR8FFSM1J
xEfBz80OwEfDjM/BzUnIw8GATM/Dx8PMxEhAq8BAQIEBAUFBwYICAgJCQoKCgkKCAkIBwCHBQUEAw
MBAQEBawMEBQUHBwCICQgKCQoKCgkKCAgIBgcFBQQEAgH+CwEBAgQEBQUHBggICakJCgoKCQoICQg
HBwCFBQQDAwEBAQEDAwQFBQcHBwgJCAoJCgoKCQkICAgGBwUFBAQCAQJ8AwUIWAwD3n0BAwMGBgYI
CAMEBQYHBwkJCgsLDA0NDQ4ODQ4MDAwLCgkJCAYGBQMDKAgIBwYFBAECvLsICAYHBQMD/beAAwQFB
```

```

QcHCAkKCgsLDA0MDg0NDQwLCwsJCQkHBwUFAwKE/eMBAQoJCQkJCAcHBgYFBAMDAQEBAQMDBAUGBg
cHCAkJCQkKCgoJCQgICAcGBgQFAwICAgIDBAUFBgcHCAkJCQoLCgkJCQkIBwcGBgUEAwMBAQEBAwM
EBQYGBwcICQkJCQoKCgkJCAgIBwYGBAUDAgICAgMEBQUGBwcICQkJCgGuAQIGehYJBKYp/10ICAcG
BQQAQ0NDQwLCgoJCAgGBQUDAgIDBQUGCAGJCgoLDA0NDQECBAUGBwQI1foBAGQFBgcH/iwNDawLC
woJCQgHBgUEAwEBAQEDBAUGBwgJCQoLCwwMDQJJAABAgAAAAAD8wOWAAYAqgBaAGwArQDuAAABBz
cfAwUhLwIHlY8PNS8CKwIPHQEHLwEjDwE1LwMjNz0BJzcfBTcfAg8BLwY3OwEfaQcVHw8/Dy8PDw4
BFR8PPw8vDw8OAxEWBgEDAgb8/wNuBAUEDQsVFBQTEhEPDw0GCwoIBgQCFhITE+wQDw8PDg4ODg0N
DQwNCwwKCwoKCQgJBwcHBgYEBQMEASFrBAQDBAMBawMDBgIDagIEBgYGBxwCAwIBFYQYGBAgFBgIWA
gQHCPCBAgQGBggKCgsMDQ4PDxAQEBAPDw4NDAsLCQgGBgQCAQECBAYGCAkLCwwNDg8PEBAQEASPDg
0MCwoKCAYGBAL+KgEEBQgKCw0PEBETFBQWFxcXfHYUFBMREQ4NDakIBgMBAQMGCakMDQ4RERMUFBY
WFxcXfHQUExEQDw0LCggFBAEXBhcFBAMDrxYWDQEBawUHCASMDQ4IERESFBQUFQQDAgECAGMEBAUG
BgYIBwgJCQoKCwsLDAwMDQ0ODQ4PDgEZawIBAQIGBQMCAQQDBgZqBgOHBQMDMAMHBwMWAQICBQYKC
hYCB1wICBAPDw4NDAsLCQgGBgQCAQECBAYGCAkLCwwNDg8PEBAQEASPDg0MCwoKCAYGawMBAQMDBg
YICgoLDA0ODw8QATMLDBYVFRQSERAPDQsKCAUEAQEEBQgKCw0PEBESFBUVFhcXfXYVFBISEA8NCwo
IBQQBAQQFCAoLDQ8QEhIUFRYXAAAAAQAQAAAAA/QDRwA/AH8AhwCRAAABFR8OPw49AS8NKwEPDQUV
Hw4/Dj0BLw0rAQ8NEwcTIRMnMSMhMxMhNSEDBzUha0YBAGMEBAQGBQcGBwgICAgICAgIBwYHBQYEB
AQDAgEBAGMEBAQGBQcGBwgICAgICAgIBwYHBQYEBAQDAgH+aAICAgQEBAYFBwYIBwgICAgICAgHBg
cFBgQEBAMCAQECAwQEBAYFBwYHCAGICAgICAcIBgcFBgQEBaICASh6jAFKjPpu/Z3NwgJZ/dzDAf8
AAQkICAgHBwcGBgUFBAQCAgEBAQECAgQEBQUGBgcHBwgICakIBwgHBwYGBQUEAwMCAQECAwMEBQUG
BgCHCAcICQgICAcHBwYGBQUEBAICAQEBAQICBAQFBQYGBwcHCAgICQgHCAcHBgYFBQQDAwIBAQIDA
wQFBQYGBwcIBwgB+wH+vQFABP5dOgGkAQEAAAADAAAAANkA5oAnQDxAR4AAAEzHwEdAR8HFQ8DIY
8HDWYdAR8WDW0dAQ8BKwIvAT0BLwc9AT8COWEfBj8HLxc/DTU/AwEfDjsBPxEvFiMPFR8BEw8CFR8
HMz8HNS8GIw8ELwQrAQ8BAGoCAGENDAwKCggHBQEBAikCAGIEAwQFDA0SBwcGAGIBAQICBgcHBxYK
CQkJCAcHBgUFBAMCAQEBAQIDAQWFBQYGBwcPEQECAhUCAQINDAsLCQgHBQICKQICAgQDBAULdhIHB
wYCAQEBAQEBAgYHBwcWCgkKCAgHBwYFBQQDAgEBAQECAwMEBAYFBgcHEBABAQED/qwUFRUVFRYWFh
YWFxYXfHcXfXcWFxYXfHYWFhYVFRUVFAQCAQICBAUGCAGJCgsLDA0MDQ0NDBk2EQYgqgYGCEsZDQ0
NDA0MCwsKCQgIBgUEAgIBaQCAQEBAwKRNRIHBqADChI1DQoFagEBAGMEBAOMew8eTw4IVxkXCwkJ
BwYCOAIBAIIDAwUGBwgJCgICAQENAQEFawIDagECAGMFAwMEBAUDBAMFAwIBAQECAwMEBAUGBgYHC
AgICQgHBwcGBgYFBQYEBAYDIgICAQECAiICBAUGBwgJCQMBAGEMAQUDAwIDAQICBAQDBAQEBAQEAW
QEAgEBQICBAMFBQUGBwcICAgJBwgHBgcGBgUFBAQEBAQIAGeBAf6RDAsLCQkICAYGBQUDAwIBAQI
DAWUFBgYICakJCwsMKSckIiAeGxoYfHqTERAPDQwLCgkIDxsJBQYFBQqNEakKCwwNDxARExQWGBob
HiAiJCcCoAMDawQECA8XPRcKCgUPFz0REakIBAMDawMCAQICAwCYAwEaBwQBAGIAAAEAAAAAA/MDN
AA0AAABDwQvAw8EHwQ/ETUnIw8LAYsEJwwGAgIwXmMXFBIIcCsKqKaEqRUclJSYnJykpKiosLC4GFg
sCAWMhISiIiIjIkJAPRwB8AQmCQMBARQuNgsMDgcIJCYnmyZOTycmJiYlJSQjIiIgHwULCAMCAQ4
RERITFBUVKy0tFgAAABIA3gABAAAAAEEEEAAAAABAAAAAABAAcAAQABAAAAAACAACACAABAAAA
AAADAACADwABAAAAAEEAAcAFgABAAAAAFAAsAHQABAAAAAAGAAcAKAABAAAAAAKACwALwABA
AAAAAALABIAWwADAAEECQAAAAIAbQADAAEECQABAA4AbwADAAEECQACAA4AfQADAAEECQADAA4AiW
ADAAEECQAEAA4AmQADAAEECQAFABYApwADAAEECQAGAA4AvQADAAEECQAKAFgAywADAAEECQALACQ
BIyBEZWZhdWx0UmVndWxhcRlZmFlbHREZWZhdWx0VmVyc2lvbiAxLjBEZWZhdWx0Rm9udCBnZW5l
cmF0ZWQgdXNpbmcuU3luY2Z1c2lvbiBNZXRYbyBTdHVkaW93d3cuc3luY2Z1c2lvbi5jb20AIABEA
GUAZgBhAHUAbAB0AFIAZQBnAHUAbABhAHIArABLAGYAYQB1AGwAdABEAGUAZgBhAHUAbAB0AFYAZQ
ByAHMAaQBvAG4AIAAxAx4AMABEAGUAZgBhAHUAbAB0AEYAbwBuAHQAIABnAGUAbgB1AHIAIYQB0AGU
AZAAgAHUAcwBpAG4AZwAgAFMAeQBwAGMAZgB1AHMAaQBvAG4AIAABNAGUAdABYAG8AIABTAHQAdQBk
AGkAbwB3AHcAdwAuAHMAeQBwAGMAZgB1AHMAaQBvAG4ALgBjAG8AbQAAAAACAAAAAoooooAAAAA
AAAAAAAAAAAAAAAAAAAAAYBAGEDAQBBQEGAQcADXRyYW5zcG9ydC12YW4ldXNlcil1b2R2R2Zn
kRc2hvcHBpbmctY2FydF8wMS0Lc3BlbmQtbW9uZXkFY2h1Y2sAAAA=) format('trueType');
    font-weight: normal;
    font-style: normal;
}
.stepper-horizontal [class^="sf-icon-"], .stepper-horizontal [class*=" sf-
icon-"] {
    font-family: 'Default' !important;
    speak: none;
    font-style: normal;
    font-weight: normal;
    font-variant: normal;
    text-transform: none;

```

```

line-height: inherit;
-webkit-font-smoothing: antialiased;
-moz-osx-font-smoothing: grayscale;
}
.stepper-horizontal {
  margin-top: 30px;
  padding: 30px;
}
.stepper-horizontal .sf-icon-cart:before { content: "\e710"; }
.stepper-horizontal .sf-icon-transport:before { content: "\e702"; }
.stepper-horizontal .sf-icon-payment:before { content: "\e706"; }
.stepper-horizontal .sf-icon-success:before { content: "\e715"; }

```

## Vertical

You can display the steps one below the other vertically by setting the [orientation](#) property to **Vertical**.

### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
  return (
    <div className="stepper-vertical">
      <StepperComponent orientation="vertical">
        <StepsDirective>
          <StepDirective iconCss= {'sf-icon-cart'} />
          <StepDirective iconCss= {'sf-icon-transport'} />
          <StepDirective iconCss= {'sf-icon-payment'} />
          <StepDirective iconCss= {'sf-icon-success'} />
        </StepsDirective>
      </StepperComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

### INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
  return (
    <div className="stepper-vertical">
      <StepperComponent orientation="vertical">
        <StepsDirective>
          <StepDirective iconCss= {'sf-icon-cart'} />
          <StepDirective iconCss= {'sf-icon-transport'} />

```

```

        <StepDirective iconCss= {'sf-icon-payment'} />
        <StepDirective iconCss= {'sf-icon-success'} />
      </StepsDirective>
    </StepperComponent>
  </div>
);
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

## INDEX.CSS

```

/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
@font-face {
  font-family: 'Default';
  src:
    url(data:application/x-font-ttf;charset=utf-
8;base64,AAEAAAAKAIAAAwAgTlMvMjlvSgcAAAEoAAAAVmNtYXCDelPaAAABmAAAAF5nbHlmEwr+
pwAAAggAAAJQaGVhZCYp2+EAADQAAAAANmhoZWEIUQQHAAAArAAAACRobXR4GAAAAAAYAAAAAYb
G9jYQhUBlAAAAH4AAAAADmlheHABFgErAAABCAAAACBuYWllluF5THQAACtgAAAIlcG9zdJ8LuoMAAA
0AAAAAbwABAAAEAAAAAFwEAAAAAAD9AABAAAAAABAAAAAABAAAAAQAARxT6wV8PPPU
ACwQAAAAAAGLy6UAAAAA4YvLpQAAAAAD9AOaAAAAACAACAAAAAABAAAAAAGAR8ABgAAAAAAGAA
AAoACgAAAP8AAAAAQAQAAZAABQAAAokCzAAAAI8CiQLMAAAB6wAyAQgAAAIABQMAAAAAAABAAAA
AAAAAAAAAAAAAAAAAUGZFZABA5wLnFQAAAAAXAQAAAAAABAAAAAABAAAAAQAQAAAAEAAAAABA
AAAAQAQAAAEAAAAAAGAAAAAUAAMAAQAABQABABKAAAAADAAIAAIBOC5wbnCOCQ5xX//wA
A5wLnBucI5xDnFf//AAAAAAAAAAAAAAAAAABAawADAAMAawADAAAAAEABAACAAMABQAAAAAAAEQAIwC
3AQkBGgAAAAFAAAAAAP0A18APwB/AIkAxgDrAAABHw8/Dy8OKwEPDQUfDz8PLw4rAQ8NAR8FFSM1J
xEfBz80OwEfDjM/BzUnIw8GATM/Dx8PMxEhAQ8BAQIEBAUFbWYICAgJCQoKCgkKCAkIBwCHBQUEAw
MBAQEBAwMEBQUHBwCICQgKCQoKCgkKJCAgIBgcFBQQEAgH+CwEBAgQEBQUHBggICakJCgoKCQoICQg
HBwCQBQDAwEBAQEDAwQFBQcHBwgJCAoJCgoKCQkICAgGBwUFBAQCAQJ8AwUIWAwD3n0BAwMGBgYI
CAMEBQYHBwkJCgsLDA0MDQ4ODQ4MDAwLCgkJCAYGBQMDKAgIBwYFBAECvLsICAYHBQMD/beAAwQFB
QcHCAkKCgsLDA0MDg0NDQwLCwsJCQkHBwUFawKE/eMBAQoJCQkJCAcHBgYFBAMDAQEBAQMDBAUGBg
cHCAkJCQkKCgoJCQgICAcGBgQFAwICAgIDBAUFbGcHCAkJCQoLCgkJCQkIBwCGBgUEAwMBAQEBAwM
EBQYGBwCICQkJCQoKCgkKJCAgIBwYGBAUDAQICAgMEBQUGBwCICQkJCgGuAQIGehYJBKYp/10ICAcG
BQCAQ0NDQwLCgoJCAgGBQUdAgIDBQUGCAgJCgoLDA0NDQECBAUGBwQI1foBAGQFBgcH/iwNDawLC
woJCQgHBgUEAwEBAQEDBAUGBwgJCQoLCwwMDQJJAAAABgAAAAAD8wOWAAYAQgBaAGwArQDuAAABBz
cfAwUhLwIHlY8PNS8CKwIPHQEHLwEjDwE1LwMjNz0BJzcfBTcfAg8BLwY3OwEfaQcVHw8/Dy8PDw4
BFR8PPw8vDw8OAxEWBgEDAg8/wNuBAUEDQsVFBQTEhEPDw0GCwoIBgQCFhITE+wQDw8PDg4ODg0N
DQwNCwwKCwoKCQgJBwCHBgYEBQMEA5FrBAQDBAMBawMDBgIDagIEBgYGBxwCAwIBFQYGBAgFBgIWA
gQHCPcBAgQGBggKCgsMDQ4PDxAQEBAPDw4NDAsLCQgGBgQCAQECBAYGCAkLCwwNDg8PEBAQEAE8PDg
0MCwoKCAYGBAL+KgEEBQgKCw0PEBETFBQWFxcXFhYUFBMREQ4NDakIBgMBAQMGCakMDQ4RERMUFBY
WFxcXFhQUExEQDw0LCgGFBAEXBhcFBAMDrxYWDQEBawUHCAsmDQ4IERESFBQUFQQDAGECAGMEBAUG
BgYIBwgJCQoKCwsLDAwMDQ0ODQ4PDgEZawIBAQIGBQMCAQQDBgZqBgOHBQMDMAMHBwMWAQICBQYKC
hYCB1wICBAPDw4NDAsLCQgGBgQCAQECBAYGCAkLCwwNDg8PEBAQEAE8PDg0MCwoKCAYGAwMBAQMDBg
YICgoLDA0ODw8QATMLDBYVFRQSERAPDQsKCAUEAQEEBQgKCw0PEBESFBUVfHcXFxYVFBISEA8NCwo
IBQQAQQAQCAoLDQ8QEHlUFYRYAAAAAQAQAAAAA/QDRwA/AH8AhwCRAAABFR8OPw49AS8NKwEPDQUV
Hw4/Dj0BLw0rAQ8NEwCTIRMnMSMhMxMhNSEDBzUha0YBAGMEBAQGBQcGBwgICAgICAgIBwYHBQYEB

```

```

AQDAgEBAGMEBAQGBQcGBwgICAgICAgIBwYHBQYEBAQDAgH+aAICAgQEBAYFBwYIBwgICAgICAgHBg
cFBgQEBAMCAQECAwQEBAYFBwYHCAGICAgICAcIBgcFBgQEBAICAsH6jAFKjPpu/Z3NwgJZ/dzDAf8
AAQkICAgHBwcGBgUFBAQCAgEBAQECAgQEBQUGBgCHBwgICAkIBwgHBwYGBQUEAwMCAQECAwMEBQUG
BgCHCAcICQgICAcHBwYGBQUEBAICAQEBAQICBAQFBQYGBwcHCAgICQgHCAcHBgYFBQQDAwIBAQIDA
wQFBQYGBwcIBwgB+wH+vQFABP5dOgGkAQEAAAADAAAAANkA5oAnQDxAR4AAAEzHwEdAR8HFQ8DIy
8HDWydAR8WDW0dAQ8BKwIvAT0BLwc9AT8COWEfBj8HLxc/DTU/AwEfDjsBPxEvFiMPFR8BEw8CFR8
HMz8HNS8GIw8ELwQrAQ8BAgoCAgENDAwKCgghBQEBAikCAgIEAwQFDA0SBwcGAgIBAQICBgCHBxYK
CQkJCAcHBgUFBAQCAQEBAQIDAQFBQYGBwcPEQECAhUCAQINDAsLCQgHBQICKQICAgQDBAULdhIHB
wYCAQEBAQEBAgYHBwcWCgkKCAGHBwYFBQQDAgEBAQECAwMEBAYFBgcHEBABAQED/qwUFRUVFRYWFh
YWFxYXFhcXFxcWFxYXFhYWFhYVFRUVFAQCAQICBAUGCAGJCgsLDA0MDQ0NDBk2EQYGqgYGCEsZDQ0
NDA0MCwsKCQgIBgUEAgIBAgQCAQEBAwKRNRIHBqADChI1DQoFAGEBAGMEBAoMEw8eTw4IVxkXCwkJ
BwYCOAIBAIIDAUGBwgJCgICAQENAEFAwIDAQECAGMFAwMEBAUDBAMFAwIBAQECAwMEBAUGBgYHC
AgICQgHBwcGBgYFBQQEBAyDIgICAQECAiICBAUGBwgJCQMBAGEMAQUDAwIDAQICBAQDBAQEBAQEAw
QEAgEBAQICBAMFBQUGBwcICAgJBwgHBgcGBgUFBAQEBAQAgEBAf6RDAsLCQkICAYGBQUDAwIBAQI
DAwUFBgYICAKJCwsMKSckIiAeGxoYFhQTERAPDQwLCgkIDxsJBQUBBQnEAKKCwWNDxARExQWGBob
HiAiJCcCoAMDawQECA8XPRcKCgUPFz0REAKIBAMDawMCAQICAwcYAwEaBwQBAGIAAAEAAAAAA/MDN
AA0AAABDwQvAw8EHwQ/ETUnIw8LAYsEJwwGAGIwXmMXFBIICCsqKaEqRUclJSYnJykpKiosLC4GFg
sCAWMhISiIiIjIkJAPRwB8AQmQMBARQuNgsMDgcIJCYnmyZOTycmJiYlJSQjIiIgHwULCAMCAQ4
RERITFBUVKy0tFgAAABIA3gABAAAAAEEAABAAAAAABAAcAAQABAAAAAACAACAABAAAA
AAADAACADWABAAAAAEEAACAFgABAAAAAFAAsAHQABAAAAAAGAAcAKAABAAAAAAKACwALWABA
AAAAAALABIawADAAEEECQAAAAIAbQADAAEEECQABAA4AbwADAAEEECQACAA4AfQADAAEEECQADAA4Aiw
ADAAEEECQAEAA4AmQADAAEEECQAFABYApwADAAEEECQAGAA4AvQADAAEEECQAKAFgAywADAAEEECQALACQ
BIyBEZWZhdWx0UmVndWxhckRlZmFlbHREZWZhdWx0VmVyc2lvbiAxLjBEZWZhdWx0Rm9udCBnZW5l
cmF0ZWQgdXNpbmcgU3luY2Z1c2lvbiBNZXRYbyBTdHVkaW93d3cuc3luY2Z1c2lvbi5jb20AIABEA
GUAZgBhAHUAbAB0AFIAZQBnAHUAbABhAHIArABLAGYAYQBLAGwAdABEAGUAZgBhAHUAbAB0AFYAZQ
ByAHMAaQBvAG4AIAAxA4AMABEAGUAZgBhAHUAbAB0AEYAbwBuAHQAIABnAGUAbgBLAHIAAYQB0AGU
AZAAGAHUAcwBpAG4AZwAgAFMAeQBwAGMAZgB1AHMAaQBvAG4AIAABNAGUAdABYAG8AIABTAHQAdQBk
AGkAbwB3AHcAdwAuAHMAeQBwAGMAZgB1AHMAaQBvAG4ALgBjAG8AbQAAAAACAAAAAaAAAAA
AAAAAAAAAAAAAAAAAAAAAYBAGEDAQQBBQEQAQcADXRyYW5zcG9ydC12YW4LdXNlcil1b2R2Rn
kRc2hvcHBpmbmctY2FydF8wMS0Lc3BlbmQtbW9uZXkFY2h1Y2sAAAA=) format('trueype');
    font-weight: normal;
    font-style: normal;
}
.stepper-vertical [class^="sf-icon-"], .stepper-vertical [class*=" sf-icon-"]
{
    font-family: 'Default' !important;
    speak: none;
    font-style: normal;
    font-weight: normal;
    font-variant: normal;
    text-transform: none;
    line-height: inherit;
    -webkit-font-smoothing: antialiased;
    -moz-osx-font-smoothing: grayscale;
}
.stepper-vertical {
    margin-top: 30px;
    padding: 30px;
    min-height: 450px;
    display: flex;
    justify-content: center;
}
.stepper-vertical .sf-icon-cart:before { content: "\e710"; }
.stepper-vertical .sf-icon-transport:before { content: "\e702"; }
.stepper-vertical .sf-icon-payment:before { content: "\e706"; }
.stepper-vertical .sf-icon-success:before { content: "\e715"; }

```



## Linear flow in React Stepper component

The Stepper component enables users to progress sequentially through each step, ensuring navigation from one step to the next in a linear way by setting the [linear](#) property to `true`. The default value is `false` allowing navigation between any steps and vice versa.

The example demonstrates the functionality of both linear and non-linear flow in the Stepper.

### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
    const linearStepper = React.useRef(null);
    const updateLinear = (args) => {
        linearStepper.current.linear =
        (//true/).test(args.currentTarget.value) ? true : false;
        linearStepper.current.reset();
    };
    return (
        <div className="stepper-linear-position">
            <div className="e-btn-group">
                <input type="radio" id="linear" name="Linear" value="true"
onClick={updateLinear} defaultChecked />
                <label className="e-btn" htmlFor="linear">Linear</label>
                <input type="radio" id="nonLinear" name="Linear"
value="false" onClick={updateLinear} />
                <label className="e-btn" htmlFor="nonLinear">Non-
Linear</label>
            </div>
            <StepperComponent ref={linearStepper} id="stepper-linear-control"
linear={true}>
                <StepsDirective>
                    <StepDirective iconCss={'sf-icon-cart'} label={'Cart'} />
                    <StepDirective iconCss={'sf-icon-transport'}
label={'Delivery Address'} />
                    <StepDirective iconCss={'sf-icon-payment'}
label={'Payment'} />
                    <StepDirective iconCss={'sf-icon-success'}
label={'Confirmation'} />
                </StepsDirective>
            </StepperComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

### INDEX.TSX

```
{% raw %}
```

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
    const linearStepper = React.useRef<StepperComponent>(null);
    const updateLinear = (args: any) => {
        linearStepper.current.linear =
        (/true/).test(args.currentTarget.value) ? true : false;
        linearStepper.current.reset();
    };
    return (
        <div className="stepper-linear-position">
            <div className="e-btn-group">
                <input type="radio" id="linear" name="Linear" value="true"
onClick={updateLinear} defaultChecked />
                <label className="e-btn" htmlFor="linear">Linear</label>
                <input type="radio" id="nonLinear" name="Linear"
value="false" onClick={updateLinear} />
                <label className="e-btn" htmlFor="nonLinear">Non-
Linear</label>
            </div>
            <StepperComponent ref={linearStepper} id="stepper-linear-control"
linear={true}>
                <StepsDirective>
                    <StepDirective iconCss={'sf-icon-cart'} label={'Cart'} />
                    <StepDirective iconCss={'sf-icon-transport'}
label={'Delivery Address'} />
                    <StepDirective iconCss={'sf-icon-payment'}
label={'Payment'} />
                    <StepDirective iconCss={'sf-icon-success'}
label={'Confirmation'} />
                </StepsDirective>
            </StepperComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

## INDEX.CSS

```

/* Represents the styles for loader */
#loader {
    color: #008cff;
    height: 40px;
    left: 45%;
    position: absolute;
    top: 45%;
    width: 30%;
}
@font-face {
    font-family: 'Default';
    src:

```

```
url(data:application/x-font-ttf;charset=utf-
8;base64,AAEAAAAKIAIAAwAgTlMvMjlvSgcAAAEoAAAAVmNtYXCDeIPaAAABmAAAAF5nbHlmEwr+
pwAAAggAAAJQaGVhZCYp2+EAADQAAAAANmhoZWEIUQQHAAAArAAAACRobXR4GAAAAAAYAAAAAYb
G9jYQhUBlAAAAH4AAAAADm1heHABFgErAAABCAAAACBuYw1luF5THQAACtgAAAIlcG9zdJ8LuoMAAA
0AAAAAbwABAAAEAAAAAFwEAAAAAAD9AABAAAAAABAAAAAABAAAAAQAARxT6wV8PPPU
ACwQAAAAAAGLy6UAAAAA4YvLpQAAAAAD9A0aAAAAACAACAAAAAABAAAAAGAR8ABgAAAAAAAgAA
AAoACgAAAP8AAAAAQAQAAZAABQAAAokCzAAAAI8CiQLMAAAB6wAyAQgAAAIABQMAAAAAAABAAA
AAAAAAAAAAAAAAAAAUGZFZABA5wLnFQQAAAAAXAQAAAAAABAAAAAABAAAAAQAABAAAAEAAAAABA
AAAAQAAAAEAAAAAAGgAAAAAMAAAUAMAAQAAABQABABKAAAAADAAIAAIABoCc5wbnCOcQ5xX//wA
A5wLnBucI5xDnFf//AAAAAAAAAAAAAAAAABAawADAAMAawADAAAAAEABAACAAMABQAAAAAAAEQAIwC
3AQkBGgAAAAFAAAAAAP0A18APwB/AIkAxgDrAAABHw8/Dy8OKwEPDQUfDz8PLw4rAQ8NAR8FFSM1J
xEfBz800wEfDjM/BzUnIw8GATM/Dx8PMxEhAq8BAQIEBAUFbWYICAgJCQoKCgkKCAkIBwcHBQIEAw
MBAQEBAwMEBQUHBwCICQgKCQoKCgkKCAgIBgcFBQqEAgH+CwEBAgQEBQUHBwGICAgJCgoKCQoICQg
HBwCfBQqDAwEBAQEDAwQFBQcHBwgJCAoJCgoKCQkICAgGBwUFBAQCAQJ8AwUIWawD3n0BAwMGBgYI
CAMEBQYHBwkJCgsLDA0NDQ4ODQ4MDAwLCgkJCAYGBQMDKAgIBwYFBAECvLsICAYHBQMD/beAAwQFB
QcHCAkKCgsLDA0MDg0NDQwLCwsJCQkHBwUFawKE/eMBAQoJCQkJCACHBgYFBAMDAQEBAQMDBAUGBg
cHCAkJCQkKCgoJCQgICAcGBgQFAwICAgIDBAUFbGcHCAkJCQoLCgkJCQkIBwCGBgUEAwMBAQEBAwM
EBQYGBwCICQkJCQoKCgkJCAGIBwYGBAUDAgICAgMEBQUGBwCICQkJCgGuAQIGehYJBKYp/10ICAcG
BQqCAQ0NDQwLCgoJCAgGBQUDAgIDBQUGCAGJCgoLDA0NDQCEBAUGBwQI1foBAGQFBgCgH/iwNDawLC
woJCQgHBgUEAwEBAQEDBAUGBwgJCQoLCwwMDQJJAABAgAAAAAD8wOWAAYAqgBaAgwArQDuAAABZ
cfAwUhLwIH1y8PNS8CKwIPHQEHLwEjDwE1LwMjNz0BJzcFBTcfAg8BLwY3OwEfAQcVHw8/Dy8PDw4
BFR8PPw8vDw8OAxEWBgEDAg8/wNuBAUEDQsVFBQTEhEPDw0GCwoIBgQCfHITE+wQDw8PDg4ODg0N
DQwNCwwKCwoKCQgJBwCfHBgYEBQMEa5FrBAQDBAMBawMDBgIDagIEBgYGBxwCAwIBFQYGBAgFBgIWA
gQHCPcBAgQGBgKCGsMDQ4PDxAQEBApDw4NDAsLCQgGBgQCAQECEBAYGCAkLCwwNDg8PEBAQEa8PDg
0MCwoKCAyGBAL+KgEEBQgKCw0PEBETFBQWFxcXFhYUFBMREQ4NDakIBgMBAQMGCakMDQ4RERMUFBY
WFxcXFhQUExEQDw0LCgGFBAEXBhcFBAMDrxYWDQEBawUHCAsMDQ4IERESFBQUFQqDAGECAGMEBAUG
BgYIBwgJCQoKCwsLDAwMDQ0ODQ4PDgEZawIBAQIGBQMCAQQDBgZqBgOHBQMDMAMHBwMWAQICBQYKC
hYCB1wICBAPDw4NDAsLCQgGBgQCAQECEBAYGCAkLCwwNDg8PEBAQEa8PDg0MCwoKCAyGAwMBAQMDBg
YICgoLDA0ODw8QATMLDBYVFRQSERAPDQsKCAUEAQEEBQgKCw0PEBESFBUVFhcXFxYVFBISEA8NCwo
IBQQAQqFCAoLDQ8QEH1UFRYXAAAAAQAaaaaa/QDRwA/AH8AhwCRAAABFR8OPw49AS8NKwEPDQUV
Hw4/Dj0BLw0rAQ8NEwCTIRmMMSMhMxMhNSEDBzUha0YBAgMEBAQGBQcGBwgICAgICAgIBwYHBQYEB
AQDAGEBAGMEBAQGBQcGBwgICAgICAgIBwYHBQYEBAQDAGH+aaICAgQEBAYFBwYIBwgICAgICAgHBg
cFBgQEBAMCAQECAwQEBAYFBwYHCAgICAgICAcIBgcFBgQEBaICAsH6jAFKjPpu/Z3NwgJZ/dzDAf8
AAQkICAgHBwCGBgUFBAQCAgEBAQECAgQEBQUGBgcHBwgICAkIBwgHBwYGBQUEAwMCAQECAwMEBQUG
BgCHCAcICQgICAcHBwYGBQUEBAICAQEBAQICBAQFBQYGBwCHCAgICQgHCAcHBgYFBQqDAwIBAQIDA
wQFBQYGBwCIBwGB+wH+vQFABP5dOgGkAQEAAAADAAAAANkA5oAnQDxAR4AAAEzHwEdAR8HFQ8DIy
8HDwYdAR8WDw0dAQ8BKwIvAT0BLwc9AT8COWEfBj8HLxc/DTU/AwEfDjsBPxEvFiMPFR8BEw8CFR8
HMz8HNS8GIw8ELwQrAQ8BAGoCAGENDAwKCgGHBQEBaIkCAGIEAwQFDA0SBwCGAgIBAQICBgcHBxYK
CQkJCAcHBgUFBAQCAQEBAQIDAQFBQYGBwCPEQECAhUCAQINDAsLCQgHBQICKQICAgQDBAULdhIHB
wYCAQEBAQEBAgYHBwCWCgkKCAgHBwYFBQqDAgEBAQECAwMEBAYFBgCHEBABAQED/qwUFRUVFRYWFh
YWFxYXFhcXFxcWFxYXFhYWFhYVFRUVFAQCAQICBAUGCAGJCgsLDA0MDQ0NDBk2EQYQGgYGCEsZDQ0
NDA0MCwsKCQgIBgUEAgIBaQCAQEBAwKRNRIHBqADChI1DQoFAgEBAgMEBAoMEw8eTw4IVxkXCwkJ
BwYCOAIBAIDAwUGBwgJCgICAQENAQEFawIDAQECAGMFAwMEBAUDBAwMFAwIBAQECAwMEBAUGBgYHC
AgICQgHBwCGBgYFBQqEBAYDIgICAQECAiICBAUGBwgJCQMBAGEMAQUDAwIDAQICBAQDBAQEBAQEAW
QEAgEBAQICBAMFBQUGBwCICAgJBwgHBgcGBgUFBAQEBOQIAGeBAf6RDAsLCQkICAYGBQUDAwIBAQI
DAwUFBgYICAKJCwsMKSckIiAeGxoYFhQTERAPDQwLCgkIDxsJBQUBQqNEakKCwwNDxARExQWGBob
HiAiJCcCwQMDAwQECA8XPRcKCgUPFz0REAKIBAMDawMCAQICAwCYawEaBwQBAgIAAAEAAAAA/MDN
AA0AAABDQAwAw8EHwQ/ETUnIw8LAYsEJwwGAgIwXmMXFBIIcCsqKaEqRUclJSYnJykpKiosLC4GFg
sCAWMhISiIiIjIkJAPRwB8AQmCQMBARQuNgsMDgcIJCYnmYzOTycmJiYlJSQjIiIgHwULCAMCAQ4
RERITFBUVKy0tFgAAABIA3gABAAAAAABAAAAEAAAAAABAAAAAABAAAAAQAABAAAAAACAABAAAA
AAADAACADwABAAAAAEEAaAFgABAAAAAFAAAsAHQABAAAAAAGAAcAKAABAAAAAABKACwALwABA
AAAAAALABIAWwADAAEECQAAAAIAbQADAAEECQABAA4AbwADAAEECQACAA4AfQADAAEECQADAA4AiW
ADAAEECQAEAA4AmQADAAEECQAFABYApwADAAEECQAGAA4AvQADAAEECQAKAFgAywADAAEECQALACQ
BIyBEZwZhdWx0UmVndWxhcRlZmF1bHREZwZhdWx0VmVyc2lvbiAxLjBEZwZhdWx0Rm9udCBnZW5l
cmF0ZWQgdXNpbmcgU3luY2Z1c2lvbiBNZXRybyBTdHVkaW93d3cuc3luY2Z1c2lvbi5jb20AIABEA
GUAZgBhAHUAbAB0AFIAZQBnAHUAbABhAHIArABLAGYAYQB1AGwAdABEAGUAZgBhAHUAbAB0AFYAZQ
ByAHMAaQBvAG4AIAAxAAC4AMABEAGUAZgBhAHUAbAB0AEYAbwBuAHQAIABnAGUAbgBlAHIAAYQB0AGU
```

```

AZAAgAHUAcwBpAG4AZwAgAFMAeQBvAGMAZgB1AHMAaQBvAG4AIABNAGUAdABYAG8AIABTAHQAdQBk
AGkAbwB3AHcAdwAuAHMAeQBvAGMAZgB1AHMAaQBvAG4ALgBjAG8AbQAAAAACAAAAAaAAAAA
AAAAAAAAAAAAAAAAAAAAAAYBAGEDAQQBBQEGAQcADXRyYW5zcG9ydC12YW4LdXNlci1tb2RpZn
kRc2hvcHBpbnctY2FydF8wMS0Lc3BlbmQtbW9uZXkFY2hlY2sAAAA=) format('truetype');
font-weight: normal;
font-style: normal;
}
#stepper-linear-control [class^="sf-icon-"], #stepper-linear-control
[class*=" sf-icon-"] {
font-family: 'Default' !important;
speak: none;
font-style: normal;
font-weight: normal;
font-variant: normal;
text-transform: none;
line-height: inherit;
-webkit-font-smoothing: antialiased;
-moz-osx-font-smoothing: grayscale;
}
.stepper-linear-position {
text-align: center;
margin-top: 30px;
padding: 30px;
}
#stepper-linear-control {
margin: 50px 100px;
}
#stepper-linear-control .sf-icon-cart:before { content: "\e710"; }
#stepper-linear-control .sf-icon-transport:before { content: "\e702"; }
#stepper-linear-control .sf-icon-payment:before { content: "\e706"; }
#stepper-linear-control .sf-icon-success:before { content: "\e715"; }

```

## Steps validation in React Stepper component

The Stepper component allows you to set the validation state for each step, displaying either a success or error icon. You can define the success state of a step by setting the `isValid` property to `true`. If set to `false`, the step will display an error state. By default, the `isValid` property is `null`.

Based on the `stepType`, the validation state icon will be displayed either as an indicator or as part of the step label/text.

### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
return (


Copyright © 2001 -2024 Syncfusion Inc.



1248


```

```

        <StepDirective iconCss= {'sf-icon-success'} />
      </StepsDirective>
    </StepperComponent>
  </div>
);
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

## INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
 '@syncfusion/ej2-react-navigations';
function App() {
  return (
    <div className="stepper-validation">
      <StepperComponent>
        <StepsDirective>
          <StepDirective iconCss= {'sf-icon-cart'} isValid={true}/>
          <StepDirective iconCss= {'sf-icon-transport'} />
          <StepDirective iconCss= {'sf-icon-payment'} />
          <StepDirective iconCss= {'sf-icon-success'} />
        </StepsDirective>
      </StepperComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

## INDEX.CSS

```

/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
@font-face {
  font-family: 'Default';
  src:
    url(data:application/x-font-ttf;charset=utf-
8;base64,AAEAAAIAIAAwAgTlMvMjlvSgcAAAEoAAAAVmNtYXCDelPaAAABmAAAAF5nbHlmEwr+
pwAAAggAAAJQaGVhZCYp2+EAADQAAAAANmhoZWEIUQQHAAArAAAACRobXR4GAAAAAAAAAYAAAAyb
G9jYQhUBlAAAAH4AAADm1heHABFgErAAABCAAAACBuYW11luF5THQAACTgAAAIlcG9zdJ8LuoMAAA
0AAAAAbwABAAAEAAAAAFwEAAAAAAD9AABAAAAAAAAAAAAAAAAAAAAABgABAAAAQAARxT6wV8PPPU
ACwQAAAAAOGly6UAAAAA4YvLpQAAAAAD9AOaAAAAACAACAAAAAAAAAAAAEAAAAGAR8ABgAAAAAAgAA
AAoACgAAAP8AAAAAAAAAAQAAZAABQAAAokCzAAAAI8CiQLMAAAB6wAyAQgAAAIABQMAAAAAAAAAAA

```

```
AAAAAAAAAAAAAAAAAUGZFZABA5wLnFQQAAAAAXAQAAAAAAAAABAAAAAAAABAAAAAQAAAAEAAAABA
AAAAQAAAAEAAAAAAAAAgAAAAAMAAUAAMAAQAAABQABABKAAAADAAIAAIABOCc5wbnCocQ5xX//wA
A5wLnBucI5xDnFf//AAAAAAAAAAAAAAAAABAAwADAAMAawADAAAAEABAACAAMABQAAAAAAAAEQaiwC
3AQkBGgAAAAFAAAAAAP0A18APwB/AIkAxgDrAAABHw8/Dy8OKwEPDQUfDz8PLw4rAQ8NAR8FFSM1J
xEfBz800wEfDjM/BzUnIw8GATM/Dx8PMxEhAq8BAQIEBAUFBwYICAgJCQoKCgkKCAkIBwcHBQUEAw
MBAQEBAwMEBQUHBwCICQgKCQoKCgkJCAgIBgcFBQQEAgH+CwEBAgQEBQUHBggICakJCgoKCQoICQg
HBwCQBQDQAwEBAQEDAwQFBQcHBwgJCAoJCgoKCQkICAgGBwUFBAQCAQJ8AwUIAwD3n0BAwMGBgYI
CAMEBQYHBwkJCgsLDA0NDQ4ODQ4MDAwLCgkJCAYGBQMDKAgIBwYFBAECvLsICAYHBQMD/beAAwQFB
QcHCAkKCgsLDA0MDg0NDQwLCwsJCQkHBwUFAwKE/eMBAQoJCQkJCACHBgYFBAMDAQEBAQMDBAUGBg
cHCAkJCQkKCgoJCQgICAcGBgQFAwICAgIDBAUFBgHCakJCQoLCgkJCQkIBwcGBgUEAwMBAQEBAwM
EBQYGBwCICQkJCQoKCgkJCAgIBwYGBAUDAgICAgMEBQUGBwCICQkJCgGuAQIGehYJBKYp/10ICAcG
BQQCAQ0NDQwLCgoJCAgGBQUDAgIDBQUGCAgJCgoLDA0NDQECBAUGBwQI1foBAGQFBgCgH/iwNDAwLC
woJCQgHBgUEAwEBAQEDBAUGBwgJCQoLCwwMDQJJAABAgAAAAAD8wOWAAYAqgBaAGwArQDuAAABBz
cfAwUhLwIHlY8PNS8CKwIPHQEHLwEjDwE1LwMjNz0BjzcfBTcfAg8BLwY30wEfAQcVHw8/Dy8PDw4
BFR8PPw8vDw8OAxEWBgEDAg8/wNuBAUEDQsVFBQTEhEPDw0GCwoIBgQCFhITE+wQDw8PDg4ODg0N
DQwNCwwKCwoKCQgJBwCHBgYEBQMEA5FrBAQDBAMBAwMDBgIDagIEBgYGBxwCAwIBFQYGBAgFBgIWA
gQHCPcBAGQGBgKCGsMDQ4PDxAQEBAPDw4NDAsLCQgGBgQCAQECBAYGCAkLCwwNDg8PEBAQEA8PDg
0MCwoKCAyGBAL+KgEEBQgKCw0PEBETFBQWFxcXFhYUFBMREQ4NDakIBgMBAQMGCakMDQ4RERMUFBY
WFxcXFhQUEExEQDw0LCggFBAEXBhcFBAMDrxYWDQEBawUHCAsmDQ4IERESFBQUFQQDagECAGMEBAUG
BgYIBwgJCQoKCwsLDAwMDQ0ODQ4PDgEZawIBAQIGBQMCQQDBgZqBgOHBQMDMAMHBwMWAQICBQYKC
hYCB1wICBAPDw4NDAsLCQgGBgQCAQECBAYGCAkLCwwNDg8PEBAQEA8PDg0MCwoKCAyGawMBAQMDBg
YICgoLDA0ODw8QATMLDBYVFRQSERAPDQsKCAUEAQEEBQgKCw0PEBESFBUVFhcXFxYVFBISEA8NCwo
IBQQBAQQFCAoLDQ8QEHlUFRYXAAAAAQAAAAA/QDRwA/AH8AhwCRAAABFR8OPw49AS8NKwEPDQUV
Hw4/Dj0BLw0rAQ8NEwcTIRMnMSMhMxMhNSEDBzUha0YBAGMEBAQGBQcGBwgICAgICAgIBwYHBQYEB
AQDagEBAgMEBAQGBQcGBwgICAgICAgIBwYHBQYEBAQDagH+aaICAgQEBAYFBwYIBwgICAgICAgHBg
cFBgQEBAMCAQECAwQEBAYFBwYHCAgICAgICAcIBgcFBgQEBaICAsH6jAFKjPpu/Z3NwgJZ/dzDAf8
AAQkICAgHBwCGBgUFBAQCAgEBAQECAgQEBQUGBgCgHBwgICakIBwgHBwYGBQUEAwMCAQECAwMEBQUG
BgCHCAcICQgICAcHBwYGBQUEBAICAQEBAQICBAQFBQYGBwCHCAgICQgHCAcHBgYFBQQDAwIBAQIDA
wQFBQYGBwCIBwgB+wH+vQFABP5dOgGkAQEAAAAADAAAAANkA5oAnQDxAR4AAAEzHwEdAR8HFQ8DIY
8HDwYdAR8WDw0dAQ8BKwIvAT0BLwc9AT8COWEfBj8HLxc/DTU/AwEfDjsBPxEvFiMPFR8BEw8CFR8
HMz8HNS8GIw8ELwQrAQ8BAGoCAGENDAwKCgHBQEBAikCAGIEAwQFDA0SBwCAGIBAQICBgcHBxYK
CQkJCACHBgUFBMCAQEBAQIDAwQFBQYGBwCPEQECAhUCAQINDAsLCQgHBQICKQICAgQDBAULDhIHB
wYCAQEBAQEBAgYHBwCWCgkKCAgHBwYFBQQDagEBAQECAwMEBAYFBgCHEBABAQED/qwUFRUVFRYWFh
YWFxYXFhcXFxcWFxYXFhYWFhYVFRUVFAQCAQICBAUGCAGJCgsLDA0MDQ0NDBk2EQYgqgYGCEsZDQ0
NDA0MCwsKCQgIBgUEAgIBAgQCAQEBAwKRNRIHBqADChI1DQoFAGEBAGMEBAOMew8eTw4IVxkXCwkJ
BwYCOAIBAIIDAwUGBwgJCgICAQENAQEFawIDAgECAGMFAwMEBAUDBAMFAwIBAQECAwMEBAUGBgYHC
AgICQgHBwCGBgYFBQQEBAYDIgICAQECAICBAUGBwgJCQMBAGEMAQUDAwIDAQICBAQDBAQEBAQEAW
QEAgEBAQICBAMFBQUGBwCICAgJBwgHBgcGBgUFBAQEBOQIAgEBAf6RDAsLCQkICAYGBQUDAwIBAQI
DAwUFBgYICakJCwsMKSckIiAeGxoYFhQTERAPDQwLCgkIDxsJBQUBQqNEakKCwwNDxARExQWGBob
HiAiJCcCoAMDawQECA8XPRcKCgUPFz0REakIBAMDawMCAQICAwCYAwEaBwQBAgIAAAEAAAAAA/MDN
AA0AAABDwQvAw8EHwQ/ETUnIw8LAYsEJwwGAgIwXmMXFBIIcCsqKaEqRUclJSYnJykpKiosLC4GFg
sCAWMhISiIiIjIkJAPRwB8AQmCQMBARQuNgsMDgcIJCYnmyZOTycmJiYlJSQJiiIgHwULCAMCAQ4
RERITFBUVKy0tFgAAABIA3gABAAAAAAAAAAAAAABAAAAAABAAcAAQABAAAAAAACAAcACAABAAAA
AAADAACADwABAAAAAAAEAAcAFgABAAAAAAFAAsAHQABAAAAAAAGAAcAKAABAAAAAAAKACwALwABA
AAAAAALABIAWwADAAEECQAAAAIAbQADAAEECQABAA4AbwADAAEECQACAA4AfQADAAEECQADAA4AiW
ADAAEECQAEAA4AmQADAAEECQAFABYApwADAAEECQAGAA4AvQADAAEECQAKAFgAywADAAEECQALACQ
BIyBEZwZhdWx0UmVndWxhcRlZmF1bHREZwZhdWx0VmVyc2lvbiAxLjBEZwZhdWx0Rm9udCBnZW5l
cmF0ZWQgdXNpbmcgU3luY2Z1c2lvbiBNZXRybyBTdHVkaW93d3cuc3luY2Z1c2lvbi5jb20AIABEA
GUAZgBhAHUAbAB0AFIAZQBnAHUAbABhAHIArABlAGYAYQB1AGwAdABEAGUAZgBhAHUAbAB0AFYAZQ
ByAHMAAQBVAG4AIAAxAAC4AMABEAGUAZgBhAHUAbAB0AEYAbwBuAHQAIAbnAGUAbgBlAHIAAYQB0AGU
AZAAgAHUAcwBpAG4AZwAgAFMAeQBwAGMAZgB1AHMAAQBVAG4AIAbnAGUAdABYAG8AIAbTAHQAdQBk
AGkAbwB3AHcAdwAuAHMAeQBwAGMAZgB1AHMAAQBVAG4ALgBjAG8AbQAAAAACAAAAAaAAAAA
AAAAAAAAAAAAAAAAAAAAAYBAGEDAQBBQEGAQcADXRyYW5zcG9ydC12YW4ldXNlcil1tb2RpZn
kRc2hvcHBpbmctY2FydF8wMS0Lc3BlbmQtbW9uZXkFY2h1Y2sAAAA=) format('trueType');
    font-weight: normal;
    font-style: normal;
}
```

```
.stepper-validation [class^="sf-icon-"], .stepper-validation [class*=" sf-
icon-"] {
  font-family: 'Default' !important;
  speak: none;
  font-style: normal;
  font-weight: normal;
  font-variant: normal;
  text-transform: none;
  line-height: inherit;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}
.stepper-validation {
  margin-top: 30px;
  padding: 30px;
}
.stepper-validation .sf-icon-cart:before { content: "\e710"; }
.stepper-validation .sf-icon-transport:before { content: "\e702"; }
.stepper-validation .sf-icon-payment:before { content: "\e706"; }
.stepper-validation .sf-icon-success:before { content: "\e715"; }
```

### Template in React Stepper component

The Stepper component allows you to customize the default appearance and content of each step, creating a personalized experience for the user. You can use the [template](#) property to set the template content for the steps.

The step model and current step index are passed as `step` and `currentStep` properties in the template context for customization.

### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {

  const getTemplate = ({step}) => {
    return <div className="template-content"><span
className={step.iconCss}></span><br /><span className="e-
label">{step.label}</span></div>
  }

  return (
    <div className="stepper-template">
      <StepperComponent activeStep={1} template={getTemplate}>
        <StepsDirective>
          <StepDirective iconCss={'sf-icon-powerpoint'}
label={'Powerpoint'} />
          <StepDirective iconCss={'sf-icon-projector'}
label={'Presentation'} />
          <StepDirective iconCss={'sf-icon-onedrive'}
label={'Backup'} />
        </StepsDirective>
      </StepperComponent>
    </div>
  );
}
```

```

        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

## INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
"@syncfusion/ej2-react-navigations";
function App() {

    const getTemplate = ({step}) => {
        return <div className="template-content"><span
className={step.iconCss}></span><br /><span className="e-
label">{step.label}</span></div>
    }

    return (
        <div className="stepper-template">
            <StepperComponent activeStep={1} template={getTemplate}>
                <StepsDirective>
                    <StepDirective iconCss={'sf-icon-powerpoint'}
label={'Powerpoint'} />
                    <StepDirective iconCss={'sf-icon-projector'}
label={'Presentation'} />
                    <StepDirective iconCss={'sf-icon-onedrive'}
label={'Backup'} />
                </StepsDirective>
            </StepperComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

## INDEX.CSS

```

/* Represents the styles for loader */
#loader {
    color: #008cff;
    height: 40px;
    left: 45%;
    position: absolute;
    top: 45%;
    width: 30%;
}
@font-face {
    font-family: 'template';
    src:

```



```
url(data:application/x-font-ttf;charset=utf-
8;base64,AAEAAAAKAIAAAwAgTlMvMjltSfUAAAEoAAAAVmNtYXNlE+dkAAABlAAAAADxnbHlm39zz
MQAAAdwAAAacaGVhZCaImHMAAADQAAAAANmhoZWEIUQQGAAAArAAAACRobXR4FAAAAAAAAAAYAAAAU
G9jYQR8BVAAAAHQAAADG1heHABFwEbAAABCAAAACBuYw1ldkXdggaACHgAAAKRcG9zdD2fuhIAAA
sMAAAAXwABAAAEAAAAAFwEAAAAAAD9AABAAAAAAAAAAAAAAAAAAAAABQABAAAAQAAPdFGf18PPPU
ACwQAAAAAAOG7KcEAAAAA4bspwQAAAAAD9AP0AAAAACAACAAAAAAAAAAAAEAAAFQ8ACAAAAAAAgAA
AAoACgAAAP8AAAAAAAAAAQAAZAABQAAAokCzAAAAI8CiQLMAAAB6wAyAQgAAAIABQMAAAAAAAAA
AAAAAAAAAAAAAAAAAUGZFZABA5wDnAwQAAAAAXAQAAAAAAAABAAAAAABAAAAAQAAAAEAAAABA
AAAAQAAAAAAACAAAAAwAAABQAAwABAAAAFAAEACgAAAAEAAQAAQAA5wP//wAA5wD//wAAAAEABAA
AAAEAAgADAAQAAAAAAVQCAGMoA04ACAAAAAAD0wP0AB8APABcAHwA+gD+AQoBDgAAAQ8HLwY9AT8G
HwYlDwcVbj0BPwI7AR8FNw8HLwY9AT8GHwYlDwYrAS8GPwcZhwUHFR8HMzcXDwIfBz8HJzcfAz8CF
w8BHwc/By8HDwMnPWl1LwcPBxUfAQcvBCMPaIc/Ay8HDwYlESERAYEHFzcZfZcnIREhJyElIQMiAQ
ECaWQEBQUFBAUDAwICAgIDAwUEBQUFBAQDAgH+mwEBAGMEBAUFBQQFAwMCAGILBgUFBQQEAWIBnAE
BAGMEBAUFBQQEBAMCAGICAwQEBAUFBQQEAWIB/rkBAQMDAwUEBQUFBAQDAgEBAQECAwQEBQUFBAUD
AwMBUAEDBgYJCQsLCAXoAgQBAQQFBwgJCwsLCwkJBgYDAQFECAgICAKIBIEBAQEDBQCICGoLDAoJC
QcFAwEBAwUHCQkKDAoJCQl4BQMCAQMFBwgKCGwLCGoIBwUDAQICPwEJCQoMBwCGCGQFAwIBAQQFBw
gJCwsLCwkJBgUEAsb89j8BZ8sr+gX7K8sBZ/x4DwOm/FoB7QUFBAQDAgEBAQECAwQEBQUFBAUDAwM
BAQEBAwMDBQQzBQUEBAMCAQEBAQIDBAQFBQUGCwICAgMEBARgBQUEAwMDAQEBAQMDAwQFBQUFBAMD
AwEBAQEDAwMEBUsFBAQEAWICAgIDBAQEBOUFBAQDAgEBAgIDBAQFBQUGCgoIBwUDAQnNBAsLCwsJC
QYFBAEBawYGCQkLCwkuBAMCAQECAVkfCwsLCQgHBQQAQFBwgJCwsLCwkJBgYDAQECBQZTCACJCA
sLCQgHBQQAQFBwgJCwsJCAUqAgcFAwEBAGrkCQgICAwKCggHBQMBAMFBwgKCMp97gIS/bDALe3
tLcACji8+AAAGAAAAAP0A+QAEwA5AEsAdwCFAiKAAAEzPwc1LwcjFxUPDisBFSMRNx8OBsM1Mx8N
Jz8CFTMVDw4vAxUzFSMVMxUjFSE/AhEvAiElIREzPwMRLwMhJREFEQEQIGcHBgUFBAECAQIEBQUGB
AYmiaECAgQEBQYGCACJCAkKCQokNlSMCwoKCQgHBwYFBAMDAGEBgn0QDAwLCwoJCQgHBgUEAwK7BA
4NnAIEBAYHCAkJCgsMDAwNDQcWDGza2traAU0FAwICAwX+swF3FQQDAgEBAgMC/nL9rgIyAgABAQ
FBgYDBYQHBgYFBQQAIEfCQoJCQkIBwcGBgUEAwICCAFIQEBAGMDBQUGBgICAKKCgV9AgMEBgYH
CAkJCwsLDA1QAGUBhA0NDAsKCgoIBwcGBQQCAGEBAGMELSA+Hz8CBAQB/wUDAId+SgECAwIBwWQDA
gFb/PdfA8gAAAAACAAAAAPzAyKAcQEIAAABDYAVHw41Pw01Lw01LxErAQ8CKwEvCQcnDwgvBSsCDx
QVDw4VHw8zLwQ1Px8fCj8ELxMPAgICCwLcGoKCQkLCgGIBgUEAwECEg8QDg0LCgkIBgUEAgEBAQI
EBQYICQoLDQ4PDg4PAh0ODgYHBgcHBQgFwMDBAYDBgcICQoMDBMCAGQCBAGBwYGBwCHCAgJCRIU
EwoSEhQCARYICw0NDQ4PDw8XFwoVExIQDw4NCQwLCwsLCwsLCwoLCwsLCwsLCwsLCwsLCwsLCwsLC
RAPDgwLCgkHBwYEBAMBAQECEBAUGBwgJCgoKCgoLCwtgCwYFAwICAwQGBwkKcW0NDg8QEgIBBAMHBw
gKCwwNDxITExMUFBUDw8ODQwMCwsJCAgJERSFBAQFBgYHCAgJCgoLDAwNFBuUFhELCwKEAgMEBQU
HBwgNDQ0PDw8QEgIBAgMFBgYICAKKCwOCgsKCgsKFA8NDAsKCQcGBAMBAQEBBAMDBAYGBwCICAKI
GBIdCA0LCgGHBgUGAhAREQkICAKICACHBgYFBAQDBQMDBQgbBwoJCAYFAwIBAZ0DBwkLDQ4QEg8GB
AQDAgICAgQEBQYGCACICQkJCgkLCgsLDAsZAwQGBgCICAKKCwsLDAwMDA0NDQsMCgoKCQcGBQQDAW
EUEBAQEBEQE8PDQwLCgkJBwYFBQECDgwREA8ODgwLCwoIBgUCAQEDBAUGBwgJCwsMBAMDawEZDQ0
NDAsLCwoJCQgICAYGCAYEAgEBAGADAAAAAPfA/QACwAPABMAAAEHFzcVMzUXNyc1IyUhESEnITUh
Adq+NY1LijW/S/6JAzX8y0MDwPxAAQC/NYp+foolvlErAc5DZwAAAAASAN4AAQAAAAAAAABAAAA
QAAAAAAQAQAAEAQAQAAAAAAgAHABEAQAQAAAAAAAwAQABgAAQAAAAAABAAQACgAAQAAAAAABQALAD
gAAQAAAAAABgQAEMAAQAAAAAACgAsAFMAAQAAAAAACwASAH8AAwABBAKAAAAACAJEAAwABBAKAAQA
gAJMAAwABBAKAAgAOALMAAwABBAKAAwAgAMEAAwABBAKABAAgAOEAAwABBAKABQAWAQEAwABBAKa
BgAgARcAAwABBAKACgBYATcAAwABBAKACwAkAY8gdGVtcGxhdGVfdXBkYXRlZlZlZlZlZlZlZlZlZl
GF0ZV9lZGRhdGVkdGVtcGxhdGVfdXBkYXRlZlZlZlZlZlZlZlZlZlZlZlZlZlZlZlZlZlZlZlZlZl
QgZ2VuZXJhdGVkIHVzaW5nIFN5bmNmdXNpb24gTWV0cm8gU3R1ZGlvd3d3LnN5bmNmdXNpb24uY29
tACAAAdABLAG0AcABsAGEAdABlAF8AdQBwAGQAYQB0AGUAZABSAGUAZwB1AGwAYQByAHQAZQBtAHAA
bABhAHQAZQBfAHUAcABkAGEAdABlAGQAdABlAG0AcABsAGEAdABlAF8AdQBwAGQAYQB0AGUAZABSAG
B0ACAgBzAGkAbwBuACAAMQAuADAAdABlAG0AcABsAGEAdABlAF8AdQBwAGQAYQB0AGUAZABSAGB0
B0ACAAZwB1AG4AZQYAGEAdABlAGQAIAB1AHMAAgBuAGcAIABTAHkAbgBjAGYAdQBzAGkAbwBuACA
ATQB1AHQAcgBvACAAUwB0AHUAZABpAG8AdwB3AHcALgBzAHkAbgBjAGYAdQBzAGkAbwBuAC4AYwBv
AG0AAAAAAgAAAAAAAkAAAAAAAkAAAAAAAkAAAAAAAkAAAAAAAkAAAAAAAkAAAAAAAkAAAAAAAk
G9yLW9sZAPwb3dlcnBvaW50CG9uZWRYaXZlDXByb2ply3RvciluZXcAAAA=)
format('trueType');
font-weight: normal;
font-style: normal;
}
```

```

.stepper-template [class^="sf-icon-"], .stepper-template [class*=" sf-icon-"]
{
  font-family: 'template' !important;
  speak: none;
  font-size: 40px;
  font-style: normal;
  font-weight: normal;
  font-variant: normal;
  text-transform: none;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}
.stepper-template .sf-icon-projector:before { content: "\e700"; }
.stepper-template .sf-icon-powerpoint:before { content: "\e701"; }
.stepper-template .sf-icon-onedrive:before { content: "\e702"; }
/* Custom styles */
.stepper-template {
  width: 75%;
  margin: 0px auto;
  min-width: 85px;
  padding: 25px 0;
}
.stepper-template .template-content {
  background: #fff;
  width: 65px;
}
/* Stepper progressbar customization */
.stepper-template .e-stepper .e-stepper-progressbar {
  height: 3px;
  top: 25px;
}
.stepper-template .e-stepper .e-stepper-progressbar .e-progressbar-value {
  background-color: #27d96d;
}
/* Stepper status customization */
.stepper-template .e-step-completed * {
  color: #19cd60;
}
.stepper-template .e-step-inprogress * {
  color: #3479f3;
}
.stepper-template .e-step-notstarted * {
  color: #bdbdbd;
}

```

## Tooltip in React Stepper component

The Stepper component supports tooltip to show additional information in the steps by setting the [showTooltip](#) property to `true`.

The tooltip appears when the user hovers over the step, providing the information such as the label or text. By default, the `showTooltip` property is `false`.

### INDEX.JSX

```

{% raw %}
import * as React from "react";

```

```

import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
 '@syncfusion/ej2-react-navigations';
function App() {
    return (
        <div className="stepper-tooltip">
            <StepperComponent showTooltip={true}>
                <StepsDirective>
                    <StepDirective iconCss= {'sf-icon-cart'} label=
{'Cart'}/>
                    <StepDirective iconCss= {'sf-icon-transport'} label=
{'Delivery Address'}/>
                    <StepDirective iconCss= {'sf-icon-payment'} label=
{'Payment'}/>
                    <StepDirective iconCss= {'sf-icon-success'} label=
{'Confirmation'}/>
                </StepsDirective>
            </StepperComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

#### INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
 '@syncfusion/ej2-react-navigations';
function App() {
    return (
        <div className="stepper-tooltip">
            <StepperComponent showTooltip={true}>
                <StepsDirective>
                    <StepDirective iconCss= {'sf-icon-cart'} label=
{'Cart'}/>
                    <StepDirective iconCss= {'sf-icon-transport'} label=
{'Delivery Address'}/>
                    <StepDirective iconCss= {'sf-icon-payment'} label=
{'Payment'}/>
                    <StepDirective iconCss= {'sf-icon-success'} label=
{'Confirmation'}/>
                </StepsDirective>
            </StepperComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

#### INDEX.CSS

```

/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
@font-face {
  font-family: 'Default';
  src:
    url(data:application/x-font-ttf;charset=utf-
8;base64,AAEAAAIAIAAAwAgTlMvMjlvSgcAAAEoAAAAVmNtYXCDelPaAAABmAAAAF5nbHlmEwr+
pwAAAggAAAJQaGVhZCYp2+EAADQAAAAANmhoZWEIUQQHAAAArAAAACRobXR4GAAAAAAAYAAAAAYb
G9jYQhUBlAAAAH4AAAAADm1heHABFgErAAABCAAAACBuYW11luF5THQAACTgAAAIlcG9zdJ8LuoMAAA
0AAAAAbwABAAAEAAAAAFwEAAAAAAD9AABAAAAAABAAAAAABgABAAAAAQAArxT6wV8PPPU
ACwQAAAAAAGLy6UAAAAA4YvLpQAAAAAD9AOaAAAAACAACAAAAAABAAAAAAGAR8ABgAAAAAAAgAA
AAoACgAAAP8AAAAAQAQAAZAABQAAAokCzAAAAI8CiQLMAAAB6wAyAQgAAAIABQMAAAAAAABAAAA
AAAAAAAAAAAAAAAAAUGZFZABA5wLnFQQAAAAAXAQAAAAAABAAAAAABAAAAAQAAAAEAAAABA
AAAAQAAAAEAAAAAAGgAAAAAMAAAUAMAAQAABQABABKAAAADAAIAAIABOCc5wbnCOCQ5xX//wA
A5wLnBucI5xDnFf//AAAAAAAAAAAAAAAAABAAwADAAMAawADAAAAAEABAACAAMABQAAAAAABEQaiwC
3AQkBGgAAAAFAAAAAAP0A18APwB/AIkAxgDrAAABHw8/Dy8OKwEPDQUfDz8PLw4rAQ8NAR8FFSM1J
xEfBz80OwEfDjM/BzUnIw8GATM/Dx8PMxEhAq8BAQIEBAUFbWYICAgJCQoKCgkKCAkIBwCHBQUEAw
MBAQEBAwMEBQUHBwCICQgKCQoKCgkJCAGIBGcFBQQUEAgH+CwEBAGQEBQUHBggICakJCgoKCQoICQg
HBwCFBQQDAwEBAQEDAwQFBQcHBWgJCAoJCgoKCQkICAgGBwUFBAQCAQJ8AwUIAwD3n0BAwMGBgYI
CAMEBQYHBWkJCgsLDA0NDQ4ODQ4MDAwLCgkJCAYGBQMDKAgIBwYFBAECvLsICAYHBQMD/beAAwQFB
QcHCAkKCgsLDA0MDg0NDQwLCwsJCQkHBwUFAwKE/eMBAQoJCQkJCACHBgYFBAMDAQEBAQMDBAUGBg
cHCAkJCQkKCgoJCQgICAcGBgQFAwICAgIDBAUFbGcHCAkJCQoLCgkJCQkIBwCGBgUEAwMBAQEBAwM
EBQYGBwCICQkJCQoKCgkJCAGIBwYGBAUDAgICAgMEBQUGBwCICQkJCgGuAQIGehYJBKYp/10ICAcG
BQQCAQ0NDQwLCgoJCAGGBQUDAgIDBQUGCAGJCgoLDA0NDQECBAUGBwQI1foBAGQFBGcH/iwNDAwLC
woJCQgHBgUEAwEBAQEDBAUGBwgJCQoLCwwMDQJJAAAAABgAAAAAD8wOWAAYAQQgBaAGwArQDuAAABBz
cfAwUhLwIHIY8PNS8CKwIPHQEHLwEjDwE1LwMjNz0BJZcfBTcfAg8BLwY30wEfAQcVHw8/Dy8PDw4
BFR8PPw8vDw8OAxEWBgEDAgb8/wNuBAUEDQsVFBQTEhEPDw0GCwoIBgQCFhITE+wQDw8PDg4ODg0N
DQwNCwwKCwoKCQgJBwCHBgyEBQMEa5FrBAQDBAMBAwMDBgIDagIEBgYGBxwCAwIBFQYGBAgFBgIWA
gQHCPcBAGQGBgKCGsMDQ4PDxAQEBApDw4NDAsLCQgGBgQCAQECAyGCAkLCwwNDg8PEBAQEa8PDg
0MCwoKCAyGBAL+KgEEBQgKCw0PEBETFBQWFxcXFhYUFBMREQ4NDakIBgMBAQMGCakMDQ4RERMUFBY
WFxcXFhQUEExEQDw0LCgGFBAEXBhcFBAMDrxYWDQEBawUHCAsmDQ4IERESFBQUFQDDAgECAGMEBAUG
BgYIBWgJCQoKCwsLDAwMDQ0ODQ4PDgEZawIBAQIGBQMCAQQDBgZqBgOHBQMDMAMHBwMWAQICBQYKC
hYCB1wICBAPDw4NDAsLCQgGBgQCAQECAyGCAkLCwwNDg8PEBAQEa8PDg0MCwoKCAyGawMBAQMDBg
YICgoLDA0ODw8QATMLDBYVFRQSERAPDQsKCAUEAQEEBQgKCw0PEBESFBUVFhcXFxYVFBISEA8NCwo
IBQQBAQQFCAoLDQ8QEhIUFRYXAAAAAQAAAAA/QDRwA/AH8AhwcRAAABFR8OPw49AS8NKwEPDQUV
Hw4/Dj0BLw0rAQ8NEwcTIRMnMSMhMxMhNSEDBzUha0YBAGMEBAQGBQcGBWgICAgICAgIBwYHBQYEB
AQDAGEBAGMEBAQGBQcGBWgICAgICAgIBwYHBQYEBAQDAGH+aAICAgQEBAYFBwYIBWgICAgICAgHBg
cFBgQEBAMCAQECAwQEBAYFBwYHCAGICAgICAcIBGcFBgQEBAICAsH6jAFKjPpu/Z3NwgJZ/dzDAf8
AAQkICAgHBwCGBgUFBAQCAgEBAQECAgQEBQUGBgHBWgICakIBWgHBwYGBQUEAwMCAQECAwMEBQUG
BgCHCAcICQgICAcHBwYGBQUEBAICAEBAQICBAQFBQYGBwCHCAgICQgHCAcHBgYFBQQDAwIBAQIDA
wQFBQYGBwCIBWgB+wH+vQFABP5dOgGkAQEAADAAAAAANKA5oAnQDxAR4AAAEzHwEdAR8HFQ8DIY
8HDWYdAR8WDw0dAQ8BKwIvAT0BLwc9AT8COWEfBj8HLxc/DTU/AwEfDjsBPxEvFiMPFR8BEw8CFR8
HMz8HNS8GIw8ELwQrAQ8BAGoCAGENDAwKCGgHBQEBAikCAGIEAwQFDA0SBwcGagIBAQICBgCHBxYK
CQkJCAcHBgUFBAQCAQEBAQIDAQFBQYGBwCPEQECAhUCAQINDAsLCQgHBQICKQICAgQDBAULdhIHB
wYCAQEBAQEBAgYHBwCWCgkKCAgHBwYFBQQDAgEBAQECAwMEBAYFBGcHEBABAQED/qwUFRUVFRYWFh
YWFxYXFhcXFxcWFxYXFhYWFhYVFRUVFAQCAQICBAUGCAGJCgsLDA0MDQ0NDBk2EQYgqgYGCEsZDQ0
NDA0MCwsKCQgIBgUEAgIBaQCAQEBAwKRNRIHBqADChI1DQoFagEBAgMEBAOMew8eTw4IVxkXCwkJ
BwYCOAIBAIIDAuUGBwgJCgICAQENAQEFawIDAQECAGMFAwMEBAUDBAMFAwIBAQECAwMEBAUGBgYHC
AgICQgHBwCGBgYFBQQUEBAYDIgICAQECAiICBAUGBwgJCQMBAGEMAQUDAwIDAQICBAQDBAQEBAQEAW
QEAgEBAQICBAMFBQUGBwCICAgJBWgHBGcGBgUFBAQEBOQIAgEBAf6RDAsLCQkICAYGBQUDAwIBAQI

```

```

DAwUFBgYICakJCwsMKSckIiAeGxoYFhQTERAPDQwLCgkIDxsJBQUFBQqNEAkKCwwNDxARExQWGBob
HiAiJccCoAMDawQECA8XPRcKCgUPFz0REakIBAMDAwMCAQICawcYAwEaBwQBAGIAAAEAAAAAA/MDN
AA0AAABDwQvAw8EHwQ/ETUnIw8LAYsEJwwGAgIwXmMXFBIICCsQKaEqRUclJSYnJykpKiosLC4GFg
sCAWMhISiIiIjIkJAPRwB8AQmCQMBARQuNgsMDgcIJCYnmyZOTycmJiYlJSQjIiIgHwULCAMCAQ4
RERITFBUVKy0tFgAAABIA3gABAAAAAAAAAAAAEAAAABAAAAAABAACAAQABAAAAAAACAACACAABAAAA
AAADAACADwABAAAAAAAEAAcAFgABAAAAAAAFAAAsAHQABAAAAAAAGAAcAKAABAAAAAAAKACwALwABA
AAAAAALABIAWwADAAEECQAAAAIAbQADAAEECQABAA4AbwADAAEECQACAA4AfQADAAEECQADAA4AiW
ADAAEECQAEAA4AmQADAAEECQAFABYApwADAAEECQAGAA4AvQADAAEECQAKAFgAywADAAEECQALACQ
BIyBEZWZhdWx0UmVndWxhckRlZmFlbHREZWZhdWx0VmVyc2lvbiAxLjBEZWZhdWx0Rm9udCBnZW5l
cmF0ZWQgdXNpbmcgU3luY2Zlc2lvbiBNZXRYbyBTdHVkaW93d3cuc3luY2Zlc2lvbi5jb20AIABEA
GUAZgBhAHUAbAB0AFIAZQBnAHUAbABhAHIArABLAGYAYQB1AGwAdABEAGUAZgBhAHUAbAB0AFYAZQ
BYAHMAaQBvAG4AIAAxAAC4AMABEAGUAZgBhAHUAbAB0AEYAbwBuAHQAIABnAGUAbgB1AHIAZQB0AGU
AZAAgAHUAcwBpAG4AZwAgAFMAeQBvAGMAZgB1AHMAaQBvAG4AIAABNAGUAdABYAG8AIAABTAHQAdQBk
AGkAbwB3AHcAdwAuAHMAeQBvAGMAZgB1AHMAaQBvAG4ALgBjAG8AbQAAAAACAAAAAAAOAAAAAA
AAAAAAAAAAAAAAAAAAAAAYBAGEDAQBBQEGAQCADXRYW5zcG9ydC12YW4LdXNlciltb2RpZn
kRc2hvcHBpbmctY2FydF8wMS0Lc3BlbmQtbW9uZXkFY2h1Y2sAAAA=) format('trueType');
font-weight: normal;
font-style: normal;
}
.stepper-tooltip [class^="sf-icon-"], .stepper-tooltip [class*=" sf-icon-"] {
font-family: 'Default' !important;
speak: none;
font-style: normal;
font-weight: normal;
font-variant: normal;
text-transform: none;
line-height: inherit;
-webkit-font-smoothing: antialiased;
-moz-osx-font-smoothing: grayscale;
}
.stepper-tooltip {
margin-top: 30px;
padding: 30px;
}
.stepper-tooltip .sf-icon-cart:before { content: "\e710"; }
.stepper-tooltip .sf-icon-transport:before { content: "\e702"; }
.stepper-tooltip .sf-icon-payment:before { content: "\e706"; }
.stepper-tooltip .sf-icon-success:before { content: "\e715"; }

```

### Tooltip template

You can use the [tooltipTemplate](#) property to specify a custom template for the tooltips, providing detailed information about the steps.

When hovering over the step, the current step model is passed in the template context, allowing you to include dynamic information about the step.

### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
function getTooltipTemplate({value}) {

```

```

        return <span className="template-content"><span className={`stepper-
icon ${value.iconCss}`}></span><span className="step-
label">{value.text}</span></span>
    }
    return (
        <div className="stepper-tooltip-template">
            <StepperComponent showTooltip={true}
tooltipTemplate={getTooltipTemplate}>
                <StepsDirective>
                    <StepDirective text= {'Review your cart items'} iconCss=
{'sf-icon-cart'} label= {'Cart'}/>
                    <StepDirective text= {'Enter your delivery address'}
iconCss= {'sf-icon-transport'} label= {'Delivery Address'}/>
                    <StepDirective text= {'Complete your purchase securely'}
iconCss= {'sf-icon-payment'} label= {'Payment'}/>
                    <StepDirective text= {'Verify your order details'}
iconCss= {'sf-icon-success'} label= {'Confirmation'}/>
                </StepsDirective>
            </StepperComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

## INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
    function getTooltipTemplate({value}) {
        return <span className="template-content"><span className={`stepper-
icon ${value.iconCss}`}></span><span className="step-
label">{value.text}</span></span>
    }
    return (
        <div className="stepper-tooltip-template">
            <StepperComponent showTooltip={true}
tooltipTemplate={getTooltipTemplate}>
                <StepsDirective>
                    <StepDirective text= {'Review your cart items'} iconCss=
{'sf-icon-cart'} label= {'Cart'}/>
                    <StepDirective text= {'Enter your delivery address'}
iconCss= {'sf-icon-transport'} label= {'Delivery Address'}/>
                    <StepDirective text= {'Complete your purchase securely'}
iconCss= {'sf-icon-payment'} label= {'Payment'}/>
                    <StepDirective text= {'Verify your order details'}
iconCss= {'sf-icon-success'} label= {'Confirmation'}/>
                </StepsDirective>
            </StepperComponent>
        </div>
    );
}

```

```

}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

## INDEX.CSS

```

/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
@font-face {
  font-family: 'Default';
  src:
    url(data:application/x-font-ttf;charset=utf-
8;base64,AAEAAAKAIAAAwAgTlMvMjlvSgcAAAEoAAAAVmNtYXCDelPaAAABmAAAAF5nbHlmEwr+
pwAAAggAAAJQaGVhZCYp2+EAADQAAAAANmhoZWEIUQQHAAAArAAAACRobXR4GAAAAAAAYAAAAAYb
G9jYQhUBlAAAAH4AAAAADmlheHABFgErAAABCAAAACBuYW1luF5THQAActgAAAIlcG9zdJ8LuoMAAA
0AAAAAbwABAAAEAAAAAFwEAAAAAAD9AABAAAAAABAAAAAABGABAAAAAQAArxT6wV8PPPU
ACwQAAAAAAOGLy6UAAAAA4YvLpQAAAAAD9AOaAAAAACAACAAAAAABAAAAAGAR8ABgAAAAAAAgAA
AAoACgAAAP8AAAAAQAQAAZAABQAAAokCzAAAAI8CiQLMAAAB6wAyAQgAAAIABQMAAAAAAABAAA
AAAAAAAAAAAAAAAAAUGZFZABA5wLnFQQAAAAAXAQAAAAAABAAAAAABAAAAAQAAAAEAAAABA
AAAAQAAAAEAAAAAAGAAAAAAMAAUAAMAAQAAABQABABKAAAADAAIAAIABOC5wbnCOC5xX//wA
A5wLnBucI5xDnFf//AAAAAAAAAAAAAAAAAABAAwADAAMAawADAAAAAEABAACAAMABQAAAAAAAEQAIwC
3AQkBGgAAAAFAAAAAAP0A18APwB/AIkAxgDrAAABHw8/Dy8OKwEPDQUfDz8PLw4rAQ8NAR8FFSM1J
xEfBz8OOwEfDjM/BzUnIw8GATM/Dx8PMxEhAq8BAQIEBAUFBwYICAgJCQoKCgkKCAkIBwcHBQUEAw
MBAQEBAwMEBQUHBwcICQgKCQoKCgkJCAgIBgcFBQQEAgH+CwEBAgQEBQUHBggICakJCgoKCQoICQg
HBwcFBQQDAwEBAQEDAwQFBQcHBwgJCAoJCgoKCQkICAgGBwUFBAQCAQJ8AwUIWAdD3n0BAwMGBgYI
CAMEBQYHBwkJCgsLDA0NDQ4ODQ4MDAwLCgkJCAYGBQMDKAgIBwYFBAECvLsICAYHBQMD/beAAwQFB
QcHCAkKCgsLDA0MDg0NDQwLCwsJCQkHBwUFAwKE/eMBAQoJCQkJCACHBgYFBAMDAQEBAQMDBAUGBg
cHCAkJCQkKCgoJCQgICAcGBgQFAwICAgIDBAUFBgHCAkJCQoLCgkJCQkIBwcGBgUEAwMBAQEBAwM
EBQYGBwcICQkJCQoKCgkJCAgIBwYGBAUDAgICAgMEBQUGBwcICQkJCgGuAQIGehYJBKYp/10ICAcG
BQQCAQ0NDQwLCgoJCAgGBQUDAgIDBQUCAgJCgoLDA0NDQECBAUGBwQI1foBAGQFBGcH/iwNDAwLC
woJCQgHBgUEAwEBAQEDBAUGBwgJCQoLCwwMDQJJAAAAABgAAAAAD8wOWAAYAqgBaAGwArQDuAAABBz
cfAwUhLwIHlY8PNS8CKwIPHQEHLwEjDwE1LwMjNz0BjZcfBTcfAg8BLwY3OwEFAQcVHw8/Dy8PDw4
BFR8PPw8vDw8OAxEWBgEDAgb8/wNuBAUEDQsVFBQTEhEPDw0GCwoIBgQCfHITE+wQDw8PDg4ODg0N
DQwNCwwKCwoKCQgJBwcHBgYEBQMEA5FrBAQDBAMBAwMDBgIDagIEBgYGBxwCAwIBFQYGBAgFBgIWA
gQHCPcBAGQGBgkCgsMDQ4PDxAQEBApDw4NDAsLCQgGBgQCAQECBAYGCAkLCwwNDg8PEBAQEA8PDg
0MCwoKCAyGBAL+KgEEBQgKCw0PEBETFBQWFxcXfHYUFBMREQ4NDakIBgMBAQMGCakMDQ4RERMUFBY
WFxcXfHQUExEQDw0LCggFBAEXBhcFBAMDrxYWDQEBAwUHCAsmDQ4IERESFBQUFQQDAgECAGMEBAUG
BgYIBwgJCQoKCwsLDAwMDQ0ODQ4PDgEZawIBAQIGBQMCAQQDBgZqBgOHBQMDMAMHBwMWAQICBQYKC
hYCB1wICBAPDw4NDAsLCQgGBgQCAQECBAYGCAkLCwwNDg8PEBAQEA8PDg0MCwoKCAyGAwMBAQMDBg
YICgoLDA0ODw8QATMLDBYVFRQSERAPDQsKCAUEAQEEBQgKCw0PEBESFBUVFhcXfYVFBISEA8NCwo
IBQQBAQQFCAoLDQ8QEhIUFRYXAAAAAQAAAAA/QDRwA/AH8AhwCRAAABFR8OPw49AS8NKwEPDQUV
Hw4/Dj0BLw0rAQ8NEwcTIRMnMSMhMxMhNSEDBzUha0YBAgMEBAQGBQcGBwgICAgICAgIBwYHBQYEB
AQDAgEBAgMEBAQGBQcGBwgICAgICAgIBwYHBQYEBAQDAgH+aAICAgQEBAYFBwYIBwgICAgICAgHBg
cFBgQEBAQCAQECawQEBAYFBwYHCAgICAgICAcIBgcFBgQEBAICAsH6jAFKjPpu/Z3NwgJZ/dzDAf8
AAQkICAgHBwcGBgUFBAQCAgEBAQECAGQEBQUGBgHbWgICakIBwgHBwYGBQUEAwMCAQECawMEBQUG
BgHCAcICQgICAcHBwYGBQUEBAICAQEBAQICBAQFBQYGBwcHCAgICQgHCAcHBgYFBQQDAwIBAQIDA
wQFBQYGBwcIBwgB+wH+vQFABP5dOgGkAQEAAAADAAAAANkA5oAnQDxAR4AAAEzHwEdAR8HFQ8DIy
8HDwYdAR8WDw0dAQ8BKwIvAT0BLwc9AT8COWEfBj8HLxc/DTU/AwEfDjsBPxEvFiMPFR8BEW8CFR8

```

```

HMz8HNS8GIw8ELwQrAQ8BAgoCAgENDAwKCggHBQEBaIkCagIEAwQFDA0SBwcGAgIBAQICBgCHBxYK
CQkJCAcHBgUFBAMCAQEBAQIDAwQFBQYGBwcPEQECaHUCAQINDAsLCQgHBQICKQICAgQDBAULdHIB
wYCAQEBAQEBAgYHBwcWCgkKCAgHBwYFBQQDAgEBAQECawMEBAYFBgcHEBABAQED/qwUFRUVFRYWFh
YWFxYXFhcXfXcWFxYXFhYWFhYVFRUVFAQCAQICBAUGCAgJCgsLDA0MDQ0NDBk2EQYGqgYGCEsZDQ0
NDA0MCwsKCQgIBgUEAgIBaQCAQEBAwKRNRIHBqADChI1DQoFAgEBAgMEBAoMEw8eTw4IVxkXCwkJ
BwYCOAIBAIIDAwUGBwgJCgICAQENAQEFawIDAgECAGMFAwMEBAUDBAMFAwIBAQECAwMEBAUGBgYHC
AgICQgHBwcGBgYFBQQEBAYDIgICAQECAiICBAUGBwgJCQMBAGEMAQUDAwIDAQICBAQDBAQEBQEAw
QEAgEBAQICBAMFBQUGBwcICAgJBwgHBgcGBgUFBAQEBQQiAgEBAf6RDAsLCQkICAYGBQUDAwIBAQI
DAwUFBgYICakJCwsMKSckIiAeGxoYFhQTERAPDQwLCgkIDxsJBQUBQqNEakKCwwNDxARExQWGBob
HiAiJCcCoAMDawQECA8XPRcKCgUPFz0REakIBAMDawMCAQICAwcYAwEaBwQBAGIAAAEAAAAAAAA/MDN
AA0AAABDwQvAw8EHwQ/ETUnIw8LAYsEJwwGAgIwXmMXFBIICCsqKaEqRUclJSYnJykpKiosLC4GFg
sCAWMhISiIiIjIkJAPRwB8AQmQMBARQuNgsMDgcIJCYnmyZOTycmJiYlJSQjIiIgHwULCAMCAQ4
RERITFBUVKy0tFgAAABIA3gABAAAAAAAAAAAAEAAAABAAAAAAAAABAAcAAQABAAAAAAAAACAACAABAAA
AAADAACADwABAAAAAAAAEAACAFgABAAAAAAAAFAAsAHQABAAAAAAAAAGAAcAKAABAAAAAAAAAKACwALwABA
AAAAAAAALABIAWwADAAEECQAAAAIAbQADAAEECQABAA4AbwADAAEECQACAA4AfQADAAEECQADAA4Aiw
ADAAEECQAEAA4AmQADAAEECQAFABYApwADAAEECQAGAA4AvQADAAEECQAKAFgAywADAAEECQALACQ
BIyBEZWZhdWx0UmVndWxhcRlZmFlbHREZWZhdWx0VmVyc2lvbiAxLjBEZWZhdWx0Rm9udCBnZW5l
cmF0ZWQgdXNpbmcgU3luY2Zlc2lvbiBNZXRYbyBTdHVkaW93d3cuc3luY2Zlc2lvbi5jb20AIABEA
GUAZgBhAHUAbAB0AFIAZQBnAHUAbABhAHIArABLAGYAYQB1AGwAdABEAGUAZgBhAHUAbAB0AFYAZQ
ByAHMAaQBvAG4AIAAxAAC4AMABEAGUAZgBhAHUAbAB0AEYAbwBuAHQAIABnAGUAbgBlAHIAAYQB0AGU
AZAAgAHUAcwBpAG4AZwAgAFMAeQBvAGMAZgB1AHMAaQBvAG4AIAABNAGUAdABYAG8AIABTAHQAdQBk
AGkAbwB3AHcAdwAuAHMAeQBvAGMAZgB1AHMAaQBvAG4ALgBjAG8AbQAAAAACAAAAAAAAAAAAoAAAAA
AAAAAAAAAAAAAAAAAAAAAAYBAGEDAQQBQEGAQCADXRyYW5zcG9ydC12YW4LdXNlci1tb2RpZn
kRc2hvcHBpbmctY2FydF8wMMS0Lc3BlbmQtbW9uZXkFY2hly2sAAAA=) format('trueType');
font-weight: normal;
font-style: normal;
}
[class^="sf-icon-"], [class*=" sf-icon-"] {
font-family: 'Default' !important;
speak: none;
font-style: normal;
font-weight: normal;
font-variant: normal;
text-transform: none;
line-height: inherit;
-webkit-font-smoothing: antialiased;
-moz-osx-font-smoothing: grayscale;
}
.sf-icon-cart:before { content: "\e710"; }
.sf-icon-transport:before { content: "\e702"; }
.sf-icon-payment:before { content: "\e706"; }
.sf-icon-success:before { content: "\e715"; }
/* Custom styles */
.stepper-tooltip-template {
margin-top: 30px;
padding: 30px;
}
.e-stepper-tooltip .template-content {
display: flex;
align-items: center;
}
.e-stepper-tooltip {
border-radius: 5px;
}
.e-stepper-tooltip .e-tip-content {
padding: 7px;
}

```



```
.e-stepper-tooltip .step-label {
  margin-left: 7px;
  font-weight: bold;
}
```

### Animation in React Stepper component

The Stepper progress state can be animated, smoothly transitioning from one step to another. You can customize the animation's **duration** and **delay** by using the [animation](#) property.

You can disable the animation by setting the [enable](#) property to **false**. By default, the value is **true**.

| Fields | Type | Description |

|-----|-----|-----|

| [duration](#) | **number** | Specifies the duration of the animated transition for each step. The default value is **2000** milliseconds. |

| [delay](#) | **number** | Specifies the delay to initiate the animated transition for each step in milliseconds. The default value is **0**. |

The example demonstrates the animation **duration** and **delay** settings for the Stepper.

#### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
"@syncfusion/ej2-react-navigations";
function App() {
  return (
    <div className="stepper-animation">
      <StepperComponent animation={{ enable: true, duration: 2000,
delay: 500 }}>
        <StepsDirective>
          <StepDirective />
          <StepDirective />
          <StepDirective />
          <StepDirective />
        </StepsDirective>
      </StepperComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

#### INDEX.TSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
"@syncfusion/ej2-react-navigations";
```

```
function App() {
  return (
    <div className="stepper-animation">
      <StepperComponent animation={{ enable: true, duration: 2000,
delay: 500 }}>
        <StepsDirective>
          <StepDirective />
          <StepDirective />
          <StepDirective />
          <StepDirective />
        </StepsDirective>
      </StepperComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

## INDEX.CSS

```
/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
```

## Globalization in React Stepper component

### Localization

The Localization library allows you to localize the default text content of the Stepper. You can change the static text content used for the **optional** property to other cultures (Arabic, Deutsch, French, etc.) by defining the **locale** value and its translation object.

| Locale key | en-US (default) |

|-----|-----|

| optional | Optional |

In this example, the **French** culture is set to Stepper and the default text is updated with the content defined by the locale key.

## INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
import { L10n } from '@syncfusion/ej2-base';
function App() {
```

```

    L10n.load({
      'fr-BE': {
        'stepper': {
          'optional': "facultatif"
        }
      }
    });
    return (
      <div className="stepper-localization">
        <StepperComponent locale="fr-BE">
          <StepsDirective>
            <StepDirective iconCss={'sf-icon-cart'} label={'Cart'} />
            <StepDirective iconCss={'sf-icon-transport'}
label={'Delivery Address'} />
            <StepDirective iconCss={'sf-icon-payment'}
label={'Payment'} optional={true}/>
            <StepDirective iconCss={'sf-icon-success'}
label={'Confirmation'} />
          </StepsDirective>
        </StepperComponent>
      </div>
    );
  }
  export default App;
  ReactDOM.render(<App />, document.getElementById("element"));
  {% endraw %}

```

## INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
import { L10n } from '@syncfusion/ej2-base';
function App() {
  L10n.load({
    'fr-BE': {
      'stepper': {
        'optional': "facultatif"
      }
    }
  });
  return (
    <div className="stepper-localization">
      <StepperComponent locale="fr-BE">
        <StepsDirective>
          <StepDirective iconCss={'sf-icon-cart'} label={'Cart'} />
          <StepDirective iconCss={'sf-icon-transport'}
label={'Delivery Address'} />
          <StepDirective iconCss={'sf-icon-payment'}
label={'Payment'} optional={true}/>
          <StepDirective iconCss={'sf-icon-success'}
label={'Confirmation'} />
        </StepsDirective>
      </StepperComponent>
    </div>
  );
}

```

```

    </div>

    );
  }
  export default App;
  ReactDOM.render(<App />, document.getElementById("element"));
  {% endraw %}

```

## INDEX.CSS

```

/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
@font-face {
  font-family: 'Default';
  src:
    url(data:application/x-font-ttf;charset=utf-
8;base64,AAEAAAKAIAAAwAgTlMvMjlvSgcAAAEoAAAAVmNtYXCDeIPaAAABmAAAAF5nbHlmEwr+
pwAAAggAAAJQaGVhZCYp2+EAADQAAAAANmhoZWEIUQQHAAAArAAAACRobXR4GAAAAAAAYAAAAAYb
G9jYQhUblAAAAH4AAAAADm1heHABFgErAAABCAAAACBuYW1luF5THQAACTgAAAIlcG9zdJ8LuoMAAA
0AAAAAbwABAAAEAAAAAFwEAAAAAAD9AABAAAAAAAAAAAAAAAAAAAAABgABAAAAAQAArxT6wV8PPPU
ACwQAAAAAAGLy6UAAAAA4YvLpQAAAAAD9AOaAAAAACAACAAAAAAAAAAAAEAAAAGAR8ABgAAAAAAAgAA
AAoACgAAAP8AAAAAAAAAAQQAaZAABQAAAokCzAAAAI8CiQLMAAAB6wAyAQgAAAIABQMAAAAAAAAA
AAAAAAAAAAAAAAAAAUGZFZABA5wLnFQQAAAAAXAQAAAAAAAABAAAAAAAAABAAAAAQAAAAEAAAABA
AAAAQAAAAEAAAAAAAAAgAAAAAMAAUAAQAABQABABKAAAADAIAAIABOC5wbnCOcQ5xX//wA
A5wLnBucI5xDnFf//AAAAAAAAAAAAAAAAABAAwADAAMAawADAAAAAEABAACAAMABQAAAAAAAAAEQaiwC
3AQkBGgAAAAFAAAAAAP0A18APwB/AIkAxgDrAAABHw8/Dy8OKwEPDQUfDz8PLw4rAQ8NAR8FFSM1J
xEfBz80OwEfDjM/BzUnIw8GATM/Dx8PMxEhAq8BAQIEBAUFbWYICAgJCQoKCgkKCAkIBwCHBQUEAw
MBAQEBAwMEBQUHBwCICQgKCQoKCgkKJCAgIBgCfBQQEAgH+CwEBAGQEBQUHBggICakJCgoKCQoICQg
HBwCfBQQDAwEBAQEDAwQFBQcHBWgJCAoJCgoKCQkICAgGBwUFBAQCAQJ8AwUIAwD3n0BAwMGBgYI
CAMEBQYHBwkJCgsLDA0NDQ4ODQ4MDAwLCgkJCAYGBQMDKAgIBwYFBAECvLsICAYHBQMD/beAAwQFB
QcHCAkKCgsLDA0MDg0NDQwLCwsJCQkHBwUFAwKE/eMBAQoJCQkICAcHBgYFBAMDAQEBAQMDBAUGBg
cHCAkJCQkKCgoJCQgICAcGBgQFAwICAgIDBAUFbGcHCAkJCQoLCgkJCQkIBwCGBgUEAwMBAQEBAwM
EBQYGBwCICQkJCQoKCgkKJCAgIBwYGBAUDAgICAgMEBQUGBwCICQkJCgGuAQIGehYJBKYp/10ICAcG
BQQCAQ0NDQwLCgoJCgAgGBQUDAgIDBQUGCAGJCgoLDA0NDQECBAUGBwQI1foBAGQFBGcH/iwNDAwLC
woJCQgHBgUEAwEBAQEEDBAUGBwgJCQoLCwwMDQJJAABABgAAAAAD8wOWAAYAQQBaAGwArQDuAAABz
cfAwUhLwIHIy8PNS8CKwIPHQEHLwEjDwE1LwMjNz0BJzcFbTcfAg8BLwY30wEfAQcVHw8/Dy8PDw4
BFR8PPw8vDw8OAxEWBgEDAgb8/wNuBAUEDQsVFBQTEhEPDw0GCwoIBgQCFhITE+wQDw8PDg4ODg0N
DQwNCwwKCwoKCQgJBwCHBgYEBQMEA5FrBAQDBAMBAwMDBgIDagIEBgYGBxwCAwIBFQYGBAgFBgIWA
gQHCPcBAGQGBggKCgsMDQ4PDxAQEBApDw4NDAsLCQgGBgQCAQECBAYGCAkLCwwNDg8PEBAQEA8PDg
0MCwoKCAYGBAL+KgEEBQgKCw0PEBETFBQWFxcXFhYUFBMREQ4NDakIBgMBAQMGCakMDQ4RERMUFBY
WFxcXFhQUEExEQDw0LCgGFBAEXBhcFBAMDrxYWDQEBawUHCAsmDQ4IERESFBQUFQQDAgECAGMEBAUG
BgYIBwgJCQoKCwsLDAwMDQ0ODQ4PDgEZawIBAQIGBQMCAQQDBgZqBgOHBQMDMAMHBwMWAQICBQYKC
hYCB1wICBAPDw4NDAsLCQgGBgQCAQECBAYGCAkLCwwNDg8PEBAQEA8PDg0MCwoKCAYGAwMBAQMDBg
YICgoLDA0ODw8QATMLDBYVFRQSERAPDQsKCAUEAQEEBQgKCw0PEBESFBUVFhcXFxYVFBISEA8NCwo
IBQQBAQQFCAoLDQ8QEhIUFRYXAAAAAQAAAAA/QDRwA/AH8AhwcRAAABFR8OPw49AS8NKwEPDQUV
Hw4/Dj0BLw0rAQ8NEwcTIRMnMSMhMxMhNSEDBzUha0YBAGMEBAQGBQcGBWgICAgICAgIBwYHBQYEB
AQDAgEBAgMEBAQGBQcGBWgICAgICAgIBwYHBQYEBAQDAgH+aAICAgQEBAYFBwYIBwgICAgICAgHBg
cFBgQEBAQCAQECawQEBAYFBwYHCAGICAgICAcIBgCfBQgEBAICAsH6jAFKjPpu/Z3NwgJZ/dzDAf8
AAQkICAgHBwCGBgUFBAQCAgEBAQEAgQEBQUGBgcHBWgICakIBwgHBwYGBQUEAwMCAQECAwMEBQUG
BgCfBQgEBAICAgHBwYGBQUEBAICAQEBAQICBAQFBQYGBwCHCAgICQgHCAcHBgYFBQDQAwIBAQIDA

```

```

wQFBQYGBwCIBwgB+wH+vQFABP5dOgGkAQEAAAADAAAAAANKA5oAnQDxAR4AAAEzHwEdAR8HFQ8DIY
8HDWYdAR8WDW0dAQ8BKwIvAT0BLwc9AT8COWEfBj8HLxc/DTU/AwEfDjsBPxEvFiMPFR8BEw8CFR8
HMz8HNS8GIw8ELwQrAQ8BAGoCAGENDAwKCgGHBQEBAikCAGIEAwQFDA0SBwcGAGIBAQICBgcHBxYK
CQkJCAcHBgUFBAMCAQEBAQIDAQWQFBQYGBwCPEQECAhUCAQINDAsLCQgHBQICKQICAgQDBAULdHIB
wYCAQEBAQEBAgYHBwCWCgkKCAgHBwYFBQQDAgEBAQECAwMEBAYFBgcHEBABAQED/qwUFRUVFRYWFh
YWFxYXFhcXfXcWFxYXFhYWFhYVFRUVFAQCAQICBAUGCAGJCgsLDA0MDQ0NDBk2EQYgGgYGCEsZDQ0
NDA0MCwsKCQgIBgUEAgIBAgQCAQEBAwKRNRIHBqADChI1DQoFAGEBAGMEBAoMEw8eTw4IVxkXCwkJ
BwYCOAIBAIIDAuUGBwgJCgICAQENAQEFawIDAgECAGMFawMEBAUDBAwFAwIBAQECAwMEBAUGBgYHC
AgICQgHBwCGBgYFBQQEBAyDIgICAQECAiICBAUGBwgJCQMBAGEMAQUDAwIDAQICBAQDBAQEBAQEAW
QEAgEBAQICBAMFBQUGBwCICAgJBwgHBgcGBgUFBAQEBQqIAGEBaf6RDAsLCQkICAYGBQUdAwIBAQI
DAWUFBgYICAKJCwsMKSckIiAeGxoYFhQTERAPDQwLCgkIDxsJBQUBQqNEAKKCwwNDxARExQWGBob
HiAiJCcCoAMDawQECA8XPRcKCgUPFz0REAKIBAMDawMCAQICAwcYAwEaBwQBAgIAAAEAAAAAA/MDN
AA0AAABDwQvAw8EHwQ/ETUnIw8LAYsEJwWgAgIwXmMXFBIIcCsqKaEqRUclJSYnJykpKiosLC4GFg
sCAWMhISiiIiIjIkJAPRwB8AQmCQMBARQuNgsMDgcIJCYnmyZOTyCmJiYlJSQjIiIgHwULCAMCAQ4
RERITFBUVKy0tFgAAABIA3gABAAAAAEEEEAAAAAABAAAAAABAAcAAQABAAAAAACAACACAABAAAA
AAADAACADwABAAAAAEEEEAAAFgABAAAAAFAAsAHQABAAAAAAGAAcAKAABAAAAAAKACwALwABA
AAAAAALABIAWwADAAEECQAAAAIAbQADAAEECQABAA4AbwADAAEECQACAA4AfQADAAEECQADAA4AiW
ADAAEECQAEAA4AmQADAAEECQAFABYApwADAAEECQAGAA4AvQADAAEECQAKAFgAywADAAEECQALACQ
BIyBEZWZhdWx0UmVndWxhcRlZmF1bHREZWZhdWx0VmVyc2lubiAxLjBEZWZhdWx0Rm9udCBnZW5l
cmF0ZWQgdXNpbmcgU3luY2Z1c2lubiBNZXRYbyBTdHVkaW93d3cuc3luY2Z1c2lubi5jb20AIABEA
GUAZgBhAHUAbAB0AFIAZQBnAHUAbABhAHIArABlAGYAYQB1AGwAdABEAGUAZgBhAHUAbAB0AFYAZQ
ByAHMAaQBvAG4AIAAxAAC4AMABEAGUAZgBhAHUAbAB0AEYAbwBuAHQAIABnAGUAbgBlAHIAAYQB0AGU
AZAAgAHUAcwBpAG4AZwAgAFMAeQBvAGMAZgB1AHMAaQBvAG4AIAABNAGUAdABYAG8AIABTAHQAdQBk
AGkAbwB3AHcAdwAuAHMAeQBvAGMAZgB1AHMAaQBvAG4ALGBlAG8AbQAAAAACAAAAAoooooAAAAA
AAAAAAAAAAAAAAAAAAAAAAYBAGEDAQBBQEGAQcADXRyYW5zcG9ydC12YW4ldXNlci1tb2RpZn
kRc2hvcHBpbmctY2FydF8wMS0Lc3BlbmQtbW9uZXkFY2hly2sAAAA=) format('true');
font-weight: normal;
font-style: normal;
}
.stepper-localization [class^="sf-icon-"], .stepper-localization [class*="
sf-icon-"] {
font-family: 'Default' !important;
speak: none;
font-style: normal;
font-weight: normal;
font-variant: normal;
text-transform: none;
line-height: inherit;
-webkit-font-smoothing: antialiased;
-moz-osx-font-smoothing: grayscale;
}
.stepper-localization {
margin-top: 30px;
padding: 30px;
}
.stepper-localization .sf-icon-cart:before { content: "\e710"; }
.stepper-localization .sf-icon-transport:before { content: "\e702"; }
.stepper-localization .sf-icon-payment:before { content: "\e706"; }
.stepper-localization .sf-icon-success:before { content: "\e715"; }

```

## RTL

RTL provides an option to switch the text direction and layout of the Stepper component from right to left by setting the [enableRtl](#) property to `true`.

## INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
    return (
        <div className="stepper-rtl">
            <StepperComponent enableRtl={true}>
                <StepsDirective>
                    <StepDirective iconCss={'sf-icon-cart'} label={'Cart'} />
                    <StepDirective iconCss={'sf-icon-transport'}
label={'Delivery Address'} />
                    <StepDirective iconCss={'sf-icon-payment'}
label={'Payment'} />
                    <StepDirective iconCss={'sf-icon-success'}
label={'Confirmation'} />
                </StepsDirective>
            </StepperComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

**INDEX.TSX**

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
    return (
        <div className="stepper-rtl">
            <StepperComponent enableRtl={true}>
                <StepsDirective>
                    <StepDirective iconCss={'sf-icon-cart'} label={'Cart'} />
                    <StepDirective iconCss={'sf-icon-transport'}
label={'Delivery Address'} />
                    <StepDirective iconCss={'sf-icon-payment'}
label={'Payment'} />
                    <StepDirective iconCss={'sf-icon-success'}
label={'Confirmation'} />
                </StepsDirective>
            </StepperComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

**INDEX.CSS**

```

/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
@font-face {
  font-family: 'Default';
  src:
    url(data:application/x-font-ttf;charset=utf-
8;base64,AAEAAAAKAIAAAwAgTlMvMjlvSgcAAAEoAAAAVmNtYXCDelIPaAAABmAAAAF5nbHlmEwr+
pwAAAggAAAJQaGVhZCYp2+EAADQAAAAANmhoZWEIUQQHAAAArAAAACRobXR4GAAAAAAAYAAAAAYb
G9jYQhUBlAAAAH4AAAAADm1heHABFgErAAABCAAAACBuYW1luF5THQAACTgAAAIlcG9zdJ8LuoMAAA
0AAAAAbwABAAAEAAAAAFwEAAAAAAD9AABAAAAAAAAAAAAAAAAAAAAABgABAAAAAQAArxT6wV8PPPU
ACwQAAAAAAOGLy6UAAAAA4YvLpQAAAAAD9AoaAAAAACAACAAAAAAAAAAAAEAAAAGAR8ABgAAAAAAAgAA
AAoACgAAAP8AAAAAAAAAAQAAZAABQAAAokCzAAAAI8CiQLMAAAB6wAyAQgAAAIABQMAAAAAAAAAAA
AAAAAAAAAAAAAAAAAUGZFZABA5wLnFQQAAAAAXAQAAAAAAAAABAAAAAAAAABAAAAAQAAAAEAAAABA
AAAAQAAAAEAAAAAAAAAgAAAAAMAAAUAMAAQAAABQABABKAAAADAIAAIABOCc5wbnCOCQ5xX//wA
A5wLnBucI5xDnFf//AAAAAAAAAAAAAAAAABAAwADAAMAawADAAAAAEABAACAAMABQAAAAAAAAAEQaiwC
3AQkBGgAAAAFAAAAAAP0A18APwB/AIkAxgDrAAABHw8/Dy8OKwEPDQUfDz8PLw4rAQ8NAR8FFSM1J
xEfBz80OwEfDjM/BzUnIw8GATM/Dx8PMxEhAq8BAQIEBAUFbWYICAgJCQoKCgkKCAkIBwCHBQUEAw
MBAQEBAwMEBQUHBwCICQgKCQoKCgkJCAGIBGcFBQQUEAgH+CwEBAGQEBQUHBggICakJCgoKCQoICQg
HBwCFBQQDAwEBAQEDAwQFBQcHBwgJCAoJCgoKCQkICAgGBwUFBAQCAQJ8AwUIAwD3n0BAwMGBgYI
CAMEBQYHBwkJCgsLDA0NDQ4ODQ4MDAwLCgkJCAYGBQMDKAgIBwYFBAECvLsICAYHBQMD/beAAwQFB
QcHCAkKCgsLDA0MDg0NDQwLCwsJCQkHBwUFAwKE/eMBAQoJCQkJCACHBgYFBAMDAQEBAQMDBAUGBg
cHCAkJCQkKCgoJCQgICAcGBgQFAwICAgIDBAUFbGcHCAkJCQoLCgkJCQkIBwCGBgUEAwMBAQEBAwM
EBQYGBwCICQkJCQoKCgkJCAGIBwYGBAUDAgICAgMEBQUGBwCICQkJCgGuAQIGehYJBKYp/10ICAcG
BQQCAQ0NDQwLCgoJCAGGBQUDAgIDBQUGCAGJCgoLDA0NDQECBAUGBwQI1foBAGQFBGcH/iwNDAwLC
woJCQgHBgUEAwEBAQEDBAUGBwgJCQoLCwwMDQJJAAAAABgAAAAAD8wOWAAYAQQgBaAGwArQDuAAABZ
cfAwUhLwIHIY8PNS8CKwIPHQEHLwEjDwE1LwMjNz0BJZcfBTcfAg8BLwY30wEfAQcVHw8/Dy8PDw4
BFR8PPw8vDw8OAxEWBgEDAgb8/wNuBAUEDQsVFBQTEhEPDw0GCwoIBgQCFhITE+wQDw8PDg4ODg0N
DQwNCwwKCwoKCQgJBwCHBgyEBQMEa5FrBAQDBAMBAwMDBgIDagIEBgYGBxwCAwIBFQYGBAgFBgIWA
gQHCPcBAGQGBggKCgsMDQ4PDxAQEBApDw4NDAsLCQgGBgQCAQECAyGCAkLCwwNDg8PEBAQEa8PDg
0MCwoKCAyGBAL+KgEEBQgKCw0PEBETFBQWFxcXFhYUFBMREQ4NDakIBgMBAQMGCakMDQ4RERMUFBY
WFxcXFhQUEExEQDw0LCgGFBAEXBhcFBAMDrxYWDQEBawUHCAsmDQ4IERESFBQUFQQDAgECAGMEBAUG
BgYIBwgJCQoKCwsLDAwMDQ0ODQ4PDgEZawIBAQIGBQMCAQQDBgZqBgOHBQMDMAMHBwMWAQICBQYKC
hYCB1wICBAPDw4NDAsLCQgGBgQCAQECAyGCAkLCwwNDg8PEBAQEa8PDg0MCwoKCAyGawMBAQMDBg
YICgoLDA0ODw8QATMLDBYVFRQSERAPDQsKCAUEAQEEBQgKCw0PEBESFBUVFhcXFxYVFBISEA8NCwo
IBQQBAQQFCAoLDQ8QEhIUFRYXAAAAAAQAAAAA/QDRwA/AH8AhwcRAAABFR8OPw49AS8NKwEPDQUV
Hw4/Dj0BLw0rAQ8NEwcTIRMnMSMhMxMhNSEDBzUha0YBAGMEBAQGBQcGBwgICAgICAgIBwYHBQYEB
AQDAgEBAgMEBAQGBQcGBwgICAgICAgIBwYHBQYEBAQDAgH+aAICAgQEBAYFBwYIBwgICAgICAgHBg
cFBgQEBAQCAQECAwQEBAYFBwYHCAGICAgICAcIBGcFBgQEBAICAsH6jAFKjPpu/Z3NwgJZ/dzDAf8
AAQkICAgHBwCGBgUFBAQCAgEBAQECAgQEBQUGBgHBwgICakIBwgHBwYGBQUEAwMCAQECAwMEBQUG
BgCHCAcICQgICAcHBwYGBQUEBAICAEBAQICBAQFBQYGBwCHCAgICQgHCAcHBgYFBQQDAwIBAQIDA
wQFBQYGBwCIBwgB+wH+vQFABP5dOgGkAQEAADAAAAAANKA5oAnQDxAR4AAAEzHwEdAR8HFQ8DIY
8HDWYdAR8WDw0dAQ8BKwIvAT0BLwc9AT8COWEfBj8HLxc/DTU/AwEfDjsBPxEvFiMPFR8BEw8CFR8
HMz8HNS8GIw8ELwQrAQ8BAGoCAGENDAwKCGgHBQEBAikCAGIEAwQFDA0SBwcGAgIBAQICBgCHBxYK
CQkJCACHBgUFBAQCAQEBAQIDAQFBQYGBwCPEQECAhUCAQINDAsLCQgHBQICKQICAgQDBAULdhIHB
wYCAQEBAQEBAgYHBwCWCgkKCAgHBwYFBQQDAgEBAQECAwMEBAYFBGcHEBABAQED/qwUFRUVFRYWFh
YWFxYXFhcXFxcWFxYXFhYWFhYVFRUVFAQCAQICBAUGCAGJCgsLDA0MDQ0NDBk2EQYgqgYGCEsZDQ0
NDA0MCwsKCQgIBgUEAgIBaQCAQEBAwkrNRiHBqADChI1DQoFagEBAgMEBAOMew8eTw4IVxkXCwkJ
BwYCOAIBAIIDAuUGBwgJCgICAQENAQEFawIDAQECAGMFAwMEBAUDBAMFAwIBAQECAwMEBAUGBgYHC
AgICQgHBwCGBgYFBQQUEBAYDIgICAQECAiICBAUGBwgJCQMBAGEMAQUDAwIDAQICBAQDBAQEBAQEAW
QEAgEBAQICBAMFBQUGBwCICAgJBwgHBGcGBgUFBAQEBOQIAGeBAf6RDAsLCQkICAYGBQUDAwIBAQI

```

```

DAwUFBgYICakJCwsMKSckIiAeGxoYFhQTERAPDQwLCgkIDxsJBQUFBQQnEakKCwwNDxARExQWGBob
HiAiJCcCoAMDawQECA8XPRcKCgUPFz0REakIBAMDAwMCAQICAwcYAwEaBwQBAGIAAAEAAAAAA/MDN
AA0AAABDwQvAw8EHwQ/ETUnIw8LAYsEJwwGAgIwXmMXFBIIcCsqKaEqRUclJSYnJykpKiosLC4GFg
sCAWMhISiIiIjIkJAPRwB8AQmCQMBARQuNgsMDgcIJCYnmyZOTycmJiYlJSQjIiIgHwULCAMCAQ4
RERITFBUVKy0tFgAAABIA3gABAAAAAAAAAAAAEAAAABAAAAAAAAABAACAAQABAAAAAAAAACAACAABAAAA
AAADAACADwABAAAAAAAAAAEAAcAFgABAAAAAAAAFAAAsAHQABAAAAAAAAAGAAcAKAABAAAAAAAAAKACwALwABA
AAAAAAAALABIAWwADAAEECQAAAAIAbQADAAEECQABAA4AbwADAAEECQACAA4AfQADAAEECQADAA4AiW
ADAAEECQAEAA4AmQADAAEECQAFABYApwADAAEECQAGAA4AvQADAAEECQAKAFgAywADAAEECQALACQ
BIyBEZWZhdWx0UmVndWxhckRlZmF1bHREZWZhdWx0VmVyc2lvbiAxLjBEZWZhdWx0Rm9udCBnZW5l
cmF0ZWQgdXNpbmcgU3luY2Z1c2lvbiBNZXRYbyBTdHVkaW93d3cuc3luY2Z1c2lvbi5jb20AIABEA
GUAZgBhAHUAbAB0AFIAZQBnAHUAbABhAHIArABLAGYAYQB1AGwAdABEAGUAZgBhAHUAbAB0AFYAZQ
ByAHMAaQBvAG4AIAAxAAC4AMABEAGUAZgBhAHUAbAB0AEYAbwBuAHQAIABnAGUAbgB1AHIAZQB0AGU
AZAAgAHUAcwBpAG4AZwAgAFMAeQBvAGMAZgB1AHMAaQBvAG4AIAABNAGUAdABYAG8AIAABTAHQAdQBk
AGkAbwB3AHcAdwAuAHMAeQBvAGMAZgB1AHMAaQBvAG4ALgBjAG8AbQAAAAACAAAAAooooAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAYBAGEDAQQBBQEGAQCADXRYW5zcG9ydC12YW4LdXNlciltb2RpZn
kRc2hvcHBpbmctY2FydF8wMS0Lc3BlbmQtbW9uZXkFY2h1Y2sAAAA=) format('trueType');
  font-weight: normal;
  font-style: normal;
}
.stepper-rtl [class^="sf-icon-"], .stepper-rtl [class*=" sf-icon-"] {
  font-family: 'Default' !important;
  speak: none;
  font-style: normal;
  font-weight: normal;
  font-variant: normal;
  text-transform: none;
  line-height: inherit;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}
.stepper-rtl {
  margin-top: 30px;
  padding: 30px;
}
.stepper-rtl .sf-icon-cart:before { content: "\e710"; }
.stepper-rtl .sf-icon-transport:before { content: "\e702"; }
.stepper-rtl .sf-icon-payment:before { content: "\e706"; }
.stepper-rtl .sf-icon-success:before { content: "\e715"; }

```

## Accessibility in React Stepper component

The Stepper component followed the accessibility guidelines and standards, including [ADA](#), [Section 508](#), [WCAG 2.2](#) standards, and [WCAG roles](#) that are commonly used to evaluate accessibility.

The accessibility compliance for the Stepper component is outlined below.

| Accessibility Criteria | Compatibility |

| -- | -- |

| [WCAG 2.2](#) Support |  |

| [Section 508](#) Support |  |



| Screen Reader Support | `` |

| Right-To-Left Support | `` |

| Color Contrast | `` |

| Mobile Device Support | `` |

| Keyboard Navigation Support | `` |

| [Accessibility Checker](#) Validation | `` |

| [Axe-core](#) Accessibility Validation | `` |

```
<style>
.post .post-content img {
display: inline-block;
margin: 0.5em 0;
}
</style>

<div> - All
features of the component meet the requirement.</div>

<div> - Some features of the component do not meet the requirement.</div>

<div> - The
component does not meet the requirement.</div>
```

### WAI-ARIA attributes

The following ARIA attributes are used in the Stepper component:

- | Attributes    | Purpose   |
|---------------|---|
| aria-label    | Attribute provide a descriptive text that labels the interactive element for accessibility. |
| aria-current  | Attribute denotes the currently active step in the Stepper.                                 |
| aria-disabled | Indicates when the step is disabled and cannot be interacted.                               |

### Keyboard interaction

The following keyboard shortcuts are supported by the Stepper component.

- | Press     | To do this |
|-----------|------------|
| ---   --- |            |

- | Up Arrow | Focuses the previous step in a vertical Stepper. |
- | Down Arrow | Focuses the next step in a vertical Stepper. |
- | Left Arrow | Focuses the previous step in a horizontal Stepper. |
- | Right Arrow | Focuses the next step in a horizontal Stepper. |
- | Home | Focuses on the first step of the Stepper. |
- | End | Focuses on the last step of the Stepper. |
- | Enter / Space | Activates the currently focused step. |

### Ensuring accessibility

The Stepper component's accessibility levels are ensured through an [accessibility-checker](#) and [axe-core](#) software tools during automated testing.

The accessibility compliance of the Stepper component is shown in the following sample. Open the [sample](#) in a new window to evaluate the accessibility of the Stepper component with accessibility tools.

See also

- [Accessibility in Syncfusion React components](#)

### Events in React Stepper component

This section describes the Stepper events that will be triggered when an appropriate actions are performed. The following events are available in the Stepper component.

#### created

The Stepper component triggers the [created](#) event when the control rendering is completed.

#### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
  return (
    <div className="stepper-created">
      <StepperComponent created={() => { /* Your required action here
*/ } } >>
        <StepsDirective>
          <StepDirective/>
          <StepDirective/>
          <StepDirective/>
          <StepDirective/>
        </StepsDirective>
      </StepperComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

**INDEX.TSX**

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
 '@syncfusion/ej2-react-navigations';
function App() {
  return (
    <div className="stepper-created">
      <StepperComponent created={() => { /* Your required action here
*/ } }>
        <StepsDirective>
          <StepDirective/>
          <StepDirective/>
          <StepDirective/>
          <StepDirective/>
        </StepsDirective>
      </StepperComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

**INDEX.CSS**

```
/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
.stepper-created {
  margin-top: 30px;
  padding: 30px;
}
```

**stepChanged**

The Stepper component triggers the [stepChanged](#) event after the active step is changed.

**INDEX.JSX**

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
 '@syncfusion/ej2-react-navigations';
function App() {
  return (
```

```

        <div className="stepper-step-changed">
            <StepperComponent stepChanged={ (args) => { alert("Step Changed
from "+args.previousStep + " to " + args.activeStep) } }>
                <StepsDirective>
                    <StepDirective/>
                    <StepDirective/>
                    <StepDirective/>
                    <StepDirective/>
                </StepsDirective>
            </StepperComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

### INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective,
StepperChangedEventArgs } from '@syncfusion/ej2-react-navigations';
function App() {
    return (
        <div className="stepper-step-changed">
            <StepperComponent stepChanged={ (args: StepperChangedEventArgs) =>
{ alert("Step Changed from " + args.previousStep + " to " + args.activeStep)
}}} >
                <StepsDirective>
                    <StepDirective />
                    <StepDirective />
                    <StepDirective />
                    <StepDirective />
                </StepsDirective>
            </StepperComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

### INDEX.CSS

```

/* Represents the styles for loader */
#loader {
    color: #008cff;
    height: 40px;
    left: 45%;
    position: absolute;
    top: 45%;
    width: 30%;
}
.stepper-step-changed {

```

```
margin-top: 30px;
padding: 30px;
}
```

### stepChanging

The Stepper component triggers the [stepChanging](#) event before the active step change.

#### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
  return (
    <div className="stepper-step-changing">
      <StepperComponent stepChanging={ (args) => { alert("Step Changing
from "+args.previousStep + " to " + args.activeStep) } }>
        <StepsDirective>
          <StepDirective/>
          <StepDirective/>
          <StepDirective/>
          <StepDirective/>
        </StepsDirective>
      </StepperComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

#### INDEX.TSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective,
StepperChangingEventArgs } from '@syncfusion/ej2-react-navigations';
function App() {
  return (
    <div className="stepper-step-changing">
      <StepperComponent stepChanging={ (args: StepperChangingEventArgs)
=> { alert("Step Changing from "+args.previousStep + " to " +
args.activeStep) } }>
        <StepsDirective>
          <StepDirective/>
          <StepDirective/>
          <StepDirective/>
          <StepDirective/>
        </StepsDirective>
      </StepperComponent>
    </div>
  );
}
}
```

```
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

## INDEX.CSS

```
/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
.stepper-step-changing {
  margin-top: 30px;
  padding: 30px;
}
```

## stepClick

The Stepper component triggers the [stepClick](#) event when the step is clicked.

## INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
  return (
    <div className="stepper-step-click">
      <StepperComponent stepClick={(args) => { /* Your required action
here */ }}>
        <StepsDirective>
          <StepDirective/>
          <StepDirective/>
          <StepDirective/>
          <StepDirective/>
        </StepsDirective>
      </StepperComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

## INDEX.TSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
```

```
import { StepperComponent, StepsDirective, StepDirective,
StepperClickEventArgs } from '@syncfusion/ej2-react-navigations';
function App() {
    return (
        <div className="stepper-step-click">
            <StepperComponent stepClick={ (args: StepperClickEventArgs) => {
/* Your required action here */ } }>
                <StepsDirective>
                    <StepDirective/>
                    <StepDirective/>
                    <StepDirective/>
                    <StepDirective/>
                </StepsDirective>
            </StepperComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

## INDEX.CSS

```
/* Represents the styles for loader */
#loader {
    color: #008cff;
    height: 40px;
    left: 45%;
    position: absolute;
    top: 45%;
    width: 30%;
}
.stepper-step-click {
    margin-top: 30px;
    padding: 30px;
}
```

## beforeStepRender

The Stepper component triggers the [beforeStepRenderLink to the Video](#) event before rendering each step.

## INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective } from
 '@syncfusion/ej2-react-navigations';
function App() {
    return (
        <div className="stepper-before-render">
            <StepperComponent beforeStepRender={ (args) => { /* Your required
action here */ } }>
                <StepsDirective>
                    <StepDirective/>

```

```

                <StepDirective/>
                <StepDirective/>
                <StepDirective/>
            </StepsDirective>
        </StepperComponent>
    </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

## INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StepperComponent, StepsDirective, StepDirective,
StepperRenderingEventArgs } from '@syncfusion/ej2-react-navigations';
function App() {
    return (
        <div className="stepper-before-render">
            <StepperComponent beforeStepRender={ (args:
StepperRenderingEventArgs) => { /* Your required action here */ } }>
                <StepsDirective>
                    <StepDirective/>
                    <StepDirective/>
                    <StepDirective/>
                    <StepDirective/>
                </StepsDirective>
            </StepperComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

## INDEX.CSS

```

/* Represents the styles for loader */
#loader {
    color: #008cff;
    height: 40px;
    left: 45%;
    position: absolute;
    top: 45%;
    width: 30%;
}
.stepper-before-render {
    margin-top: 30px;
    padding: 30px;
}

```



## Stock Chart

<!-- markdownlint-disable MD036 -->

### Getting Started

This section explains you the steps required to create a simple Stock Chart and demonstrate the basic usage of the Stock Chart control.

To get start quickly with React Stock Chart, you can check on this video:

### Dependencies

Below is the list of minimum dependencies required to use the chart component.

```
`javascript
|-- @syncfusion/ej2-react-charts
|-- @syncfusion/ej2-base
|-- @syncfusion/ej2-data
|-- @syncfusion/ej2-charts
|-- @syncfusion/ej2-react-base
|-- @syncfusion/ej2-pdf-export
|-- @syncfusion/ej2-file-utils
|-- @syncfusion/ej2-compression
|-- @syncfusion/ej2-svg-base
|-- @syncfusion/ej2-navigations
|-- @syncfusion/ej2-calendars
|-- @syncfusion/ej2-popups
|-- @syncfusion/ej2-lists
|-- @syncfusion/ej2-inputs
|-- @syncfusion/ej2-buttons
|-- @syncfusion/ej2-splitbuttons
`,`
```

### Installation and configuration

You can use [create-react-app](#) to setup the applications.

To install `create-react-app` run the following command.

```
`
npm install -g create-react-app
`,`
```

- To set-up a React application in TypeScript environment, run the following command.

```
`  
create-react-app quickstart --template typescript  
cd quickstart  
npm start  
`
```

- To set-up a React application in JavaScript environment, run the following command.

```
`  
create-react-app quickstart  
cd quickstart  
npm start  
`
```

- Install Syncfusion packages using below command.

```
`  
npm install @syncfusion/ej2-react-charts --save  
`
```

### Add Stock Chart to the Project

Now, you can start adding Chart component in the application.

For getting started, add the Chart component in `src/App.tsx` file using following code.

#### INDEX.JSX

```
import { StockChartComponent } from '@syncfusion/ej2-react-charts';  
import * as React from 'react';  
function App() {  
  return (<StockChartComponent />);  
}  
export default App;
```

#### INDEX.TSX

```
import { StockChartComponent } from '@syncfusion/ej2-react-charts';  
import * as React from 'react';  
function App() {  
  return (<StockChartComponent />);  
}  
export default App;
```

Now run the `npm start` command in the console, it will run your application and open the browser window.

npm start

The below example shows a basic Chart.

### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    return (<StockChartComponent id='stockchart' primaryXAxis={{
        valueType: 'DateTime',
        majorGridLines: { width: 0 }, majorTickLines: { color:
'transparent' },
        crosshairTooltip: { enable: true }
    }} primaryYAxis={{
        labelFormat: 'n0',
        lineStyle: { width: 0 }, rangePadding: 'None',
        majorTickLines: { width: 0 }
    }} height='350'>
        <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,
EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator, Export,
AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle'>
                </StockChartSeriesDirective>
            </StockChartSeriesCollectionDirective>
        </StockChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

### INDEX.TSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import {
```

```

    StockChartComponent, StockChartSeriesCollectionDirective,
    StockChartSeriesDirective, Inject, ITooltipRenderEventArgs,
    IStockChartEventArgs, ChartTheme,
    DateTime, Tooltip, RangeTooltip, Crosshair, LineSeries, SplineSeries,
    CandleSeries, HiloOpenCloseSeries, HiloSeries, RangeAreaSeries, Trendlines
} from '@syncfusion/ej2-react-charts';
import {
    EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
    MomentumIndicator, SmaIndicator, AtrIndicator,
    AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
    Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    return (
        <StockChartComponent id='stockchart'
            primaryXAxis={{
                valueType: 'DateTime',
                majorGridLines: { width: 0 }, majorTickLines: { color:
'transparent' },
                crosshairTooltip: { enable: true }
            }}
            primaryYAxis={{
                labelFormat: 'n0',
                lineStyle: { width: 0 }, rangePadding: 'None',
                majorTickLines: { width: 0 }
            }}
            height='350'
        >
            <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,
                EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
                MomentumIndicator, SmaIndicator, AtrIndicator, Export,
                AccumulationDistributionIndicator, MacdIndicator,
                StochasticIndicator]} />
            <StockChartSeriesCollectionDirective>
                <StockChartSeriesDirective dataSource={chartData}
type='Candle'>
                    </StockChartSeriesDirective>
                </StockChartSeriesCollectionDirective>
            </StockChartComponent>
        )
    );
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% enddraw %}

```

### Module Injection

Stock Chart component are segregated into individual feature-wise modules. In order to use a particular feature, you need to inject its feature service in the AppModule. In the current application, we are going to modify the above basic chart to visualize stock value of a company.

For this application we are going to use candle series, tooltip, data label, DateTime axis

feature of the chart. Please find the relevant feature service name and description as follows.

- **CandleSeries** - Inject this module in to **services** to use candle series.
- **Tooltip** - Inject this module in to **services** to use tooltip feature.
- **DataLabel** - Inject this module in to **services** to use datalabel feature.
- **DateTime** - Inject this module in to **services** to use DateTime feature.

These modules should be injected to the **services** section as follows,

#### **INDEX.JSX**

```
import { DateTime, StockChartComponent, DataLabel, CandleSeries, Tooltip }
  from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return <StockChartComponent id='stockcharts'>
    <Inject services={[CandleSeries, Tooltip, DataLabel, Category]} />
  </StockChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

#### **INDEX.TSX**

```
import { StockChartComponent, DataLabel, CandleSeries, Tooltip } from
  '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return <StockChartComponent id='stockcharts'>
    <Inject services={[CandleSeries, Tooltip, DataLabel, Category]} />
  </StockChartComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

#### Populate Chart with Data

This section explains how to plot below JSON data to the Stock Chart. Please find the below imported datasource.

#### **INDEX.JSX**

```
export let data = [{
  "x": new Date('2012-04-02T00:00:00.000Z'),
  "open": 320.705719,
  "high": 324.074066,
  "low": 317.737732,
  "close": 323.783783,
  "volume": 45638000
}, {
  "x": new Date('2012-04-03T00:00:00.000Z'),
  "open": 323.028015,
  "high": 324.299286,
  "low": 319.639648,
  "close": 321.631622,
```

```

"volume": 40857000
}, {
"x": new Date('2012-04-04T00:00:00.000Z'),
"open": 319.544556,
"high": 319.819824,
"low": 315.865875,
"close": 317.892883,
"volume": 32519000
}, {
"x": new Date('2012-04-05T00:00:00.000Z'),
"open": 316.436432,
"high": 318.533539,
"low": 314.599609,
"close": 316.476471,
"volume": 46327000
}];

```

### INDEX.TSX

```

export let data: any[] = [{
"x": new Date('2012-04-02T00:00:00.000Z'),
"open": 320.705719,
"high": 324.074066,
"low": 317.737732,
"close": 323.783783,
"volume": 45638000
}, {
"x": new Date('2012-04-03T00:00:00.000Z'),
"open": 323.028015,
"high": 324.299286,
"low": 319.639648,
"close": 321.631622,
"volume": 40857000
}, {
"x": new Date('2012-04-04T00:00:00.000Z'),
"open": 319.544556,
"high": 319.819824,
"low": 315.865875,
"close": 317.892883,
"volume": 32519000
}, {
"x": new Date('2012-04-05T00:00:00.000Z'),
"open": 316.436432,
"high": 318.533539,
"low": 314.599609,
"close": 316.476471,
"volume": 46327000
}];

```

Add a series object to the chart by using [series](#) property and then set the JSON data to [dataSource](#) property.

Since the JSON contains DateTime data, set the [valueType](#) for horizontal axis to `DateTime`. By default, the axis valueType is `Numeric`.

**INDEX.JSX**

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    return (<StockChartComponent id='stockchart' primaryXAxis={{
        valueType: 'DateTime',
        majorGridLines: { width: 0 }, majorTickLines: { color:
'transparent' },
        crosshairTooltip: { enable: true }
    }} primaryYAxis={{
        labelFormat: 'n0',
        lineStyle: { width: 0 }, rangePadding: 'None',
        majorTickLines: { width: 0 }
    }} height='350'>
        <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,
EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator, Export,
AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle'>
                </StockChartSeriesDirective>
            </StockChartSeriesCollectionDirective>
        </StockChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

**INDEX.TSX**

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import {
    StockChartComponent, StockChartSeriesCollectionDirective,
    StockChartSeriesDirective, Inject, ITooltipRenderEventArgs,
    IStockChartEventArgs, ChartTheme,
    DateTime, Tooltip, RangeTooltip, Crosshair, LineSeries, SplineSeries,
    CandleSeries, HiloOpenCloseSeries, HiloSeries, RangeAreaSeries, Trendlines
}
```

```

} from '@syncfusion/ej2-react-charts';
import {
  EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
  MomentumIndicator, SmaIndicator, AtrIndicator,
  AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
  Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  return (
    <StockChartComponent id='stockchart'
      primaryXAxis={{
        valueType: 'DateTime',
        majorGridLines: { width: 0 }, majorTickLines: { color:
'transparent' },
        crosshairTooltip: { enable: true }
      }}
      primaryYAxis={{
        labelFormat: 'n0',
        lineStyle: { width: 0 }, rangePadding: 'None',
        majorTickLines: { width: 0 }
      }}
      height='350'
    >
      <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,
        EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator, Export,
        AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]} />
      <StockChartSeriesCollectionDirective>
        <StockChartSeriesDirective dataSource={chartData}
type='Candle'>
          </StockChartSeriesDirective>
        </StockChartSeriesCollectionDirective>
      </StockChartComponent>
    )
  );
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

### Add Stock Chart Title

You can add a title using [title](#) property to the Stock Chart to provide quick information to the user about the data plotted in the chart.

### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,

```



```
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    return (<StockChartComponent id='stockchart' primaryXAxis={{
        valueType: 'DateTime',
        majorGridLines: { width: 0 }, majorTickLines: { color:
'transparent' },
        crosshairTooltip: { enable: true }
    }} primaryYAxis={{
        labelFormat: 'n0',
        lineStyle: { width: 0 }, rangePadding: 'None',
        majorTickLines: { width: 0 }
    }} title='Sales Analysis' height='350'>
        <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,
        EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator, Export,
        AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle'>
                </StockChartSeriesDirective>
            </StockChartSeriesCollectionDirective>
        </StockChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

## INDEX.TSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import {
    StockChartComponent, StockChartSeriesCollectionDirective,
    StockChartSeriesDirective, Inject, ITooltipRenderEventArgs,
    IStockChartEventArgs, ChartTheme,
    DateTime, Tooltip, RangeTooltip, Crosshair, LineSeries, SplineSeries,
    CandleSeries, HiloOpenCloseSeries, HiloSeries, RangeAreaSeries, Trendlines
} from '@syncfusion/ej2-react-charts';
import {
    EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
    MomentumIndicator, SmaIndicator, AtrIndicator,
    AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
    Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
```

```
function App() {
  return (
    <StockChartComponent id='stockchart'
      primaryXAxis={{
        valueType: 'DateTime',
        majorGridLines: { width: 0 }, majorTickLines: { color:
'transparent' },
        crosshairTooltip: { enable: true }
      }}
      primaryYAxis={{
        labelFormat: 'n0',
        lineStyle: { width: 0 }, rangePadding: 'None',
        majorTickLines: { width: 0 }
      }}
      title= 'Sales Analysis'
      height='350'
    >
      <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,
EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator, Export,
AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]} />
      <StockChartSeriesCollectionDirective>
        <StockChartSeriesDirective dataSource={chartData}
type='Candle'>
          </StockChartSeriesDirective>
        </StockChartSeriesCollectionDirective>
      </StockChartComponent>
    )
  );
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

### Add Crosshair

Crosshair has a vertical and horizontal line to view the value of the axis at mouse or touch position.

Crosshair lines can be enabled by using [enable](#) property in the `crosshair`. Likewise tooltip label for an axis can be enabled by using [enable](#) property of `crosshairTooltip` in the corresponding axis.

### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
```

```

import { chartData } from 'datasource.ts';
function App() {
  return (<StockChartComponent id='stockchart' primaryXAxis={{
    valueType: 'DateTime',
    majorGridLines: { width: 0 }, majorTickLines: { color:
'transparent' },
    crosshairTooltip: { enable: true }
  }} primaryYAxis={{
    labelFormat: 'n0',
    lineStyle: { width: 0 }, rangePadding: 'None',
    majorTickLines: { width: 0 }
  }} title='Sales Analysis' crosshair={{ enable: true }} height='350'>
    <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,
    EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator, Export,
    AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle'>
        </StockChartSeriesDirective>
      </StockChartSeriesCollectionDirective>
    </StockChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

## INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import {
  StockChartComponent, StockChartSeriesCollectionDirective,
  StockChartSeriesDirective, Inject, ITooltipRenderEventArgs,
  IStockChartEventArgs, ChartTheme,
  DateTime, Tooltip, RangeTooltip, Crosshair, LineSeries, SplineSeries,
  CandleSeries, HiloOpenCloseSeries, HiloSeries, RangeAreaSeries, Trendlines
} from '@syncfusion/ej2-react-charts';
import {
  EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
  MomentumIndicator, SmaIndicator, AtrIndicator,
  AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
  Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  return (
    <StockChartComponent id='stockchart'
      primaryXAxis={{
        valueType: 'DateTime',

```

```

        majorGridLines: { width: 0 }, majorTickLines: { color:
'transparent' },
        crosshairTooltip: { enable: true }
    }}
    primaryYAxis={{
        labelFormat: 'n0',
        lineStyle: { width: 0 }, rangePadding: 'None',
        majorTickLines: { width: 0 }
    }}
    title= 'Sales Analysis'
    crosshair={{ enable: true }}
    height='350'
>
    <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,
        EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator, Export,
        AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]} />
    <StockChartSeriesCollectionDirective>
        <StockChartSeriesDirective dataSource={chartData}
type='Candle'>
            </StockChartSeriesDirective>
        </StockChartSeriesCollectionDirective>
    </StockChartComponent>
)
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

### Add Trackball

Trackball is used to track a data point closest to the mouse or touch position. Trackball marker indicates the closest point and trackball tooltip displays the information about the point. To use trackball feature, we need to inject **Crosshair** and **Tooltip** module into the **services**.

Trackball can be enabled by setting the [enable](#) property of the crosshair to true and [shared](#) property in tooltip to true in chart.

### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { trackData } from 'datasource.ts';
function App() {

```

```

    return (<StockChartComponent id='stockchart' primaryXAxis={{
      valueType: 'DateTime',
    }} height='350' title='Sales Analysis' tooltip={{ enable: true,
shared: true, format: '${series.name} : ${point.x} : ${point.y}' }}
crosshair={{ enable: true, lineType: 'Vertical' }}>
      <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,
      EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator, Export,
      AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
      <StockChartSeriesCollectionDirective>
        <StockChartSeriesDirective dataSource={trackData}
xName='x' yName='y' name='John' type='Line' width={2} marker={{ visible: true
}}>
        </StockChartSeriesDirective>
        <StockChartSeriesDirective dataSource={trackData}
xName='x' yName='y1' name='Andrew' type='Line' width={2} marker={{ visible:
true }}></StockChartSeriesDirective>
        <StockChartSeriesDirective dataSource={trackData}
xName='x' yName='y2' name='Thomas' type='Line' width={2} marker={{ visible:
true }}></StockChartSeriesDirective>
        <StockChartSeriesDirective dataSource={trackData}
xName='x' yName='y3' name='Mark' type='Line' width={2} marker={{ visible:
true }}></StockChartSeriesDirective>
        <StockChartSeriesDirective dataSource={trackData}
xName='x' yName='y4' name='William' type='Line' width={2} marker={{ visible:
true }}></StockChartSeriesDirective>
      </StockChartSeriesCollectionDirective>
    </StockChartComponent>;
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

## INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import {
  StockChartComponent, StockChartSeriesCollectionDirective,
  StockChartSeriesDirective, Inject, ITooltipRenderEventArgs,
  IStockChartEventArgs, ChartTheme,
  DateTime, Tooltip, RangeTooltip, Crosshair, LineSeries, SplineSeries,
  CandleSeries, HiloOpenCloseSeries, HiloSeries, RangeAreaSeries, Trendlines
} from '@syncfusion/ej2-react-charts';
import {
  EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
  MomentumIndicator, SmaIndicator, AtrIndicator,
  AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
  Export
} from '@syncfusion/ej2-react-charts';
import { trackData } from 'datasource.ts';

```

```
function App() {
  return (
    <StockChartComponent id='stockchart'
      primaryXAxis={{
        valueType: 'DateTime',
      }}
      height='350'
      title= 'Sales Analysis'
      tooltip={{ enable: true, shared: true, format:
        '${series.name} : ${point.x} : ${point.y}' }}
      crosshair={{ enable: true, lineType: 'Vertical' }}
    >
      <Inject services={[DateTime, Tooltip, RangeTooltip,
        Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
        HiloSeries, RangeAreaSeries, Trendlines,
          EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
          MomentumIndicator, SmaIndicator, AtrIndicator, Export,
          AccumulationDistributionIndicator, MacdIndicator,
          StochasticIndicator]} />
      <StockChartSeriesCollectionDirective>
        <StockChartSeriesDirective dataSource={trackData}
          xName='x' yName='y' name='John' type='Line' width={2} marker={{ visible: true
        }}>
        </StockChartSeriesDirective>
        <StockChartSeriesDirective dataSource={trackData}
          xName='x' yName='y1' name='Andrew' type='Line' width={2} marker={{ visible:
          true }}></StockChartSeriesDirective>
        <StockChartSeriesDirective dataSource={trackData}
          xName='x' yName='y2' name='Thomas' type='Line' width={2} marker={{ visible:
          true }}></StockChartSeriesDirective>
        <StockChartSeriesDirective dataSource={trackData}
          xName='x' yName='y3' name='Mark' type='Line' width={2} marker={{ visible:
          true }}></StockChartSeriesDirective>
        <StockChartSeriesDirective dataSource={trackData}
          xName='x' yName='y4' name='William' type='Line' width={2} marker={{ visible:
          true }}></StockChartSeriesDirective>
      </StockChartSeriesCollectionDirective>
    </StockChartComponent>
  )
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

You can refer to our [React Stock Chart](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Stock Chart example](#) that shows you how to present and manipulate data.

<!-- markdownlint-disable MD036 -->

## Working with data in React Stock chart component

Stock Chart can visualise data bound from local or remote data.

### Local Data

You can bind a simple JSON data to the chart using [dataSource](#) property in series.

**INDEX.JSX**

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    return (<StockChartComponent id='stockchart' primaryXAxis={{
        valueType: 'DateTime',
        majorGridLines: { width: 0 }, majorTickLines: { color:
'transparent' },
        crosshairTooltip: { enable: true }
    }} primaryYAxis={{
        labelFormat: 'n0',
        lineStyle: { width: 0 }, rangePadding: 'None',
        majorTickLines: { width: 0 }
    }} title='Sales Analysis' bearFillColor='#2ecd71'
bullFillColor='#e74c3d' height='350'>
        <Inject services=[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,
EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator, Export,
AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]]/>
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
            </StockChartSeriesDirective>
        </StockChartSeriesCollectionDirective>
    </StockChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

**INDEX.TSX**

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import {
    StockChartComponent, StockChartSeriesCollectionDirective,
    StockChartSeriesDirective, Inject, ITooltipRenderEventArgs,
    IStockChartEventArgs, ChartTheme,
```

```

    DateTime, Tooltip, RangeTooltip, Crosshair, LineSeries, SplineSeries,
    CandleSeries, HiloOpenCloseSeries, HiloSeries, RangeAreaSeries, Trendlines
  } from '@syncfusion/ej2-react-charts';
import {
    EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
    MomentumIndicator, SmaIndicator, AtrIndicator,
    AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
    Export
  } from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  return (
    <StockChartComponent id='stockchart'
      primaryXAxis={{
        valueType: 'DateTime',
        majorGridLines: { width: 0 }, majorTickLines: { color:
'transparent' },
        crosshairTooltip: { enable: true }
      }}
      primaryYAxis={{
        labelFormat: 'n0',
        lineStyle: { width: 0 }, rangePadding: 'None',
        majorTickLines: { width: 0 }
      }}
      title= 'Sales Analysis'
      bearFillColor='#2ecd71'
      bullFillColor='#e74c3d'
      height='350'
    >
      <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,
        EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
        MomentumIndicator, SmaIndicator, AtrIndicator, Export,
        AccumulationDistributionIndicator, MacdIndicator,
        StochasticIndicator]} />
      <StockChartSeriesCollectionDirective>
        <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
          </StockChartSeriesDirective>
        </StockChartSeriesCollectionDirective>
      </StockChartComponent>
    )
  );
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% enddraw %}

```

See Also

- [Series Types](#)



## Chart dimensions in React Stock chart component

### Size for Container

Stock Chart can render to its container size. You can set the size via inline or CSS as demonstrated below.

```
,  
  
<div id="stockcharts" style="width:650px; height:350px"></div>  
,  
  
`ts  
import * as React from "react";  
import * as ReactDOM from "react-dom";  
import { StockChartComponent }  
from '@syncfusion/ej2-react-charts';  
function App() {  
  return <StockChartComponent id='stockcharts'>  
    </StockChartComponent>  
  };  
  export default App;  
  ReactDOM.render(<App />, document.getElementById("stockcharts"));  
,  
  
`ts  
import * as React from "react";  
import * as ReactDOM from "react-dom";  
import { StockChartComponent } from '@syncfusion/ej2-react-charts';  
function App() {  
  return <StockChartComponent id='stockcharts'>  
    </StockChartComponent>;  
  }  
  ;  
  export default App;  
  ReactDOM.render(<App />, document.getElementById("stockcharts"));  
,
```

### INDEX.JSX

```
{% raw %}  
import * as React from "react";  
import * as ReactDOM from "react-dom";
```

```

import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    return (<StockChartComponent id='stockchart' primaryXAxis={{
        valueType: 'DateTime',
        majorGridLines: { width: 0 }, majorTickLines: { color:
'transparent' },
        crosshairTooltip: { enable: true }
    }} primaryYAxis={{
        labelFormat: 'n0',
        lineStyle: { width: 0 }, rangePadding: 'None',
        majorTickLines: { width: 0 }
    }} title='Sales Analysis' bearFillColor='#2ecd71'
bullFillColor='#e74c3d' height='350'>
        <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,
EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator, Export,
AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
                </StockChartSeriesDirective>
            </StockChartSeriesCollectionDirective>
        </StockChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

**INDEX.TSX**

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import {
    StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, ITooltipRenderEventArgs,
IStockChartEventArgs, ChartTheme,
    DateTime, Tooltip, RangeTooltip, Crosshair, LineSeries, SplineSeries,
CandleSeries, HiloOpenCloseSeries, HiloSeries, RangeAreaSeries, Trendlines
} from '@syncfusion/ej2-react-charts';
import {
    EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,

```

```

    AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
    Export
  } from '@syncfusion/ej2-react-charts';
  import { chartData } from 'datasource.ts';
  function App() {
    return (
      <StockChartComponent id='stockchart'
        primaryXAxis={{
          valueType: 'DateTime',
          majorGridLines: { width: 0 }, majorTickLines: { color:
'transparent' },
          crosshairTooltip: { enable: true }
        }}
        primaryYAxis={{
          labelFormat: 'n0',
          lineStyle: { width: 0 }, rangePadding: 'None',
          majorTickLines: { width: 0 }
        }}
        title= 'Sales Analysis'
        bearFillColor='#2ecd71'
        bullFillColor='#e74c3d'
        height='350'
      >
        <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,
          EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator, Export,
          AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]} />
        <StockChartSeriesCollectionDirective>
          <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
            </StockChartSeriesDirective>
          </StockChartSeriesCollectionDirective>
        </StockChartComponent>
      )
    );
  };
  export default App;
  ReactDOM.render(<App />, document.getElementById("charts"));
  {% endraw %}

```

### Size for Stock Chart

You can also set size for stock chart directly through [width](#) and [height](#) properties.

<!-- markdownlint-disable MD036 -->

#### In Pixel

<!-- markdownlint-disable MD036 -->

You can set the size of chart in pixel as demonstrated below.

#### INDEX.JSX

```

{% raw %}
import * as React from "react";

```

```

import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    return (<StockChartComponent id='stockchart' primaryXAxis={{
        valueType: 'DateTime',
        majorGridLines: { width: 0 }, majorTickLines: { color:
'transparent' },
        crosshairTooltip: { enable: true }
    }} primaryYAxis={{
        labelFormat: 'n0',
        lineStyle: { width: 0 }, rangePadding: 'None',
        majorTickLines: { width: 0 }
    }} title='Sales Analysis' height='350' width='650'>
        <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,
EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator, Export,
AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
                </StockChartSeriesDirective>
            </StockChartSeriesCollectionDirective>
        </StockChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

## INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import {
    StockChartComponent, StockChartSeriesCollectionDirective,
    StockChartSeriesDirective, Inject, ITooltipRenderEventArgs,
    IStockChartEventArgs, ChartTheme,
    DateTime, Tooltip, RangeTooltip, Crosshair, LineSeries, SplineSeries,
    CandleSeries, HiloOpenCloseSeries, HiloSeries, RangeAreaSeries, Trendlines
} from '@syncfusion/ej2-react-charts';
import {
    EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
    MomentumIndicator, SmaIndicator, AtrIndicator,

```

```

    AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
    Export
  } from '@syncfusion/ej2-react-charts';
  import { chartData } from 'datasource.ts';
  function App() {
    return (
      <StockChartComponent id='stockchart'
        primaryXAxis={{
          valueType: 'DateTime',
          majorGridLines: { width: 0 }, majorTickLines: { color:
'transparent' },
          crosshairTooltip: { enable: true }
        }}
        primaryYAxis={{
          labelFormat: 'n0',
          lineStyle: { width: 0 }, rangePadding: 'None',
          majorTickLines: { width: 0 }
        }}
        title= 'Sales Analysis'
        height='350'
        width='650'
      >
        <Inject services={[DateTime, Tooltip, RangeTooltip,
        Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
        HiloSeries, RangeAreaSeries, Trendlines,
        EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
        MomentumIndicator, SmaIndicator, AtrIndicator, Export,
        AccumulationDistributionIndicator, MacdIndicator,
        StochasticIndicator]} />
        <StockChartSeriesCollectionDirective>
          <StockChartSeriesDirective dataSource={chartData}
            type='Candle' animation={{ enable: true }}>
          </StockChartSeriesDirective>
        </StockChartSeriesCollectionDirective>
      </StockChartComponent>
    );
  };
  export default App;
  ReactDOM.render(<App />, document.getElementById("charts"));
  {% endraw %}

```

### In Percentage

By setting value in percentage, Stock Chart gets its dimension with respect to its container. For example, when the height is '50%', Stock Chart renders to half of the container height.

### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';

```

```

import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    return (<StockChartComponent id='stockchart' primaryXAxis={{
        valueType: 'DateTime',
        majorGridLines: { width: 0 }, majorTickLines: { color:
'transparent' },
        crosshairTooltip: { enable: true }
    }} primaryYAxis={{
        labelFormat: 'n0',
        lineStyle: { width: 0 }, rangePadding: 'None',
        majorTickLines: { width: 0 }
    }} title='Sales Analysis' height='90%' width='80%'>
        <Inject services=[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,
        EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator, Export,
        AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator] />
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
                </StockChartSeriesDirective>
            </StockChartSeriesCollectionDirective>
        </StockChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

**INDEX.TSX**

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import {
    StockChartComponent, StockChartSeriesCollectionDirective,
    StockChartSeriesDirective, Inject, ITooltipRenderEventArgs,
    IStockChartEventArgs, ChartTheme,
    DateTime, Tooltip, RangeTooltip, Crosshair, LineSeries, SplineSeries,
    CandleSeries, HiloOpenCloseSeries, HiloSeries, RangeAreaSeries, Trendlines
} from '@syncfusion/ej2-react-charts';
import {
    EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
    MomentumIndicator, SmaIndicator, AtrIndicator,
    AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
    Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    return (

```

```

        <StockChartComponent id='stockchart'
          primaryXAxis={{
            valueType: 'DateTime',
            majorGridLines: { width: 0 }, majorTickLines: { color:
'transparent' },
            crosshairTooltip: { enable: true }
          }}
          primaryYAxis={{
            labelFormat: 'n0',
            lineStyle: { width: 0 }, rangePadding: 'None',
            majorTickLines: { width: 0 }
          }}
          title= 'Sales Analysis'
          height='90%'
          width='80%'
        >
          <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,
            EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator, Export,
            AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]} />
          <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
              </StockChartSeriesDirective>
            </StockChartSeriesCollectionDirective>
          </StockChartComponent>
        )
      };
      export default App;
      ReactDOM.render(<App />, document.getElementById("charts"));
    {% endraw %}

```

Note: When you do not specify the size, it takes 450px as the height and window size as its width.

<!-- markdownlint-disable MD036 -->

## Axis types in React Stock chart component

### DateTime axis

DateTime axis uses date time scale and displays the date time values as axis labels in the specified format. To use DateTime axis, set the [valueType](#) of axis to `DateTime`.

### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,

```

```

AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    return (<StockChartComponent id='stockchart' primaryXAxis={{
        valueType: 'DateTime',
    }} height='350'>
        <Inject services={[DateTime, Tooltip, RangeTooltip,
        Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
        HiloSeries, RangeAreaSeries, Trendlines,
        EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
        MomentumIndicator, SmaIndicator, AtrIndicator, Export,
        AccumulationDistributionIndicator, MacdIndicator,
        StochasticIndicator]}/>
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle'>
                </StockChartSeriesDirective>
            </StockChartSeriesCollectionDirective>
        </StockChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

## INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import {
    StockChartComponent, StockChartSeriesCollectionDirective,
    StockChartSeriesDirective, Inject, ITooltipRenderEventArgs,
    IStockChartEventArgs, ChartTheme,
    DateTime, Tooltip, RangeTooltip, Crosshair, LineSeries, SplineSeries,
    CandleSeries, HiloOpenCloseSeries, HiloSeries, RangeAreaSeries, Trendlines
} from '@syncfusion/ej2-react-charts';
import {
    EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
    MomentumIndicator, SmaIndicator, AtrIndicator,
    AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
    Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    return (
        <StockChartComponent id='stockchart'
            primaryXAxis={{
                valueType: 'DateTime',
            }}
            height='350'
        >
            <Inject services={[DateTime, Tooltip, RangeTooltip,
            Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
            HiloSeries, RangeAreaSeries, Trendlines,

```



```

        EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
        MomentumIndicator, SmaIndicator, AtrIndicator, Export,
        AccumulationDistributionIndicator, MacdIndicator,
        StochasticIndicator]] />
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle'>
                </StockChartSeriesDirective>
            </StockChartSeriesCollectionDirective>
        </StockChartComponent>
    )
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

Note: To use this axis, we need to inject DateTime using `<Inject services={{[DateTime]}}>` method and set the [valueType](#) of axis to `DateTime`.

#### DateTimeCategory axis

DateTimeCategory axis in the stock chart is used to display only business days. To use DateTimeCategory axis, set the [valueType](#) of axis to `DateTimeCategory`.

#### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTimeCategory, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, Logarithmic } from '@syncfusion/ej2-
react-charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { series2 } from 'datasource.ts';
function App() {
    const primaryxAxis = {
        valueType: 'DateTimeCategory',
        majorGridLines: { width: 0 },
        crosshairTooltip: { enable: true }
    };
    const crosshair = { enable: true };
    const tooltip = {
        enable: true,
        header: 'AAPL Stock Price'
    };
    return (<StockChartComponent id='stockchart'
        primaryXAxis={primaryxAxis}
        crosshair={crosshair}
        tooltip={tooltip}
        height='350'>

```

```

    <Inject services={[DateTimeCategory, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, Logarithmic,
        EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator, Export,
        AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]} />
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={series2} type='Line'
xName='x' yName='y'>
        </StockChartSeriesDirective>
      </StockChartSeriesCollectionDirective>
    </StockChartComponent>;
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

## INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import {
  StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, ITooltipRenderEventArgs,
IStockChartEventArgs, ChartTheme,
  DateTimeCategory, Tooltip, RangeTooltip, Crosshair, AxisModel,
CrosshairSettingsModel, TooltipSettingsModel, LineSeries, SplineSeries,
CandleSeries, HiloOpenCloseSeries, HiloSeries, RangeAreaSeries, Trendlines
} from '@syncfusion/ej2-react-charts';
import {
  EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
  AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
Export
} from '@syncfusion/ej2-react-charts';
import { series2 } from 'datasource.ts';
function App() {
  const primaryxAxis: AxisModel = {
    valueType: 'DateTimeCategory',
    majorGridLines: { width: 0 },
    crosshairTooltip: { enable: true }
  };
  const crosshair: CrosshairSettingsModel = { enable: true };
  const tooltip: TooltipSettingsModel = {
    enable: true,
    header: 'AAPL Stock Price'
  };
  return (
    <StockChartComponent id='stockchart'
      primaryXAxis={primaryxAxis}
      crosshair={crosshair}
      tooltip={tooltip}
      height='350'

```

```

    <Inject services={[DateTimeCategory, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,
      EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator, Export,
      AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]} />
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={series2} type='Line'
xName='x' yName='y'>
        </StockChartSeriesDirective>
      </StockChartSeriesCollectionDirective>
    </StockChartComponent>
  )
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

Note: To use this axis, we need to inject `DateTimeCategory` using `<Inject services={[DateTimeCategory]}>` method and set the `valueType` of axis to `DateTimeCategory`.

### Logarithmic axis

`<!-- markdownlint-disable MD033 -->`

Logarithmic axis uses logarithmic scale and it is very useful in visualizing data, when it has numerical values in both lower order of magnitude (eg:  $10^{-6}$ ) and higher order of magnitude (eg:  $10^6$ ). To use Logarithmic axis, set the `valueType` of axis to `Logarithmic`.

### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, Logarithmic } from '@syncfusion/ej2-react-charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  return (<StockChartComponent id='stockchart' primaryXAxis={{
    valueType: 'DateTime',
  }} primaryYAxis={{
    valueType: 'Logarithmic',
  }} height='350'>
    <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, Logarithmic,
      EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator, Export,

```

```

        AccumulationDistributionIndicator, MacdIndicator,
        StochasticIndicator]]/>
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Line' xName='date' yName='high'>
            </StockChartSeriesDirective>
        </StockChartSeriesCollectionDirective>
    </StockChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

## INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import {
    StockChartComponent, StockChartSeriesCollectionDirective,
    StockChartSeriesDirective, Inject, ITooltipRenderEventArgs,
    IStockChartEventArgs, ChartTheme,
    DateTime, Tooltip, RangeTooltip, Crosshair, LineSeries, SplineSeries,
    CandleSeries, HiloOpenCloseSeries, HiloSeries, RangeAreaSeries, Trendlines,
    Logarithmic
} from '@syncfusion/ej2-react-charts';
import {
    EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
    MomentumIndicator, SmaIndicator, AtrIndicator,
    AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
    Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    return (
        <StockChartComponent id='stockchart'
            primaryXAxis={{
                valueType: 'DateTime',
            }}
            primaryYAxis={{
                valueType: 'Logarithmic',
            }}
            height='350'
        >
            <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, Logarithmic,
EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator, Export,
AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]} />
            <StockChartSeriesCollectionDirective>
                <StockChartSeriesDirective dataSource={chartData}
type='Line' xName='date' yName='high' >
                </StockChartSeriesDirective>

```

```

        </StockChartSeriesCollectionDirective>
      </StockChartComponent>
    )
  };
  export default App;
  ReactDOM.render(<App />, document.getElementById("charts"));
  {% endraw %}

```

Note: To use this axis, we need to inject Logarithmic using `<Inject services={{[Logarithmic]}}>` method and set the `valueType` of axis to `Logarithmic`.

See also

- [Axis Customization](#)

## Axis customization in React Stock chart component

### Axis Crossing

An axis can be positioned in the Stock Chart area using `crossesAt` and `[crossesInAxis]` (<https://ej2.syncfusion.com/react/documentation/api/stock-chart/stockChartAxisModel/#crossesinaxis>) properties. The `[crossesAt]` (<https://ej2.syncfusion.com/react/documentation/api/stock-chart/stockChartAxisModel/#crossesat>) property specifies the values (datetime, numeric, or logarithmic) at which the axis line has to be intersected with the vertical axis or vice-versa, and the `[crossesInAxis]` (<https://ej2.syncfusion.com/react/documentation/api/stock-chart/stockChartAxisModel/#crossesinaxis>) property specifies the axis name with which the axis line has to be crossed.

### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
  StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
  Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
  HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
  MomentumIndicator, SmaIndicator, AtrIndicator,
  AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  return (
    <StockChartComponent id='stockchart' primaryXAxis={{
      valueType: 'DateTime',
      crossesAt: 115
    }} height='350'>
      <Inject services={[DateTime, Tooltip, RangeTooltip,
        Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
        HiloSeries, RangeAreaSeries, Trendlines,
        EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
        MomentumIndicator, SmaIndicator, AtrIndicator, Export,
        AccumulationDistributionIndicator, MacdIndicator,
        StochasticIndicator]}/>
      <StockChartSeriesCollectionDirective>

```

```

        <StockChartSeriesDirective dataSource={chartData}
type='Candle'>
        </StockChartSeriesDirective>
    </StockChartSeriesCollectionDirective>
</StockChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

## INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import {
    StockChartComponent, StockChartSeriesCollectionDirective,
    StockChartSeriesDirective, Inject, ITooltipRenderEventArgs,
    IStockChartEventArgs, ChartTheme,
    DateTime, Tooltip, RangeTooltip, Crosshair, LineSeries, SplineSeries,
    CandleSeries, HiloOpenCloseSeries, HiloSeries, RangeAreaSeries, Trendlines
} from '@syncfusion/ej2-react-charts';
import {
    EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
    MomentumIndicator, SmaIndicator, AtrIndicator,
    AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
    Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    return (
        <StockChartComponent id='stockchart'
            primaryXAxis={{
                valueType: 'DateTime',
                crossesAt: 115
            }}
            height='350'
        >
            <Inject services={[DateTime, Tooltip, RangeTooltip,
                Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
                HiloSeries, RangeAreaSeries, Trendlines,
                EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
                MomentumIndicator, SmaIndicator, AtrIndicator, Export,
                AccumulationDistributionIndicator, MacdIndicator,
                StochasticIndicator]} />
            <StockChartSeriesCollectionDirective>
                <StockChartSeriesDirective dataSource={chartData}
type='Candle' >
                </StockChartSeriesDirective>
            </StockChartSeriesCollectionDirective>
        </StockChartComponent>
    );
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

```
{% endraw %}
```

## Title

You can add a title to the axis using [title](#) property to provide quick information to the user about the data plotted in the axis. Title style can be customized using [titleStyle](#) property of the axis.

## INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    return (<StockChartComponent id='stockchart' primaryXAxis={{
        valueType: 'DateTime',
        title: 'AAPL Historical',
    }} height='350'>
        <Inject services=[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,
EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator, Export,
AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]]/>
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle'>
                </StockChartSeriesDirective>
            </StockChartSeriesCollectionDirective>
        </StockChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

## INDEX.TSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import {
    StockChartComponent, StockChartSeriesCollectionDirective,
    StockChartSeriesDirective, Inject, ITooltipRenderEventArgs,
    IStockChartEventArgs, ChartTheme,
```

```

    DateTime, Tooltip, RangeTooltip, Crosshair, LineSeries, SplineSeries,
    CandleSeries, HiloOpenCloseSeries, HiloSeries, RangeAreaSeries, Trendlines
  } from '@syncfusion/ej2-react-charts';
import {
    EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
    MomentumIndicator, SmaIndicator, AtrIndicator,
    AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
    Export
  } from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  return (
    <StockChartComponent id='stockchart'
      primaryXAxis={{
        valueType: 'DateTime',
        title: 'AAPL Historical',
      }}
      height='350'
    >
      <Inject services={[DateTime, Tooltip, RangeTooltip,
        Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
        HiloSeries, RangeAreaSeries, Trendlines,
        EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
        MomentumIndicator, SmaIndicator, AtrIndicator, Export,
        AccumulationDistributionIndicator, MacdIndicator,
        StochasticIndicator]} />
      <StockChartSeriesCollectionDirective>
        <StockChartSeriesDirective dataSource={chartData}
          type='Candle' >
          </StockChartSeriesDirective>
        </StockChartSeriesCollectionDirective>
      </StockChartComponent>
    )
  );
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

### Tick Lines Customization

You can customize the **width**, **color** and **size** of the minor and major tick lines, using [majorTickLines](#) and [minorTickLines](#) properties in the axis.

### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
  StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
  Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
  HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
  charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
  MomentumIndicator, SmaIndicator, AtrIndicator,
  AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
  } from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';

```



```
function App() {
  const primaryXAxis = {
    valueType: 'DateTime',
    majorTickLines: { color: 'blue', width: 5 },
    minorTickLines: { color: 'red', width: 0 }
  };
  const primaryYAxis = {
    majorTickLines: { color: 'green', width: 5 },
    minorTickLines: { color: 'red', width: 0 }
  };
  return (<StockChartComponent id='stockchart' primaryXAxis={primaryXAxis}
    primaryYAxis={primaryYAxis} height='350'>
    <Inject services={[DateTime, Tooltip, RangeTooltip,
    Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
    HiloSeries, RangeAreaSeries, Trendlines,
    EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
    MomentumIndicator, SmaIndicator, AtrIndicator, Export,
    AccumulationDistributionIndicator, MacdIndicator,
    StochasticIndicator]}/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle'>
        </StockChartSeriesDirective>
      </StockChartSeriesCollectionDirective>
    </StockChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

## INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import {
  StockChartComponent, StockChartSeriesCollectionDirective,
  StockChartSeriesDirective, Inject, ITooltipRenderEventArgs,
  IStockChartEventArgs, ChartTheme,
  DateTime, Tooltip, RangeTooltip, Crosshair, LineSeries, SplineSeries,
  CandleSeries, HiloOpenCloseSeries, HiloSeries, RangeAreaSeries, Trendlines
} from '@syncfusion/ej2-react-charts';
import {
  EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
  MomentumIndicator, SmaIndicator, AtrIndicator,
  AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
  Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryXAxis: AxisModel= {
    valueType: 'DateTime',
    majorTickLines : { color : 'blue', width : 5 },
    minorTickLines : {color : 'red',width : 0 }
  };
  const primaryYAxis: AxisModel= {
    majorTickLines : { color : 'green', width : 5 },
```

```

        minorTickLines : {color : 'red',width : 0 }
    };
    return (
        <StockChartComponent id='stockchart'
            primaryXAxis={ primaryXAxis }
            primaryYAxis={ primaryYAxis }
            height='350'
        >
            <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,
            EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator, Export,
            AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]} />
            <StockChartSeriesCollectionDirective>
                <StockChartSeriesDirective dataSource={chartData}
type='Candle' >
                    </StockChartSeriesDirective>
                </StockChartSeriesCollectionDirective>
            </StockChartComponent>
        )
    );
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

### Grid Lines Customization

You can customize the **width**, **color** and **dashArray** of the minor and major grid lines, using [majorGridLines](#) and [minorGridLines](#) properties in the axis.

### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    const primaryXAxis = {
        valueType: 'DateTime',
        majorGridLines: { color: 'blue', width: 5 },
        minorGridLines: { color: 'red', width: 0 }
    };
    const primaryYAxis = {
        majorGridLines: { color: 'green', width: 5 },
        minorGridLines: { color: 'red', width: 0 }
    };
    return (<StockChartComponent id='stockchart' primaryXAxis={primaryXAxis}
primaryYAxis={primaryYAxis} height='350'>

```

```

        <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,
            EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator, Export,
            AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle'>
                </StockChartSeriesDirective>
            </StockChartSeriesCollectionDirective>
        </StockChartComponent>;
    }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import {
    StockChartComponent, StockChartSeriesCollectionDirective,
    StockChartSeriesDirective, Inject, ITooltipRenderEventArgs,
    IStockChartEventArgs, ChartTheme,
    DateTime, Tooltip, RangeTooltip, Crosshair, LineSeries, SplineSeries,
    CandleSeries, HiloOpenCloseSeries, HiloSeries, RangeAreaSeries, Trendlines
} from '@syncfusion/ej2-react-charts';
import {
    EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
    MomentumIndicator, SmaIndicator, AtrIndicator,
    AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
    Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    const primaryXAxis: AxisModel = {
        valueType: 'DateTime',
        majorGridLines : { color : 'blue', width : 5 },
        minorGridLines : {color : 'red',width : 0 }
    };
    const primaryYAxis: AxisModel = {
        majorGridLines : { color : 'green', width : 5 },
        minorGridLines : {color : 'red',width : 0 }
    };
    return (
        <StockChartComponent id='stockchart'
            primaryXAxis={ primaryXAxis }
            primaryYAxis={ primaryYAxis }
            height='350'
        >
            <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,

```

```

        EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
        MomentumIndicator, SmaIndicator, AtrIndicator, Export,
        AccumulationDistributionIndicator, MacdIndicator,
        StochasticIndicator]] />
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle' >
                </StockChartSeriesDirective>
            </StockChartSeriesCollectionDirective>
        </StockChartComponent>
    )
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

### Multiple Axis

In addition to primary X and Y axis, we can add n number of axis to the Stock Chart. Series can be associated with this [axis](#) , by mapping with axis's unique name.

#### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, RangeTooltip, Crosshair,
LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries, HiloSeries,
RangeAreaSeries, Trendlines, Tooltip } from '@syncfusion/ej2-react-charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
Export, StockChartAxesDirective, StockChartAxisDirective } from
 '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    const primaryxAxis = {
        valueType: 'DateTime',
        majorGridLines: { width: 0 }
    };
    const primaryyAxis = {
        majorTickLines: { width: 0 }
    };
    return (<StockChartComponent id='stockcharts' primaryXAxis={primaryxAxis}
primaryYAxis={primaryyAxis} title='Multiple Axis' height='350'>
        <Inject services=[DateTime, RangeTooltip, Tooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]]/>
        <StockChartAxesDirective>
            <StockChartAxisDirective rowIndex={0} name='yAxis1'>
                </StockChartAxisDirective>
            </StockChartAxesDirective>
        <StockChartSeriesCollectionDirective>

```

```

        <StockChartSeriesDirective dataSource={chartData}
type='Line' xName='date' yName='high' animation={{ enable: true }}>
        </StockChartSeriesDirective>
        <StockChartSeriesDirective
dataSource={chartData} yAxisName='yAxis1' type='Line' animation={{ enable:
true }} xName='date' yName='low'>
        </StockChartSeriesDirective>
    </StockChartSeriesCollectionDirective>
</StockChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

## INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AxisModel } from '@syncfusion/ej2-react-charts';
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, RangeTooltip, Crosshair,
LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries, HiloSeries,
RangeAreaSeries, Trendlines, Tooltip, AxesDirective, AxisDirective } from
 '@syncfusion/ej2-react-charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
Export,
StockChartAxesDirective, StockChartAxisDirective } from '@syncfusion/ej2-
react-charts';
import { chartData } from 'datasource.ts';
function App() {
    const primaryxAxis: AxisModel= {
        valueType: 'DateTime',
        majorGridLines: { width: 0 }
    };
    const primaryyAxis: AxisModel= {
        majorTickLines: { width: 0 }
    };
    return( <StockChartComponent id='stockcharts'
primaryXAxis={ primaryxAxis }
primaryYAxis={ primaryyAxis }
title='Multiple Axis'
height='350'>
        <Inject services={[DateTime, RangeTooltip, Tooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
        <StockChartAxesDirective>
            <StockChartAxisDirective rowIndex={0} name='yAxis1'>
            </StockChartAxisDirective>
        </StockChartAxesDirective>
    </StockChartComponent>);
}

```

```

        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Line' xName='date' yName='high' animation={{ enable: true }}>
            </StockChartSeriesDirective>
            <StockChartSeriesDirective
dataSource={chartData} yAxisName='yAxis1' type='Line' animation={{ enable:
true }} xName='date' yName='low'>
            </StockChartSeriesDirective>
        </StockChartSeriesCollectionDirective>
    </StockChartComponent>

    );
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% enddraw %}

```

### Inversed Axis

When an axis is inversed, highest value of the axis comes closer to origin and vice versa. To place an axis in inversed manner set this property [isInversed](#) to true.

### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, RangeTooltip, Tooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    const primaryXAxis = {
        valueType: 'DateTime',
        isInversed: true
    };
    const primaryYAxis = {
        isInversed: true
    };
    return (<StockChartComponent id='stockchart' primaryXAxis={primaryXAxis}
primaryYAxis={primaryYAxis} height='350'>
        <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,
EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator, Export,
AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle'>
            </StockChartSeriesDirective>
        </StockChartSeriesCollectionDirective>
    </StockChartComponent>

```

```

        </StockChartComponent>);
    }
    ;
    export default App;
    ReactDOM.render(<App />, document.getElementById("charts"));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AxisModel } from '@syncfusion/ej2-react-charts';
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, RangeTooltip, Tooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    const primaryXAxis: AxisModel = {
        valueType: 'DateTime',
        isInversed: true
    };
    const primaryYAxis: AxisModel = {
        isInversed: true
    };
    return (
        <StockChartComponent id='stockchart'
            primaryXAxis={ primaryXAxis }
            primaryYAxis={ primaryYAxis }
            height='350'
        >
            <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,
EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator, Export,
AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]} />
            <StockChartSeriesCollectionDirective>
                <StockChartSeriesDirective dataSource={chartData}
type='Candle' >
                    </StockChartSeriesDirective>
                </StockChartSeriesCollectionDirective>
            </StockChartComponent>
        )
    );
    export default App;
    ReactDOM.render(<App />, document.getElementById("charts"));

```

## Opposed Position

To place an axis opposite from its original position, set [opposedPosition](#) property of the axis to true.

**INDEX.JSX**

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, RangeTooltip, Tooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    const primaryXAxis = {
        valueType: 'DateTime',
        opposedPosition: true
    };
    return (<StockChartComponent id='stockchart' primaryXAxis={primaryXAxis}
height='350'>
        <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,
            EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator, Export,
            AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle'>
                </StockChartSeriesDirective>
            </StockChartSeriesCollectionDirective>
        </StockChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

**INDEX.TSX**

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AxisModel } from '@syncfusion/ej2-react-charts';
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, RangeTooltip, Tooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    const primaryXAxis: AxisModel= {

```



```

        valueType: 'DateTime',
        opposedPosition: true
    };

    return (
        <StockChartComponent id='stockchart'
            primaryXAxis={ primaryXAxis }
            height='350'
        >
            <Inject services={[DateTime, Tooltip, RangeTooltip,
                Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
                HiloSeries, RangeAreaSeries, Trendlines,
                EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
                MomentumIndicator, SmaIndicator, AtrIndicator, Export,
                AccumulationDistributionIndicator, MacdIndicator,
                StochasticIndicator]} />
            <StockChartSeriesCollectionDirective>
                <StockChartSeriesDirective dataSource={chartData}
type='Candle' >
                    </StockChartSeriesDirective>
                </StockChartSeriesCollectionDirective>
            </StockChartComponent>
        )
    );
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

### Series types in React Stock chart component

Essential JS 2 StockChart supports 6 major types of series namely **Line**, **Spline**, **Hilo**, **HiloOpenClose**, **Hollow Candle** and **Candle** . By using the series dropdown button you can navigate between the above listed series types.

<!-- markdownlint-disable MD036 -->

#### Line

To render a line series, use series **type** as **Line** and inject **LineSeries** module using **<Inject services={[LineSeries]}>** method.

#### Spline

To render a spline series, use series **type** as **Spline** and inject **SplineSeries** module using **<Inject services={[SplineSeries]}>** method.

#### Hilo

To render a hilo series, use series **type** as **Hilo** and inject **HiloSeries** module using **<Inject services={[HiloSeries]}>** method.

#### HiloOpenClose

To render a hiloOpenClose series, use series **type** as **HiloOpenClose** and inject **HiloOpenCloseSeries** module using **<Inject services={[HiloOpenCloseSeries]}>** method.

#### HollowCandle

To render a hollowcandle series, use series **type** as **Candle** and set **enableSolidCandle** as false. Now inject **CandleSeries** module using **<Inject services={[CandleSeries]}>** method.

## Candle

To render a candle series, use series [type](#) as `Candle` and inject `CandleSeries` module using `<Inject services=[[CandleSeries]]>` method.

### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    return (<StockChartComponent id='stockchart' primaryXAxis={{
        valueType: 'DateTime',
    }} height='350' indicatorType=[] trendlineType=[]
exportType=[]>
        <Inject services=[[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,
EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator, Export,
AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]]/>
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle'>
            </StockChartSeriesDirective>
        </StockChartSeriesCollectionDirective>
    </StockChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

### INDEX.TSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import {
    StockChartComponent, StockChartSeriesCollectionDirective,
    StockChartSeriesDirective, Inject, ITooltipRenderEventArgs,
    IStockChartEventArgs, ChartTheme,
    DateTime, Tooltip, RangeTooltip, Crosshair, LineSeries, SplineSeries,
    CandleSeries, HiloOpenCloseSeries, HiloSeries, RangeAreaSeries, Trendlines
} from '@syncfusion/ej2-react-charts';
import {
```

```

    EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
    MomentumIndicator, SmaIndicator, AtrIndicator,
    AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
    Export
  } from '@syncfusion/ej2-react-charts';
  import { chartData } from 'datasource.ts';
  function App() {
    return (
      <StockChartComponent id='stockchart'
        primaryXAxis={{
          valueType: 'DateTime',
        }}
        height='350'
        indicatorType={}
        trendlineType={}
        exportType={}
      >
        <Inject services=[[DateTime, Tooltip, RangeTooltip,
        Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
        HiloSeries, RangeAreaSeries, Trendlines,
        EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
        MomentumIndicator, SmaIndicator, AtrIndicator, Export,
        AccumulationDistributionIndicator, MacdIndicator,
        StochasticIndicator]] />
        <StockChartSeriesCollectionDirective>
          <StockChartSeriesDirective dataSource={chartData}
            type='Candle'>
            </StockChartSeriesDirective>
          </StockChartSeriesCollectionDirective>
        </StockChartComponent>
      )
    );
  };
  export default App;
  ReactDOM.render(<App />, document.getElementById("charts"));
  {% endraw %}

```

<!-- markdownlint-disable MD036 -->

## Trend lines in React Stock chart component

Trendlines are used to show the direction and speed of price.

StockChart supports 6 types of trendlines namely

Linear, Exponential, Logarithmic, Polynomial, Power, Moving Average. By using trendline dropdown button you can add or remove the required trendline type.

### Linear

A linear trendline is a best fit straight line that is used with simpler data sets. To render a linear trendline, use trendline [type](#) as [Linear](#) and inject [Trendlines](#) module using `<Inject services=[[Trendlines]]>`.

### Exponential

An exponential trendline is a curved line that is most useful when data values rise or fall at increasingly higher rates. You cannot create an exponential trendline, if your data contains zero or negative values.

To render an exponential trendline, use trendline [type](#) as **Exponential** and inject **Trendlines** module using `<Inject services={{Trendlines}}>`.

### Logarithmic

A logarithmic trendline is a best-fit curved line that is most useful when the rate of change in the data increases or decreases quickly and then levels out. A logarithmic trendline can use negative and/or positive values.

To render a logarithmic trendline, use trendline [type](#) as **Logarithmic** and inject **Trendlines** module using `<Inject services={{Trendlines}}>`.

### Polynomial

A polynomial trendline is a curved line that is used when data fluctuates.

To render a polynomial trendline, use trendline [type](#) as **Polynomial** and inject **Trendlines** module using `<Inject services={{Trendlines}}>`.

**polynomialOrder** used to define the polynomial value.

### Power

A power trendline is a curved line that is best used with data sets that compare measurements that increase at a specific rate.

To render a power trendline, use trendline [type](#) as **Power** and inject **Trendlines** module using `<Inject services={{Trendlines}}>`.

### Moving Average

A moving average trendline smoothen out fluctuations in data to show a pattern or trend more clearly.

To render a moving average trendline, use trendline [type](#) as **MovingAverage** and inject **Trendlines** module using `<Inject services={{Trendlines}}>`.

**period** property defines the period to find the moving average.

## INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  return (
    <StockChartComponent id='stockchart' primaryXAxis={{
      title: 'Months'
    }} primaryYAxis={{
      title: 'Rupees against Dollars',
      interval: 5
    }} tooltip={{ enable: true }} height='350' title='Historical Indian
Rupee Rate (INR USD)' indicatorType={[]} exportType={[]} seriesType={[]}>
      <Inject services={{DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines}}/>
    </StockChartComponent>
  );
}
```

```

        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle'>
                </StockChartSeriesDirective>
            </StockChartSeriesCollectionDirective>
        </StockChartComponent>);
    }
    ;
    export default App;
    ReactDOM.render(<App />, document.getElementById("charts"));
    {% endraw %}

```

## INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import {
    StockChartComponent, StockChartSeriesCollectionDirective,
    StockChartSeriesDirective, Inject,
    DateTime, Tooltip, RangeTooltip, Crosshair, LineSeries, SplineSeries,
    CandleSeries, HiloOpenCloseSeries, HiloSeries, RangeAreaSeries, Trendlines
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    return (
        <StockChartComponent id='stockchart'
            primaryXAxis={{
                title: 'Months'
            }}
            primaryYAxis={{
                title: 'Rupees against Dollars',
                interval: 5
            }}
            tooltip={{enable:true}}
            height='350'
            title='Historical Indian Rupee Rate (INR USD)'
            indicatorType={[]}
            exportType={[]}
            seriesType={[]}
        >
            <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines]} />
            <StockChartSeriesCollectionDirective>
                <StockChartSeriesDirective dataSource={chartData}
type='Candle'>
                    </StockChartSeriesDirective>
                </StockChartSeriesCollectionDirective>
            </StockChartComponent>
        );
    };
    export default App;
    ReactDOM.render(<App />, document.getElementById("charts"));
    {% endraw %}

```

<!-- markdownlint-disable MD036 -->

## Technical indicators in React Stock chart component

A technical indicator is a mathematical calculation based on historic price, volume or open interest information that aims to forecast financial market direction.

StockChart supports 10 types of technical indicators namely Accumulation Distribution, ATR, EMA, SMA, TMA, Momentum, MACD, RSI, Stochastic, Bollinger Band. By using indicator dropdown box you can add and remove the required indicators types.

### Accumulation Distribution

Accumulation Distribution combines price and volume to show how money may be flowing into or out of stock. To render a Accumulation Distribution Indicator, use indicator [type](#) as AccumulationDistribution and inject AccumulationDistributionIndicator module using `<Inject services={[AccumulationDistributionIndicator]}>`. To calculate the signal line [volume](#) field is additionally added with dataSource.

### Average True Range (ATR)

ATR measures the stock volatility by comparing the current value with the previous value. To render a Average True Range (ATR) Indicator, use indicator [type](#) as Atr and inject AtrIndicator module using `<Inject services={[AtrIndicator]}>`.

### Exponential Moving Average (EMA)

Moving average Indicators are used to define the direction of the trend. To render a EMA Indicator, use indicator [type](#) as Ema and inject EmaIndicator module using `<Inject services={[EmaIndicator]}>`.

### Momentum

Momentum shows the speed at which the price of the stock is changing. To render a Momentum indicator, use indicator [type](#) as Momentum and inject MomentumIndicator module using `<Inject services={[MomentumIndicator]}>` method. Momentum indicator will be represented by two lines (upperLine, signalLine). In momentum indicator the upperBand value is always renders at the value 100.

### Moving Average Convergence Divergence (MACD)

MACD is based on the difference between two EMA's. To render a MACD Indicator, use indicator [type](#) as Macd and inject MacdIndicator module using `<Inject services={[MacdIndicator]}>`. MACD indicator will be represented by MACD line, signal line, MACD histogram. MACD histogram is used to differentiate MACD line and signal line.

### Relative Strength Index (RSI)

RSI shows how strongly a stock is moving in its current direction. To render a RSI Indicator, use indicator [type](#) as Rsi and inject RsiIndicator module using `<Inject services={[Rsindicator]}>`. RSI indicator will be represented by three lines (upperBand, lowerBand, signalLine). The upperBand and lowerBand values are customized by [overBought](#) and [overSold](#) properties of indicator and the signalLine is calculated by RSI formula.

### Simple Moving Average (SMA)

Moving average Indicators are used to define the direction of the trend. To render a SMA Indicator, use indicator [type](#) as Sma and inject SmaIndicator module using `<Inject services={[SmaIndicator]}>`.

### Stochastic

It shows how a stock is, when compared to previous state. To render a Stochastic indicator, use indicator [type](#) as `Stochastic` and inject `StochasticIndicator` module using `Chart.Inject(StochasticIndicator)` method. stochastic indicator will be represented by four lines (upperLine, lowerLine, periodLine, signalLine).

In stochastic indicator the upperBand value and lowerBand value is customized by [overBought](#) and [overSold](#) properties of indicators and the periodLine and signalLine is render based on stochastic formula.

### Triangular Moving Average (TMA)

Moving average Indicators are used to define the direction of the trend. To render a TMA Indicator, use indicator [type](#) as `Tma` and inject `TmaIndicator` module using `<Inject services=[[TmaIndicator]]>`.

### Bollinger Band

A StockChart overlay that shows the upper and lower limits of normal price movements based on the standard deviation of prices. To render a Bollinger Band, use indicator [type](#) as `BollingerBand` and inject `BollingerBands` module using `<Inject services=[[BollingerBands]]>` method. Bollinger band will be represented by three lines (upperLine, lowerLine, signalLine). The default values of the Bollinger Band [period](#) is 14 and [standardDeviations](#) is 2.

### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, CandleSeries, StockChartIndicatorsDirective,
StockChartIndicatorDirective, LineSeries, SplineSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries } from '@syncfusion/ej2-react-charts';
import { BollingerBands } from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const upperline = { color: '#ffb735', width: 1 };
  const lowerline = { color: '#f2ec2f', width: 1 };
  return (<StockChartComponent id='stockchart' primaryXAxis={{
    majorGridLines: { color: 'transparent' }, crosshairTooltip: {
enable: true }
    }} primaryYAxis={{
    lineStyle: { color: 'transparent' },
    majorTickLines: { color: 'transparent', width: 0 }
    }} crosshair={{ enable: true }} tooltip={{ enable: true }}
seriesType={[]} exportType={[]} trendlineType={[]} height='350'>
    <Inject services=[[DateTime, Tooltip, RangeTooltip,
Crosshair, CandleSeries, BollingerBands,
LineSeries, SplineSeries, HiloOpenCloseSeries, HiloSeries,
RangeAreaSeries]]/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle' name='Apple Inc' bearFillColor='#00226C'
bullFillColor='#0450C2' fill='blue'>
      </StockChartSeriesDirective>
    </StockChartSeriesCollectionDirective>
    <StockChartIndicatorsDirective>
```

```

        <StockChartIndicatorDirective type='BollingerBands'
field='Close' seriesName='Apple Inc' fill='#606eff' xName='date' high='high'
low='low' open='open' close='close' period={10} upperLine={upperline}
lowerLine={lowerline}>
        </StockChartIndicatorDirective>
    </StockChartIndicatorsDirective>
</StockChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

## INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import {
    StockChartComponent, StockChartSeriesCollectionDirective,
    StockChartSeriesDirective, Inject,
    DateTime, Tooltip, RangeTooltip, Crosshair, CandleSeries, Trendlines,
    StockChartIndicatorsDirective, StockChartIndicatorDirective, LineSeries,
    SplineSeries, HiloOpenCloseSeries, HiloSeries, RangeAreaSeries
} from '@syncfusion/ej2-react-charts';
import { BollingerBands } from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    const upperline: Object = { color: '#ffb735', width: 1 };
    const lowerline: Object = { color: '#f2ec2f', width: 1 };
    return (
        <StockChartComponent id='stockchart'
            primaryXAxis={{
                majorGridLines: { color: 'transparent' },
crosshairTooltip: { enable: true }
            }}
            primaryYAxis={{
                lineStyle: { color: 'transparent' },
                majorTickLines: { color: 'transparent', width: 0 }
            }}
            crosshair={{ enable: true }}
            tooltip={{ enable: true }}
            seriesType=[]
            exportType=[]
            trendlineType=[]
            height='350'
        >
            <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, CandleSeries, BollingerBands,
                LineSeries, SplineSeries, HiloOpenCloseSeries, HiloSeries,
RangeAreaSeries]} />
            <StockChartSeriesCollectionDirective>
                <StockChartSeriesDirective dataSource={chartData}
type='Candle' name='Apple Inc'
                    bearFillColor='#00226C' bullFillColor="#0450C2"
fill='blue'>

```



```

        </StockChartSeriesDirective>
      </StockChartSeriesCollectionDirective>
      <StockChartIndicatorsDirective>
        <StockChartIndicatorDirective type='BollingerBands'
field='Close' seriesName='Apple Inc' fill='#606eff' xName='date' high='high'
low='low' open='open' close='close' period={10} upperLine={upperline}
lowerLine={lowerline}>
        </StockChartIndicatorDirective>
      </StockChartIndicatorsDirective >
    </StockChartComponent>
  )
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

### Tool tip in React Stock chart component

<!-- markdownlint-disable MD036 -->

StockChart will display details about the points through tooltip, when the mouse is moved over the point.

#### Default tooltip

By default, tooltip is not visible. Enable the tooltip by setting [enable](#) property to true and by injecting [Tooltip](#) module using `<Inject services=[[Tooltip]]>`.

#### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryXAxis = {
    valueType: 'DateTime',
  };
  const crosshair = { enable: true };
  const tooltip = { enable: true };
  return <StockChartComponent id='stockcharts' primaryXAxis={primaryXAxis}
crosshair={crosshair} tooltip={tooltip} height='350' title='AAPL Stock
Price'>
    <Inject services=[[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]]/>
    <StockChartSeriesCollectionDirective>

```

```

                                <StockChartSeriesDirective dataSource={chartData}
type='Candle'>
                                </StockChartSeriesDirective>
                                </StockChartSeriesCollectionDirective>
                                </StockChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AxisModel, CrosshairSettingsModel, TooltipSettingsModel }
from '@syncfusion/ej2-react-charts';
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    const primaryXAxis: AxisModel = {
        valueType: 'DateTime',
    };
    const crosshair: CrosshairSettingsModel = { enable: true };
    const tooltip: TooltipSettingsModel = { enable: true };
    return <StockChartComponent id='stockcharts'
        primaryXAxis={primaryXAxis}
        crosshair={crosshair}
        tooltip={tooltip}
        height='350'
        title='AAPL Stock Price'>
        <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle'>
            </StockChartSeriesDirective>
        </StockChartSeriesCollectionDirective>
    </StockChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

### Format the tooltip

By default, tooltip shows information of x and y value in points. In addition to that, you can show more information in tooltip. For example the format `${series.name} ${point.x}` shows series name and point x value.

#### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryxAxis = {
    valueType: 'DateTime',
    majorGridLines: { width: 0 },
    majorTickLines: { color: 'transparent' }
  };
  const primaryyAxis = {
    labelFormat: 'n0',
    majorTickLines: { width: 0 }
  };
  const crosshair = { enable: true };
  const tooltip = { enable: true,
    header: 'Stock price',
    format: '<b>${point.x} : ${point.y}</b>' };
  return <StockChartComponent id='stockcharts' primaryXAxis={primaryxAxis}
primaryYAxis={primaryyAxis} crosshair={crosshair} tooltip={tooltip}
height='350' title='AAPL Stock Price'>
    <Inject services=[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator] />
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
      </StockChartSeriesDirective>
    </StockChartSeriesCollectionDirective>
  </StockChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

#### INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AxisModel, CrosshairSettingsModel, TooltipSettingsModel }
from '@syncfusion/ej2-react-charts';
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryxAxis: AxisModel= {
    valueType: 'DateTime',
    majorGridLines: { width: 0 },
    majorTickLines: { color: 'transparent' }
  };
  const primaryyAxis: AxisModel= {
    labelFormat: 'n0',
    majorTickLines: { width: 0 }
  };
  const crosshair: CrosshairSettingsModel ={ enable: true } ;
  const tooltip: TooltipSettingsModel ={ enable: true,
    header: 'Stock price',
    format: '<b>${point.x} : ${point.y}</b>' } ;
  return <StockChartComponent id='stockcharts'
    primaryXAxis={ primaryxAxis }
    primaryYAxis={ primaryyAxis }
    crosshair={crosshair}
    tooltip={tooltip}
    height='350'
    title='AAPL Stock Price'>
    <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
      </StockChartSeriesDirective>
    </StockChartSeriesCollectionDirective>
  </StockChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

### Position the tooltip

By default, the tooltip is positioned at the left side of the stock chart. You can move the tooltip along with the mouse by setting **Nearest** to the [position](#) property.

### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryxAxis = {
    valueType: 'DateTime',
    majorGridLines: { width: 0 },
    majorTickLines: { color: 'transparent' }
  };
  const primaryyAxis = {
    labelFormat: 'n0',
    majorTickLines: { width: 0 }
  };
  const crosshair = { enable: true };
  const tooltip = {
    enable: true,
    shared: true,
    position: 'Nearest'
  };
  return <StockChartComponent id='stockcharts' primaryXAxis={primaryxAxis}
primaryYAxis={primaryyAxis} crosshair={crosshair} tooltip={tooltip}
height='350' title='AAPL Stock Price'>
    <Inject services={[DateTime, Tooltip, RangeTooltip, Crosshair,
LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries, HiloSeries,
RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator, BollingerBands,
TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator, Export,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator]} />
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData} type='Candle'
animation={{ enable: true }}>
    </StockChartSeriesDirective>
    </StockChartSeriesCollectionDirective>
  </StockChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

### INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AxisModel, CrosshairSettingsModel, TooltipSettingsModel }
    from '@syncfusion/ej2-react-charts';
import { StockChartComponent, StockChartSeriesCollectionDirective,
    StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
    Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
    HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
    MomentumIndicator, SmaIndicator, AtrIndicator,
    AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    const primaryxAxis: AxisModel = {
        valueType: 'DateTime',
        majorGridLines: { width: 0 },
        majorTickLines: { color: 'transparent' }
    };
    const primaryyAxis: AxisModel = {
        labelFormat: 'n0',
        majorTickLines: { width: 0 }
    };
    const crosshair: CrosshairSettingsModel = { enable: true };
    const tooltip: TooltipSettingsModel = {
        enable: true,
        shared: true,
        position: 'Nearest'
    };
    return <StockChartComponent id='stockcharts'
        primaryXAxis={primaryxAxis}
        primaryYAxis={primaryyAxis}
        crosshair={crosshair}
        tooltip={tooltip}
        height='350'
        title='AAPL Stock Price'>
        <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]} />
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
            </StockChartSeriesDirective>
        </StockChartSeriesCollectionDirective>
    </StockChartComponent>
    </>
    export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

### Customize the appearance of the tooltip

The [fill](#) and [border](#) properties are used to customize the background color and border of the tooltip respectively. The [textStyle](#) property in the tooltip is used to customize the font of the tooltip text.

#### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryxAxis = {
    valueType: 'DateTime',
    majorGridLines: { width: 0 },
    majorTickLines: { color: 'transparent' }
  };
  const primaryyAxis = {
    labelFormat: 'n0',
    majorTickLines: { width: 0 }
  };
  const crosshair = { enable: true };
  const tooltip = {
    enable: true,
    format: ' ${point.x} : ${point.y}',
    fill: '#7bb4eb',
    border: { width: 2, color: 'grey' }
  };
  return <StockChartComponent id='stockcharts' primaryXAxis={primaryxAxis}
primaryYAxis={primaryyAxis} crosshair={crosshair} tooltip={tooltip}
height='350' title='AAPL Stock Price'>
    <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
xName='date' type='Candle' animation={{ enable: true }}>
        </StockChartSeriesDirective>
      </StockChartSeriesCollectionDirective>
    </StockChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

**INDEX.TSX**

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AxisModel, CrosshairSettingsModel, TooltipSettingsModel }
from '@syncfusion/ej2-react-charts';
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    const primaryxAxis: AxisModel= {
        valueType: 'DateTime',
        majorGridLines: { width: 0 },
        majorTickLines: { color: 'transparent' }
    };
    const primaryyAxis: AxisModel= {
        labelFormat: 'n0',
        majorTickLines: { width: 0 }
    };
    const crosshair: CrosshairSettingsModel = { enable: true };
    const tooltip: TooltipSettingsModel = {
        enable: true,
        format: ' ${point.x} : ${point.y}',
        fill: '#7bb4eb',
        border: {width: 2, color: 'grey'}
    };
    return <StockChartComponent id='stockcharts'
        primaryXAxis={ primaryxAxis }
        primaryYAxis={ primaryyAxis }
        crosshair={crosshair}
        tooltip={tooltip}
        height='350'
        title='AAPL Stock Price'>
        <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
xName='date' type='Candle' animation={{ enable: true }}>
                </StockChartSeriesDirective>
            </StockChartSeriesCollectionDirective>
        </StockChartComponent>
    </return>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```



## Cross hair in React Stock chart component

Crosshair has a vertical and horizontal line to view the value of the axis at mouse or touch position.

Crosshair lines can be enabled by using [enable](#) property in the `crosshair`.

### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryxAxis = {
    valueType: 'DateTime',
    majorGridLines: { width: 0 },
    majorTickLines: { color: 'transparent' }
  };
  const primaryyAxis = {
    labelFormat: 'n0',
    majorTickLines: { width: 0 }
  };
  const crosshair = { enable: true };
  return <StockChartComponent id='stockcharts' primaryXAxis={primaryxAxis}
primaryYAxis={primaryyAxis} crosshair={crosshair} height='350' title='AAPL
Stock Price'>
    <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
      </StockChartSeriesDirective>
    </StockChartSeriesCollectionDirective>
  </StockChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

### INDEX.TSX

```
{% raw %}
```

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AxisModel, CrosshairSettingsModel, FontModel }
  from '@syncfusion/ej2-react-charts';
import { StockChartComponent, StockChartSeriesCollectionDirective,
  StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
  Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
  HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
  charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
  MomentumIndicator, SmaIndicator, AtrIndicator,
  AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryxAxis: AxisModel= {
    valueType: 'DateTime',
    majorGridLines: { width: 0 },
    majorTickLines: { color: 'transparent' }
  };
  const primaryyAxis: AxisModel= {
    labelFormat: 'n0',
    majorTickLines: { width: 0 }
  };
  const crosshair: CrosshairSettingsModel = { enable: true };
  return <StockChartComponent id='stockcharts'
    primaryXAxis={ primaryxAxis }
    primaryYAxis={ primaryyAxis }
    crosshair={crosshair}
    height='350'
    title='AAPL Stock Price'>
    <Inject services={[DateTime, Tooltip, RangeTooltip,
  Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
  HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
  BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
  Export, AccumulationDistributionIndicator, MacdIndicator,
  StochasticIndicator]}/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
      </StockChartSeriesDirective>
    </StockChartSeriesCollectionDirective>
  </StockChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

### Tooltip for axis

Tooltip label for an axis can be enabled by using [enable](#) property of `crosshairTooltip` in the corresponding axis.

### INDEX.JSX

```

{% raw %}
import * as React from "react";

```

```

import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryxAxis = {
    valueType: 'DateTime',
    majorGridLines: { width: 0 },
    majorTickLines: { color: 'transparent' },
    crosshairTooltip: { enable: true }
  };
  const primaryyAxis = {
    majorTickLines: { width: 0 },
    crosshairTooltip: { enable: true }
  };
  const crosshair = { enable: true };
  return <StockChartComponent id='stockcharts' primaryXAxis={primaryxAxis}
primaryYAxis={primaryyAxis} crosshair={crosshair} height='350' title='AAPL
Stock Price'>
    <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
      </StockChartSeriesDirective>
    </StockChartSeriesCollectionDirective>
  </StockChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

## INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AxisModel, CrosshairSettingsModel, FontModel }
from '@syncfusion/ej2-react-charts';
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';

```

```

import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryxAxis: AxisModel= {
    valueType: 'DateTime',
    majorGridLines: { width: 0 },
    majorTickLines: { color: 'transparent' },
    crosshairTooltip: { enable: true }
  };
  const primaryyAxis: AxisModel= {
    majorTickLines: { width: 0 },
    crosshairTooltip: { enable: true }
  };
  const crosshair: CrosshairSettingsModel = { enable: true };
  return <StockChartComponent id='stockcharts'
    primaryXAxis={ primaryxAxis }
    primaryYAxis={ primaryyAxis }
    crosshair={crosshair}
    height='350'
    title='AAPL Stock Price'>
    <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
      </StockChartSeriesDirective>
    </StockChartSeriesCollectionDirective>
  </StockChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

### Customization

The [fill](#) and [textStyle](#) property of the [crosshairTooltip](#) is used to customize the background color and font style of the crosshair label respectively. Color and width of the crosshair line can be customized by using the [line](#) property in the crosshair.

### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';

```

```

import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryxAxis = {
    valueType: 'DateTime',
    majorGridLines: { width: 0 },
    majorTickLines: { color: 'transparent' },
    crosshairTooltip: { enable: true }
  };
  const primaryyAxis = {
    labelFormat: 'n0',
    majorTickLines: { width: 0 },
    crosshairTooltip: { enable: true }
  };
  const crosshair = { enable: true, line: { width: 2, color: 'green' } };
  return <StockChartComponent id='stockcharts' primaryXAxis={primaryxAxis}
primaryYAxis={primaryyAxis} crosshair={crosshair} height='350' title='AAPL
Stock Price'>
    <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
        </StockChartSeriesDirective>
      </StockChartSeriesCollectionDirective>
    </StockChartComponent>;
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

## INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AxisModel, CrosshairSettingsModel, FontModel }
from '@syncfusion/ej2-react-charts';
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';

```

```
function App() {
  const primaryxAxis: AxisModel= {
    valueType: 'DateTime',
    majorGridLines: { width: 0 },
    majorTickLines: { color: 'transparent' },
    crosshairTooltip: { enable: true }
  };
  const primaryyAxis: AxisModel= {
    labelFormat: 'n0',
    majorTickLines: { width: 0 },
    crosshairTooltip: { enable: true }
  };
  const crosshair: CrosshairSettingsModel = { enable: true, line: { width:
2, color: 'green' } };
  return <StockChartComponent id='stockcharts'
    primaryXAxis={ primaryxAxis }
    primaryYAxis={ primaryyAxis }
    crosshair={crosshair}
    height='350'
    title='AAPL Stock Price'>
    <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
        </StockChartSeriesDirective>
      </StockChartSeriesCollectionDirective>
    </StockChartComponent>
  </>
  export default App;
  ReactDOM.render(<App />, document.getElementById("charts"));
  {% endraw %}
```

Note: To use crosshair feature, we need to inject **Crosshair** module `<Inject services={[Crosshair]}>` method.

### Legend in React Stock chart component

Legend provides information about the series rendered in the Stock Chart. Legend can be added to a Stock Chart by enabling the [visible](#) option in the [legendSettings](#).

#### Position and Alignment

By using the [position](#) property, legend can be placed at **Left**, **Right**, **Top**, **Bottom** or **Custom** of the Stock Chart. The legend is positioned at the bottom of the Stock Chart, by default.

#### INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
```

```
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
Export, StockLegend } from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryXAxis = {
    valueType: 'DateTime',
  };
  const crosshair = { enable: true };
  const tooltip = { enable: true };
  const legendSettings = { visible: true, position: 'Top' };
  return <StockChartComponent id='stockcharts' primaryXAxis={primaryXAxis}
crosshair={crosshair} tooltip={tooltip} legendSettings={legendSettings}
indicatorType={[]} trendlineType={[]} height='350' title='AAPL Stock Price'>
    <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, StockLegend, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle' name='AAPL'>
        </StockChartSeriesDirective>
      </StockChartSeriesCollectionDirective>
    </StockChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

**INDEX.TSX**

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AxisModel, CrosshairSettingsModel, TooltipSettingsModel }
from '@syncfusion/ej2-react-charts';
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
Export, StockLegend } from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryXAxis: AxisModel= {
    valueType: 'DateTime',
  };
  const crosshair: CrosshairSettingsModel = { enable: true };
  const tooltip: TooltipSettingsModel = { enable: true };

```

```

const legendSettings: LegendSettingsModel = { visible: true, position:
'Top' };

return <StockChartComponent id='stockcharts'
  primaryXAxis={ primaryXAxis }
  crosshair={crosshair}
  tooltip={tooltip}
  legendSettings={legendSettings}
  indicatorType={[]}
  trendlineType ={[]}
  height='350'
  title='AAPL Stock Price'>
  <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines,EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, StockLegend, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle' name='AAPL'>
        </StockChartSeriesDirective>
      </StockChartSeriesCollectionDirective>
    </StockChartComponent>
  </>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

[Custom](#) position is used to position the legend anywhere in the Stock Chart using x, y coordinates.

#### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
Export, StockLegend } from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryXAxis = {
    valueType: 'DateTime',
  };
  const crosshair = { enable: true };
  const tooltip = { enable: true };
  const legendSettings = { visible: true, position: 'Custom', location: {
x: 200, y: 20 } };
  return <StockChartComponent id='stockcharts' primaryXAxis={primaryXAxis}
crosshair={crosshair} tooltip={tooltip} legendSettings={legendSettings}
indicatorType={[]} trendlineType={[]} height='350' title='AAPL Stock Price'>
    <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,

```



```
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, StockLegend, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]]/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle' name='AAPL'>
    </StockChartSeriesDirective>
  </StockChartSeriesCollectionDirective>
</StockChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

## INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AxisModel, CrosshairSettingsModel, TooltipSettingsModel }
from '@syncfusion/ej2-react-charts';
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
Export, StockLegend } from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryXAxis: AxisModel= {
    valueType: 'DateTime',
  };
  const crosshair: CrosshairSettingsModel = { enable: true };
  const tooltip: TooltipSettingsModel = { enable: true };
  const legendSettings: LegendSettingsModel = { visible: true,
position: 'Custom', location: { x: 200, y: 20 } };
  return <StockChartComponent id='stockcharts'
    primaryXAxis={ primaryXAxis }
    crosshair={crosshair}
    tooltip={tooltip}
    legendSettings={legendSettings}
    indicatorType={[]}
    trendlineType ={}
    height='350'
    title='AAPL Stock Price'>
    <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, StockLegend, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]]/>
    <StockChartSeriesCollectionDirective>
```

```

                                <StockChartSeriesDirective dataSource={chartData}
type='Candle' name='AAPL'>
                                </StockChartSeriesDirective>
                                </StockChartSeriesCollectionDirective>
                                </StockChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

<!-- markdownlint-disable MD036 -->

### Legend Alignment

<!-- markdownlint-disable MD036 -->

The legend can be align as **Center**, **Far** or **Near** to the Stock Chart using [alignment](#) property.

### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
Export, StockLegend } from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    const primaryXAxis = {
        valueType: 'DateTime',
    };
    const crosshair = { enable: true };
    const tooltip = { enable: true };
    const legendSettings = { visible: true, position: 'Bottom', alignment:
'Near' };
    return <StockChartComponent id='stockcharts' primaryXAxis={primaryXAxis}
crosshair={crosshair} tooltip={tooltip} legendSettings={legendSettings}
indicatorType={[]} trendlineType={[]} height='350' title='AAPL Stock Price'>
        <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, StockLegend, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle' name='AAPL'>
            </StockChartSeriesDirective>
        </StockChartSeriesCollectionDirective>
    </StockChartComponent>;
}
;
export default App;

```

```
ReactDOM.render(<App />, document.getElementById("charts"));
```

## INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AxisModel, CrosshairSettingsModel, TooltipSettingsModel }
  from '@syncfusion/ej2-react-charts';
import { StockChartComponent, StockChartSeriesCollectionDirective,
  StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
  Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
  HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
  charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
  MomentumIndicator, SmaIndicator, AtrIndicator,
  AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
  Export, StockLegend } from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryXAxis: AxisModel = {
    valueType: 'DateTime',
  };
  const crosshair: CrosshairSettingsModel = { enable: true };
  const tooltip: TooltipSettingsModel = { enable: true };
  const legendSettings: LegendSettingsModel = { visible: true,
    position: 'Bottom', alignment: 'Near' };
  return <StockChartComponent id='stockcharts'
    primaryXAxis={ primaryXAxis }
    crosshair={crosshair}
    tooltip={tooltip}
    legendSettings={legendSettings}
    indicatorType={[]}
    trendlineType ={[]}
    height='350'
    title='AAPL Stock Price'>
    <Inject services={[DateTime, Tooltip, RangeTooltip,
    Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
    HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
    BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
    Export, StockLegend, AccumulationDistributionIndicator, MacdIndicator,
    StochasticIndicator]}/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle' name='AAPL'>
      </StockChartSeriesDirective>
    </StockChartSeriesCollectionDirective>
  </StockChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

## Customization

To change the legend icon shape, [legendShape](#) property in the [series](#) can be used. By default legend icon shape is `seriesType`.

**INDEX.JSX**

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
Export, StockLegend } from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    const primaryXAxis = {
        valueType: 'DateTime',
    };
    const crosshair = { enable: true };
    const tooltip = { enable: true };
    const legendSettings = { visible: true };
    return <StockChartComponent id='stockcharts' primaryXAxis={primaryXAxis}
crosshair={crosshair} tooltip={tooltip} legendSettings={legendSettings}
indicatorType={[]} trendlineType={[]} height='350' title='AAPL Stock Price'>
        <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, StockLegend, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle' name='AAPL' legendShape='Pentagon'>
            </StockChartSeriesDirective>
        </StockChartSeriesCollectionDirective>
    </StockChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

**INDEX.TSX**

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AxisModel, CrosshairSettingsModel, TooltipSettingsModel }
from '@syncfusion/ej2-react-charts';
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
Export, StockLegend } from '@syncfusion/ej2-react-charts';

```

```
import { chartData } from 'datasource.ts';
function App() {
  const primaryXAxis: AxisModel = {
    valueType: 'DateTime',
  };
  const crosshair: CrosshairSettingsModel = { enable: true };
  const tooltip: TooltipSettingsModel = { enable: true };
  const legendSettings: LegendSettingsModel = { visible: true };
  return <StockChartComponent id='stockcharts'
    primaryXAxis={primaryXAxis}
    crosshair={crosshair}
    tooltip={tooltip}
    legendSettings={legendSettings}
    indicatorType={[]}
    trendlineType = [[]]
    height='350'
    title='AAPL Stock Price'>
    <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, StockLegend, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle' name='AAPL' legendShape= 'Pentagon'>
        </StockChartSeriesDirective>
      </StockChartSeriesCollectionDirective>
    </StockChartComponent>
  </>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

### Legend Size

By default, legend takes 20% - 25% of the Stock Chart's height horizontally, when it is placed on top or bottom position and 20% - 25% of the width vertically, while placing on left or right position of the Stock Chart. The default legend size can be changed by using the [width](#) and [height](#) property of the `legendSettings`.

### INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
Export, StockLegend } from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryXAxis = {
```

```

        valueType: 'DateTime',
    };
    const crosshair = { enable: true };
    const tooltip = { enable: true };
    const legendSettings = { visible: true, width: '500', height: '50',
border: { width: 1, color: 'pink' } };
    return <StockChartComponent id='stockcharts' primaryXAxis={primaryXAxis}
crosshair={crosshair} tooltip={tooltip} legendSettings={legendSettings}
indicatorType={[]} trendlineType={[]} height='350' title='AAPL Stock Price'>
        <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, StockLegend, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle' name='AAPL'>
                </StockChartSeriesDirective>
            </StockChartSeriesCollectionDirective>
        </StockChartComponent>;
    }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AxisModel, CrosshairSettingsModel, TooltipSettingsModel }
from '@syncfusion/ej2-react-charts';
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
Export, StockLegend } from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    const primaryXAxis: AxisModel = {
        valueType: 'DateTime',
    };
    const crosshair: CrosshairSettingsModel = { enable: true };
    const tooltip: TooltipSettingsModel = { enable: true };
    const legendSettings: LegendSettingsModel = { visible: true, width:
'500', height: '50', border: { width: 1, color: 'pink' } };
    return <StockChartComponent id='stockcharts'
        primaryXAxis={ primaryXAxis }
        crosshair={crosshair}
        tooltip={tooltip}
        legendSettings={legendSettings}
        indicatorType={[]}

```

```

        trendlineType = {[[]]}
        height='350'
        title='AAPL Stock Price'>
        <Inject services={[DateTime, Tooltip, RangeTooltip,
        Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
        HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
        BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
        Export, StockLegend, AccumulationDistributionIndicator, MacdIndicator,
        StochasticIndicator]}/>
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle' name='AAPL'>
            </StockChartSeriesDirective>
        </StockChartSeriesCollectionDirective>
    </StockChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

### Legend Item Size

The size of the legend items can customized by using the [shapeHeight](#) and [shapeWidth](#) property.

### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
Export, StockLegend } from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    const primaryXAxis = {
        valueType: 'DateTime',
    };
    const crosshair = { enable: true };
    const tooltip = { enable: true };
    const legendSettings = { visible: true, shapeHeight: 15, shapeWidth: 15
};
    return <StockChartComponent id='stockcharts' primaryXAxis={primaryXAxis}
crosshair={crosshair} tooltip={tooltip} legendSettings={legendSettings}
indicatorType={[]} trendlineType={[]} height='350' title='AAPL Stock Price'>
        <Inject services={[DateTime, Tooltip, RangeTooltip,
        Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
        HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
        BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
        Export, StockLegend, AccumulationDistributionIndicator, MacdIndicator,
        StochasticIndicator]}/>
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle' name='AAPL'>

```

```

        </StockChartSeriesDirective>
      </StockChartSeriesCollectionDirective>
    </StockChartComponent>;
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AxisModel, CrosshairSettingsModel, TooltipSettingsModel }
  from '@syncfusion/ej2-react-charts';
import { StockChartComponent, StockChartSeriesCollectionDirective,
  StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
  Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
  HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
  charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
  MomentumIndicator, SmaIndicator, AtrIndicator,
  AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
  Export, StockLegend } from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryXAxis: AxisModel = {
    valueType: 'DateTime',
  };
  const crosshair: CrosshairSettingsModel = { enable: true };
  const tooltip: TooltipSettingsModel = { enable: true };
  const legendSettings: LegendSettingsModel = { visible: true, shapeHeight:
15, shapeWidth: 15 };
  return <StockChartComponent id='stockcharts'
    primaryXAxis={primaryXAxis}
    crosshair={crosshair}
    tooltip={tooltip}
    legendSettings={legendSettings}
    indicatorType={[]}
    trendlineType={[]}
    height='350'
    title='AAPL Stock Price'>
    <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, StockLegend, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle' name='AAPL'>
      </StockChartSeriesDirective>
    </StockChartSeriesCollectionDirective>
  </StockChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```



### Collapsing Legend Item

By default, series name will be displayed as legend. To skip the legend for a particular series, empty string to the series name can be given.

#### INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
Export, StockLegend } from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    const primaryXAxis = {
        valueType: 'DateTime',
    };
    const crosshair = { enable: true };
    const tooltip = { enable: true };
    const legendSettings = { visible: true };
    return <StockChartComponent id='stockcharts' primaryXAxis={primaryXAxis}
crosshair={crosshair} tooltip={tooltip} legendSettings={legendSettings}
indicatorType={[]} trendlineType={[]} height='350' title='AAPL Stock Price'>
        <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, StockLegend, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle'>
            </StockChartSeriesDirective>
        </StockChartSeriesCollectionDirective>
    </StockChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

#### INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AxisModel, CrosshairSettingsModel, TooltipSettingsModel }
from '@syncfusion/ej2-react-charts';
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
```

```
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
Export, StockLegend } from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryXAxis: AxisModel = {
    valueType: 'DateTime',
  };
  const crosshair: CrosshairSettingsModel = { enable: true };
  const tooltip: TooltipSettingsModel = { enable: true };
  const legendSettings: LegendSettingsModel = { visible: true };
  return <StockChartComponent id='stockcharts'
    primaryXAxis={ primaryXAxis }
    crosshair={crosshair}
    tooltip={tooltip}
    legendSettings={legendSettings}
    indicatorType={[]}
    trendlineType ={[]}
    height='350'
    title='AAPL Stock Price'>
    <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, StockLegend, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle'>
      </StockChartSeriesDirective>
    </StockChartSeriesCollectionDirective>
  </StockChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

### Legend Title

The title for legend can be set using [title](#) property in [legendSettings](#). Customize the [fontStyle](#), [size](#), [fontWeight](#), [color](#), [textAlignment](#), [fontFamily](#), [opacity](#) and [textOverflow](#) of legend title. [titlePosition](#) is used to set the legend position in [Top](#), [Left](#) and [Right](#) position. [maximumTitleWidth](#) is used to set the width of the legend title. By default, it will be [100px](#).

### INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-charts';
```

```

import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
Export, StockLegend } from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    const primaryXAxis = {
        valueType: 'DateTime',
    };
    const crosshair = { enable: true };
    const tooltip = { enable: true };
    const legendSettings = { visible: true, title: 'Countries',
titlePosition: 'Top', titleStyle: { fontFamily: 'verdana', fontStyle:
'Normal', fontWeight: 'Normal', size: '15px', textAlign: 'Center', color:
'blue', textOverflow: 'None' }, maximumTitleWidth: 150 };
    return <StockChartComponent id='stockcharts' primaryXAxis={primaryXAxis}
crosshair={crosshair} tooltip={tooltip} legendSettings={legendSettings}
indicatorType={[]} trendlineType={[]} height='350' title='AAPL Stock Price'>
        <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, StockLegend, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle' name='AAPL'>
                </StockChartSeriesDirective>
            </StockChartSeriesCollectionDirective>
        </StockChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AxisModel, CrosshairSettingsModel, TooltipSettingsModel }
from '@syncfusion/ej2-react-charts';
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
Export, StockLegend } from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    const primaryXAxis: AxisModel = {
        valueType: 'DateTime',
    };
    const crosshair: CrosshairSettingsModel = { enable: true };

```

```

const tooltip: TooltipSettingsModel = { enable: true };
const legendSettings: LegendSettingsModel = { visible: true, title:
'Countries', titlePosition: 'Top', titleStyle: { fontFamily: 'verdana',
fontStyle: 'Normal', fontWeight: 'Normal', size: '15px', textAlign:
'Center', color: 'blue', textOverflow: 'None' }, maximumTitleWidth: 150 };
return <StockChartComponent id='stockcharts'
  primaryXAxis={ primaryXAxis }
  crosshair={crosshair}
  tooltip={tooltip}
  legendSettings={legendSettings}
  indicatorType={[]}
  trendlineType ={}
  height='350'
  title='AAPL Stock Price'>
  <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, StockLegend, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle' name='AAPL'>
        </StockChartSeriesDirective>
      </StockChartSeriesCollectionDirective>
    </StockChartComponent>
  </>
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Note: To use legend feature, we need to inject **StockLegend** module into the **services**.

### Period selector in React Stock chart component

The period selector allows to select a range with specified periods. By default the period selector is enabled in stock chart.

#### Periods

Periods is an array of objects that allows users to specify the range of **periods** (../api/stock-chart/stockChartModel/#periods). The **interval** property specifies the count value of the button, and the **text** property specifies the text to be displayed on button. The **intervalType** property allows users to customize the intervals of the buttons. The **intervalType** property supports the following interval types:

- Auto
- Years
- Quarter
- Months
- Weeks
- Days
- Hours
- Minutes
- Seconds

**INDEX.JSX**

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryxAxis = {
    valueType: 'DateTime',
    majorGridLines: { width: 0 },
    majorTickLines: { color: 'transparent' }
  };
  const primaryyAxis = {
    labelFormat: 'n0',
    majorTickLines: { width: 0 }
  };
  const crosshair = { enable: true };
  const periodselector = [
    { text: '1M', interval: 1, intervalType: 'Months' },
    { text: '3M', interval: 3, intervalType: 'Months' },
    { text: '6M', interval: 6, intervalType: 'Months' }, { text: 'YTD' },
    { text: '1Y', interval: 1, intervalType: 'Years' },
    { text: '2Y', interval: 2, intervalType: 'Years', selected: true }, {
text: 'All' }
  ];
  return <StockChartComponent id='stockcharts' primaryXAxis={primaryxAxis}
primaryYAxis={primaryyAxis} crosshair={crosshair} periods={periodselector}
height='350' title='AAPL Stock Price'>
    <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
        </StockChartSeriesDirective>
      </StockChartSeriesCollectionDirective>
    </StockChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

**INDEX.TSX**

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AxisModel, CrosshairSettingsModel, PeriodsModel }
from '@syncfusion/ej2-react-charts';
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryxAxis: AxisModel= {
    valueType: 'DateTime',
    majorGridLines: { width: 0 },
    majorTickLines: { color: 'transparent' }
  };
  const primaryyAxis: AxisModel= {
    labelFormat: 'n0',
    majorTickLines: { width: 0 }
  };
  const crosshair: CrosshairSettingsModel = { enable: true };
  const periodselector: PeriodsModel = [
    { text: '1M', interval: 1, intervalType: 'Months' },
    { text: '3M', interval: 3, intervalType: 'Months' },
    { text: '6M', interval: 6, intervalType: 'Months' }, { text: 'YTD' },
    { text: '1Y', interval: 1, intervalType: 'Years' },
    { text: '2Y', interval: 2, intervalType: 'Years', selected: true }, {
text: 'All' }
  ];
  return <StockChartComponent id='stockcharts'
    primaryXAxis={ primaryxAxis }
    primaryYAxis={ primaryyAxis }
    crosshair={crosshair}
    periods={periodselector}
    height='350'
    title='AAPL Stock Price'>
    <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
      </StockChartSeriesDirective>
    </StockChartSeriesCollectionDirective>
  </StockChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

### Visibility of period selector

The [enablePeriodSelector](#) property allows users to toggle the visibility of period selector.

#### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryxAxis = {
    valueType: 'DateTime',
    majorGridLines: { width: 0 },
    majorTickLines: { color: 'transparent' },
    crosshairTooltip: { enable: true }
  };
  const primaryyAxis = {
    majorTickLines: { width: 0 },
    crosshairTooltip: { enable: true }
  };
  const crosshair = { enable: true };
  return <StockChartComponent id='stockcharts' enablePeriodSelector={false}
primaryXAxis={primaryxAxis} primaryYAxis={primaryyAxis} crosshair={crosshair}
height='350' title='AAPL Stock Price'>
    <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
        </StockChartSeriesDirective>
      </StockChartSeriesCollectionDirective>
    </StockChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

#### INDEX.TSX

```
{% raw %}
import * as React from "react";
```

```

import * as ReactDOM from "react-dom";
import { AxisModel, CrosshairSettingsModel, FontModel }
  from '@syncfusion/ej2-react-charts';
import { StockChartComponent, StockChartSeriesCollectionDirective,
  StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
  Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
  HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
  charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
  MomentumIndicator, SmaIndicator, AtrIndicator,
  AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryxAxis: AxisModel = {
    valueType: 'DateTime',
    majorGridLines: { width: 0 },
    majorTickLines: { color: 'transparent' },
    crosshairTooltip: { enable: true }
  };
  const primaryyAxis: AxisModel = {
    majorTickLines: { width: 0 },
    crosshairTooltip: { enable: true }
  };
  const crosshair: CrosshairSettingsModel = { enable: true };
  return <StockChartComponent id='stockcharts'
    enablePeriodSelector={false}
    primaryXAxis={ primaryxAxis }
    primaryYAxis={ primaryyAxis }
    crosshair={crosshair}
    height='350'
    title='AAPL Stock Price'>
    <Inject services=[DateTime, Tooltip, RangeTooltip,
  Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
  HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
  BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
  Export, AccumulationDistributionIndicator, MacdIndicator,
  StochasticIndicator]]/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
      </StockChartSeriesDirective>
    </StockChartSeriesCollectionDirective>
  </StockChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

### Range selector in React Stock chart component

The left and right thumb of RangeNavigator are used to indicate the selected range in the large collection of data. Following are the ways you can select a range.

- By dragging the thumbs.
- By tapping on the labels.



- By setting the start and end through Date Range button.

Following code example shows the [enableSelector](#) property allows users to toggle the visibility of range selector.

#### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryxAxis = {
    valueType: 'DateTime',
    majorGridLines: { width: 0 },
    majorTickLines: { color: 'transparent' },
    crosshairTooltip: { enable: true }
  };
  const primaryyAxis = {
    majorTickLines: { width: 0 },
    crosshairTooltip: { enable: true }
  };
  const crosshair = { enable: true };
  return <StockChartComponent id='stockcharts' enableSelector={true}
primaryXAxis={primaryxAxis} primaryYAxis={primaryyAxis} crosshair={crosshair}
height='350' title='AAPL Stock Price'>
    <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
      </StockChartSeriesDirective>
    </StockChartSeriesCollectionDirective>
  </StockChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

#### INDEX.TSX

```
{% raw %}
```

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AxisModel, CrosshairSettingsModel, FontModel }
from '@syncfusion/ej2-react-charts';
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    const primaryxAxis: AxisModel= {
        valueType: 'DateTime',
        majorGridLines: { width: 0 },
        majorTickLines: { color: 'transparent' },
        crosshairTooltip: { enable: true }
    };
    const primaryyAxis: AxisModel= {
        majorTickLines: { width: 0 },
        crosshairTooltip: { enable: true }
    };
    const crosshair: CrosshairSettingsModel = { enable: true };
    return <StockChartComponent id='stockcharts'
        enableSelector={true}
        primaryXAxis={ primaryxAxis }
        primaryYAxis={ primaryyAxis }
        crosshair={crosshair}
        height='350'
        title='AAPL Stock Price'>
        <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
            </StockChartSeriesDirective>
        </StockChartSeriesCollectionDirective>
    </StockChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

### Export print in React Stock chart component

The rendered stock chart can be exported to JPEG, PNG, SVG, or PDF format using the export dropdown button in the period selector toolbar. You can choose the required format using the export dropdown button in stock-chart.

The rendered stock chart can be printed directly using print button in period selector toolbar.

### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryxAxis = {
    valueType: 'DateTime',
    majorGridLines: { width: 0 },
    majorTickLines: { color: 'transparent' }
  };
  const primaryyAxis = {
    labelFormat: 'n0',
    majorTickLines: { width: 0 }
  };
  return <StockChartComponent id='stockcharts' primaryXAxis={primaryxAxis}
primaryYAxis={primaryyAxis} height='350' title='AAPL Stock Price'>
    <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
        </StockChartSeriesDirective>
      </StockChartSeriesCollectionDirective>
    </StockChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

### INDEX.TSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AxisModel, CrosshairSettingsModel, FontModel }
from '@syncfusion/ej2-react-charts';
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
```

```
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryxAxis: AxisModel= {
    valueType: 'DateTime',
    majorGridLines: { width: 0 },
    majorTickLines: { color: 'transparent' }
  };
  const primaryyAxis: AxisModel= {
    labelFormat: 'n0',
    majorTickLines: { width: 0 }
  };
  return <StockChartComponent id='stockcharts'
    primaryXAxis={ primaryxAxis }
    primaryYAxis={ primaryyAxis }
    height='350'
    title='AAPL Stock Price'>
    <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
    </StockChartSeriesDirective>
    </StockChartSeriesCollectionDirective>
  </StockChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

### Disable Export and print

To empty the value of `exportType` to disable the Export button.

### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
```

```

import { chartData } from 'datasource.ts';
function App() {
  const primaryxAxis = {
    valueType: 'DateTime',
    majorGridLines: { width: 0 },
    majorTickLines: { color: 'transparent' }
  };
  const primaryyAxis = {
    majorTickLines: { width: 0 }
  };
  return <StockChartComponent id='stockcharts' primaryXAxis={primaryxAxis}
primaryYAxis={primaryyAxis} exportType={[]} height='350' title='AAPL Stock
Price'>
    <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
    </StockChartSeriesDirective>
    </StockChartSeriesCollectionDirective>
  </StockChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

## INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AxisModel, CrosshairSettingsModel, FontModel }
from '@syncfusion/ej2-react-charts';
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryxAxis: AxisModel= {
    valueType: 'DateTime',
    majorGridLines: { width: 0 },
    majorTickLines: { color: 'transparent' }
  };
  const primaryyAxis: AxisModel= {
    majorTickLines: { width: 0 }
  };

```

```

    };

    return <StockChartComponent id='stockcharts'
      primaryXAxis={ primaryxAxis }
      primaryYAxis={ primaryyAxis }
      exportType={[]}
      height='350'
      title='AAPL Stock Price'>
      <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
      <StockChartSeriesCollectionDirective>
        <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
          </StockChartSeriesDirective>
        </StockChartSeriesCollectionDirective>
      </StockChartComponent>
    };
    export default App;
    ReactDOM.render(<App />, document.getElementById("charts"));
    {% endraw %}

```

## Appearance in React Stock chart component

### Stock Chart Title

Stock Chart can be given a title using [title](#) property, to show the information about the data plotted.

### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryxAxis = {
    valueType: 'DateTime',
    majorGridLines: { width: 0 },
    majorTickLines: { color: 'transparent' },
    crosshairTooltip: { enable: true }
  };
  const primaryyAxis = {
    labelFormat: 'n0',
    lineStyle: { width: 0 },
    rangePadding: 'None',
    majorTickLines: { width: 0 }
  };

```

```

const crosshair = { enable: true };
const titlestyle = { fontFamily: "Arial", color: '#424242 ' };
const tooltip = { enable: true };
return <StockChartComponent id='stockcharts' primaryXAxis={primaryxAxis}
primaryYAxis={primaryyAxis} tooltip={tooltip} crosshair={crosshair}
height='350' title='AAPL Stock Price' titleStyle={titlestyle}>
    <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
    <StockChartSeriesCollectionDirective>
        <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
            </StockChartSeriesDirective>
        </StockChartSeriesCollectionDirective>
    </StockChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

**INDEX.TSX**

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AxisModel, CrosshairSettingsModel, FontModel, TooltipSettingsModel }
from '@syncfusion/ej2-react-charts';
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    const primaryxAxis: AxisModel= {
        valueType: 'DateTime',
        majorGridLines: { width: 0 },
        majorTickLines: { color: 'transparent' },
        crosshairTooltip: { enable: true }
    };
    const primaryyAxis: AxisModel= {
        labelFormat: 'n0',
        lineStyle: { width: 0 },
        rangePadding: 'None',
        majorTickLines: { width: 0 }
    };
    const crosshair: CrosshairSettingsModel = { enable: true };
    const titlestyle: FontModel = {fontFamily: "Arial", color: '#424242 '};

```

```

const tooltip: TooltipSettingsModel = { enable: true };
return <StockChartComponent id='stockcharts'
  primaryXAxis={ primaryxAxis }
  primaryYAxis={ primaryyAxis }
  tooltip={tooltip}
  crosshair={crosshair}
  height='350'
  title='AAPL Stock Price'
  titleStyle = { titlestyle }>
  <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]} />
  <StockChartSeriesCollectionDirective>
    <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
    </StockChartSeriesDirective>
  </StockChartSeriesCollectionDirective>
</StockChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

<!-- markdownlint-disable MD036 -->

### Title Customization

The `textStyle` property of chart title provides options to customize the `size`, `color`, `fontFamily`, `fontWeight`, `fontStyle`, `opacity`, `textAlignment` and `textOverflow`.

### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
  const primaryxAxis = {
    valueType: 'DateTime',
    majorGridLines: { width: 0 },
    majorTickLines: { color: 'transparent' }
  };
  const primaryyAxis = {
    labelFormat: 'n0',
    majorTickLines: { width: 0 }
  };

```



```

    };
    const crosshair = { enable: true };
    const titlestyle = { fontFamily: "Arial", color: '#424242 ',
textOverflow: 'Wrap' };
    return <StockChartComponent id='stockcharts' primaryXAxis={primaryxAxis}
primaryYAxis={primaryyAxis} title='AAPL Stock Price' height='350'
titleStyle={titlestyle}>
        <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}>
            <StockChartSeriesCollectionDirective>
                <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
                    </StockChartSeriesDirective>
                </StockChartSeriesCollectionDirective>
            </StockChartComponent>;
        }
    };
    export default App;
    ReactDOM.render(<App />, document.getElementById("charts"));
    {% endraw %}

```

## INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AxisModel, CrosshairSettingsModel, FontModel }
from '@syncfusion/ej2-react-charts';
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    const primaryxAxis: AxisModel= {
        valueType: 'DateTime',
        majorGridLines: { width: 0 },
        majorTickLines: { color: 'transparent' }
    };
    const primaryyAxis: AxisModel= {
        labelFormat: 'n0',
        majorTickLines: { width: 0 }
    };
    const crosshair: CrosshairSettingsModel = { enable: true };
    const titlestyle: FontModel = {fontFamily: "Arial", color: '#424242
',textOverflow: 'Wrap'};
    return <StockChartComponent id='stockcharts'

```

```

        primaryXAxis={ primaryxAxis }
        primaryYAxis={ primaryyAxis }
        title='AAPL Stock Price'
        height='350'
        titleStyle = { titlestyle }>
        <Inject services={[DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator,
BollingerBands, TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator,
Export, AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
            </StockChartSeriesDirective>
        </StockChartSeriesCollectionDirective>
    </StockChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

### Stock Chart Theme

Changing Stock Chart theme will affect background color, grid lines, tooltip colors and appearance.

[theme](#) property of Stock chart is shipped with several built-in themes such as **Material**, **Fabric**, **Bootstrap**, **HighContrastLight**, **MaterialDark**, **FabricDark**, **FabricDark**, **HighContrast** and **BootstrapDark**.

### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, RangeTooltip, Crosshair,
LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries, HiloSeries,
RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    const primaryxAxis = {
        valueType: 'DateTime',
        majorGridLines: { width: 0 },
        majorTickLines: { color: 'transparent' }
    };
    const primaryyAxis = {
        labelFormat: 'n0',
        majorTickLines: { width: 0 }
    };
    const crosshair = { enable: true };

```

```

    return <StockChartComponent id='stockcharts' primaryXAxis={primaryxAxis}
primaryYAxis={primaryyyAxis} title='AAPL Stock Price' height='350'
theme='HighContrast'>
        <Inject services={[DateTime, RangeTooltip, Crosshair,
LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries, HiloSeries,
RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator, BollingerBands,
TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator, Export,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator]}/>
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
                </StockChartSeriesDirective>
            </StockChartSeriesCollectionDirective>
        </StockChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

## INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AxisModel, CrosshairSettingsModel, FontModel }
from '@syncfusion/ej2-react-charts';
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, Tooltip, RangeTooltip,
Crosshair, LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries,
HiloSeries, RangeAreaSeries, Trendlines } from '@syncfusion/ej2-react-
charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    const primaryxAxis: AxisModel= {
        valueType: 'DateTime',
        majorGridLines:{ width: 0 },
        majorTickLines: { color: 'transparent' }
    };
    const primaryyyAxis: AxisModel= {
        labelFormat: 'n0',
        majorTickLines: { width: 0 }
    };
    const crosshair: CrosshairSettingsModel ={ enable: true } ;
    return <StockChartComponent id='stockcharts'
        primaryXAxis={ primaryxAxis }
        primaryYAxis={ primaryyyAxis }
        title='AAPL Stock Price'
        height='350'
        theme='HighContrast'>
        <Inject services={[DateTime, RangeTooltip, Crosshair,
LineSeries, SplineSeries, CandleSeries, HiloOpenCloseSeries, HiloSeries,

```

```

RangeAreaSeries, Trendlines, EmaIndicator, RsiIndicator, BollingerBands,
TmaIndicator, MomentumIndicator, SmaIndicator, AtrIndicator, Export,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator]]/>
    <StockChartSeriesCollectionDirective>
      <StockChartSeriesDirective dataSource={chartData}
type='Candle' animation={{ enable: true }}>
    </StockChartSeriesDirective>
  </StockChartSeriesCollectionDirective>
</StockChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

## See Also

- [Axis Customization](#)

<!-- markdownlint-disable MD036 -->

## Stock events in React Stock chart component

Stock Events visualizes stock events in stock chart. 'SplineSeries' is used to represent selected data value. You can customize the specific data value using `stockEvents` event.

### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
StockChartSeriesDirective, Inject, DateTime, StripLine, LineSeries,
SplineSeries, CandleSeries, HiloOpenCloseSeries, HiloSeries, RangeAreaSeries,
Trendlines, Crosshair } from '@syncfusion/ej2-react-charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator,
AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { aapl } from 'datasource.ts';
function App() {
  const primaryxAxis = {
    valueType: 'DateTime', majorGridLines: { color: 'transparent' },
    crosshairTooltip: { enable: true }
  };
  const primaryyAxis = {
    lineStyle: { color: 'transparent' },
    majorTickLines: { color: 'transparent' },
    crosshairTooltip: { enable: true }
  };
  return <StockChartComponent id='stockcharts' primaryXAxis={primaryxAxis}
primaryYAxis={primaryyAxis} indicatorType={[]} seriesType={[]}
trendlineType={[]} title='AAPL Stock Price' height='350' stockEvents=[
    { date: new Date(2012, 3, 1), text: 'Q2', description: '2012
Quarter2 starts', type: 'Flag' },
    {
      date: new Date(2012, 3, 20), text: 'Open', description:
'Markets opened', textStyle: { color: 'white' },

```

```

        background: '#f48a21', border: { color: '#f48a21' }
    },
    {
        date: new Date(2012, 6, 1), text: 'Q3', description: '2013
Quarter3 starts', type: 'Flag',
        textStyle: { color: 'white' }, background: '#6c6d6d', border:
{ color: '#6c6d6d' }
    },
    {
        date: new Date(2012, 9, 1), text: 'Q4', description: '2013
Quarter4 starts', type: 'Flag',
        textStyle: { color: 'white' }, background: '#6c6d6d', border:
{ color: '#6c6d6d' }
    },
    {
        date: new Date(2012, 7, 30), text: 'G', description: 'Google
stocks bought',
        textStyle: { color: 'white' }, background: '#f48a21', border:
{ color: '#f48a21' }
    },
    {
        date: new Date(2012, 10, 1), text: 'Y', description: 'Yahoo
stocks sold', type: 'Square',
        textStyle: { color: 'white' }, background: '#841391', border:
{ color: '#841391' }
    },
    {
        date: new Date(2012, 12, 0), text: 'Y2', description: 'Year
2013', type: 'Pin', showOnSeries: false,
        textStyle: { color: 'white' }, background: '#6322e0', border:
{ color: '#6322e0' }
    },
    {
        date: new Date(2013, 3, 1), text: 'Q2', description: '2013
Quarter2 starts', type: 'Flag',
        textStyle: { color: 'white' }, background: '#6c6d6d', border:
{ color: '#6c6d6d' }
    },
    {
        date: new Date(2013, 3, 20), text: 'Q2', description: 'Surge
in Stocks', type: 'ArrowUp',
        textStyle: { color: 'white' }, background: '#3ab0f9', border:
{ color: '#3ab0f9' }
    },
    {
        date: new Date(2013, 6, 1), text: 'Q3', description: '2013
Quarter3 starts', type: 'Flag',
        textStyle: { color: 'white' }, background: '#6c6d6d', border:
{ color: '#6c6d6d' }
    },
    {
        date: new Date(2013, 9, 1), text: 'Q4', description: '2013
Quarter4 starts', type: 'Flag',
        textStyle: { color: 'white' }, background: '#6c6d6d', border:
{ color: '#6c6d6d' }
    },
    {

```

```

        date: new Date(2013, 12, 0), text: 'Y3', description: 'Year
2014', type: 'Pin', showOnSeries: false,
        textStyle: { color: 'white' }, background: '#6322e0', border:
{ color: '#6322e0' }
    },
    {
        date: new Date(2014, 3, 1), text: 'Q2', description: '2014
Quarter2 starts', type: 'ArrowDown',
        textStyle: { color: 'white' }, background: '#3ab0f9', border:
{ color: '#3ab0f9' }
    },
    {
        date: new Date(2014, 6, 1), text: 'Q3', description: '2014
Quarter3 starts',
        textStyle: { color: 'white' }, background: '#f48a21', border:
{ color: '#f48a21' }
    },
    {
        date: new Date(2014, 9, 1), text: 'Q4', description: '2014
Quarter4 starts', type: 'Flag',
        textStyle: { color: 'white' }, background: '#6c6d6d', border:
{ color: '#6c6d6d' }
    },
    {
        date: new Date(2014, 12, 0), text: 'Y4', description: 'Year
2015', type: 'Pin', showOnSeries: false,
        textStyle: { color: 'white' }, background: '#6322e0', border:
{ color: '#6322e0' }
    },
    {
        date: new Date(2014, 2, 2), text: 'End', description:
'Markets closed', type: 'ArrowDown',
        textStyle: { color: 'white' }, background: '#3ab0f9', border:
{ color: '#3ab0f9' }
    },
    {
        date: new Date('2015-01-07'), text: 'A', description: 'This
is event description',
        textStyle: { color: 'white' }, background: '#f48a21', border:
{ color: '#f48a21' }
    },
    {
        date: new Date(2015, 1, 2), text: 'Q1', description: 'Add
longer text',
        textStyle: { color: 'white' }, background: '#dd3c9f', border:
{ color: '#dd3c9f' }, type: 'Text'
    },
    {
        date: new Date(2015, 2, 12), text: 'Close', description:
'Markets closed',
        textStyle: { color: 'white' }, background: '#f48a21', border:
{ color: '#f48a21' }
    }
  ]]>
  <Inject services={[DateTime, StripLine, LineSeries, Crosshair,
SplineSeries, CandleSeries, HiloOpenCloseSeries, HiloSeries, RangeAreaSeries,
Trendlines,

```

```

        EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
        MomentumIndicator, SmaIndicator, AtrIndicator, Export,
        AccumulationDistributionIndicator, MacdIndicator,
        StochasticIndicator]]/>
        <StockChartSeriesCollectionDirective>
            <StockChartSeriesDirective dataSource={aapl} type='Spline'
            xName='x' width={2} yName='high' name='google' close='high'>
                </StockChartSeriesDirective>
            </StockChartSeriesCollectionDirective>
        </StockChartComponent>;
    }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

### INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import {
    StockChartComponent, StockChartSeriesCollectionDirective,
    StockChartSeriesDirective, Inject,
    DateTime, StripLine, LineSeries, SplineSeries, CandleSeries,
    HiloOpenCloseSeries, HiloSeries,
    RangeAreaSeries, Trendlines, ChartTheme, IStockChartEventArgs, Crosshair
} from '@syncfusion/ej2-react-charts';
import {
    EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
    MomentumIndicator, SmaIndicator, AtrIndicator,
    AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
    Export
} from '@syncfusion/ej2-react-charts';
import { aapl } from 'datasource.ts';
function App() {
    const primaryxAxis: AxisModel = {
        valueType: 'DateTime', majorGridLines: { color: 'transparent' },
        crosshairTooltip: { enable: true }
    };
    const primaryyAxis: AxisModel = {
        lineStyle: { color: 'transparent' },
        majorTickLines: { color: 'transparent' },
        crosshairTooltip: { enable: true }
    };
    return <StockChartComponent id='stockcharts'
        primaryXAxis={primaryxAxis}
        primaryYAxis={primaryyAxis}
        indicatorType={[]}
        seriesType={[]}
        trendlineType={[]}
        title='AAPL Stock Price'
        height='350'
        stockEvents=[
            { date: new Date(2012, 3, 1), text: 'Q2', description: '2012
Quarter2 starts', type: 'Flag' },
            {

```

```

        date: new Date(2012, 3, 20), text: 'Open', description:
'Markets opened', textStyle: { color: 'white' },
        background: '#f48a21', border: { color: '#f48a21' }
    },
    {
        date: new Date(2012, 6, 1), text: 'Q3', description:
'2013 Quarter3 starts', type: 'Flag',
        textStyle: { color: 'white' }, background: '#6c6d6d',
border: { color: '#6c6d6d' }
    },
    {
        date: new Date(2012, 9, 1), text: 'Q4', description:
'2013 Quarter4 starts', type: 'Flag',
        textStyle: { color: 'white' }, background: '#6c6d6d',
border: { color: '#6c6d6d' }
    },
    {
        date: new Date(2012, 7, 30), text: 'G', description:
'Google stocks bought',
        textStyle: { color: 'white' }, background: '#f48a21',
border: { color: '#f48a21' }
    },
    {
        date: new Date(2012, 10, 1), text: 'Y', description:
'Yahoo stocks sold', type: 'Square',
        textStyle: { color: 'white' }, background: '#841391',
border: { color: '#841391' }
    },
    {
        date: new Date(2012, 12, 0), text: 'Y2', description:
'Year 2013', type: 'Pin', showOnSeries: false,
        textStyle: { color: 'white' }, background: '#6322e0',
border: { color: '#6322e0' }
    },
    {
        date: new Date(2013, 3, 1), text: 'Q2', description:
'2013 Quarter2 starts', type: 'Flag',
        textStyle: { color: 'white' }, background: '#6c6d6d',
border: { color: '#6c6d6d' }
    },
    {
        date: new Date(2013, 3, 20), text: 'Q2', description:
'Surge in Stocks', type: 'ArrowUp',
        textStyle: { color: 'white' }, background: '#3ab0f9',
border: { color: '#3ab0f9' }
    },
    {
        date: new Date(2013, 6, 1), text: 'Q3', description:
'2013 Quarter3 starts', type: 'Flag',
        textStyle: { color: 'white' }, background: '#6c6d6d',
border: { color: '#6c6d6d' }
    },
    {
        date: new Date(2013, 9, 1), text: 'Q4', description:
'2013 Quarter4 starts', type: 'Flag',
        textStyle: { color: 'white' }, background: '#6c6d6d',
border: { color: '#6c6d6d' }
    }

```



```

    },
    {
      date: new Date(2013, 12, 0), text: 'Y3', description:
'Year 2014', type: 'Pin', showOnSeries: false,
      textStyle: { color: 'white' }, background: '#6322e0',
border: { color: '#6322e0' }
    },
    {
      date: new Date(2014, 3, 1), text: 'Q2', description:
'2014 Quarter2 starts', type: 'ArrowDown',
      textStyle: { color: 'white' }, background: '#3ab0f9',
border: { color: '#3ab0f9' }
    },
    {
      date: new Date(2014, 6, 1), text: 'Q3', description:
'2014 Quarter3 starts',
      textStyle: { color: 'white' }, background: '#f48a21',
border: { color: '#f48a21' }
    },
    {
      date: new Date(2014, 9, 1), text: 'Q4', description:
'2014 Quarter4 starts', type: 'Flag',
      textStyle: { color: 'white' }, background: '#6c6d6d',
border: { color: '#6c6d6d' }
    },
    {
      date: new Date(2014, 12, 0), text: 'Y4', description:
'Year 2015', type: 'Pin', showOnSeries: false,
      textStyle: { color: 'white' }, background: '#6322e0',
border: { color: '#6322e0' }
    },
    {
      date: new Date(2014, 2, 2), text: 'End', description:
'Markets closed', type: 'ArrowDown',
      textStyle: { color: 'white' }, background: '#3ab0f9',
border: { color: '#3ab0f9' }
    },
    {
      date: new Date('2015-01-07'), text: 'A', description:
'This is event description',
      textStyle: { color: 'white' }, background: '#f48a21',
border: { color: '#f48a21' }
    },
    {
      date: new Date(2015, 1, 2), text: 'Q1', description: 'Add
longer text',
      textStyle: { color: 'white' }, background: '#dd3c9f',
border: { color: '#dd3c9f' }, type: 'Text'
    },
    {
      date: new Date(2015, 2, 12), text: 'Close', description:
'Markets closed',
      textStyle: { color: 'white' }, background: '#f48a21',
border: { color: '#f48a21' }
    }
  ]}
>

```

```

    <Inject services={[DateTime, StripLine, LineSeries, Crosshair,
    SplineSeries, CandleSeries, HiloOpenCloseSeries, HiloSeries, RangeAreaSeries,
    Trendlines,
        EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
    MomentumIndicator, SmaIndicator, AtrIndicator, Export,
        AccumulationDistributionIndicator, MacdIndicator,
    StochasticIndicator]} />
    <StockChartSeriesCollectionDirective>
        <StockChartSeriesDirective dataSource={aapl} type='Spline'
    xName='x' width={2} yName='high' name='google' close='high'>
        </StockChartSeriesDirective>
    </StockChartSeriesCollectionDirective>
</StockChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

### Stock Events for individual series

By default, stock events will be showed for all series. Now, you can set the stock events for particular series using `seriesIndexes` property.

#### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { StockChartComponent, StockChartSeriesCollectionDirective,
    StockChartSeriesDirective, Inject, DateTime, StripLine, LineSeries,
    SplineSeries, CandleSeries, HiloOpenCloseSeries, HiloSeries, RangeAreaSeries,
    Trendlines, Crosshair } from '@syncfusion/ej2-react-charts';
import { EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
    MomentumIndicator, SmaIndicator, AtrIndicator,
    AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator, Export
} from '@syncfusion/ej2-react-charts';
import { chartData } from 'datasource.ts';
function App() {
    const primaryXAxis = {
        valueType: 'DateTime', majorGridLines: { color: 'transparent' },
        crosshairTooltip: { enable: true }
    };
    const primaryYAxis = {
        lineStyle: { color: 'transparent' }, majorTickLines: { color:
        'transparent' }, crosshairTooltip: { enable: true }
    };
    return <StockChartComponent id='stockcharts' primaryXAxis={primaryXAxis}
    primaryYAxis={primaryYAxis} indicatorType={[]} seriesType={[]}
    trendlineType={[]} title='AAPL Stock Price' height='350' stockEvents={[
        { date: new Date(2012, 3, 1), text: 'Q2', description: '2012
        Quarter2 starts', type: 'Flag', background: '#6c6d6d', border: { color:
        '#6c6d6d' }, seriesIndexes: [0] },
        { date: new Date(2012, 3, 20), text: 'Open', description:
        'Markets opened', textStyle: { color: 'white' },
            background: '#f48a21', border: { color: '#f48a21' },
        seriesIndexes: [0] },
        { date: new Date(2012, 6, 1), text: 'Q3', description: '2013
        Quarter3 starts', type: 'Flag',

```

```

        textStyle: { color: 'white' }, background: '#6c6d6d', border:
{ color: '#6c6d6d' }, seriesIndexes: [0] },
        { date: new Date(2012, 9, 1), text: 'Q4', description: '2013
Quarter4 starts', type: 'Flag',
        textStyle: { color: 'white' }, background: '#6c6d6d', border:
{ color: '#6c6d6d' }, seriesIndexes: [0] },
        { date: new Date(2012, 7, 30), text: 'G', description: 'Google
stocks bought',
        textStyle: { color: 'white' }, background: '#f48a21', border:
{ color: '#f48a21' }, seriesIndexes: [0] },
        { date: new Date(2012, 10, 1), text: 'Y', description: 'Yahoo
stocks sold', type: 'Square',
        textStyle: { color: 'white' }, background: '#841391', border:
{ color: '#841391' }, seriesIndexes: [0] },
        { date: new Date(2012, 12, 0), text: 'Y2', description: 'Year
2013', type: 'Pin', showOnSeries: false,
        textStyle: { color: 'white' }, background: '#6322e0', border:
{ color: '#6322e0' }, seriesIndexes: [0] },
        { date: new Date(2013, 3, 1), text: 'Q2', description: '2013
Quarter2 starts', type: 'Flag',
        textStyle: { color: 'white' }, background: '#6c6d6d', border:
{ color: '#6c6d6d' }, seriesIndexes: [0] },
        { date: new Date(2013, 3, 20), text: 'Q2', description: 'Surge in
Stocks', type: 'ArrowUp',
        textStyle: { color: 'white' }, background: '#3ab0f9', border:
{ color: '#3ab0f9' }, seriesIndexes: [1] },
        { date: new Date(2013, 6, 1), text: 'Q3', description: '2013
Quarter3 starts', type: 'Flag',
        textStyle: { color: 'white' }, background: '#6c6d6d', border:
{ color: '#6c6d6d' }, seriesIndexes: [1] },
        { date: new Date(2013, 9, 1), text: 'Q4', description: '2013
Quarter4 starts', type: 'Flag',
        textStyle: { color: 'white' }, background: '#6c6d6d', border:
{ color: '#6c6d6d' }, seriesIndexes: [1] },
        { date: new Date(2013, 12, 0), text: 'Y3', description: 'Year
2014', type: 'Pin', showOnSeries: false,
        textStyle: { color: 'white' }, background: '#6322e0', border:
{ color: '#6322e0' }, seriesIndexes: [1] },
        { date: new Date(2014, 3, 1), text: 'Q2', description: '2014
Quarter2 starts', type: 'ArrowDown',
        textStyle: { color: 'white' }, background: '#3ab0f9', border:
{ color: '#3ab0f9' }, seriesIndexes: [1] },
        { date: new Date(2014, 6, 1), text: 'Q3', description: '2014
Quarter3 starts',
        textStyle: { color: 'white' }, background: '#f48a21', border:
{ color: '#f48a21' }, seriesIndexes: [2] },
        { date: new Date(2014, 9, 1), text: 'Q4', description: '2014
Quarter4 starts', type: 'Flag',
        textStyle: { color: 'white' }, background: '#6c6d6d', border:
{ color: '#6c6d6d' }, seriesIndexes: [2] },
        { date: new Date(2014, 12, 0), text: 'Y4', description: 'Year
2015', type: 'Pin', showOnSeries: false,
        textStyle: { color: 'white' }, background: '#6322e0', border:
{ color: '#6322e0' }, seriesIndexes: [2] },
        { date: new Date(2014, 2, 2), text: 'End', description: 'Markets
closed', type: 'ArrowDown',

```

```

        textStyle: { color: 'white' }, background: '#3ab0f9', border:
    { color: '#3ab0f9' }, seriesIndexes: [2] },
    { date: new Date('2015-01-07'), text: 'A', description: 'This is
event description',
        textStyle: { color: 'white' }, background: '#f48a21', border:
    { color: '#f48a21' }, seriesIndexes: [2] },
    { date: new Date(2015, 1, 2), text: 'Q1', description: 'Add
longer text',
        textStyle: { color: 'white' }, background: '#dd3c9f', border:
    { color: '#dd3c9f' }, type: 'Text', seriesIndexes: [2] },
    { date: new Date(2015, 2, 12), text: 'Close', description:
'Markets closed',
        textStyle: { color: 'white' }, background: '#f48a21', border:
    { color: '#f48a21' }, seriesIndexes: [2] }
    ]}>
    <Inject services={[DateTime, StripLine, LineSeries, Crosshair,
SplineSeries, CandleSeries, HiloOpenCloseSeries, HiloSeries, RangeAreaSeries,
Trendlines,
        EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator, Export,
        AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]}/>
    <StockChartSeriesCollectionDirective>
        <StockChartSeriesDirective dataSource={chartData}
type='Spline' xName='date' width={2} yName='high'>
        </StockChartSeriesDirective>
        <StockChartSeriesDirective dataSource={chartData}
type='Spline' xName='date' width={2} yName='low'>
        </StockChartSeriesDirective>
        <StockChartSeriesDirective dataSource={chartData}
type='Spline' xName='date' width={2} yName='open'>
        </StockChartSeriesDirective>
    </StockChartSeriesCollectionDirective>
</StockChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import {
    StockChartComponent, StockChartSeriesCollectionDirective,
    StockChartSeriesDirective, Inject,
    DateTime, StripLine, LineSeries, SplineSeries, CandleSeries,
    HiloOpenCloseSeries, HiloSeries,
    RangeAreaSeries, Trendlines, ChartTheme, IStockChartEventArgs, Crosshair
} from '@syncfusion/ej2-react-charts';
import {
    EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
    MomentumIndicator, SmaIndicator, AtrIndicator,
    AccumulationDistributionIndicator, MacdIndicator, StochasticIndicator,
    Export
} from '@syncfusion/ej2-react-charts';

```

```

import { chartData } from 'datasource.ts';
function App() {
  const primaryxAxis: AxisModel = {
    valueType: 'DateTime', majorGridLines: { color: 'transparent' },
    crosshairTooltip: { enable: true }
  };
  const primaryyAxis: AxisModel = {
   LineStyle: { color: 'transparent' }, majorTickLines: { color:
'transparent' }, crosshairTooltip: { enable: true }
  };
  return <StockChartComponent id='stockcharts'
    primaryXAxis={primaryxAxis}
    primaryYAxis={primaryyAxis}
    indicatorType={[]}
    seriesType={[]}
    trendlineType={[]}
    title='AAPL Stock Price'
    height='350'
    stockEvents=[
      { date: new Date(2012, 3, 1), text: 'Q2', description: '2012
Quarter2 starts', type: 'Flag', background: '#6c6d6d', border: { color:
'#6c6d6d' }, seriesIndexes: [0] },
      { date: new Date(2012, 3, 20), text: 'Open', description:
'Markets opened', textStyle: { color: 'white' },
        background: '#f48a21', border: { color: '#f48a21' },
seriesIndexes: [0] },
      { date: new Date(2012, 6, 1), text: 'Q3', description: '2013
Quarter3 starts', type: 'Flag',
        textStyle: { color: 'white' }, background: '#6c6d6d', border: {
color: '#6c6d6d' }, seriesIndexes: [0] },
      { date: new Date(2012, 9, 1), text: 'Q4', description: '2013
Quarter4 starts', type: 'Flag',
        textStyle: { color: 'white' }, background: '#6c6d6d', border: {
color: '#6c6d6d' }, seriesIndexes: [0] },
      { date: new Date(2012, 7, 30), text: 'G', description: 'Google
stocks bought',
        textStyle: { color: 'white' }, background: '#f48a21', border: {
color: '#f48a21' }, seriesIndexes: [0] },
      { date: new Date(2012, 10, 1), text: 'Y', description: 'Yahoo
stocks sold', type: 'Square',
        textStyle: { color: 'white' }, background: '#841391', border: {
color: '#841391' }, seriesIndexes: [0] },
      { date: new Date(2012, 12, 0), text: 'Y2', description: 'Year
2013', type: 'Pin', showOnSeries: false,
        textStyle: { color: 'white' }, background: '#6322e0', border: {
color: '#6322e0' }, seriesIndexes: [0] },
      { date: new Date(2013, 3, 1), text: 'Q2', description: '2013
Quarter2 starts', type: 'Flag',
        textStyle: { color: 'white' }, background: '#6c6d6d', border: {
color: '#6c6d6d' }, seriesIndexes: [0] },
      { date: new Date(2013, 3, 20), text: 'Q2', description: 'Surge in
Stocks', type: 'ArrowUp',
        textStyle: { color: 'white' }, background: '#3ab0f9', border: {
color: '#3ab0f9' }, seriesIndexes: [1] },
      { date: new Date(2013, 6, 1), text: 'Q3', description: '2013
Quarter3 starts', type: 'Flag',

```

```

        textStyle: { color: 'white' }, background: '#6c6d6d', border: {
color: '#6c6d6d' }, seriesIndexes: [1] },
        { date: new Date(2013, 9, 1), text: 'Q4', description: '2013
Quarter4 starts', type: 'Flag',
        textStyle: { color: 'white' }, background: '#6c6d6d', border: {
color: '#6c6d6d' }, seriesIndexes: [1] },
        { date: new Date(2013, 12, 0), text: 'Y3', description: 'Year
2014', type: 'Pin', showOnSeries: false,
        textStyle: { color: 'white' }, background: '#6322e0', border: {
color: '#6322e0' }, seriesIndexes: [1] },
        { date: new Date(2014, 3, 1), text: 'Q2', description: '2014
Quarter2 starts', type: 'ArrowDown',
        textStyle: { color: 'white' }, background: '#3ab0f9', border: {
color: '#3ab0f9' }, seriesIndexes: [1] },
        { date: new Date(2014, 6, 1), text: 'Q3', description: '2014
Quarter3 starts',
        textStyle: { color: 'white' }, background: '#f48a21', border: {
color: '#f48a21' }, seriesIndexes: [2] },
        { date: new Date(2014, 9, 1), text: 'Q4', description: '2014
Quarter4 starts', type: 'Flag',
        textStyle: { color: 'white' }, background: '#6c6d6d', border: {
color: '#6c6d6d' }, seriesIndexes: [2] },
        { date: new Date(2014, 12, 0), text: 'Y4', description: 'Year
2015', type: 'Pin', showOnSeries: false,
        textStyle: { color: 'white' }, background: '#6322e0', border: {
color: '#6322e0' }, seriesIndexes: [2] },
        { date: new Date(2014, 2, 2), text: 'End', description: 'Markets
closed', type: 'ArrowDown',
        textStyle: { color: 'white' }, background: '#3ab0f9', border: {
color: '#3ab0f9' }, seriesIndexes: [2] },
        { date: new Date('2015-01-07'), text: 'A', description: 'This is
event description',
        textStyle: { color: 'white' }, background: '#f48a21', border: {
color: '#f48a21' }, seriesIndexes: [2] },
        { date: new Date(2015, 1, 2), text: 'Q1', description: 'Add
longer text',
        textStyle: { color: 'white' }, background: '#dd3c9f', border: {
color: '#dd3c9f' }, type: 'Text', seriesIndexes: [2] },
        { date: new Date(2015, 2, 12), text: 'Close', description:
'Markets closed',
        textStyle: { color: 'white' }, background: '#f48a21', border: {
color: '#f48a21' }, seriesIndexes: [2] }
    ]}
    >
    <Inject services={[DateTime, StripLine, LineSeries, Crosshair,
SplineSeries, CandleSeries, HiloOpenCloseSeries, HiloSeries, RangeAreaSeries,
Trendlines,
        EmaIndicator, RsiIndicator, BollingerBands, TmaIndicator,
MomentumIndicator, SmaIndicator, AtrIndicator, Export,
        AccumulationDistributionIndicator, MacdIndicator,
StochasticIndicator]} />
    <StockChartSeriesCollectionDirective>
        <StockChartSeriesDirective dataSource={chartData}
type='Spline' xName='date' width={2} yName='high'>
        </StockChartSeriesDirective>
        <StockChartSeriesDirective dataSource={chartData}
type='Spline' xName='date' width={2} yName='low'>

```

```

        </StockChartSeriesDirective>
        <StockChartSeriesDirective dataSource={chartData}
type='Spline' xName='date' width={2} yName='open'>
        </StockChartSeriesDirective>
    </StockChartSeriesCollectionDirective>
</StockChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

See Also

- [Series Types](#)

### Accessibility in React Stock chart component

The Stock chart component followed the accessibility guidelines and standards, including [ADA, Section 508](#), [WCAG 2.2](#) standards, and [WCAG roles](#) that are commonly used to evaluate accessibility.

The accessibility compliance for the Stock chart component is outlined below.

| Accessibility Criteria | Compatibility |

| -- | -- |

| [WCAG 2.2](#) Support |  |

| [Section 508](#) Support |  |

| Screen Reader Support |  |

| Right-To-Left Support |  |

| Color Contrast |  |

| Mobile Device Support |  |

| Keyboard Navigation Support |  |

| [Accessibility Checker](#) Validation |  |

| [Axe-core](#) Accessibility Validation |  |

<style>

.post .post-content img {

display: inline-block;

```
margin: 0.5em 0;
}
```

```
</style>
```

```
<div> - All
features of the component meet the requirement.</div>
```

```
<div> - Some features of the component do not meet the requirement.</div>
```

```
<div> - The component does not meet the requirement.</div>
```

### WAI-ARIA attributes

The Stock chart component followed the [WAI-ARIA](#) patterns to meet the accessibility. The following ARIA attributes are used in the Stock chart component:

- `img` (role)
- `button` (role)
- `region` (role)
- `aria-label` (attribute)
- `aria-hidden` (attribute)
- `aria-pressed` (attribute)

### Keyboard interaction

The Stock chart component followed the [keyboard interaction](#) guideline, making it easy for people who use assistive technologies (AT) and those who completely rely on keyboard navigation. The following keyboard shortcuts are supported by the Stock chart component.

| **Press** | **To do this** |

| --- | --- |

| **Alt + J** | Moves the focus to the Stock chart element. |

| **Tab** | Moves the focus to the next element in the Stock chart. |

| **Shift + Tab** | Moves the focus to the previous element in the Stock chart. |

| **Down Arrow** | Moves the focus to the data point left side from the selected point. |

| **Up Arrow** | Moves the focus to the data point right side from the selected point. |

| **ESC** | Cancel the tooltip for the data point. |

| **Ctrl + P** | Prints the Stock chart. |

### Ensuring accessibility

The Stock chart component's accessibility levels are ensured through an [accessibility-checker](#) and [axe-core](#) software tools during automated testing.

The accessibility compliance of the Stock chart component is shown in the following sample. Open the [sample](#) in a new window to evaluate the accessibility of the Stock chart component with accessibility tools.



See also

- [Accessibility in Syncfusion React components](#)

## Switch

### Getting Started

This section explains how to create a simple Switch, and configure its available functionalities in React, using React quickstart application.

### Dependencies

The following list of dependencies are required to use Switch component in your application.

```
`javascript
|-- @syncfusion/ej2-react-buttons
|-- @syncfusion/ej2-react-base
|-- @syncfusion/ej2-base
|-- @syncfusion/ej2-buttons
\`
```

### Installation and Configuration

You can use [create-react-app](#) to setup the applications. To install `create-react-app` run the following command.

```
`bash
npm install -g create-react-app
\`
```

To set-up a React application in TypeScript environment, run the following command.

```
`bash
npx create-react-app my-app --template typescript
cd my-app
npm start
\`
```

To set-up a React application in JavaScript environment, run the following command.

```
`bash
npx create-react-app my-app
cd my-app
npm start
\`
```

### Adding Syncfusion packages

All the available Essential JS 2 packages are published in [npmjs.com](#) public registry.

To install Switch component, use the following command

```
`bash
npm install @syncfusion/ej2-react-buttons --save
`
```

### Adding CSS Reference

Import the Switch component's required CSS references as follows in `src/App.css`.

```
`css
@import "../node_modules/@syncfusion/ej2-base/styles/material.css";
@import "../node_modules/@syncfusion/ej2-buttons/styles/material.css";
`
```

### Adding Switch component to the Application

To include Switch component in your application import the `SwitchComponent` from `ej2-react-buttons` package in `App.tsx`.

Add Switch component in application as shown in below code example.

```
`ts
// Import the Switch.
import { SwitchComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import './App.css';
// To render Switch.
function App() {
  return ( <div style={{marginTop: '150px'}}><SwitchComponent /></div>);
}
export default App;
`
```

### Run the application

Run the application in the browser using the following command:

```
`npm start
`
```

The following example shows a basic Switch component.

### **APP.JSX**

```
import { enableRipple } from '@syncfusion/ej2-base';
import { SwitchComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
```

```
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render Switch with checked state.
function App() {
  return (<SwitchComponent checked={true}/>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('switch'));
```

### APP.TSX

```
import { enableRipple } from '@syncfusion/ej2-base';
import { SwitchComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render Switch with checked state.
function App() {
  return (
    <SwitchComponent checked={true} />
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('switch'));
```

### See Also

- [How to customize the switch appearance](#)

### Accessibility in React Switch component

The Switch component followed the accessibility guidelines and standards, including [ADA](#), [Section 508](#), [WCAG 2.2](#) standards, and [WCAG roles](#) that are commonly used to evaluate accessibility.

The accessibility compliance for the Switch component is outlined below.

| Accessibility Criteria | Compatibility |

| -- | -- |

| [WCAG 2.2](#) Support |  |

| [Section 508](#) Support |  |

| Screen Reader Support |  |

| Right-To-Left Support |  |

| Color Contrast |  |

```

| Mobile Device Support |  |

| Keyboard Navigation Support |  |

| Accessibility Checker Validation |  |

| Axe-core Accessibility Validation |  |

<style>

.post .post-content img {
display: inline-block;
margin: 0.5em 0;
}

</style>

<div> - All
features of the component meet the requirement.</div>

<div> - Some features of the component do not meet the requirement.</div>

<div> - The component does not meet the requirement.</div>

```

### WAI-ARIA attributes

The Switch component followed the [WAI-ARIA](#) patterns to meet the accessibility. The following ARIA attributes are used in the Switch component:

| Attributes | Purpose |

| --- | --- |

| **role** | Indicates the switch component. |

| **aria-disabled** | Indicates that the element is perceivable but disabled, so it is not editable or otherwise operable. |

### Keyboard interaction

The Switch component followed the [keyboard interaction](#) guideline, making it easy for people who use assistive technologies (AT) and those who completely rely on keyboard navigation. The following keyboard shortcuts are supported by the Switch component.

| **Press** | **To do this** |

| --- | --- |

| **Space** | When the switch has focus, pressing the Space key changes the state of the switch. |

### Ensuring accessibility

The Switch component's accessibility levels are ensured through an [accessibility-checker](#) and [axe-core](#) software tools during automated testing.

The accessibility compliance of the Switch component is shown in the following sample. Open the [sample](#) in a new window to evaluate the accessibility of the Switch component with accessibility tools.

See also

- [Accessibility in Syncfusion React components](#)

### Style and appearance in React Switch component

To modify the Switch appearance, you need to override the default CSS of Switch component. Please find the list of CSS classes and its corresponding section in Switch. Also, you have an option to create your own custom theme for the controls using our [Theme Studio](#).

#### CSS Class | Purpose of Class

`.e-switch-wrapper .e-switch-inner` | To customize the line of the switch in off mode

`.e-switch-wrapper .e-switch-handle` | To customize the handle of the switch in off mode

`.e-switch-wrapper:not(.e-switch-disabled):hover .e-switch-handle:not(.e-switch-active)` | To customize the handle of the switch in off mode when hover

`.e-switch-wrapper:not(.e-switch-disabled):hover .e-switch-inner:not(.e-switch-active)` | To customize the line of the switch in off mode when hover

`.e-switch-wrapper .e-switch-handle.e-switch-active` | To customize the handle of the switch in on mode

`.e-switch-wrapper .e-switch-on` | To customize the line of the switch in on mode

`.e-switch-wrapper:hover .e-switch-handle.e-switch-active` | To customize the handle of the switch in on mode when hover

`.e-switch-wrapper:hover .e-switch-inner.e-switch-active .e-switch-on` | To customize the line of the switch in on mode when hover

### How To

#### Set text in React Switch component

This section explains how to set [onLabel](#) and [offLabel](#) texts on Switch. In the following example, `onLabel` is set as `ON` and `offLabel` is set as `OFF`.

#### APP.JSX

```
import { SwitchComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render Switch.
function App() {
    return (<SwitchComponent onLabel="ON" offLabel="OFF" checked={true}/>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('switch'));
```

**APP.TSX**

```
import { SwitchComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render Switch.
function App() {
  return (
    <SwitchComponent onLabel="ON" offLabel="OFF" checked={true} />
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('switch'));
```

Switch does not have text support for material themes, and does not support long custom text.

## Change size in React Switch component

The different Switch sizes available are default and small. To reduce the size of default Switch to small, set the [cssClass](#) property to `e-small`.

**APP.JSX**

```
import { enableRipple } from '@syncfusion/ej2-base';
import { SwitchComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render Switch.
function App() {
  return (<table className='size'>
    <tr>
      <td className='lSize'>Small</td>
      <td>
        <SwitchComponent cssClass="e-small"/>
      </td>
    </tr>
    <tr>
      <td className='lSize'>Default</td>
      <td>
        <SwitchComponent />
      </td>
    </tr>
  </table>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('switch'));
```

**APP.TSX**

```
import { enableRipple } from '@syncfusion/ej2-base';
import { SwitchComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render Switch.
function App() {
```

```

return (
  <table className='size'>
    <tr>
      <td className='lSize'>Small</td>
      <td>
        <SwitchComponent cssClass="e-small" />
      </td>
    </tr>
    <tr>
      <td className='lSize'>Default</td>
      <td>
        <SwitchComponent />
      </td>
    </tr>
  </table>
);
}
export default App;
ReactDOM.render(<App />, document.getElementById('switch'));

```

### Customize the appearance of a switch in React Switch component

You can customize the appearance of the Switch component using the CSS rules. Define your own CSS rules according to your requirement and assign the class name to the [cssClass](#) property.

#### Customize Switch bar and handle

Switch bar and handle can be customized as per requirement using CSS rules. Switch bar and handle customized using `cssClass` property. In the following sample, the `border-radius` CSS property for the Switch bar (`e-switch-inner`) and handle (`e-switch-handle`) elements was changed border radius circle to square shape.

#### APP.JSX

```

{% raw %}
import { SwitchComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render Switch.
function App() {
  return (
    <div className="switch-control">
      <div>
        <h3>Customizing Shape</h3>
      </div>
      <div>
        <label htmlFor="switch1" style={{ padding: "10px 85px 10px 7px" }}> Square Switch </label>
        <SwitchComponent id="switch1" cssClass="square"/>
      </div>
      <div>
        <label htmlFor="switch2" style={{ padding: "10px 76px 10px 7px" }}> Bar and handle </label>
        <SwitchComponent id="switch2" cssClass="custom-switch" checked={true}/>
      </div>
    </div>
  );
}

```

```

        <label htmlFor='switch3' style={{ padding: "10px 96px
10px 7px" }}> Handle text </label>
        <SwitchComponent id="switch3" cssClass="handle-text"/>
      </div>
    </div>);
  }
  export default App;
  ReactDOM.render(<App />, document.getElementById('switch'));
  {% endraw %}

```

## APP.TSX

```

{% raw %}
import { SwitchComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render Switch.
function App() {
  return (
    <div className="switch-control">
      <div>
        <h3>Customizing Shape</h3>
      </div>
      <div>
        <label htmlFor="switch1" style={{ padding: "10px 85px
10px 7px" }}> Square Switch </label>
        <SwitchComponent id="switch1" cssClass="square" />
      </div>
      <div>
        <label htmlFor='switch2' style= {{ padding: "10px 76px
10px 7px" }}> Bar and handle </label>
        <SwitchComponent id="switch2" cssClass="custom-switch"
checked={true} />
      </div>
      <div>
        <label htmlFor='switch3' style= {{ padding: "10px 96px
10px 7px" }}> Handle text </label>
        <SwitchComponent id="switch3" cssClass="handle-text" />
      </div>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('switch'));
{% endraw %}

```

### Color the Switch

Switch colors can be customized as per the requirement using CSS rules. Switch bar and handle colors customized using `cssClass` property. In the following sample, the Switch bar (`e-switch-inner`) element background and border colors were changed from default colors.

## APP.JSX

```

{% raw %}
import * as React from 'react';

```



```

import * as ReactDOM from 'react-dom';
import { SwitchComponent } from '@syncfusion/ej2-react-buttons';
// To render Switch.
function App() {
    return (<div className="switch-control">
        <div>
            <h3>Customizing Color</h3>
        </div>
        <div>
            <label htmlFor="switch1" style={{ padding: "10px 85px 10px 0"
}}> Custom color </label>
            <SwitchComponent id="switch1" cssClass='bar-color' />
        </div>
        <div>
            <label htmlFor='switch2' style={{ padding: "10px 90px 10px 0"
}}> Handle color </label>
            <SwitchComponent id="switch2" cssClass='handle-color'
checked={true} />
        </div>
        <div>
            <label htmlFor='switch3' style={{ padding: "10px 103px 10px
0" }}> iOS Switch </label>
            <SwitchComponent id="switch3" cssClass='custom-iOS'
checked={true} />
        </div>
    </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('switch'));
{% endraw %}

```

**APP.TSX**

```

{% raw %}
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { SwitchComponent } from '@syncfusion/ej2-react-buttons';
// To render Switch.
function App() {
    return (
        <div className="switch-control">
            <div>
                <h3>Customizing Color</h3>
            </div>
            <div>
                <label htmlFor="switch1" style={{ padding: "10px 85px 10px 0"
}}> Custom color </label>
                <SwitchComponent id="switch1" cssClass='bar-color' />
            </div>
            <div>
                <label htmlFor='switch2' style={{ padding: "10px 90px 10px 0"
}}> Handle color </label>
                <SwitchComponent id="switch2" cssClass='handle-color'
checked={true} />
            </div>
            <div>

```

```

        <label htmlFor='switch3' style={{ padding: "10px 103px 10px
0" }}> iOS Switch </label>
        <SwitchComponent id="switch3" cssClass='custom-iOS'
checked={true} />
    </div>
</div>
);
}
export default App;
ReactDOM.render(<App />, document.getElementById('switch'));
{% endraw %}

```

### Enable ripple for switch label in React Switch component

By default, label with ripple effect is not available in Switch. You can achieve this by using `rippleMouseHandler` method.

The following example illustrates how to enable ripple effect for labels in Switch component.

#### APP.JSX

```

import { enableRipple } from '@syncfusion/ej2-base';
import { rippleMouseHandler } from '@syncfusion/ej2-buttons';
import { SwitchComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render Switch with checked state.
function App() {
    function onCreated() {
        // Function to handle ripple effect for Switch with label.
        document.querySelector('.lSize label').addEventListener('mouseup',
rippleHandler);
        document.querySelector('.lSize label').addEventListener('mousedown',
rippleHandler);
        function rippleHandler(e) {
            const rippleSpan =
e.target.parentElement.nextElementSibling.querySelector('.e-ripple-
container');
            if (rippleSpan) {
                rippleMouseHandler(e, rippleSpan);
            }
        }
    }
    return (<table className='size'>
        <tr>
            <td className='lSize'>
                <label htmlFor='switch1'>USB Tethering</label>
            </td>
            <td>
                <SwitchComponent id="switch1" created={onCreated}
checked={true}/>
            </td>
        </tr>
    </table>);
}
export default App;

```

```
ReactDOM.render(<App />, document.getElementById('switch'));
```

### APP.TSX

```
import { enableRipple } from '@syncfusion/ej2-base';
import { rippleMouseHandler } from '@syncfusion/ej2-buttons';
import { SwitchComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render Switch with checked state.
function App() {
    function onCreated(): void {
        // Function to handle ripple effect for Switch with label.
        (document.querySelector('.lSize label') as
        HTMLElement).addEventListener('mouseup', rippleHandler);
        (document.querySelector('.lSize label') as
        HTMLElement).addEventListener('mousedown', rippleHandler);
        function rippleHandler(e: MouseEvent): void {
            const rippleSpan = (e.target as
            HTMLElement).parentElement.nextElementSibling.querySelector('.e-ripple-
            container');
            if (rippleSpan) {
                rippleMouseHandler(e, rippleSpan);
            }
        }
    }
    return (
        <table className='size'>
            <tr>
                <td className='lSize'>
                    <label htmlFor='switch1'>USB Tethering</label>
                </td>
                <td>
                    <SwitchComponent id="switch1" created={onCreated}
checked={true} />
                </td>
            </tr>
        </table>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('switch'));
```

While accessing HTML Elements we got the exception 'object is possibly null' due to 'strictNullChecks' option. So you can disable it in 'tsconfig.json' file.

### Enable rtl in React Switch component

Switch component has RTL support. This can be achieved by setting [enableRtl](#) as `true`.

The following example illustrates how to enable right-to-left support in Switch component.

### APP.JSX

```
import { enableRipple } from '@syncfusion/ej2-base';
import { SwitchComponent } from '@syncfusion/ej2-react-buttons';
```

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render Switch with checked state.
function App() {
    return (<SwitchComponent enableRtl={true} checked={true}/>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('switch'));
```

**APP.TSX**

```
import { enableRipple } from '@syncfusion/ej2-base';
import { SwitchComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render Switch with checked state.
function App() {
    return (
        <SwitchComponent enableRtl={true} checked={true} />
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('switch'));
```

## Set disabled state in React Switch component

Switch can be disabled by setting the [disabled](#) property to true.

The following example illustrates how to disable support in Switch component.

**APP.JSX**

```
import { enableRipple } from '@syncfusion/ej2-base';
import { SwitchComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render Switch with checked state.
function App() {
    return (<SwitchComponent disabled={true}/>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('switch'));
```

**APP.TSX**

```
import { enableRipple } from '@syncfusion/ej2-base';
import { SwitchComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render Switch with checked state.
function App() {
    return (
```

```

    <SwitchComponent disabled={true} />
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('switch'));

```

### Submit name and value in form in React Switch component

The name attribute of the Switch is used to group Switches. When the Switches are grouped in form, the checked items value attribute will post to the server on form submit. The disabled and unchecked Switch values will not be sent to the server on form submit.

In the following code snippet, USB and Wi-Fi in the checked state, and Bluetooth is in disabled state. Values that are in checked state only be sent on form submit.

### APP.JSX

```

{% raw %}
import { enableRipple } from '@syncfusion/ej2-base';
import { ButtonComponent, SwitchComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render Switch.
function App() {
  return (
    <form>
      <div className="switch-control">
        <div>
          <h3>Form Submit</h3>
        </div>
        <div>
          <label htmlFor="switch1" style={{ padding: "10px 85px 10px 0" }}> USB </label>
          <SwitchComponent id="switch1" name='Tethering' value='USB' checked={true}/>
        </div>
        <div>
          <label htmlFor="switch2" style={{ padding: "10px 80px 10px 0" }}> Wi-Fi </label>
          <SwitchComponent id="switch2" name='Hotspot' value='Wi-Fi' checked={true}/>
        </div>
        <div>
          <label htmlFor="switch3" style={{ padding: "10px 53px 10px 0" }}> Bluetooth </label>
          <SwitchComponent id="switch3" name='Tethering' value='Bluetooth' disabled={true}/>
        </div>
        <div>
          <ButtonComponent isPrimary={true}>Submit</ButtonComponent>
        </div>
      </div>
    </form>
  );
}
export default App;

```

```
ReactDOM.render(<App />, document.getElementById('switch'));
{% endraw %}
```

### APP.TSX

```
{% raw %}
import { enableRipple } from '@syncfusion/ej2-base';
import { ButtonComponent, SwitchComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render Switch.
function App() {
    return (
        <form>
            <div className="switch-control">
                <div>
                    <h3>Form Submit</h3>
                </div>
                <div>
                    <label htmlFor="switch1" style={{ padding: "10px 85px 10px 0" }}> USB </label>
                    <SwitchComponent id="switch1" name='Tethering' value='USB' checked={true} />
                </div>
                <div>
                    <label htmlFor="switch2" style={{ padding: "10px 80px 10px 0" }}> Wi-Fi </label>
                    <SwitchComponent id="switch2" name='Hotspot' value='Wi-Fi' checked={true} />
                </div>
                <div>
                    <label htmlFor="switch3" style={{ padding: "10px 53px 10px 0" }}> Bluetooth </label>
                    <SwitchComponent id="switch3" name='Tethering' value='Bluetooth' disabled={true} />
                </div>
                <div>
                    <ButtonComponent
                        isPrimary={true}>Submit</ButtonComponent>
                </div>
            </div>
        </form>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('switch'));
{% endraw %}
```

Change switch state using toggle method in React Switch component

This section explains about how to toggle between the switch states using [toggle](#) method.

### APP.JSX

```
{% raw %}
```

```
import { enableRipple } from '@syncfusion/ej2-base';
import { SwitchComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render Switch with checked state.
function App() {
    let switchRef;
    function created() {
        switchRef.toggle();
    }
    return (<SwitchComponent id="switch" ref={(scope) => { switchRef = scope;
}} enableRtl={true} checked={true} created={created}/>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('switch'));
{% endraw %}
```

### APP.TSX

```
{% raw %}
import { enableRipple } from '@syncfusion/ej2-base';
import { SwitchComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render Switch with checked state.
function App() {
    let switchRef: SwitchComponent;
    function created(): void {
        switchRef.toggle();
    }
    return (
        <SwitchComponent id="switch" ref={(scope) => { switchRef = scope as
SwitchComponent; }} enableRtl={true} checked={true} created={created}/>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('switch'));
{% endraw %}
```

Switch triggers [change](#) event on every state stage to perform custom operations.

## Tab

### Getting Started

This section briefly explains about how to create a simple [Link to the Video](#) component and configuring the Tab header content in React.

To get started quickly with React Tabs using Create React App, you can check out this video:

### Dependencies

The following is the list of dependencies required to use the Tab component in your application.

```
`javascript
```

```
|-- @syncfusion/ej2-react-navigations
|-- @syncfusion/ej2-base
|-- @syncfusion/ej2-react-base
|-- @syncfusion/ej2-navigations
|-- @syncfusion/ej2-buttons
|-- @syncfusion/ej2-popups
\
```

### Setup for Local Development

To set-up a React application, choose any of the following ways. The best and easiest way is to use the [create-react-app](#). It sets up your development environment in JavaScript and improvise your application for production. Refer to the [installation instructions](#) of `create-react-app`.

```
`bash
npx create-react-app my-app
cd my-app
npm start
\
```

or

```
`bash
yarn create react-app my-app
cd my-app
yarn start
\
```

To set-up a React application in `TypeScript` environment, run the following command.

```
`bash
npx create-react-app my-app --template typescript
cd my-app
npm start
\
```

Besides using the [npm](#) package runner tool, also create an application from the `npm init`. To begin with the `npm init`, upgrade the `npm` version to `npm 6+`.

```
`bash
npm init react-app my-app
cd my-app
npm start
```



### Adding Syncfusion packages

All the available Essential JS 2 packages are published in [npmjs.com](https://www.npmjs.com) public registry.

To install Tab component, use the following command

```
npm install @syncfusion/ej2-react-navigations --save
```

### Adding CSS reference

Add components style as given below in `src/App.css`.

```
`css
@import '../node_modules/@syncfusion/ej2-base/styles/material.css';
@import '../node_modules/@syncfusion/ej2-buttons/styles/material.css';
@import '../node_modules/@syncfusion/ej2-popups/styles/material.css';
@import '../node_modules/@syncfusion/ej2-react-navigations/styles/material.css';
```

To refer `App.css` in the application then import it in the `src/App.tsx` file

### Initialize the Tab using JSON items collection

The Tab can be rendered by defining a JSON array. The item is rendered with [header](#) text and [content](#) for each Tab.

- Import the Tab component to your `src/App.tsx` file using following code.

### INDEX.JSX

```
import { TabComponent, TabItemDirective, TabItemsDirective } from
 '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
const ReactApp = () => {
  let headerText = [{ text: "Twitter" }, { text: "Facebook" }, { text:
    "WhatsApp" }];
  const content0 = () => {
    return <div>
      Twitter is an online social networking service that enables users to
      send and read short 140-character messages called "tweets". Registered users
      can read and post tweets, but those who are unregistered can only read them.
      Users access Twitter through the website interface, SMS or mobile device app
      Twitter Inc. is based in San Francisco and has more than 25 offices around
      the world. Twitter was created in March 2006 by Jack Dorsey, Evan Williams,
      Biz Stone, and Noah Glass and launched in July 2006. The service rapidly
      gained worldwide popularity, with more than 100 million users posting 340
      million tweets a day in 2012. The service also handled 1.6 billion search
      queries per day.
    </div>;
  }
}
```

```

const content1 = () => {
  return <div>
    Facebook is an online social networking service headquartered in Menlo
    Park, California. Its website was launched on February 4, 2004, by Mark
    Zuckerberg with his Harvard College roommates and fellow students Eduardo
    Saverin, Andrew McCollum, Dustin Moskovitz and Chris Hughes. The founders had
    initially limited the website membership to Harvard students, but later
    expanded it to colleges in the Boston area, the Ivy League, and Stanford
    University. It gradually added support for students at various other
    universities and later to high-school students.
  </div>;
}
const content2 = () => {
  return <div>
    WhatsApp Messenger is a proprietary cross-platform instant messaging
    client for smartphones that operates under a subscription business model. It
    uses the Internet to send text messages, images, video, user location and
    audio media messages to other users using standard cellular mobile numbers.
    As of February 2016, WhatsApp had a user base of up to one billion, [10]
    making it the most globally popular messaging application. WhatsApp Inc.,
    based in Mountain View, California, was acquired by Facebook Inc. on February
    19, 2014, for approximately US$19.3 billion.
  </div>;
}
return (<TabComponent heightAdjustMode='Auto'>
  <TabItemsDirective>
    <TabItemDirective header={headerText[0]} content={content0}/>
    <TabItemDirective header={headerText[1]} content={content1}/>
    <TabItemDirective header={headerText[2]} content={content2}/>
  </TabItemsDirective>
</TabComponent>);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

## INDEX.TSX

```

import { TabComponent, TabItemDirective, TabItemsDirective } from
'@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
const ReactApp = () => {
  let headerText: any = [{ text: "Twitter" }, { text: "Facebook" }, { text:
  "WhatsApp" }];
  const content0 = () => {
    return <div>
      Twitter is an online social networking service that enables users to
      send and read short 140-character messages called "tweets". Registered users
      can read and post tweets, but those who are unregistered can only read them.
      Users access Twitter through the website interface, SMS or mobile device app
      Twitter Inc. is based in San Francisco and has more than 25 offices around
      the world. Twitter was created in March 2006 by Jack Dorsey, Evan Williams,
      Biz Stone, and Noah Glass and launched in July 2006. The service rapidly
      gained worldwide popularity, with more than 100 million users posting 340
      million tweets a day in 2012. The service also handled 1.6 billion search
      queries per day.
    </div>
  }
}

```

```

    </div>;
  }
  const content1 = () => {
    return <div>
      Facebook is an online social networking service headquartered in Menlo
      Park, California. Its website was launched on February 4, 2004, by Mark
      Zuckerberg with his Harvard College roommates and fellow students Eduardo
      Saverin, Andrew McCollum, Dustin Moskovitz and Chris Hughes. The founders had
      initially limited the website membership to Harvard students, but later
      expanded it to colleges in the Boston area, the Ivy League, and Stanford
      University. It gradually added support for students at various other
      universities and later to high-school students.
    </div>;
  }
  const content2 = () => {
    return <div>
      WhatsApp Messenger is a proprietary cross-platform instant messaging
      client for smartphones that operates under a subscription business model. It
      uses the Internet to send text messages, images, video, user location and
      audio media messages to other users using standard cellular mobile numbers.
      As of February 2016, WhatsApp had a user base of up to one billion, [10]
      making it the most globally popular messaging application. WhatsApp Inc.,
      based in Mountain View, California, was acquired by Facebook Inc. on February
      19, 2014, for approximately US$19.3 billion.
    </div>;
  }
  return (
    <TabComponent heightAdjustMode='Auto'>
      <TabItemsDirective>
        <TabItemDirective header={headerText[0]} content={content0} />
        <TabItemDirective header={headerText[1]} content={content1} />
        <TabItemDirective header={headerText[2]} content={content2} />
      </TabItemsDirective>
    </TabComponent>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

- Run the application in the browser using the following command.

```

,
npm start
,

```

In the above sample code, **element** is the **id** of the HTML element in a page to which the Tab is initialized.

### Initialize the Tab using HTML elements

The Tab component can be rendered based on the given HTML element using **<TabComponent>** tag. Header section must be enclosed with in a wrapper element using **e-tab-header** class and corresponding content must be mapped with **e-content** class.

You need to follow the below structure of HTML elements to render the Tab,

`<TabComponent id='defaultTab'>` --> Root Tab element

`<div class="e-tab-header">` --> Tab header

`<div>` --> Header Item

`</div>`

`</div>`

`<div class="e-content">` --> Tab content

`<div>` --> Content Item

`</div>`

`</div>`

`</TabComponent>`

## INDEX.JSX

```
import { TabComponent } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
const ReactApp = () => {
  return (<TabComponent id='defaultTab'>
    <div className="e-tab-header">
      <div> Twitter </div>
      <div> Facebook </div>
      <div> WhatsApp </div>
    </div>
    <div className="e-content">
      <div>
        Twitter is an online social networking service that enables users
        to send and read short 140-character messages called 'tweets'. Registered
        users can read and post tweets, but those who are unregistered can only read
        them. Users access Twitter through the website interface, SMS or mobile
        device app Twitter Inc. is based in San Francisco and has more than 25
        offices around the world. Twitter was created in March 2006 by Jack Dorsey,
        Evan Williams, Biz Stone, and Noah Glass and launched in July 2006. The
        service rapidly gained worldwide popularity, with more than 100 million users
        posting 340 million tweets a day in 2012.The service also handled 1.6 billion
        search queries per day.
      </div>
      <div>
        Facebook is an online social networking service headquartered in
        Menlo Park, California. Its website was launched on February 4, 2004, by Mark
        Zuckerberg with his Harvard College roommates and fellow students Eduardo
        Saverin, Andrew McCollum, Dustin Moskovitz and Chris Hughes.The founders had
        initially limited the website's membership to Harvard students, but later
        expanded it to colleges in the Boston area, the Ivy League, and Stanford
        University. It gradually added support for students at various other
        universities and later to high-school students.
      </div>
    </div>
  </TabComponent>)
```

```

    <div>
      WhatsApp Messenger is a proprietary cross-platform instant
      messaging client for smartphones that operates under a subscription business
      model. It uses the Internet to send text messages, images, video, user
      location and audio media messages to other users using standard cellular
      mobile numbers. As of February 2016, WhatsApp had a user base of up to one
      billion, [10] making it the most globally popular messaging application.
      WhatsApp Inc., based in Mountain View, California, was acquired by Facebook
      Inc. on February 19, 2014, for approximately US$19.3 billion.
    </div>
  </div>
</TabComponent>);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

## INDEX.TSX

```

import { TabComponent } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
const ReactApp = () => {
  return (
    <TabComponent id='defaultTab'>
      <div className="e-tab-header">
        <div> Twitter </div>
        <div> Facebook </div>
        <div> WhatsApp </div>
      </div>
      <div className="e-content">
        <div>
          Twitter is an online social networking service that enables users
          to send and read short 140-character messages called 'tweets'. Registered
          users can read and post tweets, but those who are unregistered can only read
          them. Users access Twitter through the website interface, SMS or mobile
          device app Twitter Inc. is based in San Francisco and has more than 25
          offices around the world. Twitter was created in March 2006 by Jack Dorsey,
          Evan Williams, Biz Stone, and Noah Glass and launched in July 2006. The
          service rapidly gained worldwide popularity, with more than 100 million users
          posting 340 million tweets a day in 2012. The service also handled 1.6 billion
          search queries per day.
        </div>
        <div>
          Facebook is an online social networking service headquartered in
          Menlo Park, California. Its website was launched on February 4, 2004, by Mark
          Zuckerberg with his Harvard College roommates and fellow students Eduardo
          Saverin, Andrew McCollum, Dustin Moskovitz and Chris Hughes. The founders had
          initially limited the website's membership to Harvard students, but later
          expanded it to colleges in the Boston area, the Ivy League, and Stanford
          University. It gradually added support for students at various other
          universities and later to high-school students.
        </div>
        <div>
          WhatsApp Messenger is a proprietary cross-platform instant
          messaging client for smartphones that operates under a subscription business
          model. It uses the Internet to send text messages, images, video, user

```

```
location and audio media messages to other users using standard cellular
mobile numbers. As of February 2016, WhatsApp had a user base of up to one
billion,[10] making it the most globally popular messaging application.
WhatsApp Inc., based in Mountain View, California, was acquired by Facebook
Inc. on February 19, 2014, for approximately US$19.3 billion.
    </div>
  </div>
</TabComponent>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);
```

## See Also

- [How to load tab with DataSource](#)

## Adaptive in React Tab component

The following section explains about rendering Tab when its width exceeds the viewable area or particularly in a given [width](#).

The available modes are as follows:

- Scrollable
- Popup

### Scrollable

The default overflow mode is Scrollable. Scrollable display mode supports displaying the Tab header items in a single line with horizontal scrolling enabled, when the item overflows to the available space.

- The right and left navigation arrow is added at the start and end of the Tab header through which user can navigate

towards overflowed items of the Tab header.

- You can also see the overflowed items using touch and swipe action on the header and content section.
- By default, navigation icon in the left direction is disabled, you can see the overflowed items by moving in the right direction.
- By clicking the arrow or by holding the arrow continuously, you can see the overflowed items.



- In devices the navigation icons are not available. You can touch and swipe to see the overflowed items of the Tab header.



## INDEX.JSX

```
import { TabComponent, TabItemDirective, TabItemsDirective } from
'@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
const ReactApp = () => {
  let headerText = [{ text: "HTML" }, { text: "C Sharp(C#)" }, { text: "Java"
}, { text: "VB.Net" }, { text: "Xamarin" }, { text: "ASP.NET" }, { text:
"ASP.NET MVC" }, { text: "JavaScript" }];
  const htmlContent = () => {
    return <div>
      HyperText Markup Language, commonly referred to as HTML, is the
standard markup
      language used to create web pages. Along with CSS, and JavaScript, HTML
is a cornerstone
      technology, used by most websites to create visually engaging web
pages, user interfaces
      for web applications, and user interfaces for many mobile
applications.[1] Web browsers
      can read HTML files and render them into visible or audible web pages.
HTML describes
      the structure of a website semantically along with cues for
presentation, making it a
      markup language, rather than a programming language.
    </div>;
  }
}
```

```

    }
    const csharpContent = () => {
      return <div>
        'C# is intended to be a simple, modern, general-purpose, object-
oriented
        programming language. Its development team is led by Anders Hejlsberg.
The most recent
        version is C# 5.0, which was released on August 15, 2012.
      </div>;
    }
    const javaContent = () => {
      return <div>
        Java is a set of computer software and specifications developed by Sun
Microsystems,
        later acquired by Oracle Corporation, that provides a system for
developing application
        software and deploying it in a cross-mobile phones to platform
computing environment. Java
        is used in a wide variety of computing platforms from embedded devices
and enterprise servers
        and supercomputers. While less common, Java applets run in secure,
sandboxed environments to
        provide many features of native applications and can be embedded in
HTML pages.
      </div>;
    }
    const vbcNetContent = () => {
      return <div>
        The command-line compiler, VBC.EXE, is installed as part of the
freeware .NET
        Framework SDK. Mono also includes a command-line VB.NET compiler. The
most recent version
        is VB 2012, which was released on August 15, 2012.
      </div>;
    }
    const xamarinContent = () => {
      return <div>
        Xamarin is a San Francisco, California based software company created
in May
        2011[3] by the engineers that created Mono,[4] Mono for Android and
MonoTouch that are
        cross-platform implementations of the Common Language Infrastructure
(CLI) and Common
        Language Specifications (often called Microsoft .NET). With a C#-shared
codebase, developers
        can use Xamarin tools to write native Android, iOS, and Windows apps
with native user interfaces
        and share code across multiple platforms.[5] Xamarin has over 1 million
developers in more
        than 120 countries around the World as of May 2015.'
      </div>;
    }
    const aspContent = () => {
      return <div>
        ASP.NET is an open-source server-side web application framework
designed for web

```



```

        development to produce dynamic web pages. It was developed by Microsoft
        to allow programmers
        to build dynamic web sites, web applications and web services. It was
        first released in January
        2002 with version 1.0 of the .NET Framework, and is the successor to
        Microsoft\'\s Active Server
        Pages (ASP) technology. ASP.NET is built on the Common Language Runtime
        (CLR), allowing
        programmers to write ASP.NET code using any supported .NET language.
        The ASP.NET SOAP extension
        framework allows ASP.NET components to process SOAP messages.
    </div>;
}
const mvcContent = () => {
    return <div>
        The ASP.NET MVC is a web application framework developed by Microsoft,
        which implements
        the model-view-controller (MVC) pattern. It is open-source software,
        apart from the ASP.NET Web
        Forms component which is proprietary. In the later versions of ASP.NET,
        ASP.NET MVC, ASP.NET web
        API, and ASP.NET Web Pages (a platform using only Razor pages) will
        merge into a unified MVC 6.
        The project is called ASP.NET vNext.
    </div>;
}
const jsContent = () => {
    return <div>
        JavaScript (JS) is an interpreted computer programming language. It was
        originally
        implemented as part of web browsers so that client-side scripts could
        interact with the
        user, control the browser, communicate asynchronously, and alter the
        document content that
        was displayed.[5] More recently, however, it has become common in both
        game development
        and the creation of desktop applications.
    </div>;
}
return (<TabComponent heightAdjustMode='Auto' overflowMode='Scrollable'>
    <TabItemsDirective>
        <TabItemDirective header={headerText[0]} content={htmlContent} />
        <TabItemDirective header={headerText[1]} content={csharpContent} />
        <TabItemDirective header={headerText[2]} content={javaContent} />
        <TabItemDirective header={headerText[3]} content={vbNetContent} />
        <TabItemDirective header={headerText[4]} content={xamarinContent} />
        <TabItemDirective header={headerText[5]} content={aspContent} />
        <TabItemDirective header={headerText[6]} content={mvcContent} />
        <TabItemDirective header={headerText[7]} content={jsContent} />
    </TabItemsDirective>
</TabComponent>);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

## INDEX.TSX

```

import { TabComponent, TabItemDirective, TabItemsDirective } from
 '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
const ReactApp = () => {
  let headerText: any = [{ text: "HTML" }, { text: "C Sharp(C#)" }, { text:
"Java" }, { text: "VB.Net" }, { text: "Xamarin" }, { text: "ASP.NET" }, {
text: "ASP.NET MVC" }, { text: "JavaScript" }];
  const htmlContent = () => {
    return <div>
      HyperText Markup Language, commonly referred to as HTML, is the standard
markup
      language used to create web pages. Along with CSS, and JavaScript, HTML is
a cornerstone
      technology, used by most websites to create visually engaging web pages,
user interfaces
      for web applications, and user interfaces for many mobile applications.[1]
Web browsers
      can read HTML files and render them into visible or audible web pages. HTML
describes
      the structure of a website semantically along with cues for presentation,
making it a
      markup language, rather than a programming language.
    </div>;
  }
  const csharpContent = () => {
    return <div>
      'C# is intended to be a simple, modern, general-purpose, object-oriented
programming language. Its development team is led by Anders Hejlsberg. The
most recent
      version is C# 5.0, which was released on August 15, 2012.
    </div>;
  }
  const javaContent = () => {
    return <div>
      Java is a set of computer software and specifications developed by Sun
Microsystems,
      later acquired by Oracle Corporation, that provides a system for developing
application
      software and deploying it in a cross-mobile phones to platform computing
environment. Java
      is used in a wide variety of computing platforms from embedded devices and
enterprise servers
      and supercomputers. While less common, Java applets run in secure,
sandboxed environments to
      provide many features of native applications and can be embedded in HTML
pages.
    </div>;
  }
  const vbcNetContent = () => {
    return <div>
      The command-line compiler, VBC.EXE, is installed as part of the freeware
.NET
      Framework SDK. Mono also includes a command-line VB.NET compiler. The most
recent version
      is VB 2012, which was released on August 15, 2012.
    </div>;
  }
}

```

```

}
const xamarinContent = () => {
    return <div>
        Xamarin is a San Francisco, California based software company created in
        May
        2011[3] by the engineers that created Mono,[4] Mono for Android and
        MonoTouch that are
        cross-platform implementations of the Common Language Infrastructure (CLI)
        and Common
        Language Specifications (often called Microsoft .NET). With a C#-shared
        codebase, developers
        can use Xamarin tools to write native Android, iOS, and Windows apps with
        native user interfaces
        and share code across multiple platforms.[5] Xamarin has over 1 million
        developers in more
        than 120 countries around the World as of May 2015.'
    </div>;
}
const aspContent = () => {
    return <div>
        ASP.NET is an open-source server-side web application framework designed
        for web
        development to produce dynamic web pages. It was developed by Microsoft to
        allow programmers
        to build dynamic web sites, web applications and web services. It was first
        released in January
        2002 with version 1.0 of the .NET Framework, and is the successor to
        Microsoft\'\'s Active Server
        Pages (ASP) technology. ASP.NET is built on the Common Language Runtime
        (CLR), allowing
        programmers to write ASP.NET code using any supported .NET language. The
        ASP.NET SOAP extension
        framework allows ASP.NET components to process SOAP messages.
    </div>;
}
const mvcContent = () => {
    return <div>
        The ASP.NET MVC is a web application framework developed by Microsoft,
        which implements
        the model-view-controller (MVC) pattern. It is open-source software, apart
        from the ASP.NET Web
        Forms component which is proprietary. In the later versions of ASP.NET,
        ASP.NET MVC, ASP.NET web
        API, and ASP.NET Web Pages (a platform using only Razor pages) will merge
        into a unified MVC 6.
        The project is called ASP.NET vNext.
    </div>;
}
const jsContent = () => {
    return <div>
        JavaScript (JS) is an interpreted computer programming language. It was
        originally
        implemented as part of web browsers so that client-side scripts could
        interact with the
        user, control the browser, communicate asynchronously, and alter the
        document content that

```

```

    was displayed.[5] More recently, however, it has become common in both game
    development
    and the creation of desktop applications.
  </div>;
}

return (
  <TabComponent heightAdjustMode='Auto' overflowMode='Scrollable'>
    <TabItemsDirective>
      <TabItemDirective header={headerText[0]} content={htmlContent} />
      <TabItemDirective header={headerText[1]} content={csharpContent} />
      <TabItemDirective header={headerText[2]} content={javaContent} />
      <TabItemDirective header={headerText[3]} content={vbNetContent} />
      <TabItemDirective header={headerText[4]} content={xamarinContent} />
      <TabItemDirective header={headerText[5]} content={aspContent} />
      <TabItemDirective header={headerText[6]} content={mvcContent} />
      <TabItemDirective header={headerText[7]} content={jsContent} />
    </TabItemsDirective>
  </TabComponent>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

## INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Tab</title>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta name="description" content="Essential JS 2 for React Components" />
  <meta name="author" content="Syncfusion" />
  <link href="index.css" rel="stylesheet" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/material.css"
rel="stylesheet" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
  <script src="systemjs.config.js"></script>
</head>
<body>
  <div id='container'>
    <div id='element'>
      <div id='loader'>Loading</div>
    </div>
  </div>
</body>
</html>

```

## Popup

The Popup is the another type of [overflowMode](#) in which the Tab container holds the items that can be placed within the available space. The rest of the overflowing items for which there is no space to fit within the viewing area are moved to overflow popup container.

- The items placed in popup can be viewed by opening the popup with the help of drop-down icon given at the end of the Tab header.
- If the popup height exceeds the height of the visible area, you can scroll through the popup items and select one.

HTML

C SHARP(C#)

JAVA

VB.NET

XAMARIN



HyperText Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages. Along with CSS, and JavaScript, HTML is a cornerstone technology, used by most websites to create visually engaging web pages, user interfaces for web applications, and user interfaces for many mobile applications.[1] Web browsers can read HTML files and render them into visible or audible web pages. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language.

### INDEX.JSX

```
import { TabComponent, TabItemDirective, TabItemsDirective } from
 '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
const ReactApp = () => {
  let headerText = [{ text: "HTML" }, { text: "C Sharp(C#)" }, { text: "Java"
}, { text: "VB.Net" }, { text: "Xamarin" }, { text: "ASP.NET" }, { text:
"ASP.NET MVC" }, { text: "JavaScript" }];
  const htmlContent = () => {
    return <div>
      HyperText Markup Language, commonly referred to as HTML, is the
standard markup
      language used to create web pages. Along with CSS, and JavaScript, HTML
is a cornerstone
      technology, used by most websites to create visually engaging web
pages, user interfaces
      for web applications, and user interfaces for many mobile
applications.[1] Web browsers
      can read HTML files and render them into visible or audible web pages.
HTML describes
      the structure of a website semantically along with cues for
presentation, making it a
      markup language, rather than a programming language.
    </div>;
  }
  const csharpContent = () => {
    return <div>
      'C# is intended to be a simple, modern, general-purpose, object-
oriented
      programming language. Its development team is led by Anders Hejlsberg.
The most recent
      version is C# 5.0, which was released on August 15, 2012.
    </div>;
  }
  const javaContent = () => {
    return <div>
```

```

    Java is a set of computer software and specifications developed by Sun Microsystems,
    later acquired by Oracle Corporation, that provides a system for
    developing application
    software and deploying it in a cross-mobile phones to platform
    computing environment. Java
    is used in a wide variety of computing platforms from embedded devices
    and enterprise servers
    and supercomputers. While less common, Java applets run in secure,
    sandboxed environments to
    provide many features of native applications and can be embedded in
    HTML pages.
    </div>;
  }
  const vbcNetContent = () => {
    return <div>
      The command-line compiler, VBC.EXE, is installed as part of the
      freeware .NET
      Framework SDK. Mono also includes a command-line VB.NET compiler. The
      most recent version
      is VB 2012, which was released on August 15, 2012.
    </div>;
  }
  const xamarinContent = () => {
    return <div>
      Xamarin is a San Francisco, California based software company created
      in May
      2011[3] by the engineers that created Mono,[4] Mono for Android and
      MonoTouch that are
      cross-platform implementations of the Common Language Infrastructure
      (CLI) and Common
      Language Specifications (often called Microsoft .NET). With a C#-shared
      codebase, developers
      can use Xamarin tools to write native Android, iOS, and Windows apps
      with native user interfaces
      and share code across multiple platforms.[5] Xamarin has over 1 million
      developers in more
      than 120 countries around the World as of May 2015.'
    </div>;
  }
  const aspContent = () => {
    return <div>
      ASP.NET is an open-source server-side web application framework
      designed for web
      development to produce dynamic web pages. It was developed by Microsoft
      to allow programmers
      to build dynamic web sites, web applications and web services. It was
      first released in January
      2002 with version 1.0 of the .NET Framework, and is the successor to
      Microsoft\'\'s Active Server
      Pages (ASP) technology. ASP.NET is built on the Common Language Runtime
      (CLR), allowing
      programmers to write ASP.NET code using any supported .NET language.
      The ASP.NET SOAP extension
      framework allows ASP.NET components to process SOAP messages.
    </div>;
  }
}

```

```

const mvcContent = () => {
  return <div>
    The ASP.NET MVC is a web application framework developed by Microsoft,
    which implements
    the model-view-controller (MVC) pattern. It is open-source software,
    apart from the ASP.NET Web
    Forms component which is proprietary. In the later versions of ASP.NET,
    ASP.NET MVC, ASP.NET web
    API, and ASP.NET Web Pages (a platform using only Razor pages) will
    merge into a unified MVC 6.
    The project is called ASP.NET vNext.
  </div>;
}
const jsContent = () => {
  return <div>
    JavaScript (JS) is an interpreted computer programming language. It was
    originally
    implemented as part of web browsers so that client-side scripts could
    interact with the
    user, control the browser, communicate asynchronously, and alter the
    document content that
    was displayed.[5] More recently, however, it has become common in both
    game development
    and the creation of desktop applications.
  </div>;
}
return (<TabComponent heightAdjustMode='Auto' overflowMode='Popup'>
  <TabItemsDirective>
    <TabItemDirective header={headerText[0]} content={htmlContent} />
    <TabItemDirective header={headerText[1]} content={csharpContent} />
    <TabItemDirective header={headerText[2]} content={javaContent} />
    <TabItemDirective header={headerText[3]} content={vbcNetContent} />
    <TabItemDirective header={headerText[4]} content={xamarinContent} />
    <TabItemDirective header={headerText[5]} content={aspContent} />
    <TabItemDirective header={headerText[6]} content={mvcContent} />
    <TabItemDirective header={headerText[7]} content={jsContent} />
  </TabItemsDirective>
</TabComponent>);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

## INDEX.TSX

```

import { TabComponent, TabItemDirective, TabItemsDirective } from
'@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
const ReactApp = () => {
  let headerText: any = [{ text: "HTML" }, { text: "C Sharp(C#)" }, { text:
"Java" }, { text: "VB.Net" }, { text: "Xamarin" }, { text: "ASP.NET" }, {
text: "ASP.NET MVC" }, { text: "JavaScript" }];
  const htmlContent = () => {
    return <div>
      HyperText Markup Language, commonly referred to as HTML, is the standard
      markup
    
```

```

    language used to create web pages. Along with CSS, and JavaScript, HTML is
    a cornerstone
    technology, used by most websites to create visually engaging web pages,
    user interfaces
    for web applications, and user interfaces for many mobile applications.[1]
    Web browsers
    can read HTML files and render them into visible or audible web pages. HTML
    describes
    the structure of a website semantically along with cues for presentation,
    making it a
    markup language, rather than a programming language.
  </div>;
}
const csharpContent = () => {
  return <div>
    'C# is intended to be a simple, modern, general-purpose, object-oriented
    programming language. Its development team is led by Anders Hejlsberg. The
    most recent
    version is C# 5.0, which was released on August 15, 2012.
  </div>;
}
const javaContent = () => {
  return <div>
    Java is a set of computer software and specifications developed by Sun
    Microsystems,
    later acquired by Oracle Corporation, that provides a system for developing
    application
    software and deploying it in a cross-mobile phones to platform computing
    environment. Java
    is used in a wide variety of computing platforms from embedded devices and
    enterprise servers
    and supercomputers. While less common, Java applets run in secure,
    sandboxed environments to
    provide many features of native applications and can be embedded in HTML
    pages.
  </div>;
}
const vbcNetContent = () => {
  return <div>
    The command-line compiler, VBC.EXE, is installed as part of the freeware
    .NET
    Framework SDK. Mono also includes a command-line VB.NET compiler. The most
    recent version
    is VB 2012, which was released on August 15, 2012.
  </div>;
}
const xamarinContent = () => {
  return <div>
    Xamarin is a San Francisco, California based software company created in
    May
    2011[3] by the engineers that created Mono,[4] Mono for Android and
    MonoTouch that are
    cross-platform implementations of the Common Language Infrastructure (CLI)
    and Common
    Language Specifications (often called Microsoft .NET). With a C#-shared
    codebase, developers

```



```

    can use Xamarin tools to write native Android, iOS, and Windows apps with
    native user interfaces
    and share code across multiple platforms.[5] Xamarin has over 1 million
    developers in more
    than 120 countries around the World as of May 2015.'
</div>;
}
const aspContent = () => {
    return <div>
        ASP.NET is an open-source server-side web application framework designed
        for web
        development to produce dynamic web pages. It was developed by Microsoft to
        allow programmers
        to build dynamic web sites, web applications and web services. It was first
        released in January
        2002 with version 1.0 of the .NET Framework, and is the successor to
        Microsoft\'s Active Server
        Pages (ASP) technology. ASP.NET is built on the Common Language Runtime
        (CLR), allowing
        programmers to write ASP.NET code using any supported .NET language. The
        ASP.NET SOAP extension
        framework allows ASP.NET components to process SOAP messages.
    </div>;
}
const mvcContent = () => {
    return <div>
        The ASP.NET MVC is a web application framework developed by Microsoft,
        which implements
        the model-view-controller (MVC) pattern. It is open-source software, apart
        from the ASP.NET Web
        Forms component which is proprietary. In the later versions of ASP.NET,
        ASP.NET MVC, ASP.NET web
        API, and ASP.NET Web Pages (a platform using only Razor pages) will merge
        into a unified MVC 6.
        The project is called ASP.NET vNext.
    </div>;
}
const jsContent = () => {
    return <div>
        JavaScript (JS) is an interpreted computer programming language. It was
        originally
        implemented as part of web browsers so that client-side scripts could
        interact with the
        user, control the browser, communicate asynchronously, and alter the
        document content that
        was displayed.[5] More recently, however, it has become common in both game
        development
        and the creation of desktop applications.
    </div>;
}
return (
    <TabComponent heightAdjustMode='Auto' overflowMode='Popup'>
        <TabItemsDirective>
            <TabItemDirective header={headerText[0]} content={htmlContent} />
            <TabItemDirective header={headerText[1]} content={csharpContent} />
            <TabItemDirective header={headerText[2]} content={javaContent} />
            <TabItemDirective header={headerText[3]} content={vbNetContent} />
        </TabItemsDirective>
    </TabComponent>
)

```

```

        <TabItemDirective header={headerText[4]} content={xamarinContent} />
        <TabItemDirective header={headerText[5]} content={aspContent} />
        <TabItemDirective header={headerText[6]} content={mvcContent} />
        <TabItemDirective header={headerText[7]} content={jsContent} />
    </TabItemsDirective>
</TabComponent>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

## INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Syncfusion React Tab</title>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta name="description" content="Essential JS 2 for React Components" />
    <meta name="author" content="Syncfusion" />
    <link href="index.css" rel="stylesheet" />
    <link href="https://cdn.syncfusion.com/ej2/20.3.56/material.css"
rel="stylesheet" />
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
    <script src="systemjs.config.js"></script>
</head>
<body>
    <div id='container'>
        <div id='element'>
            <div id='loader'>Loading</div>
        </div>
    </div>
</body>
</html>

```

See Also

- [How to prevent content swipe selection](#)
- [Collapsible Tab](#)

## Header in React Tab component

This section explains about modifying the style of Tab header, and configuring its icons and positions.

### Styles

You can customize header styles by adding predefined classes in the Tab root element. The pre-defined CSS class names are as follows:

- **e-fill:** The Selected Tab header background is set as solid fill.
- **e-background:** Tab header has a solid fill background, and the selected header has a highlighted border.

- **e-background e-accent:** Tab header has a solid fill background, and the selected header has a highlighted border with accent color.

If the above custom style classes are not included in the root element, the default style is applied to the Tab items.

#### INDEX.JSX

```
import { ChangeEventArgs, DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { TabComponent, TabItemDirective, TabItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from "react";
import { useRef } from 'react';
import * as ReactDOM from 'react-dom';

const ReactApp = () => {
  let tabObj = useRef(null);
  let headerStyles = useRef(null)
  let headerText = [{ text: "Twitter" }, { text: "Facebook" }, { text: "WhatsApp" }];
  let fields = { text: 'text', value: 'value' };
  let headerData = [
    { 'value': 'default', 'text': 'Default' }, { 'value': 'fill', 'text': 'Fill' },
    { 'value': 'background', 'text': 'Background' }, { 'value': 'accent', 'text': 'Accent' }
  ];
  let hdrVal = 'default';
  const changeHeaderStyles = (e) => {
    removeStyleClass();
    const name = headerStyles.current.value;
    if (name === 'fill') {
      tabObj.current.element.classList.add('e-fill');
    } else if (name === 'background') {
      tabObj.current.element.classList.add('e-background');
    } else if (name === 'accent') {
      tabObj.current.element.classList.add('e-background');
      tabObj.current.element.classList.add('e-accent');
    }
  }
  const removeStyleClass = () => {
    tabObj.current.element.classList.remove('e-fill');
    tabObj.current.element.classList.remove('e-background');
    tabObj.current.element.classList.remove('e-accent');
  }
  const content0 = () => {
    return <div>
      Twitter is an online social networking service that enables users to send and read short 140-character messages called "tweets". Registered users can read and post tweets, but those who are unregistered can only read them. Users access Twitter through the website interface, SMS or mobile device app Twitter Inc. is based in San Francisco and has more than 25 offices around the world. Twitter was created in March 2006 by Jack Dorsey,
```

```

    Evan Williams, Biz Stone, and Noah Glass and launched in July 2006. The
    service rapidly gained worldwide popularity,
    with more than 100 million users posting 340 million tweets a day in
2012.The service also handled 1.6 billion
    search queries per day.
  </div>;
}
const content1 = () => {
  return <div>
    Facebook is an online social networking service headquartered in Menlo
    Park, California. Its website was
    launched on February 4, 2004, by Mark Zuckerberg with his Harvard
    College roommates and fellow students Eduardo
    Saverin, Andrew McCollum, Dustin Moskovitz and Chris Hughes.The
    founders had initially limited the website
    membership to Harvard students, but later expanded it to colleges in
    the Boston area, the Ivy League, and Stanford
    University. It gradually added support for students at various other
    universities and later to high-school students.
  </div>;
}
const content2 = () => {
  return <div>
    WhatsApp Messenger is a proprietary cross-platform instant messaging
    client for smartphones that operates
    under a subscription business model. It uses the Internet to send text
    messages, images, video, user location and
    audio media messages to other users using standard cellular mobile
    numbers. As of February 2016, WhatsApp had a user
    base of up to one billion,[10] making it the most globally popular
    messaging application. WhatsApp Inc., based in
    Mountain View, California, was acquired by Facebook Inc. on February
19, 2014, for approximately US$19.3 billion.
  </div>;
}
return (
  <div>
    <DropDownListComponent ref={headerStyles} dataSource={headerData}
    fields={fields} value={hdrVal} width={'150'} change={changeHeaderStyles} />
    <br /><br />
    <TabComponent heightAdjustMode='Auto' height={320} ref={tabObj}>
      <TabItemsDirective>
        <TabItemDirective header={headerText[0]} content={content0} />
        <TabItemDirective header={headerText[1]} content={content1} />
        <TabItemDirective header={headerText[2]} content={content2} />
      </TabItemsDirective>
    </TabComponent>
  </div>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

## INDEX.TSX

```

import { ChangeEventArgs, DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { TabComponent, TabItemDirective, TabItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from "react";
import { useRef } from 'react';
import * as ReactDOM from 'react-dom';
const ReactApp = () => {
    let tabObj = useRef<TabComponent>(null);
    let headerStyles = useRef<DropDownListComponent>(null)
    let headerText: any = [{ text: "Twitter" }, { text: "Facebook" }, { text: "WhatsApp" }];
    let fields: object = { text: 'text', value: 'value' };
    let headerData: any = [
        { 'value': 'default', 'text': 'Default' }, { 'value': 'fill', 'text': 'Fill' },
        { 'value': 'background', 'text': 'Background' }, { 'value': 'accent', 'text': 'Accent' }
    ];
    let hdrVal: string = 'default';
    const changeHeaderStyles = (e: ChangeEventArgs): void => {
        removeStyleClass();
        const name: string = headerStyles.current.value as string;
        if (name === 'fill') {
            tabObj.current.element.classList.add('e-fill');
        } else if (name === 'background') {
            tabObj.current.element.classList.add('e-background');
        } else if (name === 'accent') {
            tabObj.current.element.classList.add('e-background');
            tabObj.current.element.classList.add('e-accent');
        }
    }
    const removeStyleClass = (): void => {
        tabObj.current.element.classList.remove('e-fill');
        tabObj.current.element.classList.remove('e-background');
        tabObj.current.element.classList.remove('e-accent');
    }
    const content0 = () => {
        return <div>
            Twitter is an online social networking service that enables users to send and read short 140-character messages called "tweets". Registered users can read and post tweets, but those who are unregistered can only read them. Users access Twitter through the website interface, SMS or mobile device app Twitter Inc. is based in San Francisco and has more than 25 offices around the world. Twitter was created in March 2006 by Jack Dorsey, Evan Williams, Biz Stone, and Noah Glass and launched in July 2006. The service rapidly gained worldwide popularity, with more than 100 million users posting 340 million tweets a day in 2012. The service also handled 1.6 billion search queries per day.
        </div>;
    }
    const content1 = () => {
        return <div>

```

```

    Facebook is an online social networking service headquartered in Menlo
    Park, California. Its website was
    launched on February 4, 2004, by Mark Zuckerberg with his Harvard
    College roommates and fellow students Eduardo
    Saverin, Andrew McCollum, Dustin Moskovitz and Chris Hughes.The
    founders had initially limited the website
    membership to Harvard students, but later expanded it to colleges in
    the Boston area, the Ivy League, and Stanford
    University. It gradually added support for students at various other
    universities and later to high-school students.
  </div>;
}
const content2 = () => {
  return <div>
    WhatsApp Messenger is a proprietary cross-platform instant messaging
    client for smartphones that operates
    under a subscription business model. It uses the Internet to send text
    messages, images, video, user location and
    audio media messages to other users using standard cellular mobile
    numbers. As of February 2016, WhatsApp had a user
    base of up to one billion,[10] making it the most globally popular
    messaging application. WhatsApp Inc., based in
    Mountain View, California, was acquired by Facebook Inc. on February
    19, 2014, for approximately US$19.3 billion.
  </div>;
}
return (
  <div>
    <DropDownListComponent ref={headerStyles} dataSource={headerData}
    fields={fields} value={hdrVal} width={'150'} change={changeHeaderStyles} />
    <br /><br />
    <TabComponent heightAdjustMode='Auto' height={320} ref={tabObj}>
      <TabItemsDirective>
        <TabItemDirective header={headerText[0]} content={content0} />
        <TabItemDirective header={headerText[1]} content={content1} />
        <TabItemDirective header={headerText[2]} content={content2} />
      </TabItemsDirective>
    </TabComponent>
  </div>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

### Icon positions

You can customize the position of the Tab header icons using the [iconPosition](#) property. This property depends on the header items [iconCss](#) property. By default, Tab header icon is placed on left position. The position values are as follows:

- **Left:** Icon is placed on the left of the Tab header item.
- **Right:** Icon is placed on the right of the Tab header item.
- **Top:** Icon is placed on the top of the Tab header item.
- **Bottom:** Icon is placed on the bottom of the Tab header item.

**INDEX.JSX**

```

import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { TabComponent, TabItemDirective, TabItemsDirective } from
 '@syncfusion/ej2-react-navigations';
import * as React from "react";
import { useRef } from 'react';
import * as ReactDOM from 'react-dom';
const ReactApp = () => {
    const tabRef = useRef(null);
    const dropRef = useRef(null);
    let headerText = [{ text: "Twitter", iconCss: 'e-twitter' }, { text:
"Facebook", iconCss: 'e-facebook' }, { text: "WhatsApp", iconCss: 'e-
whatsapp' }];
    let fields = { text: 'text', value: 'value' };
    let headerData = [
        { 'value': 'left', 'text': 'Left' }, { 'value': 'right', 'text': 'Right'
    },
        { 'value': 'top', 'text': 'Top' }, { 'value': 'bottom', 'text': 'Bottom'
    }
    ];
    let value = 'left';
    const TwitterContent = () => {
        return <div>
            Twitter is an online social networking service that enables users to
            send and read short 140-character messages called "tweets". Registered users
            can read and post tweets, but those who are unregistered can only read them.
            Users access Twitter through the website interface, SMS or mobile device app
            Twitter Inc. is based in San Francisco and has more than 25 offices around
            the world. Twitter was created in March 2006 by Jack Dorsey, Evan Williams,
            Biz Stone, and Noah Glass and launched in July 2006. The service rapidly
            gained worldwide popularity, with more than 100 million users posting 340
            million tweets a day in 2012. The service also handled 1.6 billion search
            queries per day.
        </div>;
    }
    const facebookContent = () => {
        return <div>
            Facebook is an online social networking service headquartered in Menlo
            Park, California. Its website was launched on February 4, 2004, by Mark
            Zuckerberg with his Harvard College roommates and fellow students Eduardo
            Saverin, Andrew McCollum, Dustin Moskovitz and Chris Hughes. The founders had
            initially limited the website membership to Harvard students, but later
            expanded it to colleges in the Boston area, the Ivy League, and Stanford
            University. It gradually added support for students at various other
            universities and later to high-school students.
        </div>;
    }
    const whatsappContent = () => {
        return <div>
            WhatsApp Messenger is a proprietary cross-platform instant messaging
            client for smartphones that operates under a subscription business model. It
            uses the Internet to send text messages, images, video, user location and
            audio media messages to other users using standard cellular mobile numbers.
            As of February 2016, WhatsApp had a user base of up to one billion, [10]
            making it the most globally popular messaging application. WhatsApp Inc.,

```

```

based in Mountain View, California, was acquired by Facebook Inc. on February
19, 2014, for approximately US$19.3 billion.
</div>;
}
const onChange = () => {
  let items = tabRef.current.items;
  for (let i = 0; i < items.length; i++) {
    items[i].header.iconPosition = dropRef.current.value;
  };
}
return (
  <div>
    <DropDownListComponent ref={dropRef} dataSource={headerData}
fields={fields} value={value} width={'150'} change={onChange} />
    <br /><br />
    <TabComponent ref={tabRef}>
      <TabItemsDirective>
        <TabItemDirective header={headerText[0]} content={TwitterContent}
/>
        <TabItemDirective header={headerText[1]} content={facebookContent}
/>
        <TabItemDirective header={headerText[2]} content={whatsappContent}
/>
      </TabItemsDirective>
    </TabComponent>
  </div>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

## INDEX.TSX

```

import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { TabComponent, TabItemDirective, TabItemsDirective } from
 '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import { useRef } from 'react';
import * as ReactDOM from 'react-dom';
const ReactApp = () => {
  const tabRef = useRef<TabComponent>(null);
  const dropRef = useRef<DropDownListComponent>(null);
  let headerText: any = [{ text: 'Twitter', iconCss: 'e-twitter' }, { text:
'Facebook', iconCss: 'e-facebook' }, { text: 'WhatsApp', iconCss: 'e-
whatsapp' }];
  let fields: object = { text: 'text', value: 'value' };
  let headerData: any = [
    { 'value': 'left', 'text': 'Left' }, { 'value': 'right', 'text': 'Right'
},
    { 'value': 'top', 'text': 'Top' }, { 'value': 'bottom', 'text': 'Bottom'
}
  ];
  let value: string = 'left';
  const twitterContent = () => {
    return <div>

```



```

    Twitter is an online social networking service that enables users to
    send and read short 140-character messages called "tweets". Registered users
    can read and post tweets, but those who are unregistered can only read them.
    Users access Twitter through the website interface, SMS or mobile device app
    Twitter Inc. is based in San Francisco and has more than 25 offices around
    the world. Twitter was created in March 2006 by Jack Dorsey, Evan Williams,
    Biz Stone, and Noah Glass and launched in July 2006. The service rapidly
    gained worldwide popularity, with more than 100 million users posting 340
    million tweets a day in 2012.The service also handled 1.6 billion search
    queries per day.
    </div>;
  }
  const facebookContent = () => {
    return <div>
      Facebook is an online social networking service headquartered in Menlo
      Park, California. Its website was launched on February 4, 2004, by Mark
      Zuckerberg with his Harvard College roommates and fellow students Eduardo
      Saverin, Andrew McCollum, Dustin Moskovitz and Chris Hughes.The founders had
      initially limited the website membership to Harvard students, but later
      expanded it to colleges in the Boston area, the Ivy League, and Stanford
      University. It gradually added support for students at various other
      universities and later to high-school students.
    </div>;
  }
  const whatsappContent = () => {
    return <div>
      WhatsApp Messenger is a proprietary cross-platform instant messaging
      client for smartphones that operates under a subscription business model. It
      uses the Internet to send text messages, images, video, user location and
      audio media messages to other users using standard cellular mobile numbers.
      As of February 2016, WhatsApp had a user base of up to one billion,[10]
      making it the most globally popular messaging application. WhatsApp Inc.,
      based in Mountain View, California, was acquired by Facebook Inc. on February
      19, 2014, for approximately US$19.3 billion.
    </div>;
  }
  const onChange = (): void => {
    let items: any = tabRef.current.items;
    for (let i: number = 0; i < items.length; i++) {
      items[i].header.iconPosition = dropRef.current.value;
    };
  }
  return (
    <div>
      <DropDownListComponent ref={dropRef} dataSource={headerData}
      fields={fields} value={value} width={'150'} change={onChange} />
      <br /><br />
      <TabComponent ref={tabRef}>
        <TabItemsDirective>
          <TabItemDirective header={headerText[0] } content={twitterContent}
        />
          <TabItemDirective header={headerText[1] } content={facebookContent}
        />
          <TabItemDirective header={headerText[2] } content={whatsappContent}
        />
        </TabItemsDirective>
      </TabComponent>
    </div>
  )

```

```

    </div>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

### See Also

- [How to customize selected tab styles](#)

### Localization in React Tab component

Localization library allows to localize the default text content of the Tab to different cultures using the [locale](#) property. In Tab, the close button's tooltip text alone will be localize based on culture. The close button is shown on tab header when enabled [showCloseButton](#) property.

| Locale key | en-US (default) |

|-----|-----|

| closeButtonTitle | Close |

### Loading translations

To load translation object in an application use `load` function of `L10n` class.

In the below sample, `French` culture is set to Tab and change the close button's tooltip text.

### INDEX.JSX

```

import { L10n } from '@syncfusion/ej2-base';
import { TabComponent, TabItemDirective, TabItemsDirective } from
 '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
L10n.load({
  'fr-BE': {
    'tab': {
      'closeButtonTitle': "Fermer"
    }
  }
});
const ReactApp = () => {
  const headerText = [{ text: "HTML" }, { text: "C Sharp(C#)" }, { text:
    "Java" }];
  const htmlContent = () => {
    return <div>
      HyperText Markup Language, commonly referred to as HTML, is the
      standard markup language used to create web pages. Along with CSS, and
      JavaScript, HTML is a cornerstone technology, used by most websites to create
      visually engaging web pages, user interfaces for web applications, and user
      interfaces for many mobile applications.[1] Web browsers can read HTML files
      and render them into visible or audible web pages. HTML describes the
      structure of a website semantically along with cues for presentation, making
      it a markup language, rather than a programming language.
    </div>;
  }
  const csharpContent = () => {

```

```

    return <div>
      C# is intended to be a simple, modern, general-purpose, object-oriented
      programming language. Its development team is led by Anders Hejlsberg. The
      most recent version is C# 5.0, which was released on August 15, 2012.
    </div>;
  }
  const javaContent = () => {
    return <div>
      Java is a set of computer software and specifications developed by Sun
      Microsystems, later acquired by Oracle Corporation, that provides a system
      for developing application software and deploying it in a cross-mobile phones
      to platform computing environment. Java is used in a wide variety of
      computing platforms from embedded devices and enterprise servers and
      supercomputers. While less common, Java applets run in secure, sandboxed
      environments to provide many features of native applications and can be
      embedded in HTML pages.
    </div>;
  }
  return (<TabComponent heightAdjustMode='Auto' locale="fr-BE"
showCloseButton={true}>
  <TabItemsDirective>
    <TabItemDirective header={headerText[0]} content={htmlContent}/>
    <TabItemDirective header={headerText[1]} content={csharpContent}/>
    <TabItemDirective header={headerText[2]} content={javaContent}/>
  </TabItemsDirective>
</TabComponent>);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

## INDEX.TSX

```

import { L10n } from '@syncfusion/ej2-base';
import { TabComponent, TabItemDirective, TabItemsDirective } from
 '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
L10n.load({
  'fr-BE': {
    'tab': {
      'closeButtonTitle': "Fermer"
    }
  }
});
const ReactApp = () => {
  const headerText: any = [{ text: "HTML" }, { text: "C Sharp(C#)" }, { text:
  "Java" }];
  const htmlContent = () => {
    return <div>
      HyperText Markup Language, commonly referred to as HTML, is the
      standard markup language used to create web pages. Along with CSS, and
      JavaScript, HTML is a cornerstone technology, used by most websites to create
      visually engaging web pages, user interfaces for web applications, and user
      interfaces for many mobile applications.[1] Web browsers can read HTML files
      and render them into visible or audible web pages. HTML describes the
    </div>
  }

```

```

structure of a website semantically along with cues for presentation, making
it a markup language, rather than a programming language.
    </div>;
}
const csharpContent = () => {
    return <div>
        C# is intended to be a simple, modern, general-purpose, object-oriented
        programming language. Its development team is led by Anders Hejlsberg. The
        most recent version is C# 5.0, which was released on August 15, 2012.
    </div>;
}
const javaContent = () => {
    return <div>
        Java is a set of computer software and specifications developed by Sun
        Microsystems, later acquired by Oracle Corporation, that provides a system
        for developing application software and deploying it in a cross-mobile phones
        to platform computing environment. Java is used in a wide variety of
        computing platforms from embedded devices and enterprise servers and
        supercomputers. While less common, Java applets run in secure, sandboxed
        environments to provide many features of native applications and can be
        embedded in HTML pages.
    </div>;
}
return (
    <TabComponent heightAdjustMode='Auto' locale="fr-BE"
    showCloseButton={true}>
        <TabItemsDirective>
            <TabItemDirective header={headerText[0]} content={htmlContent} />
            <TabItemDirective header={headerText[1]} content={csharpContent} />
            <TabItemDirective header={headerText[2]} content={javaContent} />
        </TabItemsDirective>
    </TabComponent>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

## INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Syncfusion React Tab</title>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta name="description" content="Essential JS 2 for React Components" />
    <meta name="author" content="Syncfusion" />
    <link href="index.css" rel="stylesheet" />
    <link href="https://cdn.syncfusion.com/ej2/20.3.56/material.css"
    rel="stylesheet" />
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
    ipt>
    <script src="systemjs.config.js"></script>
</head>
<body>

```

```
<div id='container'>
  <div id='element'>
    <div id='loader'>Loading</div>
  </div>
</div>
</body>
</html>
```

## Orientation in React Tab component

This section explains about modifying the position and modes of Tab header.

It allows placing the header section inside the Tab component at different positions by using the [headerPlacement](#) property. The available positions are as follows:

- **Top:** Tab header items can be arranged horizontally, and their content can be placed after the header.
- **Bottom:** Tab header items can be arranged horizontally, and their content can be placed before the header.
- **Left:** Tab header items can be arranged vertically, and their content can be placed after the header.
- **Right:** Tab header items can be arranged vertically, and their content can be placed before the header.

It is also adaptable to the available space when the tab items exceed the view space. You can customize the modes by using [overflowMode](#) property. The available modes are as follows:

- Scrollable
- Popup

## INDEX.JSX

```
import * as React from "react";
import { useRef } from 'react';
import * as ReactDOM from 'react-dom';
import { ChangeEventArgs, DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { TabComponent, TabItemDirective, TabItemsDirective } from '@syncfusion/ej2-react-navigations';
const ReactApp = () => {
  const tabObj = useRef(null);
  const headerPosition = useRef(null);
  const orientation = useRef(null);
  let headerText= [{ text: "HTML" }, { text: "C Sharp(C#)" }, { text: "Java" }, { text: "VB.Net" }, { text: "Xamarin" }, { text: "ASP.NET" }, { text: "ASP.NET MVC" }, { text: "JavaScript" }];
  let fields = { text: 'text', value: 'value' };
  let headerData = [{ 'value': 'top', 'text': 'Top' }, { 'value': 'bottom', 'text': 'Bottom' }, { 'value': 'left', 'text': 'Left' }, { 'value': 'right', 'text': 'Right' }];
};
```

```

let orientationData = [{ 'value': 'scrollable', 'text': 'Scrollable' }, {
'value': 'popup', 'text': 'Popup' }];
let hdrVal = 'top';
let orientationVal = 'scrollable';
const changeOrientationMode = (e) => {
  let placement = headerPosition.current.value.charAt(0).toUpperCase() +
headerPosition.current.value.slice(1);
  tabObj.current.headerPlacement = placement;
  tabObj.current.dataBind();
}
const changeOverflowMode = (e) => {
  let placement = orientation.current.value.charAt(0).toUpperCase() +
orientation.current.value.slice(1);
  tabObj.current.overflowMode = placement;
  tabObj.current.dataBind();
}
const htmlcontent = () => {
  return <div>
    HyperText Markup Language, commonly referred to as HTML, is the
standard markup
    language used to create web pages. Along with CSS, and JavaScript, HTML
is a cornerstone
    technology, used by most websites to create visually engaging web
pages, user interfaces
    for web applications, and user interfaces for many mobile
applications.[1] Web browsers
    can read HTML files and render them into visible or audible web pages.
HTML describes
    the structure of a website semantically along with cues for
presentation, making it a
    markup language, rather than a programming language.
  </div>;
}
const csharpContent = () => {
  return <div>
    C# is intended to be a simple, modern, general-purpose, object-oriented
programming language. Its development team is led by Anders Hejlsberg. The
most recent version is C# 5.0, which was released on August 15, 2012.
  </div>;
}
const javaContent = () => {
  return <div>
    Java is a set of computer software and specifications developed by Sun
Microsystems, later acquired by Oracle
    Corporation, that provides a system for developing application software
and deploying it in a cross-platform computing
    environment. Java is used in a wide variety of computing platforms from
embedded devices and mobile phones to
    enterprise servers and supercomputers. While less common, Java applets
run in secure, sandboxed environments to
    provide many features of native applications and can be embedded in
HTML pages.
  </div>;
}
const vbNetcontent = () => {
  return <div>

```

```

    The command-line compiler, VBC.EXE, is installed as part of the
    freeware .NET
    Framework SDK. Mono also includes a command-line VB.NET compiler. The
    most recent version
    is VB 2012, which was released on August 15, 2012.
  </div>;
}
const xamarinContent = () => {
  return <div>
    Xamarin is a San Francisco, California based software company created
    in May
    2011[3] by the engineers that created Mono,[4] Mono for Android and
    MonoTouch that are
    cross-platform implementations of the Common Language Infrastructure
    (CLI) and Common
    Language Specifications (often called Microsoft .NET). With a C#-shared
    codebase, developers
    can use Xamarin tools to write native Android, iOS, and Windows apps
    with native user interfaces
    and share code across multiple platforms.[5] Xamarin has over 1 million
    developers in more
    than 120 countries around the World as of May 2015.
  </div>;
}
const aspNetContent = () => {
  return <div>
    ASP.NET is an open-source server-side web application framework
    designed for web
    development to produce dynamic web pages. It was developed by Microsoft
    to allow programmers
    to build dynamic web sites, web applications and web services. It was
    first released in January
    2002 with version 1.0 of the .NET Framework, and is the successor to
    Microsoft Active Server
    Pages (ASP) technology. ASP.NET is built on the Common Language Runtime
    (CLR), allowing
    programmers to write ASP.NET code using any supported .NET language.
    The ASP.NET SOAP extension
    framework allows ASP.NET components to process SOAP messages.
  </div>;
}
const mvcContent = () => {
  return <div>
    The ASP.NET MVC is a web application framework developed by Microsoft,
    which implements
    the model-view-controller (MVC) pattern. It is open-source software,
    apart from the ASP.NET Web
    Forms component which is proprietary. In the later versions of ASP.NET,
    ASP.NET MVC, ASP.NET Web
    API, and ASP.NET Web Pages (a platform using only Razor pages) will
    merge into a unified MVC 6.
    The project is called ASP.NET vNext.
  </div>;
}
const javaScriptContent = () => {
  return <div>

```

```

        JavaScript (JS) is an interpreted computer programming language. It was
        originally
        implemented as part of web browsers so that client-side scripts could
        interact with the
        user, control the browser, communicate asynchronously, and alter the
        document content that
        was displayed.[5] More recently, however, it has become common in both
        game development
        and the creation of desktop applications.
    </div>;
}
return (
    <div>
        <div className='row' id='header'>
            <label> Header Position </label>
            <div>
                <DropDownListComponent ref={headerPosition} dataSource={headerData}
fields={fields} value={hdrVal} width={'150'} change={changeOrientationMode}
/>
            </div>
        </div>
        <div className='row' id='mode'>
            <label> Mode </label>
            <div>
                <DropDownListComponent ref={orientation}
dataSource={orientationData} fields={fields} value={orientationVal}
width={'150'} change={changeOverflowMode} />
            </div>
        </div>
        <br /><br />
        <TabComponent id='default' heightAdjustMode='None' height={150}
width='700' ref={tabObj}>
            <TabItemsDirective>
                <TabItemDirective header={headerText[0]} content={htmlContent} />
                <TabItemDirective header={headerText[1]} content={csharpContent} />
                <TabItemDirective header={headerText[2]} content={javaContent} />
                <TabItemDirective header={headerText[3]} content={vbNetContent} />
                <TabItemDirective header={headerText[4]} content={xamarinContent}
/>
                <TabItemDirective header={headerText[5]} content={aspNetContent} />
                <TabItemDirective header={headerText[6]} content={mvcContent} />
                <TabItemDirective header={headerText[7]}
content={javascriptContent} />
            </TabItemsDirective>
        </TabComponent>
    </div>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

## INDEX.TSX

```

import * as React from 'react';
import { useRef } from 'react';
import * as ReactDOM from 'react-dom';

```



```

import { ChangeEventArgs, DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { TabComponent, TabItemDirective, TabItemsDirective } from '@syncfusion/ej2-react-navigations';
const ReactApp = () => {
    const tabObj = useRef<TabComponent>(null);
    const headerPosition = useRef<DropDownListComponent>(null);
    const orientation = useRef<DropDownListComponent>(null);
    let headerText: any = [{ text: "HTML" }, { text: "C Sharp(C#)" }, { text: "Java" }, { text: "VB.Net" }, { text: "Xamarin" }, { text: "ASP.NET" }, { text: "ASP.NET MVC" }, { text: "JavaScript" }];
    let fields: object = { text: 'text', value: 'value' };
    let headerData: any = [{ 'value': 'top', 'text': 'Top' }, { 'value': 'bottom', 'text': 'Bottom' }, { 'value': 'left', 'text': 'Left' }, { 'value': 'right', 'text': 'Right' }];
    let orientationData: { [key: string]: Object }[] = [{ 'value': 'scrollable', 'text': 'Scrollable' }, { 'value': 'popup', 'text': 'Popup' }];
    let hdrVal: string = 'top';
    let orientationVal: string = 'scrollable';
    const changeOrientationMode = (e: ChangeEventArgs): void => {
        let placement: string = (headerPosition.current.value as string).charAt(0).toUpperCase() + (headerPosition.current.value as string).slice(1);
        tabObj.current.headerPlacement = placement as any;
        tabObj.current.dataBind();
    }
    const changeOverflowMode = (e: ChangeEventArgs): void => {
        let placement: string = (orientation.current.value as string).charAt(0).toUpperCase() + (orientation.current.value as string).slice(1);
        tabObj.current.overflowMode = placement as any;
        tabObj.current.dataBind();
    }
    const htmlcontent = () => {
        return <div>
            HyperText Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages. Along with CSS, and JavaScript, HTML is a cornerstone technology, used by most websites to create visually engaging web pages, user interfaces for web applications, and user interfaces for many mobile applications.[1] Web browsers can read HTML files and render them into visible or audible web pages. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language.
        </div>;
    }
    const csharpContent = () => {
        return <div>
            C# is intended to be a simple, modern, general-purpose, object-oriented programming language. Its development team is led by Anders Hejlsberg. The most recent version is C# 5.0, which was released on August 15, 2012.
        </div>;
    }
}

```

```

    </div>;
  }
  const javaContent = () => {
    return <div>
      Java is a set of computer software and specifications developed by Sun
      Microsystems, later acquired by Oracle
      Corporation, that provides a system for developing application software
      and deploying it in a cross-platform computing
      environment. Java is used in a wide variety of computing platforms from
      embedded devices and mobile phones to
      enterprise servers and supercomputers. While less common, Java applets
      run in secure, sandboxed environments to
      provide many features of native applications and can be embedded in
      HTML pages.
    </div>;
  }
  const vbNetcontent = () => {
    return <div>
      The command-line compiler, VBC.EXE, is installed as part of the
      freeware .NET
      Framework SDK. Mono also includes a command-line VB.NET compiler. The
      most recent version
      is VB 2012, which was released on August 15, 2012.
    </div>;
  }
  const xamarinContent = () => {
    return <div>
      Xamarin is a San Francisco, California based software company created
      in May
      2011[3] by the engineers that created Mono,[4] Mono for Android and
      MonoTouch that are
      cross-platform implementations of the Common Language Infrastructure
      (CLI) and Common
      Language Specifications (often called Microsoft .NET). With a C#-shared
      codebase, developers
      can use Xamarin tools to write native Android, iOS, and Windows apps
      with native user interfaces
      and share code across multiple platforms.[5] Xamarin has over 1 million
      developers in more
      than 120 countries around the World as of May 2015.
    </div>;
  }
  const aspNetContent = () => {
    return <div>
      ASP.NET is an open-source server-side web application framework
      designed for web
      development to produce dynamic web pages. It was developed by Microsoft
      to allow programmers
      to build dynamic web sites, web applications and web services. It was
      first released in January
      2002 with version 1.0 of the .NET Framework, and is the successor to
      Microsoft Active Server
      Pages (ASP) technology. ASP.NET is built on the Common Language Runtime
      (CLR), allowing
      programmers to write ASP.NET code using any supported .NET language.
      The ASP.NET SOAP extension
      framework allows ASP.NET components to process SOAP messages.
    </div>;
  }

```

```

    </div>;
  }
  const mvcContent = () => {
    return <div>
      The ASP.NET MVC is a web application framework developed by Microsoft,
      which implements
      the model-view-controller (MVC) pattern. It is open-source software,
      apart from the ASP.NET Web
      Forms component which is proprietary. In the later versions of ASP.NET,
      ASP.NET MVC, ASP.NET Web
      API, and ASP.NET Web Pages (a platform using only Razor pages) will
      merge into a unified MVC 6.
      The project is called ASP.NET vNext.
    </div>;
  }
  const javascriptContent = () => {
    return <div>
      JavaScript (JS) is an interpreted computer programming language. It was
      originally
      implemented as part of web browsers so that client-side scripts could
      interact with the
      user, control the browser, communicate asynchronously, and alter the
      document content that
      was displayed.[5] More recently, however, it has become common in both
      game development
      and the creation of desktop applications.
    </div>;
  }
  return (
    <div>
      <div className='row' id='header'>
        <label> Header Position </label>
        <div>
          <DropDownListComponent ref={headerPosition} dataSource={headerData}
fields={fields} value={hdrVal} width={'150'} change={changeOrientationMode}
/>
        </div>
      </div>
      <div className='row' id='mode'>
        <label> Mode </label>
        <div>
          <DropDownListComponent ref={orientation}
dataSource={orientationData} fields={fields} value={orientationVal}
width={'150'} change={changeOverflowMode} />
        </div>
      </div>
      <br /><br />
      <TabComponent id='default' heightAdjustMode='None' height={150}
width='700' ref={tabObj}>
        <TabItemsDirective>
          <TabItemDirective header={headerText[0]} content={htmlContent} />
          <TabItemDirective header={headerText[1]} content={csharpContent} />
          <TabItemDirective header={headerText[2]} content={javaContent} />
          <TabItemDirective header={headerText[3]} content={vbNetContent} />
          <TabItemDirective header={headerText[4]} content={xamarinContent}
/>
          <TabItemDirective header={headerText[5]} content={aspNetContent} />

```

```

        <TabItemDirective header={headerText[6]} content={mvcContent} />
        <TabItemDirective header={headerText[7]}
content={javascriptContent} />
      </TabItemsDirective>
    </TabComponent>
  </div>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

## INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Tab</title>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta name="description" content="Essential JS 2 for React Components" />
  <meta name="author" content="Syncfusion" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/material.css"
rel="stylesheet" />
  <link href="index.css" rel="stylesheet" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
  <script src="systemjs.config.js"></script>
</head>
<body>
  <div id='container'>
    <div id='element'>
      <div id='loader'>Loading</div>
    </div>
  </div>
</body>
</html>

```

## Drag and drop in React Tab component

The Tab component allows you to drag and drop any item by setting `allowDragAndDrop` to `true`. Items can be reordered to any place by dragging and dropping them onto the desired location.

- If you need to prevent dragging action for a particular item, the `onDragStart` event can be used which will trigger when the item drag is started. If you need to prevent dropping action for a particular item, the `dragged` event can be used which will trigger when the drag action is stopped.
- The `dragArea` defines the area in which the draggable element movement will be occurring. Outside that area will be restricted for the draggable element movement.
- The `onDragStart` event will be triggered before dragging the item from Tab.
- The `dragging` event will be triggered when the Tab item is being dragged.
- The `dragged` event will be triggered when the Tab item is dropped on the target element successfully.

In the following sample, the [allowDragAndDrop](#) property is enabled.

#### INDEX.JSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { TabComponent, TabItemDirective, TabItemsDirective } from
 '@syncfusion/ej2-react-navigations';
const ReactApp = () => {
  let headerText = [{ text: "India" }, { text: "Australia" }, { text: "USA"
 }, { text: "France" }];
  const indiaContent = () => {
    return <div>
      India officially the Republic of India, is a country in South Asia. It
 is the seventh-largest country by area, the second-most populous country with
 over 1.2 billion people, and the most populous democracy in the world.
 Bounded by the Indian Ocean on the south, the Arabian Sea on the south-west,
 and the Bay of Bengal on the south-east, it shares land borders with Pakistan
 to the west; China, Nepal, and Bhutan to the north-east; and Burma and
 Bangladesh to the east. In the Indian Ocean, India is in the vicinity of Sri
 Lanka and the Maldives; in addition, India Andaman and Nicobar Islands share
 a maritime border with Thailand and Indonesia.
    </div>;
  }
  const australiaContent = () => {
    return <div>
      Australia, officially the Commonwealth of Australia, is a country
 comprising the mainland of the Australian continent, the island of Tasmania
 and numerous smaller islands. It is the world sixth-largest country by total
 area. Neighboring countries include Indonesia, East Timor and Papua New
 Guinea to the north; the Solomon Islands, Vanuatu and New Caledonia to the
 north-east; and New Zealand to the south-east.
    </div>;
  }
  const unitedContent = () => {
    return <div>
      The United States of America (USA or U.S.A.), commonly called the
 United States (US or U.S.) and America, is a federal republic consisting of
 fifty states and a federal district. The 48 contiguous states and the federal
 district of Washington, D.C. are in central North America between Canada and
 Mexico. The state of Alaska is west of Canada and east of Russia across the
 Bering Strait, and the state of Hawaii is in the mid-North Pacific. The
 country also has five populated and nine unpopulated territories in the
 Pacific and the Caribbean.
    </div>;
  }
  const franceContent = () => {
    return <div>
      France, officially the French Republic is a sovereign state comprising
 territory in western Europe and several overseas regions and territories. The
 European part of France, called Metropolitan France, extends from the
 Mediterranean Sea to the English Channel and the North Sea, and from the
 Rhine to the Atlantic Ocean; France covers 640,679 square kilo metres and as
 of August 2015 has a population of 67 million, counting all the overseas
 departments and territories.
    </div>;
  }
}
```

```

    return (
      <TabComponent id='draggableTab' heightAdjustMode='Auto'
        allowDragAndDrop={true} dragArea="#container">
        <TabItemsDirective>
          <TabItemDirective header={headerText[0]} content={indiaContent} />
          <TabItemDirective header={headerText[1]} content={australiaContent}
        />
          <TabItemDirective header={headerText[2]} content={unitedContent} />
          <TabItemDirective header={headerText[3]} content={franceContent} />
        </TabItemsDirective>
      </TabComponent>
    );
  }
  const root = ReactDOM.createRoot(document.getElementById('element'));
  root.render(<ReactApp />);

```

## INDEX.TSX

```

import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { TabComponent, TabItemDirective, TabItemsDirective } from
  '@syncfusion/ej2-react-navigations';
const ReactApp = () => {
  let headerText: any = [{ text: "India" }, { text: "Australia" }, { text:
    "USA" }, { text: "France" }];
  const indiaContent = () => {
    return <div>
      India officially the Republic of India, is a country in South Asia. It
is the seventh-largest country by area, the second-most populous country with
      over 1.2 billion people, and the most populous democracy in the world.
      Bounded by the Indian Ocean on the south, the Arabian Sea on the south-west,
      and the Bay of Bengal on the south-east, it shares land borders with Pakistan
      to the west; China, Nepal, and Bhutan to the north-east; and Burma and
      Bangladesh to the east. In the Indian Ocean, India is in the vicinity of Sri
      Lanka and the Maldives; in addition, India Andaman and Nicobar Islands share
      a maritime border with Thailand and Indonesia.
    </div>;
  }
  const australiaContent = () => {
    return <div>
      Australia, officially the Commonwealth of Australia, is a country
      comprising the mainland of the Australian continent, the island of Tasmania
      and numerous smaller islands. It is the world sixth-largest country by total
      area. Neighboring countries include Indonesia, East Timor and Papua New
      Guinea to the north; the Solomon Islands, Vanuatu and New Caledonia to the
      north-east; and New Zealand to the south-east.
    </div>;
  }
  const unitedContent = () => {
    return <div>
      The United States of America (USA or U.S.A.), commonly called the
      United States (US or U.S.) and America, is a federal republic consisting of
      fifty states and a federal district. The 48 contiguous states and the federal
      district of Washington, D.C. are in central North America between Canada and
      Mexico. The state of Alaska is west of Canada and east of Russia across the
      Bering Strait, and the state of Hawaii is in the mid-North Pacific. The

```

```

country also has five populated and nine unpopulated territories in the
Pacific and the Caribbean.
    </div>;
  }
  const franceContent = () => {
    return <div>
      France, officially the French Republic is a sovereign state comprising
territory in western Europe and several overseas regions and territories. The
European part of France, called Metropolitan France, extends from the
Mediterranean Sea to the English Channel and the North Sea, and from the
Rhine to the Atlantic Ocean; France covers 640,679 square kilo metres and as
of August 2015 has a population of 67 million, counting all the overseas
departments and territories.
    </div>;
  }
  return (
    <TabComponent id='draggableTab' heightAdjustMode='Auto'
allowDragAndDrop={true} dragArea="#container">
      <TabItemsDirective>
        <TabItemDirective header={headerText[0]} content={indiaContent} />
        <TabItemDirective header={headerText[1]} content={australiaContent}
/>
        <TabItemDirective header={headerText[2]} content={unitedContent} />
        <TabItemDirective header={headerText[3]} content={franceContent} />
      </TabItemsDirective>
    </TabComponent>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

### Drag and drop item between tabs

It is possible to drag and drop the tab items between two tabs, by manually saving those dropped items as new tab item data through the `addTab` method of Tab and removing the dragged item through the `removeTab` method of Tab.

In this example, we have used the tab control as an external source, and the item from the tab component is dragged and dropped onto another Tab. Therefore, it is necessary to use the `onDragStart` and `dragged` event of the Tab component, where we can form an event object and save it using the `addTab` method of the Tab and remove the dragged item through `removeTab` method of Tab using the dragged item index.

### INDEX.JSX

```

import { useRef } from 'react';
import * as React from "react";
import * as ReactDOM from 'react-dom';
import { TabComponent, TabItemDirective, TabItemsDirective, DragEventArgs }
from '@syncfusion/ej2-react-navigations';
import { isNullOrUndefined } from "@syncfusion/ej2-base";
const ReactApp = () => {
  const firstTabObj = useRef(null);
  const secondTabObj = useRef(null);

```

```

    let headerText = [{ text: "India" }, { text: "Australia" }, { text: "USA"
}, { text: "France" }, { text: "HTML" }, { text: "C Sharp(C#)" }, { text:
"Java" }, { text: "VB.Net" }];
    let allowDragAndDrop = true;
    let firstTabitem = [];
    let secondTabitem = [];
    let dragItemIndex;
    let dragItemContainer;
    const firstTabdragStart = (args) => {
        firstTabitem = [firstTabObj.current.items[args.index]];
        args.draggedItem.style.visibility = 'hidden';
        dragItemContainer = args.draggedItem.closest('.e-tab');
    }
    const firstTabDragStop = (args) => {
        if (!isNullOrUndefined(args.target.closest('.e-tab')) &&
!dragItemContainer.isSameNode(args.target.closest('.e-tab'))) {
            args.cancel = true;
            let TabElement = args.target.closest('.e-tab');
            let dropItem = args.target.closest('.e-toolbar-item');
            if (TabElement !== null && dropItem !== null) {
                dragItemIndex =
Array.prototype.indexOf.call(firstTabObj.current.element.querySelectorAll('.e
-toolbar-item'), args.draggedItem);
                let dropItemContainer = args.target.closest('.e-toolbar-items');
                let dropItemIndex = (dropItemContainer !== null) ?
(Array.prototype.slice.call(dropItemContainer.querySelectorAll('.e-toolbar-
item'))).indexOf(dropItem) : -1;
                secondTabObj.current.addTab(firstTabitem, dropItemIndex);
                firstTabObj.current.removeTab(dragItemIndex);
            }
        }
    }
    const secondTabdragStart = (args) => {
        secondTabitem = [secondTabObj.current.items[args.index]];
        args.draggedItem.style.visibility = 'hidden';
        dragItemContainer = args.draggedItem.closest('.e-tab');
    }
    const secondTabDragStop = (args) => {
        if (!isNullOrUndefined(args.target.closest('.e-tab')) &&
!dragItemContainer.isSameNode(args.target.closest('.e-tab'))) {
            args.cancel = true;
            let TabElement = args.target.closest('.e-tab');
            let dropItem = args.target.closest('.e-toolbar-item');
            if (TabElement !== null && dropItem !== null) {
                dragItemIndex =
Array.prototype.indexOf.call(secondTabObj.current.element.querySelectorAll('.
e-toolbar-item'), args.draggedItem);
                let dropItemContainer = args.target.closest('.e-toolbar-items');
                let dropItemIndex = (dropItemContainer !== null) ?
(Array.prototype.slice.call(dropItemContainer.querySelectorAll('.e-toolbar-
item'))).indexOf(dropItem) : -1;
                firstTabObj.current.addTab(secondTabitem, dropItemIndex);
                secondTabObj.current.removeTab(dragItemIndex);
            }
        }
    }
}
const indiaContent = () => {

```



```

    return <div>
      India officially the Republic of India, is a country in South Asia. It
is the seventh-largest country by area, the second-most populous country with
over 1.2 billion people, and the most populous democracy in the world.
Bounded by the Indian Ocean on the south, the Arabian Sea on the south-west,
and the Bay of Bengal on the south-east, it shares land borders with Pakistan
to the west; China, Nepal, and Bhutan to the north-east; and Burma and
Bangladesh to the east. In the Indian Ocean, India is in the vicinity of Sri
Lanka and the Maldives; in addition, India Andaman and Nicobar Islands share
a maritime border with Thailand and Indonesia.
    </div>;
  }
  const ausContent = () => {
    return <div>
      Australia, officially the Commonwealth of Australia, is a country
comprising the mainland of the Australian continent, the island of Tasmania
and numerous smaller islands. It is the world sixth-largest country by total
area. Neighboring countries include Indonesia, East Timor and Papua New
Guinea to the north; the Solomon Islands, Vanuatu and New Caledonia to the
north-east; and New Zealand to the south-east.
    </div>;
  }
  const usaContent = () => {
    return <div>
      The United States of America (USA or U.S.A.), commonly called the
United States (US or U.S.) and America, is a federal republic consisting of
fifty states and a federal district. The 48 contiguous states and the federal
district of Washington, D.C. are in central North America between Canada and
Mexico. The state of Alaska is west of Canada and east of Russia across the
Bering Strait, and the state of Hawaii is in the mid-North Pacific. The
country also has five populated and nine unpopulated territories in the
Pacific and the Caribbean.
    </div>;
  }
  const franceContent = () => {
    return <div>
      France, officially the French Republic is a sovereign state comprising
territory in western Europe and several overseas regions and territories. The
European part of France, called Metropolitan France, extends from the
Mediterranean Sea to the English Channel and the North Sea, and from the
Rhine to the Atlantic Ocean; France covers 640,679 square kilo metres and as
of August 2015 has a population of 67 million, counting all the overseas
departments and territories.
    </div>;
  }
  const htmlContent = () => {
    return <div>
      HyperText Markup Language, commonly referred to as HTML, is the
standard markup language used to create web pages. Along with CSS, and
JavaScript, HTML is a cornerstone technology, used by most websites to create
visually engaging web pages, user interfaces for web applications, and user
interfaces for many mobile applications.[1] Web browsers can read HTML files
and render them into visible or audible web pages. HTML describes the
structure of a website semantically along with cues for presentation, making
it a markup language, rather than a programming language.
    </div>;
  }
}

```

```

const csharpContent = () => {
  return <div>
    C# is intended to be a simple, modern, general-purpose, object-oriented
    programming language. Its development team is led by Anders Hejlsberg. The
    most recent version is C# 5.0, which was released on August 15, 2012.
  </div>;
}

const javaContent = () => {
  return <div>
    Java is a set of computer software and specifications developed by Sun
    Microsystems, later acquired by Oracle Corporation, that provides a system
    for developing application software and deploying it in a cross-platform
    computing environment. Java is used in a wide variety of computing platforms
    from embedded devices and mobile phones to enterprise servers and
    supercomputers. While less common, Java applets run in secure, sandboxed
    environments to provide many features of native applications and can be
    embedded in HTML pages.
  </div>;
}

const vbNetContent = () => {
  return <div>
    The command-line compiler, VBC.EXE, is installed as part of the
    freeware .NET Framework SDK. Mono also includes a command-line VB.NET
    compiler. The most recent version is VB 2012, which was released on August
    15, 2012.
  </div>;
}

return (
  <div>
    <TabComponent ref={firstTabObj} id='firstTab' heightAdjustMode='Auto'
    allowDragAndDrop={allowDragAndDrop} dragArea="#container"
    onDragStart={firstTabDragStart} dragged={firstTabDragStop}>
      <TabItemsDirective>
        <TabItemDirective header={headerText[0]} content={indiaContent} />
        <TabItemDirective header={headerText[1]} content={ausContent} />
        <TabItemDirective header={headerText[2]} content={usaContent} />
        <TabItemDirective header={headerText[3]} content={franceContent} />
      </TabItemsDirective>
    </TabComponent>
    <br></br>
    <TabComponent ref={secondTabObj} id='secondTab' heightAdjustMode='Auto'
    allowDragAndDrop={allowDragAndDrop} dragArea="#container"
    onDragStart={secondTabDragStart} dragged={secondTabDragStop}>
      <TabItemsDirective>
        <TabItemDirective header={headerText[4]} content={htmlContent} />
        <TabItemDirective header={headerText[5]} content={csharpContent} />
        <TabItemDirective header={headerText[6]} content={javaContent} />
        <TabItemDirective header={headerText[7]} content={vbNetContent} />
      </TabItemsDirective>
    </TabComponent>
  </div>
);
}

const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

**INDEX.TSX**

```

import { useRef } from 'react';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { TabComponent, TabItemDirective, TabItemsDirective, DragEventArgs }
from '@syncfusion/ej2-react-navigations';
import { isNullOrUndefined } from '@syncfusion/ej2-base';
const ReactApp = () => {
    const firstTabObj = useRef<TabComponent>(null);
    const secondTabObj = useRef<TabComponent>(null);
    let headerText: any = [{ text: "India" }, { text: "Australia" }, { text:
"USA" }, { text: "France" }, { text: "HTML" }, { text: "C Sharp(C#)" }, {
text: "Java" }, { text: "VB.Net" }];
    let allowDragAndDrop: boolean = true;
    let firstTabitem: Object[] = [];
    let secondTabitem: Object[] = [];
    let dragItemIndex: number;
    let dragItemContainer: Element;
    const firstTabdragStart = (args: DragEventArgs) => {
        firstTabitem = [firstTabObj.current.items[args.index]];
        args.draggedItem.style.visibility = 'hidden';
        dragItemContainer = args.draggedItem.closest('.e-tab') as Element;
    }
    const firstTabDragStop = (args: DragEventArgs) => {
        if (!isNullOrUndefined(args.target.closest('.e-tab') as HTMLElement) &&
!dragItemContainer.isSameNode(args.target.closest('.e-tab'))) {
            args.cancel = true;
            let TabElement: Element = args.target.closest('.e-tab') as Element;
            let dropItem: Element = args.target.closest('.e-toolbar-item') as
Element;
            if (TabElement != null && dropItem != null) {
                dragItemIndex =
Array.prototype.indexOf.call(firstTabObj.current.element.querySelectorAll('.e
-toolbar-item'), args.draggedItem);
                let dropItemContainer: Element = args.target.closest('.e-toolbar-
items') as Element;
                let dropItemIndex: number = (dropItemContainer != null) ?
(Array.prototype.slice.call(dropItemContainer.querySelectorAll('.e-toolbar-
item'))).indexOf(dropItem) : -1;
                secondTabObj.current.addTab(firstTabitem, dropItemIndex);
                firstTabObj.current.removeTab(dragItemIndex);
            }
        }
    }
    const secondTabdragStart = (args: DragEventArgs) => {
        secondTabitem = [secondTabObj.current.items[args.index]];
        args.draggedItem.style.visibility = 'hidden';
        dragItemContainer = args.draggedItem.closest('.e-tab') as Element;
    }
    const secondTabDragStop = (args: DragEventArgs) => {
        if (!isNullOrUndefined(args.target.closest('.e-tab') as HTMLElement) &&
!dragItemContainer.isSameNode(args.target.closest('.e-tab'))) {
            args.cancel = true;
            let TabElement: Element = args.target.closest('.e-tab') as Element;
            let dropItem: Element = args.target.closest('.e-toolbar-item') as
Element;

```

```

    if (TabElement !== null && dropItem !== null) {
      dragItemIndex =
        Array.prototype.indexOf.call(secondTabObj.current.element.querySelectorAll('.e-toolbar-item'), args.draggedItem);
      let dropItemContainer: Element = args.target.closest('.e-toolbar-items') as Element;
      let dropItemIndex: number = (dropItemContainer !== null) ?
        (Array.prototype.slice.call(dropItemContainer.querySelectorAll('.e-toolbar-item')).indexOf(dropItem) : -1;
      firstTabObj.current.addTab(secondTabitem, dropItemIndex);
      secondTabObj.current.removeTab(dragItemIndex);
    }
  }
}

const indiaContent = () => {
  return <div>
    India officially the Republic of India, is a country in South Asia. It is the seventh-largest country by area, the second-most populous country with over 1.2 billion people, and the most populous democracy in the world. Bounded by the Indian Ocean on the south, the Arabian Sea on the south-west, and the Bay of Bengal on the south-east, it shares land borders with Pakistan to the west; China, Nepal, and Bhutan to the north-east; and Burma and Bangladesh to the east. In the Indian Ocean, India is in the vicinity of Sri Lanka and the Maldives; in addition, India Andaman and Nicobar Islands share a maritime border with Thailand and Indonesia.
  </div>;
}

const ausContent = () => {
  return <div>
    Australia, officially the Commonwealth of Australia, is a country comprising the mainland of the Australian continent, the island of Tasmania and numerous smaller islands. It is the world sixth-largest country by total area. Neighboring countries include Indonesia, East Timor and Papua New Guinea to the north; the Solomon Islands, Vanuatu and New Caledonia to the north-east; and New Zealand to the south-east.
  </div>;
}

const usaContent = () => {
  return <div>
    The United States of America (USA or U.S.A.), commonly called the United States (US or U.S.) and America, is a federal republic consisting of fifty states and a federal district. The 48 contiguous states and the federal district of Washington, D.C. are in central North America between Canada and Mexico. The state of Alaska is west of Canada and east of Russia across the Bering Strait, and the state of Hawaii is in the mid-North Pacific. The country also has five populated and nine unpopulated territories in the Pacific and the Caribbean.
  </div>;
}

const franceContent = () => {
  return <div>
    France, officially the French Republic is a sovereign state comprising territory in western Europe and several overseas regions and territories. The European part of France, called Metropolitan France, extends from the Mediterranean Sea to the English Channel and the North Sea, and from the Rhine to the Atlantic Ocean; France covers 640,679 square kilo metres and as

```

of August 2015 has a population of 67 million, counting all the overseas departments and territories.

</div>;

}

const htmlContent = () => {

return <div>

HyperText Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages. Along with CSS, and JavaScript, HTML is a cornerstone technology, used by most websites to create visually engaging web pages, user interfaces for web applications, and user interfaces for many mobile applications.[1] Web browsers can read HTML files and render them into visible or audible web pages. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language.

</div>;

}

const csharpContent = () => {

return <div>

C# is intended to be a simple, modern, general-purpose, object-oriented programming language. Its development team is led by Anders Hejlsberg. The most recent version is C# 5.0, which was released on August 15, 2012.

</div>;

}

const javaContent = () => {

return <div>

Java is a set of computer software and specifications developed by Sun Microsystems, later acquired by Oracle Corporation, that provides a system for developing application software and deploying it in a cross-platform computing environment. Java is used in a wide variety of computing platforms from embedded devices and mobile phones to enterprise servers and supercomputers. While less common, Java applets run in secure, sandboxed environments to provide many features of native applications and can be embedded in HTML pages.

</div>;

}

const vbNetContent = () => {

return <div>

The command-line compiler, VBC.EXE, is installed as part of the freeware .NET Framework SDK. Mono also includes a command-line VB.NET compiler. The most recent version is VB 2012, which was released on August 15, 2012.

</div>;

}

return (

<div>

<TabComponent ref={firstTabObj} id='firstTab' heightAdjustMode='Auto' allowDragAndDrop={allowDragAndDrop} dragArea="#container" onDragStart={firstTabdragStart} dragged={firstTabDragStop}>

<TabItemsDirective>

<TabItemDirective header={headerText[0]} content={indiaContent} />

<TabItemDirective header={headerText[1]} content={ausContent} />

<TabItemDirective header={headerText[2]} content={usaContent} />

<TabItemDirective header={headerText[3]} content={franceContent} />

</TabItemsDirective>

</TabComponent>

<br></br>

```

    <TabComponent ref={secondTabObj} id='secondTab' heightAdjustMode='Auto'
    allowDragAndDrop={allowDragAndDrop} dragArea="#container"
    onDragStart={secondTabDragStart} dragged={secondTabDragStop}>
      <TabItemsDirective>
        <TabItemDirective header={headerText[4]} content={htmlContent} />
        <TabItemDirective header={headerText[5]} content={csharpContent} />
        <TabItemDirective header={headerText[6]} content={javaContent} />
        <TabItemDirective header={headerText[7]} content={vbNetContent} />
      </TabItemsDirective>
    </TabComponent>
  </div>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

### Drag and drop items to external source

It is possible to drag and drop the items to any of the external sources from the Tab, by manually saving those dropped items as new node data through the `addNodes` method of Treeview and removing the dragged item through the `removeTab` method of Tab.

In this example, we have used the tree view control as an external source, and the item from the tab component is dragged and dropped onto the child nodes of the tree view component. Therefore, it is necessary to use the `dragged` event of the Tab component, where we can form an event object and save it using the `addNodes` method of the Treeview and remove the dragged item through the `removeTab` method of Tab using the dragged item index.

### INDEX.JSX

```

import { useRef, useEffect } from 'react';
import * as React from "react";
import * as ReactDOM from 'react-dom';
import { TabComponent, TabItemDirective, TabItemsDirective, TreeViewComponent }
  from '@syncfusion/ej2-react-navigations';
import { isNullOrUndefined } from "@syncfusion/ej2-base";
const ReactApp = () => {
  const tabObj = useRef(null);
  const treeObj = useRef(null);
  let headerText = [{ text: "India" }, { text: "Australia" }, { text: "USA"
}, { text: "France" }];
  let allowDragAndDrop = true;
  let data = [
    { text: 'Hennessey Venom', id: 'list-01' },
    { text: 'Bugatti Chiron', id: 'list-02' },
    { text: 'Bugatti Veyron Super Sport', id: 'list-03' },
    { text: 'SSC Ultimate Aero', id: 'list-04' },
    { text: 'Koenigsegg CCR', id: 'list-05' },
    { text: 'McLaren F1', id: 'list-06' },
    { text: 'Aston Martin One- 77', id: 'list-07' },
    { text: 'Jaguar XJ220', id: 'list-08' },
    { text: 'McLaren P1', id: 'list-09' },
    { text: 'Ferrari LaFerrari', id: 'list-10' },
  ];
  let field = { dataSource: data, id: 'id', text: 'text' };
  let i = 0;

```

```

useEffect(() => {
  const tabElement = tabObj.current.element;
  if (!isNullOrUndefined(tabElement)) {
    tabElement.querySelector('.e-tab-header').classList.add('e-droppable');
    tabElement.querySelector('.e-content').classList.add('tab-content');
  }
}, []);
const tabDragStop = (args) => {
  const dragTabIndex =
Array.prototype.indexOf.call(tabObj.current.element.querySelectorAll('.e-
toolbar-item'), args.draggedItem);
  const dragItem = tabObj.current.items[dragTabIndex];
  const dropNode = args.target.closest('#draggableTreeview .e-list-item');
  if (dropNode !== null && !args.target.closest('#draggableTab .e-toolbar-
item')) {
    args.cancel = true;
    const dropContainer = document.querySelector('.treeview-external-drop-
tab').querySelectorAll('.e-list-item');
    const dropIndex = Array.prototype.indexOf.call(dropContainer,
dropNode);
    const newNode = [{ id: 'list' + i, text: dragItem.header ?
dragItem.header.text : '' }];
    tabObj.current.removeTab(dragTabIndex);
    treeObj.current.addNodes(newNode, 'Treeview', dropIndex);
  }
}
const indiaContent = () => {
  return <div>
    India officially the Republic of India, is a country in South Asia. It
is the seventh-largest country by area, the second-most populous country with
over 1.2 billion people, and the most populous democracy in the world.
Bounded by the Indian Ocean on the south, the Arabian Sea on the south-west,
and the Bay of Bengal on the south-east, it shares land borders with Pakistan
to the west; China, Nepal, and Bhutan to the north-east; and Burma and
Bangladesh to the east. In the Indian Ocean, India is in the vicinity of Sri
Lanka and the Maldives; in addition, India Andaman and Nicobar Islands share
a maritime border with Thailand and Indonesia.
  </div>
}
const ausContent = () => {
  return <div>
    Australia, officially the Commonwealth of Australia, is a country
comprising the mainland of the Australian continent, the island of Tasmania
and numerous smaller islands. It is the world sixth-largest country by total
area. Neighboring countries include Indonesia, East Timor and Papua New
Guinea to the north; the Solomon Islands, Vanuatu and New Caledonia to the
north-east; and New Zealand to the south-east.
  </div>
}
const usaContent = () => {
  return <div>
    The United States of America (USA or U.S.A.), commonly called the
United States (US or U.S.) and America, is a federal republic consisting of
fifty states and a federal district. The 48 contiguous states and the federal
district of Washington, D.C. are in central North America between Canada and
Mexico. The state of Alaska is west of Canada and east of Russia across the
Bering Strait, and the state of Hawaii is in the mid-North Pacific. The

```

```

country also has five populated and nine unpopulated territories in the
Pacific and the Caribbean.
    </div>;
  }
  const franceContent = () => {
    return <div>
      France, officially the French Republic is a sovereign state comprising
territory in western Europe and several overseas regions and territories. The
European part of France, called Metropolitan France, extends from the
Mediterranean Sea to the English Channel and the North Sea, and from the
Rhine to the Atlantic Ocean; France covers 640,679 square kilo metres and as
of August 2015 has a population of 67 million, counting all the overseas
departments and territories.
    </div>;
  }
  return (
    <div>
      <TabComponent ref={tabObj} heightAdjustMode='Auto'
allowDragAndDrop={allowDragAndDrop} dragArea="#container"
dragged={tabDragStop}>
        <TabItemsDirective>
          <TabItemDirective header={headerText[0]} content={indiaContent} />
          <TabItemDirective header={headerText[1]} content={ausContent} />
          <TabItemDirective header={headerText[2]} content={usaContent} />
          <TabItemDirective header={headerText[3]} content={franceContent} />
        </TabItemsDirective>
      </TabComponent>
      <br></br>
      <TreeViewComponent id="draggableTreeview" ref={treeObj}
cssClass="treeview-external-drop-tab" fields={field} />
    </div>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

## INDEX.TSX

```

import { useRef, useEffect } from 'react';
import * as React from "react";
import * as ReactDOM from 'react-dom';
import { TabComponent, TabItemDirective, TabItemsDirective, TreeViewComponent
} from '@syncfusion/ej2-react-navigations';
import { isNullOrUndefined } from "@syncfusion/ej2-base";
const ReactApp = () => {
  const tabObj = useRef<TabComponent>(null);
  const treeObj = useRef<TreeViewComponent>(null);
  let headerText: any = [{ text: "India" }, { text: "Australia" }, { text:
"USA" }, { text: "France" }];
  let allowDragAndDrop: boolean = true;
  let data: { [key: string]: Object }[] = [
    { text: 'Hennessey Venom', id: 'list-01' },
    { text: 'Bugatti Chiron', id: 'list-02' },
    { text: 'Bugatti Veyron Super Sport', id: 'list-03' },
    { text: 'SSC Ultimate Aero', id: 'list-04' },
    { text: 'Koenigsegg CCR', id: 'list-05' },

```



```

    { text: 'McLaren F1', id: 'list-06' },
    { text: 'Aston Martin One- 77', id: 'list-07' },
    { text: 'Jaguar XJ220', id: 'list-08' },
    { text: 'McLaren P1', id: 'list-09' },
    { text: 'Ferrari LaFerrari', id: 'list-10' },
  ];
  let field: Object = { dataSource: data, id: 'id', text: 'text' };
  let i: number = 0;
  useEffect(() => {
    const tabElement = tabObj.current.element;
    if (!isNullOrUndefined(tabElement)) {
      tabElement.querySelector('.e-tab-header').classList.add('e-droppable');
      tabElement.querySelector('.e-content').classList.add('tab-content');
    }
  }, []);
  const tabDragStop = (args) => {
    const dragTabIndex =
      Array.prototype.indexOf.call(tabObj.current.element.querySelectorAll('.e-
toolbar-item'), args.draggedItem);
    const dragItem = tabObj.current.items[dragTabIndex];
    const dropNode = args.target.closest('#draggableTreeview .e-list-item');
    if (dropNode !== null && !args.target.closest('#draggableTab .e-toolbar-
item')) {
      args.cancel = true;
      const dropContainer = document.querySelector('.treeview-external-drop-
tab').querySelectorAll('.e-list-item');
      const dropIndex = Array.prototype.indexOf.call(dropContainer,
dropNode);
      const newNode = [{ id: 'list' + i, text: dragItem.header ?
dragItem.header.text : '' }];
      tabObj.current.removeTab(dragTabIndex);
      treeObj.current.addNodes(newNode, 'Treeview', dropIndex);
    }
  }
  const indiaContent = () => {
    return <div>
      India officially the Republic of India, is a country in South Asia. It
is the seventh-largest country by area, the second-most populous country with
over 1.2 billion people, and the most populous democracy in the world.
Bounded by the Indian Ocean on the south, the Arabian Sea on the south-west,
and the Bay of Bengal on the south-east, it shares land borders with Pakistan
to the west; China, Nepal, and Bhutan to the north-east; and Burma and
Bangladesh to the east. In the Indian Ocean, India is in the vicinity of Sri
Lanka and the Maldives; in addition, India Andaman and Nicobar Islands share
a maritime border with Thailand and Indonesia.
    </div>;
  }
  const ausContent = () => {
    return <div>
      Australia, officially the Commonwealth of Australia, is a country
comprising the mainland of the Australian continent, the island of Tasmania
and numerous smaller islands. It is the world sixth-largest country by total
area. Neighboring countries include Indonesia, East Timor and Papua New
Guinea to the north; the Solomon Islands, Vanuatu and New Caledonia to the
north-east; and New Zealand to the south-east.
    </div>;
  }
}

```

```

const usaContent = () => {
  return <div>
    The United States of America (USA or U.S.A.), commonly called the
    United States (US or U.S.) and America, is a federal republic consisting of
    fifty states and a federal district. The 48 contiguous states and the federal
    district of Washington, D.C. are in central North America between Canada and
    Mexico. The state of Alaska is west of Canada and east of Russia across the
    Bering Strait, and the state of Hawaii is in the mid-North Pacific. The
    country also has five populated and nine unpopulated territories in the
    Pacific and the Caribbean.
  </div>;
}
const franceContent = () => {
  return <div>
    France, officially the French Republic is a sovereign state comprising
    territory in western Europe and several overseas regions and territories. The
    European part of France, called Metropolitan France, extends from the
    Mediterranean Sea to the English Channel and the North Sea, and from the
    Rhine to the Atlantic Ocean; France covers 640,679 square kilo metres and as
    of August 2015 has a population of 67 million, counting all the overseas
    departments and territories.
  </div>;
}
return (
  <div>
    <TabComponent ref={tabObj} heightAdjustMode='Auto'
      allowDragAndDrop={allowDragAndDrop} dragArea="#container"
      dragged={tabDragStop}>
      <TabItemsDirective>
        <TabItemDirective header={headerText[0]} content={indiaContent} />
        <TabItemDirective header={headerText[1]} content={ausContent} />
        <TabItemDirective header={headerText[2]} content={usaContent} />
        <TabItemDirective header={headerText[3]} content={franceContent} />
      </TabItemsDirective>
    </TabComponent>
    <br></br>
    <TreeViewComponent id="draggableTreeview" ref={treeObj}
      cssClass="treeview-external-drop-tab" fields={field} />
  </div>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

### Drag and drop items from external source

It is possible to drag and drop the items from any of the external sources into the Tab, by manually saving those dropped items as new item data through the `addTab` method of Tab and removing the dragged node through the `removeNodes` method of Treeview.

In this example, we have used the tree view control as an external source, and the child nodes from the tree view component are dragged and dropped onto the Tab. Therefore, it is necessary to use the `nodeDragStop` event of the Treeview component, where we can form an event object and save it using the `addTab` method of the Tab and remove the dragged node through the `removeNodes` method of Treeview.

**INDEX.JSX**

```

import { useRef } from 'react';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { TabComponent, TabItemDirective, TabItemsDirective, TreeViewComponent }
  from '@syncfusion/ej2-react-navigations';
import { isNullOrUndefined } from '@syncfusion/ej2-base';
const ReactApp = () => {
  const tabObj = useRef(null);
  const treeObj = useRef(null);
  let headerText = [{ text: "India" }, { text: "Australia" }, { text: "USA"
}, { text: "France" }];
  let allowDragAndDrop = true;
  let data = [
    { text: 'Hennessey Venom', id: 'list-01' },
    { text: 'Bugatti Chiron', id: 'list-02' },
    { text: 'Bugatti Veyron Super Sport', id: 'list-03' },
    { text: 'SSC Ultimate Aero', id: 'list-04' },
    { text: 'Koenigsegg CCR', id: 'list-05' },
    { text: 'McLaren F1', id: 'list-06' },
    { text: 'Aston Martin One- 77', id: 'list-07' },
    { text: 'Jaguar XJ220', id: 'list-08' },
    { text: 'McLaren P1', id: 'list-09' },
    { text: 'Ferrari LaFerrari', id: 'list-10' },
  ];
  let field = { dataSource: data, id: 'id', text: 'text' };
  const onNodeDragStop = (args) => {
    let dropElement = args.target.closest('#draggableTab .e-toolbar-item');
    if (dropElement !== null) {
      let tabElement = tabObj.current.element;
      let dropItemIndex = [].slice.call(tabElement.querySelectorAll('.e-
toolbar-item')).indexOf(dropElement);
      let newTabItem = [{
        header: { 'text': args.draggedNodeData.text.toString() },
        content: args.draggedNodeData.text.toString() + ' Content'
      }];
      tabObj.current.addTab(newTabItem, dropItemIndex);
      treeObj.current.removeNodes([args.draggedNode]);
      args.cancel = true;
    } else {
      let dropNode = args.target.closest('#draggableTreeview .e-list-item ');
      if (!isNullOrUndefined(dropNode) && args.dropIndicator === 'e-drop-in')
    {
      args.cancel = true;
    }
  }
}
const onNodeDrag = (args) => {
  if (!isNullOrUndefined(args.target.closest('.tab-content'))) {
    args.dropIndicator = 'e-no-drop';
  } else if (!isNullOrUndefined(args.target.closest('#draggableTab .e-tab-
header'))) {
    args.dropIndicator = 'e-drop-in';
  }
}
const indiaContent = () => {

```

```

    return <div>
      India officially the Republic of India, is a country in South Asia. It
is the seventh-largest country by area, the second-most populous country with
over 1.2 billion people, and the most populous democracy in the world.
Bounded by the Indian Ocean on the south, the Arabian Sea on the south-west,
and the Bay of Bengal on the south-east, it shares land borders with Pakistan
to the west; China, Nepal, and Bhutan to the north-east; and Burma and
Bangladesh to the east. In the Indian Ocean, India is in the vicinity of Sri
Lanka and the Maldives; in addition, India Andaman and Nicobar Islands share
a maritime border with Thailand and Indonesia.
    </div>;
  }
  const ausContent = () => {
    return <div>
      Australia, officially the Commonwealth of Australia, is a country
comprising the mainland of the Australian continent, the island of Tasmania
and numerous smaller islands. It is the world sixth-largest country by total
area. Neighboring countries include Indonesia, East Timor and Papua New
Guinea to the north; the Solomon Islands, Vanuatu and New Caledonia to the
north-east; and New Zealand to the south-east.
    </div>;
  }
  const usaContent = () => {
    return <div>
      The United States of America (USA or U.S.A.), commonly called the
United States (US or U.S.) and America, is a federal republic consisting of
fifty states and a federal district. The 48 contiguous states and the federal
district of Washington, D.C. are in central North America between Canada and
Mexico. The state of Alaska is west of Canada and east of Russia across the
Bering Strait, and the state of Hawaii is in the mid-North Pacific. The
country also has five populated and nine unpopulated territories in the
Pacific and the Caribbean.
    </div>;
  }
  const franceContent = () => {
    return <div>
      France, officially the French Republic is a sovereign state comprising
territory in western Europe and several overseas regions and territories. The
European part of France, called Metropolitan France, extends from the
Mediterranean Sea to the English Channel and the North Sea, and from the
Rhine to the Atlantic Ocean; France covers 640,679 square kilo metres and as
of August 2015 has a population of 67 million, counting all the overseas
departments and territories.
    </div>;
  }
  return (
    <div>
      <TabComponent ref={tabObj} id='draggableTab' heightAdjustMode='Auto'
dragArea="#container">
        <TabItemsDirective>
          <TabItemDirective header={headerText[0]} content={indiaContent} />
          <TabItemDirective header={headerText[1]} content={ausContent} />
          <TabItemDirective header={headerText[2]} content={usaContent} />
          <TabItemDirective header={headerText[3]} content={franceContent} />
        </TabItemsDirective>
      </TabComponent>
    <br></br>
  )

```

```

    <TreeViewComponent id="draggableTreeview" ref={treeObj}
    dragArea="#container" cssClass="treeview-external-drop-tab" fields={field}
    nodeDragStop={onNodeDragStop} nodeDragging={onNodeDrag}
    allowDragAndDrop={allowDragAndDrop} />
  </div>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

## INDEX.TSX

```

import { useRef } from 'react';
import * as React from "react";
import * as ReactDOM from 'react-dom';
import { TabComponent, TabItemDirective, TabItemsDirective, TabItemModel,
TreeViewComponent, DragAndDropEventArgs } from '@syncfusion/ej2-react-
navigations';
import { isNullOrUndefined } from "@syncfusion/ej2-base";
const ReactApp = () => {
  const tabObj = useRef<TabComponent>(null);
  const treeObj = useRef<TreeViewComponent>(null);
  let headerText: any = [{ text: "India" }, { text: "Australia" }, { text:
"USA" }, { text: "France" }];
  let allowDragAndDrop: boolean = true;
  let data: { [key: string]: Object }[] = [
    { text: 'Hennessey Venom', id: 'list-01' },
    { text: 'Bugatti Chiron', id: 'list-02' },
    { text: 'Bugatti Veyron Super Sport', id: 'list-03' },
    { text: 'SSC Ultimate Aero', id: 'list-04' },
    { text: 'Koenigsegg CCR', id: 'list-05' },
    { text: 'McLaren F1', id: 'list-06' },
    { text: 'Aston Martin One- 77', id: 'list-07' },
    { text: 'Jaguar XJ220', id: 'list-08' },
    { text: 'McLaren P1', id: 'list-09' },
    { text: 'Ferrari LaFerrari', id: 'list-10' },
  ];
  let field: Object = { dataSource: data, id: 'id', text: 'text' };
  const onNodeDragStop = (args: DragAndDropEventArgs) => {
    let dropElement: Element = args.target.closest('#draggableTab .e-toolbar-
item') as Element;
    if (dropElement !== null) {
      let tabElement: HTMLElement = tabObj.current.element;
      let dropItemIndex: number =
[...].slice.call(tabElement.querySelectorAll('.e-toolbar-
item')).indexOf(dropElement as never);
      let newTabItem: TabItemModel[] = [{
        header: { 'text': args.draggedNodeData.text.toString() },
        content: args.draggedNodeData.text.toString() + ' Content'
      }];
      tabObj.current.addTab(newTabItem, dropItemIndex);
      treeObj.current.removeNodes([args.draggedNode]);
      args.cancel = true;
    } else {
      let dropNode: Element = args.target.closest('#draggableTreeview .e-
list-item') as Element;

```

```

    if (!isNullOrUndefined(dropNode) && args.dropIndicator === 'e-drop-in')
    {
        args.cancel = true;
    }
}

const onNodeDrag = (args: DragAndDropEventArgs) => {
    if (!isNullOrUndefined(args.target.closest('.tab-content') as
HTMLInputElement)) {
        args.dropIndicator = 'e-no-drop';
    } else if (!isNullOrUndefined(args.target.closest('#draggableTab .e-tab-
header') as HTMLInputElement)) {
        args.dropIndicator = 'e-drop-in';
    }
}

const indiaContent = () => {
    return <div>
        India officially the Republic of India, is a country in South Asia. It
is the seventh-largest country by area, the second-most populous country with
over 1.2 billion people, and the most populous democracy in the world.
Bounded by the Indian Ocean on the south, the Arabian Sea on the south-west,
and the Bay of Bengal on the south-east, it shares land borders with Pakistan
to the west; China, Nepal, and Bhutan to the north-east; and Burma and
Bangladesh to the east. In the Indian Ocean, India is in the vicinity of Sri
Lanka and the Maldives; in addition, India Andaman and Nicobar Islands share
a maritime border with Thailand and Indonesia.
    </div>;
}

const ausContent = () => {
    return <div>
        Australia, officially the Commonwealth of Australia, is a country
comprising the mainland of the Australian continent, the island of Tasmania
and numerous smaller islands. It is the world sixth-largest country by total
area. Neighboring countries include Indonesia, East Timor and Papua New
Guinea to the north; the Solomon Islands, Vanuatu and New Caledonia to the
north-east; and New Zealand to the south-east.
    </div>;
}

const usaContent = () => {
    return <div>
        The United States of America (USA or U.S.A.), commonly called the
United States (US or U.S.) and America, is a federal republic consisting of
fifty states and a federal district. The 48 contiguous states and the federal
district of Washington, D.C. are in central North America between Canada and
Mexico. The state of Alaska is west of Canada and east of Russia across the
Bering Strait, and the state of Hawaii is in the mid-North Pacific. The
country also has five populated and nine unpopulated territories in the
Pacific and the Caribbean.
    </div>;
}

const franceContent = () => {
    return <div>
        France, officially the French Republic is a sovereign state comprising
territory in western Europe and several overseas regions and territories. The
European part of France, called Metropolitan France, extends from the
Mediterranean Sea to the English Channel and the North Sea, and from the
Rhine to the Atlantic Ocean; France covers 640,679 square kilo metres and as

```

```

of August 2015 has a population of 67 million, counting all the overseas
departments and territories.
    </div>;
  }
  return (
    <div>
      <TabComponent ref={tabObj} id='draggableTab' heightAdjustMode='Auto'
dragArea="#container">
        <TabItemsDirective>
          <TabItemDirective header={headerText[0]} content={indiaContent} />
          <TabItemDirective header={headerText[1]} content={ausContent} />
          <TabItemDirective header={headerText[2]} content={usaContent} />
          <TabItemDirective header={headerText[3]} content={franceContent} />
        </TabItemsDirective>
      </TabComponent>
      <br></br>
      <TreeViewComponent id="draggableTreeview" ref={treeObj}
dragArea="#container" cssClass="treeview-external-drop-tab" fields={field}
nodeDragStop={onNodeDragStop} nodeDragging={onNodeDrag}
allowDragAndDrop={allowDragAndDrop} />
    </div>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

## Accessibility in React Tab component

The Tab component followed the accessibility guidelines and standards, including [ADA, Section 508](#), [WCAG 2.2](#) standards, and [WCAG roles](#) that are commonly used to evaluate accessibility.

The accessibility compliance for the Tab component is outlined below.

| Accessibility Criteria | Compatibility |

| -- | -- |

| [WCAG 2.2](#) Support |  |

| [Section 508](#) Support |  |

| Screen Reader Support |  |

| Right-To-Left Support |  |

| Color Contrast |  |

| Mobile Device Support |  |

| Keyboard Navigation Support |  |

```
| Accessibility Checker Validation |  |
| Axe-core Accessibility Validation |  |

<style>
.post .post-content img {
display: inline-block;
margin: 0.5em 0;
}
</style>

<div> - All
features of the component meet the requirement.</div>

<div> - Some features of the component do not meet the requirement.</div>

<div> - The component does not meet the requirement.</div>
```

### ARIA attributes

Tab component is designed by considering [WAI-ARIA](#) standard. Tab is supported with ARIA Accessibility which is accessible by on-screen readers, and other assistive technology devices. The following list of attributes are added in the Tab.

#### | Roles and Attributes | Functionalities |

| --- | --- |

| tablist | Attribute is set to the Tab header element that describes actual role of the element. |

| tab | Attribute is set to the Tab items element to indicates an interactive element inside a **tablist** that, when activated, displays its associated **tabpanel**. |

| tabpanel | Attribute is set to the Tab content that describes the role for viewing the active content. |

| aria-orientation | Attribute is set to the Tab header element indicates the Tab header orientation. Default value of this attribute is horizontal. |

| aria-selected | Attribute set to the Tab items to indicates the selection state for Tab items. Active Tab is set to true for this attribute. |

| aria-labelledby | Attribute is set to the Tab content element to indicates the associated Tab header for the content. |

| aria-controls | Attribute is set to the Tab items element to indicates the associated **tabpanel** for the header. |

| aria-haspopup | Attribute is set to the Popup element to indicates the popup mode in the Tab. The default value of this attribute is false. If popup mode is enabled, the attribute value is set to true. |

| aria-disabled | Attribute set to the Tab items to It indicates the disabled state of the Tab. |



### Keyboard interaction

By default, keyboard navigation is enabled. This component implements keyboard navigation support by following the WAI-ARIA practices. Once focused on the active Tab element, you can use the following key combination for interacting with the Tab.

Key	Description
Left	Moves focus to the previous Tab. If focus is on the first Tab, the focus will not move to any Tab.
Right	Moves focus to the next Tab. If focus is on the last Tab element, the focus will not move to any Tab.
Enter or Space	Selects the Tab if it is not selected. Opens the popup dropdown icon if it is focussed. Select the Tab item as active when popup item is focussed.
Esc(Escape)	Closes the popup if popup is in opened state.
Down or Up	When the popup is open and focused, it will move to previous/next Tab items of the popup in the vertical direction.
Home	Moves focus to the first Tab.
End	Moves focus to the last Tab.
Shift + F10	If popup mode is enabled, it opens the popup when the Tab is focused.
Delete	Deletes the Tab, if close button is enabled in Tab header.
Tab	To Move focus through the interactive elements.
Shift + Tab	To Move focus through the interactive elements.

### Ensuring accessibility

The Tab component's accessibility levels are ensured through an [accessibility-checker](#) and [axe-core](#) software tools during automated testing.

The accessibility compliance of the Tab component is shown in the following sample. Open the [sample](#) in a new window to evaluate the accessibility of the Tab component with accessibility tools.

See also

- [Accessibility in Syncfusion React components](#)

### Style in React Tab component

The following content provides the exact CSS structure that can be used to modify the control's appearance based on user preference.

#### Customizing Tab

Use the following CSS to customize the Tab.

```
`css
.e-tab {
```

```
border: 5px solid rgb(173, 255, 47);  
}
```

,

### Customizing the Tab items

Use the following CSS to customize the header items of Tab.

```
`css  
.e-tab .e-tab-header .e-toolbar-items {  
background: #9faed8;  
border: 2px solid blue;  
}
```

,

Use the following CSS to customize the content items of Tab.

```
`css  
.e-tab .e-content .e-item {  
color: #a78515;  
font-size: 14px;  
}
```

,

### Customizing Tab's header

Use the following CSS to customize the header of Tab control.

```
`css  
.e-tab .e-tab-header {  
background: #badfba !important;  
}
```

,

### Customizing Tab's header icon

Use the following CSS to customize the header item icon of Tab control.

```
`css  
.e-tab .e-tab-header .e-toolbar-item .e-tab-icon {  
color: #badfba !important;  
}
```

,

### Customizing Tab's content

Use the following CSS to customize the content of Tab control.

```
`css
.e-tab .e-content {
background: #d1f6d1 !important;
}
`
```

#### Customizing the hover state of Tab control

Use the following CSS to customize the tab item when hovering.

```
`css
.e-tab .e-tab-header .e-toolbar-item .e-tab-wrap:hover {
background: #d1f6d1 !important;
}
`
```

Use the following CSS to customize the tab item popup icon when hovering.

```
`css
.e-tab .e-tab-header .e-hor-nav .e-popup-up-icon:hover,
.e-tab .e-tab-header .e-hor-nav .e-popup-down-icon:hover {
background: #d1f6d1 !important;
}
`
```

#### Customizing selected item of Tab control

Use the following CSS to customize the selected tab item.

```
`css
.e-tab .e-tab-header .e-toolbar-item.e-active {
background: #d1f4d1;
}
`
```

Use the following CSS to customize the selected tab item text and icon.

```
`css
.e-tab .e-tab-header .e-toolbar-item.e-active .e-tab-text,
.e-tab .e-tab-header .e-toolbar-item.e-active .e-tab-icon {
color: green !important;
}
`
```

## How To

### Load content through post in React Tab component

The Tab supports to load external contents through **AJAX** library. Refer to the following steps.

- Import the **Ajax** module from **ej2-base** and initialize with URL path.
- Get the data from **Ajax Success** event, then initialize the Tab with retrieved external path data.

#### INDEX.JSX

```
import * as React from "react";
import { useState, useEffect } from 'react';
import * as ReactDOM from 'react-dom';
import { TabComponent, TabItemsDirective, TabItemDirective } from
 '@syncfusion/ej2-react-navigations';
function App() {
  const [headerText, setHeaderText] = useState([
    { text: "Twitter" },
    { text: "Facebook" },
    { text: "WhatsApp" }
  ]);
  const [data, setData] = useState('');
  useEffect(() => {
    const fetchData = async () => {
      const response = await fetch('./ajax.html');
      const data = await response.text();
      setData(data);
    };
    fetchData();
  }, []);
  return (
    <TabComponent heightAdjustMode='Auto' id='tabelement'>
      <TabItemsDirective>
        <TabItemDirective header={headerText[0]}
          content={'Twitter is an online social networking service that
enables users to send and read short 140-character ' +
'messages called "tweets". Registered users can read and post
tweets, but those who are unregistered can only read ' +
'them. Users access Twitter through the website interface, SMS or
mobile device app Twitter Inc. is based in San ' +
'Francisco and has more than 25 offices around the world. Twitter
was created in March 2006 by Jack Dorsey, ' +
'Evan Williams, Biz Stone, and Noah Glass and launched in July
2006. The service rapidly gained worldwide popularity, ' +
'with more than 100 million users posting 340 million tweets a
day in 2012. The service also handled 1.6 billion ' +
'search queries per day.'} />
        <TabItemDirective header={headerText[1]}
          content={'Facebook is an online social networking service
headquartered in Menlo Park, California. Its website was ' +
'launched on February 4, 2004, by Mark Zuckerberg with his
Harvard College roommates and fellow students Eduardo ' +
'Saverin, Andrew McCollum, Dustin Moskovitz and Chris Hughes. The
founders had initially limited the website\' ' +
'membership to Harvard students, but later expanded it to
colleges in the Boston area, the Ivy League, and Stanford ' +
```

```

        'University. It gradually added support for students at various
other universities and later to high-school students.'} />
        <TabItemDirective header={headerText[2]} content={data} />
    </TabItemsDirective>
</TabComponent>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

## INDEX.TSX

```

import * as React from "react";
import { useState, useEffect } from 'react';
import * as ReactDOM from 'react-dom';
import { TabComponent, TabItemsDirective, TabItemDirective } from
'@syncfusion/ej2-react-navigations';
function App() {
    const [headerText, setHeaderText] = useState([
        { text: "Twitter" },
        { text: "Facebook" },
        { text: "WhatsApp" }
    ]);
    const [data, setData] = useState('');
    useEffect(() => {
        const fetchData = async () => {
            const response = await fetch('./ajax.html');
            const data = await response.text();
            setData(data);
        };
        fetchData();
    }, []);
    return (
        <TabComponent heightAdjustMode='Auto' id='tabelement'>
            <TabItemsDirective>
                <TabItemDirective header={headerText[0]}
                    content={'Twitter is an online social networking service that
enables users to send and read short 140-character ' +
'messages called "tweets". Registered users can read and post
tweets, but those who are unregistered can only read ' +
'them. Users access Twitter through the website interface, SMS or
mobile device app Twitter Inc. is based in San ' +
'Francisco and has more than 25 offices around the world. Twitter
was created in March 2006 by Jack Dorsey, ' +
'Evan Williams, Biz Stone, and Noah Glass and launched in July
2006. The service rapidly gained worldwide popularity, ' +
'with more than 100 million users posting 340 million tweets a
day in 2012.The service also handled 1.6 billion ' +
'search queries per day.'} />
                <TabItemDirective header={headerText[1]}
                    content={'Facebook is an online social networking service
headquartered in Menlo Park, California. Its website was ' +
'launched on February 4, 2004, by Mark Zuckerberg with his
Harvard College roommates and fellow students Eduardo ' +
'Saverin, Andrew McCollum, Dustin Moskovitz and Chris Hughes.The
founders had initially limited the website\' ' +

```

```

        'membership to Harvard students, but later expanded it to
        colleges in the Boston area, the Ivy League, and Stanford ' +
        'University. It gradually added support for students at various
        other universities and later to high-school students.'} />
        <TabItemDirective header={headerText[2]} content={data} />
      </TabItemsDirective>
    </TabComponent>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

## INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Tab</title>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta name="description" content="Essential JS 2 for React Components" />
  <meta name="author" content="Syncfusion" />
  <link href="index.css" rel="stylesheet" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/material.css"
rel="stylesheet" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
  <script src="systemjs.config.js"></script>
</head>
<body>
  <div id='element'>
    <div id='loader'>Loading....</div>
  </div>
</body>
</html>

```

## Prevent content swipe selection in React Tab component

We can prevent the tab selection on touch swipe action by using the Tab [selecting](#) event. Refer the below sample with preventing swipe selection.

## INDEX.JSX

```

import * as React from "react";
import { useRef } from 'react';
import * as ReactDOM from 'react-dom/client';
import { TabComponent, TabItemDirective, TabItemsDirective } from
 '@syncfusion/ej2-react-navigations';
const App = () => {
  // eslint-disable-next-line @typescript-eslint/no-unused-vars
  const tabInstance = useRef < TabComponent > (null);
  // define the array of data
  let headertext;
  headertext = [{ text: "Twitter" }, { text: "Facebook" }, { text: "WhatsApp"
}];

```

```

const select = (e) => {
  if (e.isSwiped) {
    e.cancel = true;
  }
}

return (
  <TabComponent heightAdjustMode='Auto' id='tabelement' ref={tabInstance}
  selecting={select}>
    <TabItemsDirective>
      <TabItemDirective header={headertext[0]}
        content={'Twitter is an online social networking service that
        enables users to send and read short 140-character ' +
          'messages called "tweets". Registered users can read and post
        tweets, but those who are unregistered can only read ' +
          'them. Users access Twitter through the website interface, SMS or
        mobile device app Twitter Inc. is based in San ' +
          'Francisco and has more than 25 offices around the world. Twitter
        was created in March 2006 by Jack Dorsey, ' +
          'Evan Williams, Biz Stone, and Noah Glass and launched in July
        2006. The service rapidly gained worldwide popularity, ' +
          'with more than 100 million users posting 340 million tweets a
        day in 2012. The service also handled 1.6 billion ' +
          'search queries per day.'} />
      <TabItemDirective header={headertext[1]}
        content={'Facebook is an online social networking service
        headquartered in Menlo Park, California. Its website was ' +
          'launched on February 4, 2004, by Mark Zuckerberg with his
        Harvard College roommates and fellow students Eduardo ' +
          'Saverin, Andrew McCollum, Dustin Moskovitz and Chris Hughes. The
        founders had initially limited the website\ ' +
          'membership to Harvard students, but later expanded it to
        colleges in the Boston area, the Ivy League, and Stanford ' +
          'University. It gradually added support for students at various
        other universities and later to high-school students.'} />
      <TabItemDirective header={headertext[2]}
        content={'WhatsApp Messenger is a proprietary cross-platform
        instant messaging client for smartphones that operates ' +
          'under a subscription business model. It uses the Internet to
        send text messages, images, video, user location and ' +
          'audio media messages to other users using standard cellular
        mobile numbers. As of February 2016, WhatsApp had a user ' +
          'base of up to one billion, [10] making it the most globally
        popular messaging application. WhatsApp Inc., based in ' +
          'Mountain View, California, was acquired by Facebook Inc. on
        February 19, 2014, for approximately US$19.3 billion.'} />
    </TabItemsDirective>
  </TabComponent>
);
}

const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

## INDEX.TSX

```

import * as React from "react";
import { useRef } from 'react';

```

```

import * as ReactDOM from 'react-dom';
import { TabComponent, TabItemDirective, TabItemsDirective } from
 '@syncfusion/ej2-react-navigations';
import { SelectEventArgs } from '@syncfusion/ej2-navigations';
const App = () => {
  // eslint-disable-next-line @typescript-eslint/no-unused-vars
  const tabInstance = useRef<TabComponent>(null);
  // define the array of data
  let headertext: any;
  headertext = [{ text: "Twitter" }, { text: "Facebook" }, { text: "WhatsApp"
}];
  const select = (e: SelectEventArgs) => {
    if (e.isSwiped) {
      e.cancel = true;
    }
  }
  return (
    <TabComponent heightAdjustMode='Auto' id='tabelement' ref={tabInstance}
    selecting={select}>
      <TabItemsDirective>
        <TabItemDirective header={headertext[0]}
          content={
            'Twitter is an online social networking service that
enables users to send and read short 140-character ' +
            'messages called "tweets". Registered users can read and post
tweets, but those who are unregistered can only read ' +
            'them. Users access Twitter through the website interface, SMS or
mobile device app Twitter Inc. is based in San ' +
            'Francisco and has more than 25 offices around the world. Twitter
was created in March 2006 by Jack Dorsey, ' +
            'Evan Williams, Biz Stone, and Noah Glass and launched in July
2006. The service rapidly gained worldwide popularity, ' +
            'with more than 100 million users posting 340 million tweets a
day in 2012. The service also handled 1.6 billion ' +
            'search queries per day.' } />
          <TabItemDirective header={headertext[1]}
            content={
              'Facebook is an online social networking service
headquartered in Menlo Park, California. Its website was ' +
              'launched on February 4, 2004, by Mark Zuckerberg with his
Harvard College roommates and fellow students Eduardo ' +
              'Saverin, Andrew McCollum, Dustin Moskovitz and Chris Hughes. The
founders had initially limited the website\ ' +
              'membership to Harvard students, but later expanded it to
colleges in the Boston area, the Ivy League, and Stanford ' +
              'University. It gradually added support for students at various
other universities and later to high-school students.' } />
            <TabItemDirective header={headertext[2]}
              content={
                'WhatsApp Messenger is a proprietary cross-platform
instant messaging client for smartphones that operates ' +
                'under a subscription business model. It uses the Internet to
send text messages, images, video, user location and ' +
                'audio media messages to other users using standard cellular
mobile numbers. As of February 2016, WhatsApp had a user ' +
                'base of up to one billion, [10] making it the most globally
popular messaging application. WhatsApp Inc., based in ' +
                'Mountain View, California, was acquired by Facebook Inc. on
February 19, 2014, for approximately US$19.3 billion.' } />
              </TabItemDirective>

```



```

    </TabComponent>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

## INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Tab</title>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta name="description" content="Essential JS 2 for React Components" />
  <meta name="author" content="Syncfusion" />
  <link href="index.css" rel="stylesheet" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/material.css"
rel="stylesheet" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
  <script src="systemjs.config.js"></script>
</head>
<body>
  <div id='container'>
    <div id='element'>
      <div id='loader'>Loading</div>
    </div>
  </div>
</body>
</html>

```

### Customize selected tab styles in React Tab component

You can customize the Tab style by overriding its header and active tab CSS classes. Define HTML string for adding animation and customizing the Tab header and pass it to [text](#) property. Now you can override the style using custom CSS classes added to the Tab elements.

You can add the custom class into Tab component using [cssClass](#) property which is used to customize the Tab component.

## INDEX.JSX

```

import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { TabComponent, TabItemDirective, TabItemsDirective } from
'@syncfusion/ej2-react-navigations';
const App = () => {
  let headertext = [
    { 'text': '<div><div class="e-image e-andrew"></div><div class="e-
title fade-in">Andrew</div></div>' },
    { 'text': '<div><div class="e-image e-margaret"></div><div class="e-
title fade-in">Margaret</div></div>' },
    { 'text': '<div><div class="e-image e-janet"></div><div class="e-
title fade-in">Janet</div></div>' }

```

```

];
return (<TabComponent heightAdjustMode='Auto' id='tabelement'>
  <TabItemsDirective>
    <TabItemDirective header={headertext[0]} content={'Andrew received
his BTS commercial in 1974 and a Ph.D. in international marketing from the
University of Dallas in 1981.He is fluent in French and Italian and reads
German.He joined the company as a sales representative, was promoted to sales
manager in January 1992 and to vice president of sales in March 1993.Andrew
is a member of the Sales Management Roundtable, the Seattle Chamber of
Commerce, and the Pacific Rim Importers Association.'}/>
    <TabItemDirective header={headertext[1]} content={'Margaret holds a
BA in English literature from Concordia College (1958) and an MA from the
American Institute of Culinary Arts (1966).She was assigned to the London
office temporarily from July through November 1992.'}/>
    <TabItemDirective header={headertext[2]} content={'Janet has a BS
degree in chemistry from Boston College (1984).She has also completed a
certificate program in food retailing management.Janet was hired as a sales
associate in 1991 and promoted to sales representative in February 1992.'}/>
  </TabItemsDirective>
</TabComponent>);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

## INDEX.TSX

```

import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { TabComponent, TabItemDirective, TabItemsDirective } from
'@syncfusion/ej2-react-navigations';
const App = () => {
  let headertext: any = [
    { 'text': '<div><div id="template-wrap" class="e-image e-
andrew"></div><div class="e-title fade-in">Andrew</div></div>' },
    { 'text': '<div><div id="template-wrap" class="e-image e-
margaret"></div><div class="e-title fade-in">Margaret</div></div>' },
    { 'text': '<div><div id="template-wrap" class="e-image e-
janet"></div><div class="e-title fade-in">Janet</div></div>' }
  ];
  return (
    <TabComponent heightAdjustMode='Auto' id='tabelement'>
      <TabItemsDirective>
        <TabItemDirective header={headertext[0]}
          content={'Andrew received his BTS commercial in 1974 and a Ph.D. in
international marketing from the University of Dallas in 1981.He is fluent in
French and Italian and reads German.He joined the company as a sales
representative, was promoted to sales manager in January 1992 and to vice
president of sales in March 1993.Andrew is a member of the Sales Management
Roundtable, the Seattle Chamber of Commerce, and the Pacific Rim Importers
Association.'} />
        <TabItemDirective header={headertext[1]}
          content={'Margaret holds a BA in English literature from Concordia
College (1958) and an MA from the American Institute of Culinary Arts
(1966).She was assigned to the London office temporarily from July through
November 1992.'} />
        <TabItemDirective header={headertext[2]}

```

```

        content={'Janet has a BS degree in chemistry from Boston College
(1984).She has also completed a certificate program in food retailing
management.Janet was hired as a sales associate in 1991 and promoted to sales
representative in February 1992.'} />
      </TabItemsDirective>
    </TabComponent>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

## INDEX.CSS

```

#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
#template-wrap {
  display: flex;
}
.e-content .e-item {
  font-size: 12px;
  margin: 10px;
  text-align: justify;
}
.container {
  min-width: 350px;
  max-width: 500px;
  margin: 0 auto;
}
.e-image {
  background-size: 33px;
  width: 33px;
  height: 33px;
  margin: 0 auto;
}
.e-image.e-andrew { /* csslint allow: adjoining-classes */
  background-image:
url('https://ej2.syncfusion.com/demos/src/images/employees/3.png');
}
.e-image.e-margaret { /* csslint allow: adjoining-classes */
  background-image:
url('https://ej2.syncfusion.com/demos/src/images/employees/6.png');
}
.e-image.e-janet { /* csslint allow: adjoining-classes */
  background-image:
url('https://ej2.syncfusion.com/demos/src/images/employees/7.png');
}
.e-tab .e-toolbar-item .e-title {
  margin-top: 8px;
}
.e-toolbar .e-toolbar-items .e-toolbar-item:not(.e-separator),

```

```

.e-toolbar .e-toolbar-items .e-toolbar-item .e-tab-wrap {
  width: 125px;
  height: 50px;
}
.e-tab .e-tab-header .e-toolbar-item.e-active .e-tab-wrap { /* csslint allow:
adjoining-classes */
  background-color: #08c;
}
.e-tab .e-tab-header {
  background-color: #e6e6e6;
}
.e-tab .e-tab-header .e-toolbar-item:not(.e-active) .e-tab-wrap:hover {
  background-color: #f2f2f2;
  color: #000;
}
.e-tab .e-toolbar .e-toolbar-items .e-toolbar-item .e-text-wrap {
  height: 50px;
}
.e-tab .e-toolbar-item.e-active .e-title { /* csslint allow: adjoining-
classes */
  display: block;
  color: white;
}
.e-tab .e-toolbar-item .e-text-wrap,
.e-tab .e-toolbar-item .e-tab-text {
  width: inherit;
  text-align: center;
}
.e-tab .e-toolbar-item.e-active .e-title.fade-in {
  opacity: 1;
  animation-name: fadeInOpacity;
  animation-iteration-count: 1;
  animation-timing-function: ease-in;
  animation-duration: 0.5s;
}
@keyframes fadeInOpacity {
  0% {
    opacity: 0;
  }
  100% {
    opacity: 1;
  }
}

```

### Customize tab scroll step in React Tab component

Tab supports to customize the scrolling distance when you click the left and right side navigation icons. we can customize **ScrollStep** property for scrolling distance. Refer to the following code example.

By using Tab scrollStep property, pass a required value to customize tab scrollStep.

#### INDEX.JSX

```

import { TabComponent, TabItemDirective, TabItemsDirective } from
'@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';

```

```

const ReactApp = () => {
    const htmlContent = () => {
        return <div>
            HyperText Markup Language, commonly referred to as HTML, is the
            standard markup language used to create web pages. Along with CSS, and
            JavaScript, HTML is a cornerstone technology, used by most websites to create
            visually engaging web pages, user interfaces for web applications, and user
            interfaces for many mobile applications.[1] Web browsers can read HTML files
            and render them into visible or audible web pages. HTML describes the
            structure of a website semantically along with cues for presentation, making
            it a markup language, rather than a programming language.
        </div>;
    }
    const csharpContent = () => {
        return <div>
            C# is intended to be a simple, modern, general-purpose, object-oriented
            programming language. Its development team is led by Anders Hejlsberg. The
            most recent version is C# 5.0, which was released on August 15, 2012.
        </div>;
    }
    const javaContent = () => {
        return <div>
            Java is a set of computer software and specifications developed by Sun
            Microsystems, later acquired by Oracle Corporation, that provides a system
            for developing application software and deploying it in a cross-mobile phones
            to platform computing environment. Java is used in a wide variety of
            computing platforms from embedded devices and enterprise servers and
            supercomputers. While less common, Java applets run in secure, sandboxed
            environments to provide many features of native applications and can be
            embedded in HTML pages.
        </div>;
    }
    const vbNetContent = () => {
        return <div>
            The command-line compiler, VBC.EXE, is installed as part of the
            freeware .NET Framework SDK. Mono also includes a command-line VB.NET
            compiler. The most recent version is VB 2012, which was released on August
            15, 2012.
        </div>;
    }
    const xamarinContent = () => {
        return <div>
            Xamarin is a San Francisco, California based software company created
            in May 2011[3] by the engineers that created Mono,[4] Mono for Android and
            MonoTouch that are cross-platform implementations of the Common Language
            Infrastructure (CLI) and Common Language Specifications (often called
            Microsoft .NET). With a C#-shared codebase, developers can use Xamarin tools
            to write native Android, iOS, and Windows apps with native user interfaces
            and share code across multiple platforms.[5] Xamarin has over 1 million
            developers in more than 120 countries around the World as of May 2015.
        </div>;
    }
    const aspNetcontent = () => {
        return <div>
            ASP.NET is an open-source server-side web application framework
            designed for web development to produce dynamic web pages. It was developed
            by Microsoft to allow programmers to build dynamic web sites, web

```

applications and web services. It was first released **in** January **2002** with version **1.0** of the .NET Framework, and **is** the successor to Microsoft Active Server Pages (ASP) technology. ASP.NET **is** built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code **using any** supported .NET language. The ASP.NET SOAP extension framework allows ASP.NET components to process SOAP messages.

```

    </div>;
  }
  const mvcContent = () => {
    return <div>

```

The ASP.NET MVC **is** a web application framework developed by Microsoft, which implements the model-view-controller (MVC) pattern. It **is** open-source software, apart **from** the ASP.NET Web Forms component which **is** proprietary. In the later versions of ASP.NET, ASP.NET MVC, ASP.NET Web API, and ASP.NET Web Pages (a platform **using only** Razor pages) will merge **into** a unified MVC **6**. The project **is** called ASP.NET vNext.

```

    </div>;
  }
  const jsContent = () => {
    return <div>

```

JavaScript (JS) **is** an interpreted computer programming language. It was originally implemented **as** part of web browsers so that client-side scripts could interact with the user, control the browser, communicate asynchronously, and alter the document content that was displayed.[5] More recently, however, it has become common **in** both game development and the creation of desktop applications.

```

    </div>;
  }
  return (<div>
    <TabComponent height="250px" scrollStep={50}>
      <TabItemsDirective>
        <TabItemDirective header={{ text: "HTML" }} content={htmlContent}/>
        <TabItemDirective header={{ text: "C Sharp(C#)" }}
content={csharpContent}/>
        <TabItemDirective header={{ text: "Java" }} content={javaContent}/>
        <TabItemDirective header={{ text: "VB.Net" }}
content={vbNetContent}/>
        <TabItemDirective header={{ text: "Xamarin" }}
content={xamarinContent}/>
        <TabItemDirective header={{ text: "ASP.NET" }}
content={aspNetcontent}/>
        <TabItemDirective header={{ text: "ASP.NET MVC" }}
content={mvcContent}/>
        <TabItemDirective header={{ text: "JavaScript" }}
content={jsContent}/>
      </TabItemsDirective>
    </TabComponent>
  </div>);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

## INDEX.TSX

```

import { TabComponent, TabItemDirective, TabItemsDirective } from
'@syncfusion/ej2-react-navigations';

```

```

import * as React from 'react';
import * as ReactDOM from 'react-dom';
const ReactApp = () => {
  const htmlContent = () => {
    return <div>
      HyperText Markup Language, commonly referred to as HTML, is the
      standard markup language used to create web pages. Along with CSS, and
      JavaScript, HTML is a cornerstone technology, used by most websites to create
      visually engaging web pages, user interfaces for web applications, and user
      interfaces for many mobile applications.[1] Web browsers can read HTML files
      and render them into visible or audible web pages. HTML describes the
      structure of a website semantically along with cues for presentation, making
      it a markup language, rather than a programming language.
    </div>;
  }
  const csharpContent = () => {
    return <div>
      C# is intended to be a simple, modern, general-purpose, object-oriented
      programming language. Its development team is led by Anders Hejlsberg. The
      most recent version is C# 5.0, which was released on August 15, 2012.
    </div>;
  }
  const javaContent = () => {
    return <div>
      Java is a set of computer software and specifications developed by Sun
      Microsystems, later acquired by Oracle Corporation, that provides a system
      for developing application software and deploying it in a cross-mobile phones
      to platform computing environment. Java is used in a wide variety of
      computing platforms from embedded devices and enterprise servers and
      supercomputers. While less common, Java applets run in secure, sandboxed
      environments to provide many features of native applications and can be
      embedded in HTML pages.
    </div>;
  }
  const vbNetContent = () => {
    return <div>
      The command-line compiler, VBC.EXE, is installed as part of the
      freeware .NET Framework SDK. Mono also includes a command-line VB.NET
      compiler. The most recent version is VB 2012, which was released on August
      15, 2012.
    </div>;
  }
  const xamarinContent = () => {
    return <div>
      Xamarin is a San Francisco, California based software company created
      in May 2011[3] by the engineers that created Mono,[4] Mono for Android and
      MonoTouch that are cross-platform implementations of the Common Language
      Infrastructure (CLI) and Common Language Specifications (often called
      Microsoft .NET). With a C#-shared codebase, developers can use Xamarin tools
      to write native Android, iOS, and Windows apps with native user interfaces
      and share code across multiple platforms.[5] Xamarin has over 1 million
      developers in more than 120 countries around the World as of May 2015.
    </div>;
  }
  const aspNetContent = () => {
    return <div>

```

ASP.NET **is** an open-source server-side web application framework designed **for** web development to produce **dynamic** web pages. It was developed by Microsoft to allow programmers to build **dynamic** web sites, web applications and web services. It was first released **in** January **2002** with version **1.0** of the .NET Framework, and **is** the successor to Microsoft Active Server Pages (ASP) technology. ASP.NET **is** built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code **using any** supported .NET language. The ASP.NET SOAP extension framework allows ASP.NET components to process SOAP messages.

```

    </div>;
  }
  const mvcContent = () => {
    return <div>
      The ASP.NET MVC is a web application framework developed by Microsoft,
      which implements the model-view-controller (MVC) pattern. It is open-source
      software, apart from the ASP.NET Web Forms component which is proprietary. In
      the later versions of ASP.NET, ASP.NET MVC, ASP.NET Web API, and ASP.NET Web
      Pages (a platform using only Razor pages) will merge into a unified MVC 6.
      The project is called ASP.NET vNext.
    </div>;
  }
  const jsContent = () => {
    return <div>
      JavaScript (JS) is an interpreted computer programming language. It was
      originally implemented as part of web browsers so that client-side scripts
      could interact with the user, control the browser, communicate
      asynchronously, and alter the document content that was displayed.[5] More
      recently, however, it has become common in both game development and the
      creation of desktop applications.
    </div>;
  }
  return (<div>
    <TabComponent height="250px" scrollStep={50}>
      <TabItemsDirective>
        <TabItemDirective header={{ text: "HTML" }} content={htmlContent} />
        <TabItemDirective header={{ text: "C Sharp (C#)" }}
content={csharpContent} />
        <TabItemDirective header={{ text: "Java" }} content={javaContent} />
        <TabItemDirective header={{ text: "VB.Net" }} content={vbNetContent}
/>
        <TabItemDirective header={{ text: "Xamarin" }}
content={xamarinContent} />
        <TabItemDirective header={{ text: "ASP.NET" }}
content={aspNetContent} />
        <TabItemDirective header={{ text: "ASP.NET MVC" }}
content={mvcContent} />
        <TabItemDirective header={{ text: "JavaScript" }} content={jsContent}
/>
      </TabItemsDirective>
    </TabComponent>
  </div>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```



**INDEX.HTML**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Tab</title>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta name="description" content="Essential JS 2 for React Components" />
  <meta name="author" content="Syncfusion" />
  <link href="index.css" rel="stylesheet" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/material.css"
rel="stylesheet" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
  <script src="systemjs.config.js"></script>
</head>
<body>
  <div id='container'>
    <div id='element'>
      <div id='loader'>Loading</div>
    </div>
  </div>
</body>
</html>
```

**Create wizard using tab in React Tab component**

Tab items can be disabled dynamically by passing the index and boolean value with the [enableTab](#) method and also passing index or HTML element to select an item from the tab using [select](#) method.

Create the following contents for each tab in the wizard.

**1. Search tab:**

Created with [DropDownList](#) to select the source, destination and type of ticket. A [DatePicker](#) for choosing the date of journey.

**2. Train tab:**

Based on the selected start and end point, populated Grid with random list of available seats and train list. Initially define the columns and row selected event for validating, after the source and destination chosen update the [dataSource](#) for the Grid.

**3. Passenger tab:**

A table with Textbox, Numeric, DropDownList for adding passenger name, age, gender and preferred berth/seat. Add validation on entering passenger details to proceed.

**4. Payment tab:**

Calculate the ticket cost based on location, passenger count and ticket type. Generate data for Grid with passenger details, train number and ticket cost summary.

You can go back on each tab using buttons available in it and tabs are [disabled](#) to navigate through tab header click actions. Once you end the wizard all the data get cleared and wizard goes back to starting tab.

In the below demo, designed for simple train reservation module that enable/disable tab items based on sequential validation of each Tab content.

### INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from 'react-dom';
import { useRef } from 'react';
import { DialogComponent } from '@syncfusion/ej2-react-popups';
import { DatePickerComponent } from '@syncfusion/ej2-react-calendars';
import { NumericTextBoxComponent } from '@syncfusion/ej2-react-inputs';
import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { TabComponent, TabItemDirective, TabItemsDirective } from
 '@syncfusion/ej2-react-navigations';
import { GridComponent, ColumnsDirective, ColumnDirective } from
 '@syncfusion/ej2-react-grids';
const App = () => {
  const alertDlg = useRef(null);
  const tab = useRef(null);
  const ticketDetailGrid = useRef(null);
  const pass_gender3 = useRef(null);
  const pass_gender2 = useRef(null);
  const pass_gender1 = useRef(null);
  const pass_berth1 = useRef(null);
  const pass_berth2 = useRef(null);
  const pass_berth3 = useRef(null);
  const pass_age1 = useRef(null);
  const availTrainGrid = useRef(null);
  const ticketType = useRef(null);
  const journeyDate = useRef(null);
  const endPoint = useRef(null);
  const startPoint = useRef(null);
  const passNameFirst = useRef(null);
  const passNameSecond = useRef(null);
  const passNameThird = useRef(null);
  const err_1 = useRef(null);
  const err_2 = useRef(null);
  const err_3 = useRef(null);
  const amount = useRef(null);
  let today = new Date();
  let selectedTrain;
  let dlgTarget = document.querySelector(
    ".sb-content-tab.e-tab .e-content.sb-sample-content-area"
  );
  let dateMin = new Date(today.getTime());
  let dateMax = new Date(
    today.getTime() + 60 * 24 * 60 * 60 * 1000
  );
  let fields = { id: "id", text: "text", value: "text" };
  let autoCompleteFields = { text: "name", value: "name" };
```

```

const headerText = [
  { text: "New Booking" },
  { text: "Train List" },
  { text: "Add Passenger" },
  { text: "Make Payment" }
];

const quotas = [
  { id: "1", text: "Business Class" },
  { id: "2", text: "Economy Class" },
  { id: "4", text: "Common Class" }
];

const gender = [{ id: "1", text: "Male" }, { id: "2", text: "Female" }];
const berths = [
  { id: "1", text: "Upper" },
  { id: "2", text: "Lower" },
  { id: "3", text: "Middle" },
  { id: "4", text: "Window" },
  { id: "5", text: "Aisle" }
];

const cities = [
  { name: "Chicago", fare: 300 },
  { name: "San Francisco", fare: 125 },
  { name: "Los Angeles", fare: 175 },
  { name: "Seattle", fare: 250 },
  { name: "Florida", fare: 150 }
];

const dlgButtons = [
  {
    buttonModel: { content: "OK", isPrimary: true },
    // tslint:disable-next-line: no-empty
    click: () => {
      alertDlg.current.hide();
      tab.current.enableTab(0, true);
      tab.current.enableTab(1, false);
      tab.current.enableTab(2, false);
      tab.current.enableTab(3, false);
      tab.current.select(0);
    }
  }
];

const dlgCreated = () => {
  alertDlg.current.hide();
}

const focusIn = () => {
  journeyDate.current.show()
}

const tabSelected = (e) => {
  if (e.isSwiped) {
    e.cancel = true;
  }
}

const trainSelected = (args) => {
  selectedTrain = args.data;
}

const btnClicked = (e) => {
  switch (e.target.id) {
    case "searchNext":

```

```

    /* Validate the Source, Destination, Date and Class chosen and
    proceed only if all the fields are selected */
    if (
        startPoint.current.value != null &&
        endPoint.current.value != null &&
        ticketType.current.value != null &&
        journeyDate.current.value != null
    ) {
        if (
            startPoint.current.value &&
            startPoint.current.value === endPoint.current.value
        ) {
            err_1.current.innerText =
                "* Arrival point can't be same as Departure";
        } else {
            tab.current.enableTab(0, false);
            tab.current.enableTab(1, true);
            filterTrains(e);
            tab.current.select(1);
            err_1.current.innerText = "";
            if (err_2.current) {
                err_2.current.innerText = "";
            }
        }
    } else {
        err_1.current.innerText =
            "* Please fill all the details before proceeding";
    }
    break;
case "bookTickets":
    /* Based on the selected station generate Grid content to display
    trains available */
    if (
        availTrainGrid.current.getSelectedRecords() === undefined ||
        availTrainGrid.current.getSelectedRecords().length === 0
    ) {
        err_2.current.innerText =
            "* Select your convenient train";
    } else {
        tab.current.enableTab(2, true);
        tab.current.select(2);
        tab.current.enableTab(1, false);
        err_2.current.innerText = "";
    }
    break;
case "confirmTickets":
    /* Get the Passenger details and validate the fields must not be left
    empty */
    const name = passNameFirst.current;
    const age = pass_age1.current.value;
    const gender = pass_gender1.current.value;
    if (name.value === "" || age === "" || gender === "") {
        err_3.current.innerText =
            "* Please enter passenger details";
    } else {
        tab.current.enableTab(3, true);
        tab.current.select(3);
    }

```

```

        tab.current.enableTab(2, false);
        err_3.current.innerText = "";
        finalizeDetails(e);
    }
    break;
case "makePayment":
    alertDlg.current.show();
    break;
case "goToSearch":
    /* Go back to change class, date or boarding places */
    selectedTrain = [];
    tab.current.enableTab(0, true);
    tab.current.select(0);
    tab.current.enableTab(1, false);
    break;
case "goBackToBook":
    /* Change the preferred train chosen already */
    tab.current.enableTab(1, true);
    tab.current.select(1);
    tab.current.enableTab(2, false);
    break;
case "goBackDetails":
    /* Update passenger detail before confirming the payment */
    tab.current.enableTab(2, true);
    tab.current.select(2);
    tab.current.enableTab(3, false);
    break;
}
}
const filterTrains = (args)=> {
    /* Generating trains based on source and destination chosen */
    const result = [];
    const fromCity = startPoint.current.value;
    const toCity = endPoint.current.value;
    const count = Math.floor(Math.random() * 3 + 2);
    for (let i = 0; i < count; i++) {
        // tslint:disable-next-line: ban-types
        const details = {};
        details.TrainNo = Math.floor(Math.random() * 20 + 19000);
        details.Name = "Train " + i;
        details.Departure = fromCity;
        details.Arrival = toCity;
        details.Availability = Math.floor(Math.random() * 20 + 20);
        result.push(details);
    }
    setTimeout(() => {
        availTrainGrid.current.dataSource = result;
        availTrainGrid.current.dataBind();
    }, 100);
}
const finalizeDetails = (args) => {
    /* Get the passenger details and update table with name and other details
    for confirmation */
    const reserved = [];
    let passCount = 0;
    const name1 = passNameFirst.current.value;
    const name2 = passNameSecond.current.value;

```

```

const name3 = passNameThird.current.value;
for (let i = 1; i <= 3; i++) {
  if (name1.value !== "") {
    // tslint:disable-next-line: ban-types
    const details = {};
    const gender = i === 1
      ? pass_gender1.current.value
      : i === 2
        ? pass_gender2.current.value
        : pass_gender3.current.value;
    const berth = i === 1
      ? pass_berth1.current.value
      : i === 2
        ? pass_berth2.current.value
        : pass_berth3.current.value;
    details.TrainNo = selectedTrain.TrainNo.toString();
    details.PassName =
      i === 1 ? name1.value : i === 2 ? name2.value : name3.value;
    details.Gender = gender === "" ? "Male" : gender;
    details.Berth = berth === "" ? berth : "Any";
    if (details.PassName !== "") {
      reserved.push(details);
    }
    passCount++;
  }
  let calcFare = 0;
  // tslint:disable-next-line: forin
  for (const j in cities) {
    if (startPoint.current.value === cities[j].name) {
      calcFare = calcFare + cities[j].fare;
    }
    if (endPoint.current.value === cities[j].name) {
      calcFare = calcFare + cities[j].fare;
    }
  }
  const displayAmt = amount.current;
  if (ticketType.current.value === "Economy Class" && displayAmt) {
    displayAmt.innerText =
      "Total payable amount: $" + passCount * (300 + calcFare);
  } else if (ticketType.current.value === "Business Class" && displayAmt) {
    displayAmt.innerText =
      "Total payable amount: $" + passCount * (500 + calcFare);
  } else if (ticketType.current.value === "Common Class" && displayAmt) {
    displayAmt.innerText =
      "Total payable amount: $" + passCount * (150 + calcFare);
  }
}
setTimeout(() => {
  (ticketDetailGrid || []).current.dataSource = reserved;
  ticketDetailGrid.current.dataBind();
}, 100);
}
const content0 = () => {
  return (
    <div id="booking">
      <div className="wizard-title">Plan your journey</div>

```

```

<div className="responsive-align">
  <div className="row">
    <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6 search-item">
      <DropDownListComponent
        ref={
          startPoint
        }
        width="100%"
        dataSource={cities}
        fields={autoCompleteFields}
        placeholder="From"
        floatLabelType="Auto"
      />
    </div>
    <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6 search-item">
      <DropDownListComponent
        ref={
          endPoint
        }
        width="100%"
        dataSource={cities}
        fields={autoCompleteFields}
        placeholder="To"
        floatLabelType="Auto"
      />
    </div>
  </div>
  <div className="row">
    <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6 search-item">
      <DatePickerComponent
        ref={journeyDate}
        width="100%"
        placeholder="Journey Date"
        floatLabelType="Auto"
        min={dateMin}
        max={dateMax}
        focus={focusIn}
      />
    </div>
    <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6 search-item">
      <DropDownListComponent
        ref={ticketType}
        dataSource={quotas}
        placeholder="Ticket type"
        floatLabelType="Auto"
        fields={fields}
      />
    </div>
  </div>
  <div className="btn-container">
    <button
      id="searchNext"
      className="e-btn"
      onClick={btnClicked}
    >
      Search Train
    </button>
  </div>

```

```

        </div>
        <span ref={err_1} />
    </div>
</div>
);
}
const content1 = () => {
    return (
        <div id="selectTrain">
            <div className="wizard-title">Select the train from the list </div>
            <GridComponent
                ref={availTrainGrid}
                width="100%"
                rowSelected={trainSelected}
            >
                <ColumnsDirective>
                    <ColumnDirective
                        field="TrainNo"
                        headerText="Train No"
                        width={120}
                        type="number"
                    />
                    <ColumnDirective field="Name" headerText="Name" width={140} />
                    <ColumnDirective
                        field="Departure"
                        headerText="Departure"
                        width={120}
                    />
                    <ColumnDirective field="Arrival" headerText="Arrival" width={140}
                />
                    <ColumnDirective
                        field="Availability"
                        headerText="Availability"
                        width={140}
                        type="number"
                    />
                </ColumnsDirective>
            </GridComponent>
            <br />
            <div className="btn-container">
                <button
                    id="goToSearch"
                    className="e-btn"
                    onClick={btnClicked}
                >
                    Back
                </button>
                <button
                    id="bookTickets"
                    className="e-btn"
                    onClick={btnClicked}
                >
                    Continue
                </button>
            </div>
            <span ref={err_2} />
        </div>
    );
}

```



```

    );
}
const content2 = () => {
    return (
        <div id="details">
            <div className="details-page wizard-title">
                Enter the passenger details
            </div>
            <div id="PassengersList">
                <table id="passenger-table">
                    <colgroup>
                        <col />
                        <col />
                        <col />
                        <col />
                        <col />
                        <col />
                    </colgroup>
                    <thead>
                        <tr>
                            <th className="name-header">Name</th>
                            <th className="age-header">Age</th>
                            <th className="gender-header">Gender</th>
                            <th className="type-header">Berth Preference</th>
                        </tr>
                    </thead>
                    <tbody>
                        <tr>
                            <td>
                                <input
                                    className="e-input"
                                    ref={passNameFirst}
                                    type="text"
                                    placeholder="Passenger Name"
                                />
                            </td>
                            <td>
                                <NumericTextBoxComponent
                                    ref={
                                        pass_age1
                                    }
                                    showSpinButton={false}
                                    min={1}
                                    max={100}
                                    value={18}
                                    format="n0"
                                />
                            </td>
                            <td>
                                <DropDownListComponent
                                    ref={
                                        pass_gender1
                                    }
                                    dataSource={gender}
                                    text="Male"
                                    fields={fields}
                                />
                            </td>
                        </tr>
                    </tbody>
                </table>
            </div>
        </div>
    );
}

```

```

        </td>
        <td>
            <DropDownListComponent
                ref={
                    pass_berth1
                }
                dataSource={berths}
                placeholder="Optional"
                fields={fields}
            />
        </td>
    </tr>
    <tr>
        <td>
            <input
                ref={passNameSecond}
                className="e-input"
                type="text"
                placeholder="Passenger Name"
            />
        </td>
        <td>
            <NumericTextBoxComponent
                showSpinButton={false}
                min={1}
                max={100}
                value={18}
                format="n0"
            />
        </td>
        <td>
            <DropDownListComponent
                ref={
                    pass_gender2
                }
                dataSource={gender}
                text="Male"
                fields={fields}
            />
        </td>
        <td>
            <DropDownListComponent
                ref={
                    pass_berth2
                }
                dataSource={berths}
                placeholder="Optional"
                fields={fields}
            />
        </td>
    </tr>
    <tr>
        <td>
            <input
                ref={passNameThird}
                className="e-input"
                type="text"

```

```

        placeholder="Passenger Name"
    />
</td>
<td>
    <NumericTextBoxComponent
        showSpinButton={false}
        min={1}
        max={100}
        value={18}
        format="n0"
    />
</td>
<td>
    <DropDownListComponent
        ref={
            pass_gender3
        }
        dataSource={gender}
        text="Male"
        fields={fields}
    />
</td>
<td>
    <DropDownListComponent
        ref={
            pass_berth3
        }
        dataSource={berths}
        placeholder="Optional"
        fields={fields}
    />
</td>
</tr>
</tbody>
</table>
</div>
<br />
<div className="btn-container">
    <button
        id="goBackToBook"
        className="e-btn"
        onClick={btnClicked}
    >
        Back
    </button>
    <button
        id="confirmTickets"
        className="e-btn"
        onClick={btnClicked}
    >
        Continue
    </button>
</div>
<span ref={err_3} />
</div>
);
}

```

```

const content3 = () => {
  return (
    <div id="confirm">
      <div className="tab-title1 wizard-title">
        Confirm the details and proceed
      </div>
      <GridComponent
        ref={ticketDetailGrid}
        width="100%"
      >
        <ColumnsDirective>
          <ColumnDirective
            field="TrainNo"
            headerText="Train No"
            width={120}
            type="number"
          />
          <ColumnDirective field="PassName" headerText="Name" width={140}
        />
          <ColumnDirective field="Gender" headerText="Gender" width={120}
        />
          <ColumnDirective field="Berth" headerText="Berth" width={140} />
        </ColumnsDirective>
      </GridComponent>
      <br />
      <div ref={amount} />
      <br />
      <div className="btn-container">
        <button
          id="goBackDetails"
          className="e-btn"
          onClick={btnClicked}
        >
          Back
        </button>
        <button
          id="makePayment"
          className="e-btn"
          onClick={btnClicked}
        >
          Pay
        </button>
      </div>
    </div>
  );
}
return (
  <div>
    <div className="col-lg-12 control-section e-tab-section">
      <div className="e-sample-resize-container">
        <TabComponent
          id="tab-wizard"
          ref={tab}
          heightAdjustMode="None"
          height={390}
          selecting={tabSelected}
        >

```

```

        <TabItemsDirective>
            <TabItemDirective
                header={headerText[0]}
                content={content0}
            />
            <TabItemDirective
                header={headerText[1]}
                content={content1}
                disabled={true}
            />
            <TabItemDirective
                header={headerText[2]}
                content={content2}
                disabled={true}
            />
            <TabItemDirective
                header={headerText[3]}
                content={content3}
                disabled={true}
            />
        </TabItemsDirective>
    </TabComponent>
    <DialogComponent
        ref={alertDlg}
        header="Success"
        width={250}
        isModal={true}
        visible={false}
        showCloseIcon={true}
        content="Your payment successfully processed"
        target={dlgTarget}
        buttons={dlgButtons}
        created={dlgCreated}
    />
</div>
</div>
</div>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

## INDEX.TSX

```

import * as ReactDOM from 'react-dom';
import * as React from "react";
import { useRef } from 'react';
import { DialogComponent } from '@syncfusion/ej2-react-popups';
import { DatePickerComponent } from '@syncfusion/ej2-react-calendars';
import { NumericTextBoxComponent } from '@syncfusion/ej2-react-inputs';
import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { TabComponent, TabItemDirective, TabItemsDirective, SelectEventArgs }
from '@syncfusion/ej2-react-navigations';
import { GridComponent, RowSelectEventArgs, ColumnsDirective, ColumnDirective
} from '@syncfusion/ej2-react-grids';
const App = () => {

```

```

const alertDlg = useRef<DialogComponent>(null);
const tab = useRef<TabComponent>(null);
const ticketDetailGrid = useRef<GridComponent>(null);
const pass_gender3 = useRef<DropDownListComponent>(null);
const pass_gender2 = useRef<DropDownListComponent>(null);
const pass_gender1 = useRef<DropDownListComponent>(null);
const pass_berth1 = useRef<DropDownListComponent>(null);
const pass_berth2 = useRef<DropDownListComponent>(null);
const pass_berth3 = useRef<DropDownListComponent>(null);
const pass_age1 = useRef<NumericTextBoxComponent>(null);
const availTrainGrid = useRef<GridComponent>(null);
const ticketType = useRef<DropDownListComponent>(null);
const journeyDate = useRef<DatePickerComponent>(null);
const endPoint = useRef<DropDownListComponent>(null);
const startPoint = useRef<DropDownListComponent>(null);
const passNameFirst = useRef(null);
const passNameSecond = useRef(null);
const passNameThird = useRef(null);
const err_1 = useRef(null);
const err_2 = useRef(null);
const err_3 = useRef(null);
const amount = useRef(null);
let today: Date = new Date();
let selectedTrain: any;
let dlgTarget: HTMLElement = (document.querySelector(
    ".sb-content-tab.e-tab .e-content.sb-sample-content-area"
) as HTMLElement);
let dateMin: Date = new Date(today.getTime());
let dateMax: Date = new Date(
    today.getTime() + 60 * 24 * 60 * 60 * 1000
);
let fields: Object = { id: "id", text: "text", value: "text" };
let autoCompleteFields: Object = { text: "name", value: "name" };
const headerText: any = [
    { text: "New Booking" },
    { text: "Train List" },
    { text: "Add Passenger" },
    { text: "Make Payment" }
];
const quotas: any = [
    { id: "1", text: "Business Class" },
    { id: "2", text: "Economy Class" },
    { id: "4", text: "Common Class" }
];
const gender: any = [{ id: "1", text: "Male" }, { id: "2", text: "Female"
}];
const berths: any = [
    { id: "1", text: "Upper" },
    { id: "2", text: "Lower" },
    { id: "3", text: "Middle" },
    { id: "4", text: "Window" },
    { id: "5", text: "Aisle" }
];
const cities: any = [
    { name: "Chicago", fare: 300 },
    { name: "San Francisco", fare: 125 },
    { name: "Los Angeles", fare: 175 },

```

```

    { name: "Seattle", fare: 250 },
    { name: "Florida", fare: 150 }
  ];
  const dlgButtons: any = [
    {
      buttonModel: { content: "OK", isPrimary: true },
      // tslint:disable-next-line: no-empty
      click: () => {
        alertDlg.current.hide();
        tab.current.enableTab(0, true);
        tab.current.enableTab(1, false);
        tab.current.enableTab(2, false);
        tab.current.enableTab(3, false);
        tab.current.select(0);
      }
    }
  ];
  const dlgCreated = (): void => {
    alertDlg.current.hide();
  }
  const focusIn = (): void => {
    journeyDate.current.show()
  }
  const tabSelected = (e: SelectEventArgs) => {
    if (e.isSwiped) {
      e.cancel = true;
    }
  }
  const trainSelected = (args: RowSelectEventArgs): void => {
    selectedTrain = args.data;
  }
  const btnClicked = (e: any): void => {
    switch (e.target.id) {
      case "searchNext":
        /* Validate the Source, Destination, Date and Class chosen and
        proceed only if all the fields are selected */
        if (
          startPoint.current.value != null &&
          endPoint.current.value != null &&
          ticketType.current.value != null &&
          journeyDate.current.value != null
        ) {
          if (
            startPoint.current.value &&
            startPoint.current.value === endPoint.current.value
          ) {
            err_1.current.innerText =
              "* Arrival point can't be same as Departure";
          } else {
            tab.current.enableTab(0, false);
            tab.current.enableTab(1, true);
            filterTrains(e);
            tab.current.select(1);
            err_1.current.innerText = "";
            if (err_2.current) {
              err_2.current.innerText = "";
            }
          }
        }
      }
    }
  }

```

```

    }
    } else {
        err_1.current.innerText =
            "* Please fill all the details before proceeding";
    }
    break;
case "bookTickets":
    /* Based on the selected station generate Grid content to display
    trains available */
    if (
        availTrainGrid.current.getSelectedRecords() === undefined ||
        availTrainGrid.current.getSelectedRecords().length === 0
    ) {
        err_2.current.innerText =
            "* Select your convenient train";
    } else {
        tab.current.enableTab(2, true);
        tab.current.select(2);
        tab.current.enableTab(1, false);
        err_2.current.innerText = "";
    }
    break;
case "confirmTickets":
    /* Get the Passenger details and validate the fields must not be left
    empty */
    const name: any = passNameFirst.current;
    const age: any = pass_age1.current.value;
    const gender: any = pass_gender1.current.value;
    if (name.value === "" || age === "" || gender === "") {
        err_3.current.innerText =
            "* Please enter passenger details";
    } else {
        tab.current.enableTab(3, true);
        tab.current.select(3);
        tab.current.enableTab(2, false);
        err_3.current.innerText = "";
        finalizeDetails(e);
    }
    break;
case "makePayment":
    alertDialog.current.show();
    break;
case "goToSearch":
    /* Go back to change class, date or boarding places */
    selectedTrain = [];
    tab.current.enableTab(0, true);
    tab.current.select(0);
    tab.current.enableTab(1, false);
    break;
case "goBackToBook":
    /* Change the preferred train chosen already */
    tab.current.enableTab(1, true);
    tab.current.select(1);
    tab.current.enableTab(2, false);
    break;
case "goBackDetails":
    /* Update passenger detail before confirming the payment */

```



```

        tab.current.enableTab(2, true);
        tab.current.select(2);
        tab.current.enableTab(3, false);
        break;
    }
}

const filterTrains = (args: RowSelectEventArgs): void => {
    /* Generating trains based on source and destination chosen */
    const result: { [key: string]: Object }[] = [];
    const fromCity: string = startPoint.current.value as string;
    const toCity: string = endPoint.current.value as string;
    const count: number = Math.floor(Math.random() * 3 + 2);
    for (let i: number = 0; i < count; i++) {
        // tslint:disable-next-line: ban-types
        const details: { [key: string]: Object } = {};
        details.TrainNo = Math.floor(Math.random() * 20 + 19000);
        details.Name = "Train " + i;
        details.Departure = fromCity;
        details.Arrival = toCity;
        details.Availability = Math.floor(Math.random() * 20 + 20);
        result.push(details);
    }
    setTimeout(() => {
        availTrainGrid.current.dataSource = result;
        availTrainGrid.current.dataBind();
    }, 100);
}

const finalizeDetails = (args: RowSelectEventArgs): void => {
    /* Get the passenger details and update table with name and other details for confirmation */
    const reserved: { [key: string]: Object }[] = [];
    let passCount: any = 0;
    const name1: any = passNameFirst.current.value;
    const name2: any = passNameSecond.current.value;
    const name3: any = passNameThird.current.value;
    for (let i: number = 1; i <= 3; i++) {
        if (name1.value !== "") {
            // tslint:disable-next-line: ban-types
            const details: { [key: string]: Object } = {};
            const gender: string = (i === 1
                ? pass_gender1.current.value
                : i === 2
                ? pass_gender2.current.value
                : pass_gender3.current.value) as string;
            const berth: string = (i === 1
                ? pass_berth1.current.value
                : i === 2
                ? pass_berth2.current.value
                : pass_berth3.current.value) as string;
            details.TrainNo = selectedTrain.TrainNo.toString();
            details.PassName =
                i === 1 ? name1.value : i === 2 ? name2.value : name3.value;
            details.Gender = gender === "" ? "Male" : gender;
            details.Berth = berth === "" ? berth : "Any";
            if (details.PassName !== "") {
                reserved.push(details);
            }
        }
    }
}

```

```

        passCount++;
    }
    let calcFare: any = 0;
    // tslint:disable-next-line: forin
    for (const j in cities) {
        if (startPoint.current.value === cities[j].name) {
            calcFare = calcFare + cities[j].fare;
        }
        if (endPoint.current.value === cities[j].name) {
            calcFare = calcFare + cities[j].fare;
        }
    }
    const displayAmt: any = amount.current;
    if (ticketType.current.value === "Economy Class" && displayAmt) {
        displayAmt.innerText =
            "Total payable amount: $" + passCount * (300 + calcFare);
    } else if (ticketType.current.value === "Business Class" && displayAmt) {
        displayAmt.innerText =
            "Total payable amount: $" + passCount * (500 + calcFare);
    } else if (ticketType.current.value === "Common Class" && displayAmt) {
        displayAmt.innerText =
            "Total payable amount: $" + passCount * (150 + calcFare);
    }
    }
    setTimeout(() => {
        (ticketDetailGrid || []).current.dataSource = reserved;
        ticketDetailGrid.current.dataBind();
    }, 100);
}
const content0 = () => {
    return (
        <div id="booking">
            <div className="wizard-title">Plan your journey</div>
            <div className="responsive-align">
                <div className="row">
                    <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6 search-item">
                        <DropDownListComponent
                            ref={
                                startPoint
                            }
                            width="100%"
                            dataSource={cities}
                            fields={autoCompleteFields}
                            placeholder="From"
                            floatLabelType="Auto"
                        />
                    </div>
                    <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6 search-item">
                        <DropDownListComponent
                            ref={
                                endPoint
                            }
                            width="100%"
                            dataSource={cities}
                            fields={autoCompleteFields}
                            placeholder="To"

```

```

        floatLabelType="Auto"
    />
</div>
</div>
<div className="row">
    <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6 search-item">
        <DatePickerComponent
            ref={journeyDate}
            width="100%"
            placeholder="Journey Date"
            floatLabelType="Auto"
            min={dateMin}
            max={dateMax}
            focus={focusIn}
        />
    </div>
    <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6 search-item">
        <DropDownListComponent
            ref={ticketType}
            dataSource={quotas}
            placeholder="Ticket type"
            floatLabelType="Auto"
            fields={fields}
        />
    </div>
</div>
<div className="btn-container">
    <button
        id="searchNext"
        className="e-btn"
        onClick={btnClicked}
    >
        Search Train
    </button>
</div>
<span ref={err_1} />
</div>
</div>
);
}
const content1 = () => {
    return (
        <div id="selectTrain">
            <div className="wizard-title">Select the train from the list </div>
            <GridComponent
                ref={availTrainGrid}
                width="100%"
                rowSelected={trainSelected}
            >
                <ColumnsDirective>
                    <ColumnDirective
                        field="TrainNo"
                        headerText="Train No"
                        width={120}
                        type="number"
                    />
                    <ColumnDirective field="Name" headerText="Name" width={140} />
                </ColumnsDirective>
            </GridComponent>
        </div>
    );
};

```

```

        <ColumnDirective
            field="Departure"
            headerText="Departure"
            width={120}
        />
        <ColumnDirective field="Arrival" headerText="Arrival" width={140}
    />

    <ColumnDirective
        field="Availability"
        headerText="Availability"
        width={140}
        type="number"
    />
    </ColumnsDirective>
</GridComponent>
<br />
<div className="btn-container">
    <button
        id="goToSearch"
        className="e-btn"
        onClick={btnClicked}
    >
        Back
    </button>
    <button
        id="bookTickets"
        className="e-btn"
        onClick={btnClicked}
    >
        Continue
    </button>
</div>
<span ref={err_2} />
</div>
);
}
const content2 = () => {
    return (
        <div id="details">
            <div className="details-page wizard-title">
                Enter the passenger details
            </div>
            <div id="PassengersList">
                <table id="passenger-table">
                    <colgroup>
                        <col />
                        <col />
                        <col />
                        <col />
                        <col />
                        <col />
                    </colgroup>
                    <thead>
                        <tr>
                            <th className="name-header">Name</th>
                            <th className="age-header">Age</th>
                            <th className="gender-header">Gender</th>

```

```

        <th className="type-header">Berth Preference</th>
    </tr>
</thead>
<tbody>
    <tr>
        <td>
            <input
                className="e-input"
                ref={passNameFirst}
                type="text"
                placeholder="Passenger Name"
            />
        </td>
        <td>
            <NumericTextBoxComponent
                ref={
                    pass_age1
                }
                showSpinButton={false}
                min={1}
                max={100}
                value={18}
                format="n0"
            />
        </td>
        <td>
            <DropDownListComponent
                ref={
                    pass_gender1
                }
                dataSource={gender}
                text="Male"
                fields={fields}
            />
        </td>
        <td>
            <DropDownListComponent
                ref={
                    pass_berth1
                }
                dataSource={berths}
                placeholder="Optional"
                fields={fields}
            />
        </td>
    </tr>
    <tr>
        <td>
            <input
                ref={passNameSecond}
                className="e-input"
                type="text"
                placeholder="Passenger Name"
            />
        </td>
        <td>
            <NumericTextBoxComponent

```

```

        showSpinButton={false}
        min={1}
        max={100}
        value={18}
        format="n0"
    />
</td>
<td>
    <DropDownListComponent
        ref={
            pass_gender2
        }
        dataSource={gender}
        text="Male"
        fields={fields}
    />
</td>
<td>
    <DropDownListComponent
        ref={
            pass_berth2
        }
        dataSource={berths}
        placeholder="Optional"
        fields={fields}
    />
</td>
</tr>
<tr>
<td>
    <input
        ref={passNameThird}
        className="e-input"
        type="text"
        placeholder="Passenger Name"
    />
</td>
<td>
    <NumericTextBoxComponent
        showSpinButton={false}
        min={1}
        max={100}
        value={18}
        format="n0"
    />
</td>
<td>
    <DropDownListComponent
        ref={
            pass_gender3
        }
        dataSource={gender}
        text="Male"
        fields={fields}
    />
</td>
<td>

```

```

        <DropDownListComponent
            ref={
                pass_berth3
            }
            dataSource={berths}
            placeholder="Optional"
            fields={fields}
        />
    </td>
</tr>
</tbody>
</table>
</div>
<br />
<div className="btn-container">
    <button
        id="goBackToBook"
        className="e-btn"
        onClick={btnClicked}
    >
        Back
    </button>
    <button
        id="confirmTickets"
        className="e-btn"
        onClick={btnClicked}
    >
        Continue
    </button>
</div>
<span ref={err_3} />
</div>
);
}
const content3 = () => {
    return (
        <div id="confirm">
            <div className="tab-title1 wizard-title">
                Confirm the details and proceed
            </div>
            <GridComponent
                ref={ticketDetailGrid}
                width="100%"
            >
                <ColumnsDirective>
                    <ColumnDirective
                        field="TrainNo"
                        headerText="Train No"
                        width={120}
                        type="number"
                    />
                    <ColumnDirective field="PassName" headerText="Name" width={140}
                />
                    <ColumnDirective field="Gender" headerText="Gender" width={120}
                />
                    <ColumnDirective field="Berth" headerText="Berth" width={140} />
                </ColumnsDirective>
            </GridComponent>
        </div>
    );
};

```

```

</GridComponent>
<br />
<div ref={amount} />
<br />
<div className="btn-container">
  <button
    id="goBackDetails"
    className="e-btn"
    onClick={btnClicked}
  >
    Back
  </button>
  <button
    id="makePayment"
    className="e-btn"
    onClick={btnClicked}
  >
    Pay
  </button>
</div>
</div>
);
}
return (
  <div>
    <div className="col-lg-12 control-section e-tab-section">
      <div className="e-sample-resize-container">
        <TabComponent
          id="tab-wizard"
          ref={tab}
          heightAdjustMode="None"
          height={390}
          selecting={tabSelected}
        >
          <TabItemsDirective>
            <TabItemDirective
              header={headerText[0]}
              content={content0}
            />
            <TabItemDirective
              header={headerText[1]}
              content={content1}
              disabled={true}
            />
            <TabItemDirective
              header={headerText[2]}
              content={content2}
              disabled={true}
            />
            <TabItemDirective
              header={headerText[3]}
              content={content3}
              disabled={true}
            />
          </TabItemsDirective>
        </TabComponent>
      </div>
    </div>
  )
);

```



```

        ref={alertDlg}
        header="Success"
        width={250}
        isModal={true}
        visible={false}
        showCloseIcon={true}
        content="Your payment successfully processed"
        target={dlgTarget}
        buttons={dlgButtons}
        created={dlgCreated}
      />
    </div>
  </div>
</div>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

## INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Tab</title>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta name="description" content="Essential JS 2 for React Components" />
  <meta name="author" content="Syncfusion" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/material.css"
rel="stylesheet" />
  <link
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
rel="stylesheet">
  <link href="index.css" rel="stylesheet" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
  <script src="systemjs.config.js"></script>
</head>
<body>
  <div id='container'>
    <div id='element'>
      <div id='loader'>Loading</div>
    </div>
  </div>
</body>
</html>

```

## INDEX.CSS

```

#container {
  max-width: 700px;
  min-width: 520px;
}
#loader {

```

```

    color: #008cff;
    height: 40px;
    left: 45%;
    position: absolute;
    top: 45%;
    width: 30%;
}
.tab-control-section .e-content .e-item {
    font-size: 12px;
    padding: 10px;
    text-align: justify;
}
.tab-control-section .e-headertext {
    font-weight: 500;
}
.e-bigger .tab-control-section .e-tab .e-list-item {
    border-bottom: 1px solid gainsboro;
}
.tab-control-section .e-tab .e-list-item {
    height: 63px;
    border-bottom: 1px solid gainsboro;
}
.tab-control-section .e-tab .e-list-item:last-child {
    border-bottom: 1px solid gainsboro;
}
.e-bigger .tab-control-section .e-tab .e-list-item {
    height: 63px;
}
.tab-control-section .template-container {
    height: 100%;
    padding-top: 7px;
}
.tab-control-section .left {
    display: inline-block;
    vertical-align: top;
}
.tab-control-section .empImg {
    height: 48px;
    width: 48px;
}
.tab-control-section .name, .role {
    line-height: normal;
}
.tab-control-section .name {
    font-weight: 500;
}
.tab-control-section .role {
    color: grey;
    font-weight: 100;
}
.tab-control-section .info-div {
    padding: 7px 0 0 15px;
}
.tab-control-section .content-title {
    height: 50px;
    display: table;
    margin: 0 auto;
}

```

```

}
.tab-control-section .cnt-text {
    vertical-align: middle;
    display: table-cell;
    font-size: 18px;
    font-weight: 600;
}
.tab-control-section .e-tab .e-listview .e-text {
    display: block;
    text-align: center;
}
.tab-control-section .property-panel-content {
    padding: 10px;
}
@media only screen and (min-width: 1824px) {
    .tab-control-section .col-lg-8.adaptive { /* csslint allow: adjoining-
classes */
        width: 40%;
    }
    .e-bigger .tab-control-section .col-lg-8.adaptive { /* csslint allow:
adjoining-classes */
        width: 45%;
    }
}
.e-content .e-item {
    font-size: 12px;
    margin: 10px;
}
#amount {
    text-align: right;
    font-size: 15px;
    padding: 15px 0;
}
#passenger-table th {
    text-align: center;
    font-size: 14px;
    font-weight: 400;
    border: 1px solid gainsboro;
}
#passenger-table td,
#passenger-table th {
    padding: 10px;
}
#passenger-table td {
    border: 1px solid gainsboro;
}
.name-header {
    width: 200px;
}
.age-header {
    width: 80px;
}
.gender-header {
    width: 120px;
}
.type-header {
    width: 150px;
}

```

```

}
.btn-container {
    text-align: center;
}
.search-item {
    padding-right: 50px;
    padding-bottom: 20px;
}
.item-title {
    font-weight: 500;
    padding-top: 10px;
}
}
@media (max-width: 450px) {
    .e-sample-resize-container {
        height: 450px;
    }
    .responsive-align {
        width: 75%;
        margin: 0 auto;
    }
    .search-item {
        padding: 0 0 20px 0;
        width: 100%;
    }
}
}
#err1,
#err2,
#err3 {
    font-weight: bold;
    color: red;
    display: block;
    margin-top: 15px;
}
}
.wizard-title {
    font-size: 15px;
}
}
#PassengersList {
    overflow: auto;
}
}
#passenger-table {
    min-width: 500px;
    width: 100%;
}
}

```

### Load tab with data source in React Tab component

You can bind any data object to Tab items, by mapping it to a [header](#) and [content](#) property.

In the below demo, Data is fetched from an OData service using `DataManager`. The result data is formatted as a JSON object with `header` and `content` fields, which is set to [items](#) property of Tab.

### INDEX.JSX

```

import * as React from "react";
import { useRef, useEffect } from 'react';
import * as ReactDOM from 'react-dom';
import { DataManager, Query, ODataV4Adaptor } from '@syncfusion/ej2-data';

```

```
import { TabComponent } from '@syncfusion/ej2-react-navigations';
const SERVICE_URI =
'https://services.odata.org/V4/Northwind/Northwind.svc/Employees';
const App = () => {
  const tabInstance = useRef(null);
  useEffect(() => {
    const fetchTabItems = async () => {
      const e = await new DataManager({ url: SERVICE_URI, adaptor: new
ODataV4Adaptor() }).executeQuery(new Query().range(1, 4));
      const result = e.result;
      const mapping = { header: 'FirstName', content: 'Notes' };
      const itemsData = result.map((item) => ({
        header: { text: item[mapping.header] },
        content: item[mapping.content],
      }));
      tabInstance.current.items = itemsData;
      tabInstance.current.refresh();
    };
    if (tabInstance.current) {
      fetchTabItems();
    }
  }, [tabInstance]);
  return (
    <TabComponent heightAdjustMode='Auto' ref={tabInstance}>
    </TabComponent>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);
```

## INDEX.TSX

```
import * as React from "react";
import { useRef, useEffect } from 'react';
import * as ReactDOM from 'react-dom';
import { DataManager, Query, ODataV4Adaptor, ReturnOption } from
 '@syncfusion/ej2-data';
import { TabComponent } from '@syncfusion/ej2-react-navigations';
const SERVICE_URI =
'https://services.odata.org/V4/Northwind/Northwind.svc/Employees';
const App = () => {
  const tabInstance = useRef<TabComponent>(null);
  useEffect(() => {
    const fetchTabItems = async () => {
      const e: ReturnOption = await new DataManager({ url: SERVICE_URI,
adaptor: new ODataV4Adaptor() }).executeQuery(new Query().range(1, 4));
      const result: any = e.result;
      const mapping = { header: 'FirstName', content: 'Notes' };
      const itemsData: Object[] = result.map((item: any) => ({
        header: { text: item[mapping.header] },
        content: item[mapping.content],
      }));
      tabInstance.current!.items = itemsData;
      tabInstance.current!.refresh();
    };
    if (tabInstance.current) {
```

```

        fetchTabItems();
    }
}, [tabInstance]);
return (
    <TabComponent heightAdjustMode='Auto' ref={tabInstance}>
    </TabComponent>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

## INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Syncfusion React Tab</title>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta name="description" content="Essential JS 2 for React Components" />
    <meta name="author" content="Syncfusion" />
    <link href="index.css" rel="stylesheet" />
    <link href="https://cdn.syncfusion.com/ej2/20.3.56/material.css"
rel="stylesheet" />
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
    <script src="systemjs.config.js"></script>
</head>
<body>
    <div id='container'>
        <div id='element'>
            <div id='loader'>Loading</div>
        </div>
    </div>
</body>
</html>

```

## Add nested tabs in React Tab component

Tab supports to render the nested level of Tabs by using [content](#) property. You can add the nested Tab element inside the parent Tab [content](#) property.

## INDEX.JSX

```

import * as ReactDOM from "react-dom"
import * as React from "react";
import {
    TabComponent,
    TabItemDirective,
    TabItemsDirective
} from "@syncfusion/ej2-react-navigations";
const Default = () => {
    // Mapping Tab items Header property
    let headertext = [
        { text: "USA" },
    ]

```

```

    { text: "FRANCE" },
    { text: "AUSTRALIA" },
    { text: "NEW YORK" },
    { text: "LOS ANGELES" },
    { text: "CHICAGO" },
    { text: "PARIS" },
    { text: "MARSEILLE" },
    { text: "LYON" },
    { text: "SYDNEY" },
    { text: "MELBORNE" },
    { text: "BRISBANE" }
  ];
  const newYorkContent = () => {
    return (
      <div>
        New York City comprises 5 boroughs sitting where the Hudson River
        meets the Atlantic Ocean. At its core is Manhattan, a densely
        populated borough that's among the world's major commercial,
financial
        and cultural centers. Its iconic sites include skyscrapers such as
the
        Empire State Building and sprawling Central Park. Broadway theater is
        staged in neon-lit Times Square.
      </div>
    );
  }
  const losAngelesContent = () => {
    return (
      <div>
        Los Angeles is a sprawling Southern California city and the center of
        the nation's film and television industry. Near its iconic Hollywood
        sign, studios such as Paramount Pictures, Universal and Warner
        Brothers offer behind-the-scenes tours. On Hollywood Boulevard, TCL
        Chinese Theatre displays celebrities' hand- and footprints, the Walk
        of Fame honors thousands of luminaries and vendors sell maps to
stars'
        homes.
      </div>
    );
  }
  const chicagoContent = () => {
    return (
      <div>
        Chicago, on Lake Michigan in Illinois, is among the largest cities in
        the U.S. Famed for its bold architecture, it has a skyline punctuated
        by skyscrapers such as the iconic John Hancock Center, 1,451-ft.
        Willis Tower (formerly the Sears Tower) and the neo-Gothic Tribune
        Tower. The city is also renowned for its museums, including the Art
        Institute of Chicago with its noted Impressionist and
        Post-Impressionist works.
      </div>
    );
  }
  const parisContent = () => {
    return (
      <div>
        Paris, France capital, is a major European city and a global center

```

```

    for art, fashion, gastronomy and culture. Its 19th-century cityscape
    is crisscrossed by wide boulevards and the River Seine. Beyond such
    landmarks as the Eiffel Tower and the 12th-century, Gothic Notre-Dame
    cathedral, the city is known for its cafe culture and designer
    boutiques along the Rue du Faubourg Saint-Honoré.
  </div>
  );
}
const marseilleContent = () => {
  return (
    <div>
      Marseille, a port city in southern France, has been a crossroads of
      immigration and trade since its founding by the Greeks circa 600 B.C.
      At its heart is the Vieux-Port (Old Port), where fishmongers sell
      their catch along the boat-lined quay. Basilique
      Notre-Dame-de-la-Garde is a Romanesque-Byzantine church. Modern
      landmarks include Le Corbusier's influential Cité Radieuse complex
and
      Zaha Hadid's CMA CGM Tower.
    </div>
  );
}
const lyonContent = () => {
  return (
    <div>
      Lyon, the capital city in France's Auvergne-Rhône-Alpes region, sits
      at the junction of the Rhône and Saône rivers. Its center reflects
      2,000 years of history from the Roman Amphithéâtre des Trois Gaules,
      medieval and Renaissance architecture in Vieux (Old) Lyon, to the
      modern Confluence district on Presquîle peninsula. Traboules, covered
      passageways between buildings, connect Vieux Lyon and La Croix-Rousse
      hill.
    </div>
  );
}
const sydneyContent = () => {
  return (
    <div>
      Sydney, capital of New South Wales and one of Australia largest
      cities, is best known for its harbourfront Sydney Opera House, with a
      distinctive sail-like design. Massive Darling Harbour and the smaller
      Circular Quay port are hubs of waterside life, with the arched
Harbour
      Bridge and esteemed Royal Botanic Garden nearby. Sydney Tower's
      outdoor platform, the Skywalk, offers 360-degree views of the city
and
      suburbs.
    </div>
  );
}
const melbourneContent = () => {
  return (
    <div>
      Melbourne is the coastal capital of the southeastern Australian state
      of Victoria. At the city centre is the modern Federation Square
      development, with plazas, bars, and restaurants by the Yarra River.
In

```



```

        the Southbank area, the Melbourne Arts Precinct is the site of Arts
        Centre Melbourne - a performing arts complex - and the National
        Gallery of Victoria, with Australian and indigenous art.
    </div>
    );
}
const brisBaneContent = () => {
    return (
        <div>
            Brisbane, capital of Queensland, is a large city on the Brisbane
            River. Clustered in its South Bank cultural precinct are the
            Queensland Museum and Sciencentre, with noted interactive
exhibitions.
            Another South Bank cultural institution is Queensland Gallery of
            Modern Art, among Australia major contemporary art museums. Looming
            over the city is Mt. Coot-tha, site of Brisbane Botanic Gardens.
        </div>
    );
}
const nestedTab1 = () => {
    return (
        <TabComponent heightAdjustMode="Auto" id="nestedTab1">
            <TabItemsDirective>
                <TabItemDirective header={headertext[3]} content={newYorkContent}
/>
                <TabItemDirective header={headertext[4]}
content={losAngelesContent} />
                <TabItemDirective header={headertext[5]} content={chicagoContent}
/>
            </TabItemsDirective>
        </TabComponent>
    );
}
const nestedTab2 = () => {
    return (
        <TabComponent heightAdjustMode="Auto" id="nestedTab2">
            <TabItemsDirective>
                <TabItemDirective header={headertext[6]} content={parisContent} />
                <TabItemDirective header={headertext[7]} content={marseilleContent}
/>
                <TabItemDirective header={headertext[8]} content={lyonContent} />
            </TabItemsDirective>
        </TabComponent>
    );
}
const nestedTab3 = () => {
    return (
        <TabComponent heightAdjustMode="Auto" id="nestedTab3">
            <TabItemsDirective>
                <TabItemDirective header={headertext[9]} content={sydneyContent} />
                <TabItemDirective header={headertext[10]}
content={melbourneContent} />
                <TabItemDirective header={headertext[11]} content={brisBaneContent}
/>
            </TabItemsDirective>
        </TabComponent>
    );
}

```

```

    }
    return (
      <div className="control-pane">
        <div className="control-section tab-control-section">
          <TabComponent heightAdjustMode="Auto" id="defaultTab">
            <TabItemsDirective>
              <TabItemDirective header={headertext[0]} content={nestedTab1} />
              <TabItemDirective header={headertext[1]} content={nestedTab2} />
              <TabItemDirective header={headertext[2]} content={nestedTab3} />
            </TabItemsDirective>
          </TabComponent>
        </div>
      </div>
    );
  }
  const root = ReactDOM.createRoot(document.getElementById('element'));
  root.render(<Default />);

```

## INDEX.TSX

```

import * as ReactDOM from "react-dom"
import * as React from "react";
import {
  TabComponent,
  TabItemDirective,
  TabItemsDirective
} from "@syncfusion/ej2-react-navigations";
const Default = () => {
  // Mapping Tab items Header property
  let headertext = [
    { text: "USA" },
    { text: "FRANCE" },
    { text: "AUSTRALIA" },
    { text: "NEW YORK" },
    { text: "LOS ANGELES" },
    { text: "CHICAGO" },
    { text: "PARIS" },
    { text: "MARSEILLE" },
    { text: "LYON" },
    { text: "SYDNEY" },
    { text: "MELBORNE" },
    { text: "BRISBANE" }
  ];
  const newYorkContent = () => {
    return (
      <div>
        New York City comprises 5 boroughs sitting where the Hudson River
        meets the Atlantic Ocean. At its core is Manhattan, a densely
        populated borough that's among the world's major commercial,
        financial
        and cultural centers. Its iconic sites include skyscrapers such as
        the
        Empire State Building and sprawling Central Park. Broadway theater is
        staged in neon-lit Times Square.
      </div>
    );
  };

```

```

}
const losAngelesContent = () => {
  return (
    <div>
      Los Angeles is a sprawling Southern California city and the center of
      the nation's film and television industry. Near its iconic Hollywood
      sign, studios such as Paramount Pictures, Universal and Warner
      Brothers offer behind-the-scenes tours. On Hollywood Boulevard, TCL
      Chinese Theatre displays celebrities' hand- and footprints, the Walk
      of Fame honors thousands of luminaries and vendors sell maps to
      stars'
      homes.
    </div>
  );
}
const chicagoContent = () => {
  return (
    <div>
      Chicago, on Lake Michigan in Illinois, is among the largest cities in
      the U.S. Famed for its bold architecture, it has a skyline punctuated
      by skyscrapers such as the iconic John Hancock Center, 1,451-ft.
      Willis Tower (formerly the Sears Tower) and the neo-Gothic Tribune
      Tower. The city is also renowned for its museums, including the Art
      Institute of Chicago with its noted Impressionist and
      Post-Impressionist works.
    </div>
  );
}
const parisContent = () => {
  return (
    <div>
      Paris, France capital, is a major European city and a global center
      for art, fashion, gastronomy and culture. Its 19th-century cityscape
      is crisscrossed by wide boulevards and the River Seine. Beyond such
      landmarks as the Eiffel Tower and the 12th-century, Gothic Notre-Dame
      cathedral, the city is known for its cafe culture and designer
      boutiques along the Rue du Faubourg Saint-Honoré.
    </div>
  );
}
const marseilleContent = () => {
  return (
    <div>
      Marseille, a port city in southern France, has been a crossroads of
      immigration and trade since its founding by the Greeks circa 600 B.C.
      At its heart is the Vieux-Port (Old Port), where fishmongers sell
      their catch along the boat-lined quay. Basilique
      Notre-Dame-de-la-Garde is a Romanesque-Byzantine church. Modern
      landmarks include Le Corbusier's influential Cité Radieuse complex
      and
      Zaha Hadid's CMA CGM Tower.
    </div>
  );
}
const lyonContent = () => {
  return (
    <div>

```

```

    Lyon, the capital city in France's Auvergne-Rhône-Alpes region, sits
    at the junction of the Rhône and Saône rivers. Its center reflects
    2,000 years of history from the Roman Amphithéâtre des Trois Gaules,
    medieval and Renaissance architecture in Vieux (Old) Lyon, to the
    modern Confluence district on Presquîle peninsula. Traboules, covered
    passageways between buildings, connect Vieux Lyon and La Croix-Rousse
    hill.
  </div>
);
}
const sydneyContent = () => {
  return (
    <div>
      Sydney, capital of New South Wales and one of Australia largest
      cities, is best known for its harbourfront Sydney Opera House, with a
      distinctive sail-like design. Massive Darling Harbour and the smaller
      Circular Quay port are hubs of waterside life, with the arched
Harbour
      Bridge and esteemed Royal Botanic Garden nearby. Sydney Tower's
      outdoor platform, the Skywalk, offers 360-degree views of the city
and
      suburbs.
    </div>
  );
}
const melbourneContent = () => {
  return (
    <div>
      Melbourne is the coastal capital of the southeastern Australian state
      of Victoria. At the city centre is the modern Federation Square
      development, with plazas, bars, and restaurants by the Yarra River.
In
      the Southbank area, the Melbourne Arts Precinct is the site of Arts
      Centre Melbourne - a performing arts complex - and the National
      Gallery of Victoria, with Australian and indigenous art.
    </div>
  );
}
const brisBaneContent = () => {
  return (
    <div>
      Brisbane, capital of Queensland, is a large city on the Brisbane
      River. Clustered in its South Bank cultural precinct are the
      Queensland Museum and Sciencentre, with noted interactive
exhibitions.
      Another South Bank cultural institution is Queensland Gallery of
      Modern Art, among Australia major contemporary art museums. Looming
      over the city is Mt. Coot-tha, site of Brisbane Botanic Gardens.
    </div>
  );
}
const nestedTab1 = () => {
  return (
    <TabComponent heightAdjustMode="Auto" id="nestedTab1">
      <TabItemsDirective>
        <TabItemDirective header={headertext[3]} content={newYorkContent}
      />

```

```

        <TabItemDirective header={headertext[4]}
content={losAngelesContent} />
        <TabItemDirective header={headertext[5]} content={chicagoContent}
/>
    </TabItemsDirective>
</TabComponent>
);
}
const nestedTab2 = () => {
    return (
        <TabComponent heightAdjustMode="Auto" id="nestedTab2">
            <TabItemsDirective>
                <TabItemDirective header={headertext[6]} content={parisContent} />
                <TabItemDirective header={headertext[7]} content={marseilleContent}
/>
                <TabItemDirective header={headertext[8]} content={lyonContent} />
            </TabItemsDirective>
        </TabComponent>
    );
}
const nestedTab3 = () => {
    return (
        <TabComponent heightAdjustMode="Auto" id="nestedTab3">
            <TabItemsDirective>
                <TabItemDirective header={headertext[9]} content={sydneyContent} />
                <TabItemDirective header={headertext[10]}
content={melbourneContent} />
                <TabItemDirective header={headertext[11]} content={brisBaneContent}
/>
            </TabItemsDirective>
        </TabComponent>
    );
}
return (
    <div className="control-pane">
        <div className="control-section tab-control-section">
            <TabComponent heightAdjustMode="Auto" id="defaultTab">
                <TabItemsDirective>
                    <TabItemDirective header={headertext[0]} content={nestedTab1} />
                    <TabItemDirective header={headertext[1]} content={nestedTab2} />
                    <TabItemDirective header={headertext[2]} content={nestedTab3} />
                </TabItemsDirective>
            </TabComponent>
        </div>
    </div>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<Default />);

```

## INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Syncfusion React Tab</title>

```

```

<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<meta name="description" content="Essential JS 2 for React Components" />
<meta name="author" content="Syncfusion" />
<link href="index.css" rel="stylesheet" />
<link href="https://cdn.syncfusion.com/ej2/20.3.56/material.css"
rel="stylesheet" />
<script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
<script src="systemjs.config.js"></script>
</head>
<body>
  <div id='container'>
    <div id='element'>
      <div id='loader'>Loading</div>
    </div>
  </div>
</body>
</html>

```

### Set state persistence of the tab component in React Tab component

State persistence allows the Tab to retain the current modal value in the browser cookies for state maintenance. This action is handled through the [enablePersistence](#) property which is set to false by default. When it is set to true, some of the Tab component model values will be retained even after refreshing the page.

The following sample demonstrates how to set state persistence of the Tab component.

#### INDEX.JSX

```

import { TabComponent, TabItemDirective, TabItemsDirective } from
 '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
const ReactApp = () => {
  let headerText = [{ text: "Twitter" }, { text: "Facebook" }, { text:
  "WhatsApp" }];
  const twitterContent = () => {
    return <div>
      Twitter is an online social networking service that enables users to
      send and read short 140-character messages called "tweets". Registered users
      can read and post tweets, but those who are unregistered can only read them.
      Users access Twitter through the website interface, SMS or mobile device app
      Twitter Inc. is based in San Francisco and has more than 25 offices around
      the world. Twitter was created in March 2006 by Jack Dorsey, Evan Williams,
      Biz Stone, and Noah Glass and launched in July 2006. The service rapidly
      gained worldwide popularity, with more than 100 million users posting 340
      million tweets a day in 2012. The service also handled 1.6 billion search
      queries per day.
    </div>;
  }
  const faceBookcontent = () => {
    return <div>
      Facebook is an online social networking service headquartered in Menlo
      Park, California. Its website was launched on February 4, 2004, by Mark

```

Zuckerberg with his Harvard College roommates and fellow students Eduardo Saverin, Andrew McCollum, Dustin Moskovitz and Chris Hughes. The founders had initially limited the website membership to Harvard students, but later expanded it to colleges **in** the Boston area, the Ivy League, and Stanford University. It gradually added support **for** students at various other universities and later to high-school students.

```

    </div>;
  }
  const whatsappContent = () => {
    return <div>
      WhatsApp Messenger is a proprietary cross-platform instant messaging
      client for smartphones that operates under a subscription business model. It
      uses the Internet to send text messages, images, video, user location and
      audio media messages to other users using standard cellular mobile numbers.
      As of February 2016, WhatsApp had a user base of up to one billion, [10]
      making it the most globally popular messaging application. WhatsApp Inc.,
      based in Mountain View, California, was acquired by Facebook Inc. on February
      19, 2014, for approximately US$19.3 billion.
    </div>;
  }
  return (<TabComponent heightAdjustMode='Auto' enablePersistence={true}>
    <TabItemsDirective>
      <TabItemDirective header={headerText[0]} content={twitterContent}/>
      <TabItemDirective header={headerText[1]} content={faceBookContent}/>
      <TabItemDirective header={headerText[2]} content={whatsappContent}/>
    </TabItemsDirective>
  </TabComponent>);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

## INDEX.TSX

```

import { TabComponent, TabItemDirective, TabItemsDirective } from
'@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
const ReactApp = () => {
  let headerText: any = [{ text: "Twitter" }, { text: "Facebook" }, { text:
  "WhatsApp" }];
  const twitterContent = () => {
    return <div>
      Twitter is an online social networking service that enables users to
      send and read short 140-character messages called "tweets". Registered users
      can read and post tweets, but those who are unregistered can only read them.
      Users access Twitter through the website interface, SMS or mobile device app
      Twitter Inc. is based in San Francisco and has more than 25 offices around
      the world. Twitter was created in March 2006 by Jack Dorsey, Evan Williams,
      Biz Stone, and Noah Glass and launched in July 2006. The service rapidly
      gained worldwide popularity, with more than 100 million users posting 340
      million tweets a day in 2012. The service also handled 1.6 billion search
      queries per day.
    </div>;
  }
  const faceBookContent = () => {
    return <div>

```

Facebook **is** an online social networking service headquartered **in** Menlo Park, California. Its website was launched on February **4, 2004**, by Mark Zuckerberg with his Harvard College roommates and fellow students Eduardo Saverin, Andrew McCollum, Dustin Moskovitz and Chris Hughes. The founders had initially limited the website membership to Harvard students, but later expanded it to colleges **in** the Boston area, the Ivy League, and Stanford University. It gradually added support **for** students at various other universities and later to high-school students.

```

    </div>;
  }
  const whatsappContent = () => {
    return <div>
      WhatsApp Messenger is a proprietary cross-platform instant messaging
      client for smartphones that operates under a subscription business model. It
      uses the Internet to send text messages, images, video, user location and
      audio media messages to other users using standard cellular mobile numbers.
      As of February 2016, WhatsApp had a user base of up to one billion, [10]
      making it the most globally popular messaging application. WhatsApp Inc.,
      based in Mountain View, California, was acquired by Facebook Inc. on February
      19, 2014, for approximately US$19.3 billion.
    </div>;
  }
  return (
    <TabComponent heightAdjustMode='Auto' enablePersistence={true}>
      <TabItemsDirective>
        <TabItemDirective header={headerText[0]} content={twitterContent} />
        <TabItemDirective header={headerText[1]} content={faceBookContent} />
        <TabItemDirective header={headerText[2]} content={whatsappContent} />
      </TabItemsDirective>
    </TabComponent>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

## INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Tab</title>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta name="description" content="Essential JS 2 for React Components" />
  <meta name="author" content="Syncfusion" />
  <link href="index.css" rel="stylesheet" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/material.css"
rel="stylesheet" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
  <script src="systemjs.config.js"></script>
</head>
<body>
  <div id='container'>
    <div id='element'>

```



```

        <div id='loader'>Loading</div>
    </div>
</div>
</body>
</html>

```

### Set custom animation in React Tab component

Tab supports custom animations for both previous and next actions from the provided animation option of [Animation](#) library. The [animation](#) property also allows you to set [easing](#), [duration](#), and various other [effect](#).

Default animation is given as [SlideLeftIn](#) for [previous](#) tab animation and [SlideRightIn](#) for [next](#) tab animation. You can also disable the animation by setting the animation effect as [None](#). Also, please use the following CSS to disable indicator animation for animation effect as [None](#).

`css

```

.e-tab .e-tab-header:not(.e-vertical) .e-indicator, .e-tab .e-tab-header.e-vertical .e-indicator {
transition: none;
}
`

```

The sample demonstrates some types of animation that suits Tab. You can check all the animation effects [here](#).

### INDEX.JSX

```

import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { TabComponent, TabItemDirective, TabItemsDirective } from
 '@syncfusion/ej2-react-navigations';
import { useEffect, useRef } from 'react';
import * as React from "react";
import * as ReactDOM from 'react-dom';
const ReactApp = () => {
    const headerText = [{ text: "Twitter" }, { text: "Facebook" }, { text:
    "WhatsApp" }];
    const tabInstance = useRef(null);
    const previousInstance = useRef(null);
    const nextInstance = useRef(null);
    let animationData = ['SlideLeftIn', 'SlideRightIn', 'FadeIn', 'FadeOut',
    'FadeZoomIn', 'FadeZoomOut', 'ZoomIn', 'ZoomOut', 'None'];
    useEffect(() => {
        previousAnimationChange = previousAnimationChange;
        nextAnimationChange = nextAnimationChange;
    }, []);
    const twitterContent = () => {
        return <div>
            Twitter is an online social networking service that enables users to
            send and read short 140-character messages called "tweets". Registered users
            can read and post tweets, but those who are unregistered can only read them.
            Users access Twitter through the website interface, SMS or mobile device app
            Twitter Inc. is based in San Francisco and has more than 25 offices around
            the world. Twitter was created in March 2006 by Jack Dorsey, Evan Williams,
            Biz Stone, and Noah Glass and launched in July 2006. The service rapidly

```

gained worldwide popularity, with more than **100** million users posting **340** million tweets a day **in 2012**. The service also handled **1.6** billion search queries per day.

```

    </div>;
  }
  const faceBookContent = () => {
    return <div>
      Facebook is an online social networking service headquartered in Menlo Park, California. Its website was launched on February 4, 2004, by Mark Zuckerberg with his Harvard College roommates and fellow students Eduardo Saverin, Andrew McCollum, Dustin Moskovitz and Chris Hughes. The founders had initially limited the website membership to Harvard students, but later expanded it to colleges in the Boston area, the Ivy League, and Stanford University. It gradually added support for students at various other universities and later to high-school students.
    </div>;
  }

```

```

  const whatsAppcontent = () => {
    return <div>
      WhatsApp Messenger is a proprietary cross-platform instant messaging client for smartphones that operates under a subscription business model. It uses the Internet to send text messages, images, video, user location and audio media messages to other users using standard cellular mobile numbers. As of February 2016, WhatsApp had a user base of up to one billion, [10] making it the most globally popular messaging application. WhatsApp Inc., based in Mountain View, California, was acquired by Facebook Inc. on February 19, 2014, for approximately US$19.3 billion.
    </div>;
  }

```

```

  let previousAnimationChange = () => {
    tabInstance.current.animation.previous = { effect:
previousInstance.current.value };
  }
  let nextAnimationChange = () => {
    tabInstance.current.animation.next = { effect: nextInstance.current.value
};
  }
  const wrapStyle = { padding: '25px 0 0' };
  return (
    <div style={wrapStyle}>
      <div className='row'>
        <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
          <label> Previous Animation </label>
        </div>
        <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
          <DropDownListComponent ref={previousInstance} index={0}
change={previousAnimationChange} dataSource={animationData}
popupHeight="250px" placeholder="Select Previous animation" />
        </div>
      </div>
      <div className='row'>
        <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
          <label> Next Animation </label>
        </div>
        <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">

```

```

        <DropDownListComponent ref={nextInstance} index={0}
        change={nextAnimationChange} dataSource={animationData} popupHeight="250px"
        placeholder="Select Next animation" />
      </div>
    </div>
    <TabComponent ref={tabInstance}>
      <TabItemsDirective>
        <TabItemDirective header={headerText[0]} content={twitterContent}
      />
        <TabItemDirective header={headerText[1]} content={faceBookContent}
      />
        <TabItemDirective header={headerText[2]} content={whatsAppContent}
      />
      </TabItemsDirective>
    </TabComponent>
  </div>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

## INDEX.TSX

```

import { Effect } from '@syncfusion/ej2-base';
import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { TabComponent, TabItemDirective, TabItemsDirective } from
 '@syncfusion/ej2-react-navigations';
import * as React from "react";
import { useEffect, useRef } from 'react';
import * as ReactDOM from 'react-dom';
const ReactApp = () => {
  const headerText: any = [{ text: "Twitter" }, { text: "Facebook" }, { text:
  "WhatsApp" }];
  const tabInstance = useRef<TabComponent>(null);
  const previousInstance = useRef<DropDownListComponent>(null);
  const nextInstance = useRef<DropDownListComponent>(null);
  let animationData: string[] = ['SlideLeftIn', 'SlideRightIn', 'FadeIn',
  'FadeOut', 'FadeZoomIn', 'FadeZoomOut', 'ZoomIn', 'ZoomOut', 'None'];
  useEffect(() => {
    previousAnimationChange = previousAnimationChange;
    nextAnimationChange = nextAnimationChange;
  }, []);
  const twitterContent = () => {
    return <div>
      Twitter is an online social networking service that enables users to
      send and read short 140-character messages called "tweets". Registered users
      can read and post tweets, but those who are unregistered can only read them.
      Users access Twitter through the website interface, SMS or mobile device app
      Twitter Inc. is based in San Francisco and has more than 25 offices around
      the world. Twitter was created in March 2006 by Jack Dorsey, Evan Williams,
      Biz Stone, and Noah Glass and launched in July 2006. The service rapidly
      gained worldwide popularity, with more than 100 million users posting 340
      million tweets a day in 2012. The service also handled 1.6 billion search
      queries per day.
    </div>;
  }
}

```

```

const faceBookContent = () => {
  return <div>
    Facebook is an online social networking service headquartered in Menlo
    Park, California. Its website was launched on February 4, 2004, by Mark
    Zuckerberg with his Harvard College roommates and fellow students Eduardo
    Saverin, Andrew McCollum, Dustin Moskovitz and Chris Hughes. The founders had
    initially limited the website membership to Harvard students, but later
    expanded it to colleges in the Boston area, the Ivy League, and Stanford
    University. It gradually added support for students at various other
    universities and later to high-school students.
  </div>;
}
const whatsappcontent = () => {
  return <div>
    WhatsApp Messenger is a proprietary cross-platform instant messaging
    client for smartphones that operates under a subscription business model. It
    uses the Internet to send text messages, images, video, user location and
    audio media messages to other users using standard cellular mobile numbers.
    As of February 2016, WhatsApp had a user base of up to one billion, [10]
    making it the most globally popular messaging application. WhatsApp Inc.,
    based in Mountain View, California, was acquired by Facebook Inc. on February
    19, 2014, for approximately US$19.3 billion.
  </div>;
}
let previousAnimationChange = (): void => {
  tabInstance.current.animation.previous = { effect:
previousInstance.current.value as Effect };
}
let nextAnimationChange = (): void => {
  tabInstance.current.animation.next = { effect: nextInstance.current.value
as Effect };
}
const wrapStyle = { padding: '25px 0 0' };
return (
  <div style={wrapStyle}>
    <div className='row'>
      <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
        <label> Previous Animation </label>
      </div>
      <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
        <DropDownListComponent ref={previousInstance} index={0}
change={previousAnimationChange} dataSource={animationData}
popupHeight="250px" placeholder="Select Previous animation" />
      </div>
    </div>
    <div className='row'>
      <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
        <label> Next Animation </label>
      </div>
      <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
        <DropDownListComponent ref={nextInstance} index={0}
change={nextAnimationChange} dataSource={animationData} popupHeight="250px"
placeholder="Select Next animation" />
      </div>
    </div>
    <TabComponent ref={tabInstance}>
      <TabItemsDirective>

```

```

        <TabItemDirective header={headerText[0]} content={twitterContent}
    />
        <TabItemDirective header={headerText[1]} content={faceBookContent}
    />
        <TabItemDirective header={headerText[2]} content={whatsAppcontent}
    />
    </TabItemsDirective>
  </TabComponent>
</div>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

## INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Tab</title>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta name="description" content="Essential JS 2 for React Components" />
  <meta name="author" content="Syncfusion" />
  <link href="index.css" rel="stylesheet" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/material.css"
rel="stylesheet" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
  <script src="systemjs.config.js"></script>
</head>
<body>
  <div id='container'>
    <div id='element'>
      <div id='loader'>Loading</div>
    </div>
  </div>
</body>
</html>

```

### Load tab items dynamically in React Tab component

Tabs can be added dynamically by passing array of items and index value to the [addTab](#) method.

```
`ts
```

```
// New tab title and content inputs are fetched and stored in local variable
```

```
let title: string = document.getElementById('tab-title').value;
```

```
let content: string = document.getElementById('tab-content').value;
```

```
// Required tab item object formed by using textbox inputs
```

```
let item: Object = { header: { text: title }, content: createElement('pre', { innerHTML:
content.replace(/\n/g, '<br>\n') }).outerHTML };
```

// Item object and the index argument passed into the addTab method to add a new tab

```
this.addTab([item], this.totalItems-1);
```

,

In the following demo, you can add the tab content by clicking the +. This + icon is added on the tab header using [iconCss](#) property. Enter the new Tab heading and content details in the available text boxes, click 'Add Tab' button to pass the details as an array and here last index is calculated to append the new tab at the end.

### INDEX.JSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { TabComponent, TabItemDirective, TabItemsDirective } from
 '@syncfusion/ej2-react-navigations';
import { Tooltip } from '@syncfusion/ej2-popups';
import { enableRipple } from '@syncfusion/ej2-base';
enableRipple(true);
function App() {
  // eslint-disable-next-line @typescript-eslint/no-unused-vars
  let tabInstance;
  const tabCreated = () => {
    const tooltip = new Tooltip({
      content: 'Add Tab'
    });
    tooltip.appendTo('.e-ileft.e-icon');
    document.getElementById('btn-add').onclick = (e) => {
      const title = document.getElementById('tab-title').value;
      const content = document.getElementById('tab-content').value;
      const ele = document.createElement("pre");
      ele.innerHTML = content.replace(/\n/g, "<br>\n");
      // tslint:disable-next-line: ban-types
      const item = { header: { text: title }, content: ele.outerHTML };
      const totalItems = document.querySelectorAll('#element .e-
toolbar-item').length;

document.getElementById('tabelement').ej2_instances[0].addTab([item],
totalItems - 1);
    };
  };
  const tabSelected = (args) => {
    if (args.selectedIndex === document.querySelectorAll('#element .e-
toolbar-item').length - 1) {
      document.getElementById('tab-title').value = '';
      document.getElementById('tab-content').value = '';
    }
  };
  let headertext;
  headertext = [{ text: "Twitter" }, { 'iconCss': 'e-add-icon' }];
  return (<TabComponent id='tabelement' ref={tab => tabInstance = tab}
created={tabCreated} selected={tabSelected}>
    <TabItemsDirective>
      <TabItemDirective header={headertext[0]} content={'#tab1_content'}/>
      <TabItemDirective header={headertext[1]} content={'#form-
container'}/>
  </TabItemsDirective>
</TabComponent>);
}
```

```

    </TabItemsDirective>
  </TabComponent>);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

## INDEX.TSX

```

import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { TabComponent, TabItemDirective, TabItemsDirective, SelectEventArgs }
from '@syncfusion/ej2-react-navigations';
import { Tooltip } from '@syncfusion/ej2-popups';
import { enableRipple } from '@syncfusion/ej2-base';
enableRipple(true);
function App() {
  // eslint-disable-next-line @typescript-eslint/no-unused-vars
  let tabInstance: TabComponent | any;
  const tabCreated = (): void => {
    const tooltip: Tooltip = new Tooltip({
      content: 'Add Tab'
    });
    tooltip.appendTo('.e-ileft.e-icon');
    (document.getElementById('btn-add') as any).onclick = (e: Event) => {
      const title: string = (document.getElementById('tab-title') as
any).value;
      const content: string = (document.getElementById('tab-content') as
any).value;
      const ele = document.createElement("pre");
      ele.innerHTML = content.replace(/\n/g, "<br>\n");
      // tslint:disable-next-line: ban-types
      const item: Object = { header: { text: title }, content: ele.outerHTML
    };
      const totalItems = document.querySelectorAll('#element .e-toolbar-
item').length;
      (document.getElementById('tabelement') as
any).ej2_instances[0].addTab([item], totalItems - 1);
    };
  }
  const tabSelected = (args: SelectEventArgs): any => {
    if (args.selectedIndex === document.querySelectorAll('#element .e-
toolbar-item').length - 1) {
      (document.getElementById('tab-title') as any).value = '';
      (document.getElementById('tab-content') as any).value = '';
    }
  }
  let headertext: any;
  headertext = [{ text: "Twitter" }, { 'iconCss': 'e-add-icon' }];
  return (
    <TabComponent id='tabelement' ref={tab => tabInstance = tab}
created={tabCreated} selected={tabSelected}>
      <TabItemsDirective>
        <TabItemDirective header={headertext[0]} content={'#tab1_content'} />
        <TabItemDirective header={headertext[1]} content={'#form-container'}
      />
    </TabItemsDirective>
  );
}

```

```

    </TabComponent>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

## INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Tab</title>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta name="description" content="Essential JS 2 for React Components" />
  <meta name="author" content="Syncfusion" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/material.css"
rel="stylesheet" />
  <link href="index.css" rel="stylesheet" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
  <script src="systemjs.config.js"></script>
</head>
<body>
  <div id='container'>
    <div id="tab1_content" style="display: none">
      <ul>
        <li>Click on the "+" header to add dynamic tab items.
</li>
        <li>It displays input elements to get the new tab
information. </li>
        <li>Add details and click the "Add Tab" button to
open the newly added tab.</li>
      </ul>
    </div>
    <div id="form-container" style="display: none">
      <div class="e-float-input">
        <input type="text" id="tab-title" required="" />
        <span class="e-float-line"></span>
        <label class="e-float-text">Enter header
title</label>
      </div>
      <br />
      <div class="e-float-input">
        <textarea rows="5" type="text" id="tab-content"
required=""></textarea>
        <span class="e-float-line"></span>
        <label class="e-float-text">Enter content</label>
      </div>
      <br />
      <div class="btn-section">
        <button id="btn-add" class="e-btn">Add Tab</button>
        <br />
        <br />

```



```

<span class="info"> * Title is mandatory to add a new
Tab</span></div>
</div>
<div id='element'>
<div id='loader'>Loading</div>
</div>
</div>
</body>
</html>

```

### Create collapsible tabs in React Tab component

You can achieve collapse and expand functionality in Tab by adding/removing a custom CSS class in the click event handler for each tab.

- Define a CSS class to set the style property display as none. Here 'collapse' class is added to the content element for hiding it using [created](#) event.
- Bind the [selected](#) event for Tab to collapse the initially selected Tab item and bind custom click handler for the Tab headers.
- In the event handler, add and remove 'collapse' class to hide and show the corresponding Tab content.

### INDEX.JSX

```

import { useEffect, useRef, useState } from 'react';
import * as React from "react";
import * as ReactDOM from 'react-dom';
import { TabComponent, TabItemDirective, TabItemsDirective, SelectEventArgs }
from '@syncfusion/ej2-react-navigations';
import { enableRipple } from '@syncfusion/ej2-base';
enableRipple(true);
const App = () => {
  const tabInstance = useRef(null);
  const [trgIndex, settrgIndex] = useState(0);
  let actLine;
  useEffect(() => {
    updateCollapseClass = updateCollapseClass;
  }, []);
  let updateCollapseClass = (index) => {
    // Custom classes are added/removed from tab content and active line
    element, when the same tab item again clicked
    const cntEle = tabInstance.current.element.querySelector("#e-content-
element_" + index);
    actLine = tabInstance.current.element.querySelector('.e-indicator');
    if (cntEle.classList.contains('collapse')) {
      cntEle.classList.remove('collapse');
      actLine.classList.remove('collapse');
    } else {
      cntEle.classList.add('collapse');
      actLine.classList.add('collapse');
    }
  }
  const tabCreated = () => {

```

```

// After tab created first tab content and active line are hidden by
adding custom class
actLine = tabInstance.current.element.querySelector('.e-indicator') as
any;
if (tabInstance.current.element.querySelector(".e-content-element_0") !==
null)
    (tabInstance.current.element.querySelector(".e-content-element_0") as
any).classList.add('collapse');
actLine.classList.add('collapse');
}
const tabSelected = (e) => {
    // If next tab item selected custom class is removed from content and
active line element
    let cnttrgs = tabInstance.current.element.querySelectorAll('#element.e-
tab > .e-content > .e-item');
    for (let i = 0; i < cnttrgs.length; i++) {
        cnttrgs[i].classList.remove('collapse');
    }
    if (actLine !== undefined) { actLine.classList.remove('collapse'); }
    setrgIndex(e.selectedIndex);
    e.selectedItem.onclick = (e) => {
        updateCollapseClass(trgIndex);
    };
}
const twitterContent = () => {
    return (<div>
        Twitter is an online social networking service that enables users to
send and read short 140-character
        messages called "tweets". Registered users can read and post tweets,
but those who are unregistered can only read
        them. Users access Twitter through the website interface, SMS or mobile
device app Twitter Inc. is based in San
        Francisco and has more than 25 offices around the world. Twitter was
created in March 2006 by Jack Dorsey,
        Evan Williams, Biz Stone, and Noah Glass and launched in July 2006. The
service rapidly gained worldwide popularity,
        with more than 100 million users posting 340 million tweets a day in
2012. The service also handled 1.6 billion '
        search queries per day.
    </div>)
}
const faceBookContent = () => {
    return (<div>
        Facebook is an online social networking service headquartered in Menlo
Park, California. Its website was
        launched on February 4, 2004, by Mark Zuckerberg with his Harvard
College roommates and fellow students Eduardo
        Saverin, Andrew McCollum, Dustin Moskovitz and Chris Hughes. The
founders had initially limited the website
        membership to Harvard students, but later expanded it to colleges in
the Boston area, the Ivy League, and Stanford
        University. It gradually added support for students at various other
universities and later to high-school students.
    </div>);
}
const WhatsAppContent = () => {
    return (

```

```

        <div>
            WhatsApp Messenger is a proprietary cross-platform instant messaging
            client for smartphones that operates
            under a subscription business model. It uses the Internet to send
            text messages, images, video, user location and
            audio media messages to other users using standard cellular mobile
            numbers. As of February 2016, WhatsApp had a user
            base of up to one billion, [10] making it the most globally popular
            messaging application. WhatsApp Inc., based in
            Mountain View, California, was acquired by Facebook Inc. on February
            19, 2014, for approximately US$19.3 billion.
        </div>;
    }
    let headertext;
    headertext = [{ text: "Twitter" }, { text: "Facebook" }, { text: "WhatsApp"
}];
    return (
        <TabComponent heightAdjustMode='Auto' className="e-background"
id='element' ref={tabInstance} created={tabCreated} selected={tabSelected}>
            <TabItemsDirective>
                <TabItemDirective header={headertext[0]}
                    content={twitterContent} />
                <TabItemDirective header={headertext[1]}
                    content={faceBookContent} />
                <TabItemDirective header={headertext[2]}
                    content={WhatsAppContent} />
            </TabItemsDirective>
        </TabComponent>
    );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

## INDEX.TSX

```

import { useEffect, useRef, useState } from 'react';
import * as React from "react";
import * as ReactDOM from 'react-dom';
import { TabComponent, TabItemDirective, TabItemsDirective, SelectEventArgs }
from '@syncfusion/ej2-react-navigations';
import { enableRipple } from '@syncfusion/ej2-base';
enableRipple(true);
const App = () => {
    const tabInstance = useRef<TabComponent>(null);
    const [trgIndex, setTrgIndex] = useState<number>(0);
    let actLine: HTMLElement;
    useEffect(() => {
        updateCollapseClass = updateCollapseClass;
    }, []);
    let updateCollapseClass = (index: number): void => {
        // Custom classes are added/removed from tab content and active line
        // element, when the same tab item again clicked
        const cntEle: any = tabInstance.current.element.querySelector("#e-
content-element_" + index);
        actLine = tabInstance.current.element.querySelector('.e-indicator') as
any
    }
}

```

```

    if (cntEle.classList.contains('collapse')) {
        cntEle.classList.remove('collapse');
        actLine.classList.remove('collapse');
    } else {
        cntEle.classList.add('collapse');
        actLine.classList.add('collapse');
    }
}

const tabCreated = (): void => {
    // After tab created first tab content and active line are hidden by
    // adding custom class
    actLine = tabInstance.current.element.querySelector('.e-indicator') as
any;
    if (tabInstance.current.element.querySelector(".e-content-element_0") !==
null)
        (tabInstance.current.element.querySelector(".e-content-element_0") as
any).classList.add('collapse');
    actLine.classList.add('collapse');
}

const tabSelected = (e: SelectEventArgs): void => {
    // If next tab item selected custom class is removed from content and
    // active line element
    let cnttrgs: NodeListOf<Element> =
tabInstance.current.element.querySelectorAll('#element.e-tab > .e-content >
.e-item');
    for (let i: number = 0; i < cnttrgs.length; i++) {
        cnttrgs[i].classList.remove('collapse');
    }
    if (actLine !== undefined) { actLine.classList.remove('collapse'); }
    setrgIndex(e.selectedIndex);
    e.selectedItem.onclick = (e: Event) => {
        updateCollapseClass(trgIndex);
    };
}

const twitterContent = () => {
    return (<div>
        Twitter is an online social networking service that enables users to
        send and read short 140-character
        messages called "tweets". Registered users can read and post tweets,
        but those who are unregistered can only read
        them. Users access Twitter through the website interface, SMS or mobile
        device app Twitter Inc. is based in San
        Francisco and has more than 25 offices around the world. Twitter was
        created in March 2006 by Jack Dorsey,
        Evan Williams, Biz Stone, and Noah Glass and launched in July 2006. The
        service rapidly gained worldwide popularity,
        with more than 100 million users posting 340 million tweets a day in
2012.The service also handled 1.6 billion '
        search queries per day.
    </div>)
}

const faceBookContent = () => {
    return (<div>
        Facebook is an online social networking service headquartered in Menlo
        Park, California. Its website was
        launched on February 4, 2004, by Mark Zuckerberg with his Harvard
        College roommates and fellow students Eduardo

```

```

    Saverin, Andrew McCollum, Dustin Moskovitz and Chris Hughes. The
    founders had initially limited the website
    membership to Harvard students, but later expanded it to colleges in
    the Boston area, the Ivy League, and Stanford
    University. It gradually added support for students at various other
    universities and later to high-school students.
  </div>);
}
const WhatsAppContent = () => {
  return (
    <div>
      WhatsApp Messenger is a proprietary cross-platform instant messaging
      client for smartphones that operates
      under a subscription business model. It uses the Internet to send
      text messages, images, video, user location and
      audio media messages to other users using standard cellular mobile
      numbers. As of February 2016, WhatsApp had a user
      base of up to one billion, [10] making it the most globally popular
      messaging application. WhatsApp Inc., based in
      Mountain View, California, was acquired by Facebook Inc. on February
      19, 2014, for approximately US$19.3 billion.
    </div>);
}
let headertext: any;
headertext = [{ text: "Twitter" }, { text: "Facebook" }, { text: "WhatsApp"
}];
return (
  <TabComponent heightAdjustMode='Auto' className="e-background"
id='element' ref={tabInstance} created={tabCreated} selected={tabSelected}>
    <TabItemsDirective>
      <TabItemDirective header={headertext[0]}
        content={twitterContent} />
      <TabItemDirective header={headertext[1]}
        content={faceBookContent} />
      <TabItemDirective header={headertext[2]}
        content={WhatsAppContent} />
    </TabItemsDirective>
  </TabComponent>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

## INDEX.CSS

```

#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
.e-content .e-item {
  font-size: 12px;
  margin: 10px;
}

```

```

        text-align: justify;
    }
    .container {
        margin: 0 10px;
        min-width: 350px;
    }
    .info {
        margin: 10px;
        font-weight: bold;
    }
    .e-tab .e-content > .e-item.e-active.collapse { /* csslint allow: adjoining-classes */
        display: none;
    }
    .e-tab .e-tab-header .e-indicator.collapse { /* csslint allow: adjoining-classes */
        display: none;
    }
}

```

### Customize tab content height in React Tab component

You can change the Tab content height by using the [heightAdjustMode](#) property. By default, the Tab content [heightAdjustMode](#) property is set to **Content** value.

- **None:** Each tab content height is set based on the Tab height. This value is used only the tab component having the [height](#) property.
- **Auto:** Each tab content height will take the maximum height of all other tabs content.
- **Content:** Each tab content height is set based on their own content.
- **Fill:** Each tab content height is set based on the full height of Tabs parent element.

### INDEX.JSX

```

import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { TabComponent, TabItemDirective, TabItemsDirective } from
 '@syncfusion/ej2-react-navigations';
import { useRef } from 'react';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
const ReactApp = () => {
    let headerText = [{ text: "Twitter" }, { text: "Facebook" }, { text:
    "WhatsApp" }];
    const tabInstance = useRef(null);
    const dropInstance = useRef(null);
    let fields = { text: 'text', value: 'value' };
    let heightData = [
        { 'value': 'None', 'text': 'None' }, { 'value': 'Content', 'text':
        'Content' },
        { 'value': 'Fill', 'text': 'Fill' }, { 'value': 'Auto', 'text': 'Auto' }
    ];
    const value = 'Content';
    const twitterContent = () => {
        return <div>
            Twitter is an online social networking service that enables users to
            send and read short 140-character messages called "tweets". Registered users
            can read and post tweets, but those who are unregistered can only read them.

```

Users access Twitter through the website **interface**, SMS or mobile device app. Twitter Inc. **is** based **in** San Francisco and has more than **25** offices around the world. Twitter was created **in** March **2006** by Jack Dorsey, Evan Williams, Biz Stone, and Noah Glass and launched **in** July **2006**. The service rapidly gained worldwide popularity, with more than **100** million users posting **340** million tweets a day **in 2012**. The service also handled **1.6** billion search queries per day.

```

    </div>;
  }
  const faceBookcontent = () => {
    return <div>
      Facebook is an online social networking service headquartered in Menlo
      Park, California. Its website was launched on February 4, 2004, by Mark
      Zuckerberg with his Harvard College roommates and fellow students Eduardo
      Saverin, Andrew McCollum, Dustin Moskovitz and Chris Hughes. The founders had
      initially limited the website membership to Harvard students, but later
      expanded it to colleges in the Boston area, the Ivy League, and Stanford
      University. It gradually added support for students at various other
      universities and later to high-school students.
    </div>;
  }
  const whatsappcontent = () => {
    return <div>
      WhatsApp Messenger is a proprietary cross-platform instant messaging
      client for smartphones that operates under a subscription business model. It
      uses the Internet to send text messages, images, video, user location and
      audio media messages to other users using standard cellular mobile numbers.
      As of February 2016, WhatsApp had a user base of up to one billion, [10]
      making it the most globally popular messaging application. WhatsApp Inc.,
      based in Mountain View, California, was acquired by Facebook Inc. on February
      19, 2014, for approximately US$19.3 billion.
    </div>;
  }
  const onChange = () => {
    tabInstance.current.heightAdjustMode = dropInstance.current.value;
  }
  return (
    <div>
      <DropDownListComponent ref={dropInstance} id='contentHeight'
      dataSource={heightData} fields={fields} value={value} width={'150'}
      change={onChange} />
      <br /><br />
      <TabComponent ref={tabInstance} height="400px">
        <TabItemsDirective>
          <TabItemDirective header={headerText[0]} content={twitterContent}
        />
          <TabItemDirective header={headerText[1]} content={faceBookcontent}
        />
          <TabItemDirective header={headerText[2]} content={whatsappcontent}
        />
        </TabItemsDirective>
      </TabComponent>
    </div>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

**INDEX.TSX**

```

import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { HeightStyles, TabComponent, TabItemDirective, TabItemsDirective }
from '@syncfusion/ej2-react-navigations';
import { useRef } from 'react';
import * as React from "react";
import * as ReactDOM from 'react-dom';
const ReactApp = () => {
  let headerText: any = [{ text: "Twitter" }, { text: "Facebook" }, { text:
  "WhatsApp" }];
  const tabInstance = useRef<TabComponent>(null);
  const dropInstance = useRef<DropDownListComponent>(null);
  let fields: object = { text: 'text', value: 'value' };
  let heightData: any = [
    { 'value': 'None', 'text': 'None' }, { 'value': 'Content', 'text':
    'Content' },
    { 'value': 'Fill', 'text': 'Fill' }, { 'value': 'Auto', 'text': 'Auto' }
  ];
  const value: string = 'Content';
  const twitterContent = () => {
    return <div>
      Twitter is an online social networking service that enables users to
      send and read short 140-character messages called "tweets". Registered users
      can read and post tweets, but those who are unregistered can only read them.
      Users access Twitter through the website interface, SMS or mobile device app
      Twitter Inc. is based in San Francisco and has more than 25 offices around
      the world. Twitter was created in March 2006 by Jack Dorsey, Evan Williams,
      Biz Stone, and Noah Glass and launched in July 2006. The service rapidly
      gained worldwide popularity, with more than 100 million users posting 340
      million tweets a day in 2012. The service also handled 1.6 billion search
      queries per day.
    </div>;
  }
  const faceBookContent = () => {
    return <div>
      Facebook is an online social networking service headquartered in Menlo
      Park, California. Its website was launched on February 4, 2004, by Mark
      Zuckerberg with his Harvard College roommates and fellow students Eduardo
      Saverin, Andrew McCollum, Dustin Moskovitz and Chris Hughes. The founders had
      initially limited the website membership to Harvard students, but later
      expanded it to colleges in the Boston area, the Ivy League, and Stanford
      University. It gradually added support for students at various other
      universities and later to high-school students.
    </div>;
  }
  const whatsappContent = () => {
    return <div>
      WhatsApp Messenger is a proprietary cross-platform instant messaging
      client for smartphones that operates under a subscription business model. It
      uses the Internet to send text messages, images, video, user location and
      audio media messages to other users using standard cellular mobile numbers.
      As of February 2016, WhatsApp had a user base of up to one billion, [10]
      making it the most globally popular messaging application. WhatsApp Inc.,

```



```

based in Mountain View, California, was acquired by Facebook Inc. on February
19, 2014, for approximately US$19.3 billion.
    </div>;
  }
  const onChange = () => {
    tabInstance.current.heightAdjustMode = dropInstance.current.value as
    HeightStyles;
  }
  return (
    <div>
      <DropDownListComponent ref={dropInstance} id='contentHeight'
      dataSource={heightData} fields={fields} value={value} width={'150'}
      change={onChange} />
      <br /><br />
      <TabComponent ref={tabInstance} height="400px">
        <TabItemsDirective>
          <TabItemDirective header={headerText[0]} content={twitterContent}
        />
          <TabItemDirective header={headerText[1]} content={faceBookcontent}
        />
          <TabItemDirective header={headerText[2]} content={whatsAppcontent}
        />
        </TabItemsDirective>
      </TabComponent>
    </div>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

## INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Tab</title>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta name="description" content="Essential JS 2 for React Components" />
  <meta name="author" content="Syncfusion" />
  <link href="index.css" rel="stylesheet" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/material.css"
rel="stylesheet" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
  <script src="systemjs.config.js"></script>
</head>
<body>
  <div id='container'>
    <div id='element'>
      <div id='loader'>Loading</div>
    </div>
  </div>
</body>
</html>

```

### Render other components in tab using template in React Tab component

You can render other components inside Tab using React [Link to the Video](#). Through this, we can add content as other components directly with all functionalities to our Tab. Follow the below guidelines for using the other components as template in tab.

- Declare a template within the function returns jsx element. If the template does not need arguments no need to pass the properties.
- Assign the function as value for the template property.

To quickly get started with integrating UI components inside the React Tabs component, you can check out this video:

#### INDEX.JSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { TabComponent, TabItemDirective, TabItemsDirective } from
 '@syncfusion/ej2-react-navigations';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { DatePickerComponent, CalendarComponent } from '@syncfusion/ej2-
react-calendars';
const App = () => {
  let headertext;
  const buttonTemplate = () => {
    return (<ButtonComponent>Click me</ButtonComponent>);
  }
  const datePickerTemplate = () => {
    return (<DatePickerComponent></DatePickerComponent>);
  }
  const calendarTemplate = () => {
    return (<CalendarComponent></CalendarComponent>);
  }
  headertext = [{ text: "ButtonComponent" }, { text: "DatePickerComponent"
}, { text: "CalendarComponent" }];
  return (<div id='container'>
    <TabComponent heightAdjustMode='Auto' id='tabelement'>
      <TabItemsDirective>
        <TabItemDirective header={headertext[0]} content={buttonTemplate}/>
        <TabItemDirective header={headertext[1]}
content={datePickerTemplate}/>
        <TabItemDirective header={headertext[2]}
content={calendarTemplate}/>
      </TabItemsDirective>
    </TabComponent>
  </div>);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);
```

#### INDEX.TSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
```

```
import { TabComponent, TabItemDirective, TabItemsDirective } from
 '@syncfusion/ej2-react-navigations';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { DatePickerComponent, CalendarComponent } from '@syncfusion/ej2-
react-calendars';
const App = () => {
  let headertext: any;
  const buttonTemplate = (): JSX.Element => {
    return (<ButtonComponent>Click me</ButtonComponent>);
  }
  const datePickerTemplate = (): JSX.Element => {
    return (<DatePickerComponent></DatePickerComponent>);
  }
  const calenderTemplate = (): JSX.Element => {
    return (<CalendarComponent ></CalendarComponent>);
  }
  headertext = [{ text: "ButtonComponent" }, { text: "DatePickerComponent" },
 { text: "CalendarComponent" }];
  return (
    <div id='container'>
      <TabComponent heightAdjustMode='Auto' id='tabelement'>
        <TabItemsDirective>
          <TabItemDirective header={headertext[0]}
            content={buttonTemplate} />
          <TabItemDirective header={headertext[1]}
            content={datePickerTemplate} />
          <TabItemDirective header={headertext[2]}
            content={calenderTemplate} />
        </TabItemsDirective>
      </TabComponent>
    </div>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);
```

## INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Tab</title>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta name="description" content="Essential JS 2 for React Components" />
  <meta name="author" content="Syncfusion" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/material.css"
rel="stylesheet" />
  <link
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
rel="stylesheet">
  <link href="index.css" rel="stylesheet" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
  <script src="systemjs.config.js"></script>
```

```

</head>
<body>
  <div id='container'>
    <div id='element'>
      <div id='loader'>Loading</div>
    </div>
  </div>
</body>
</html>

```

### Passing props

The following code example demonstrates how to use `props` when rendering other components in tab component.

#### INDEX.JSX

```

import { useRef, useState } from "react";
import * as React from "react";
import * as ReactDOM from "react-dom";
import { enableRipple } from "@syncfusion/ej2-base";
import {
  TabComponent,
  TabItemDirective,
  TabItemsDirective,
  ContextMenuComponent
} from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from "@syncfusion/ej2-react-buttons";
import {
  DatePickerComponent,
  CalendarComponent
} from "@syncfusion/ej2-react-calendars";
enableRipple(true);
const App = () => {
  const contextMenuRef = useRef(null);
  const [menuItems] = useState([
    { text: "Cut" },
    { text: "Copy" },
    { text: "Paste" }
  ]);
  const btnClick = () => {
    contextMenuRef.current.open(80, 20);
  }
  let headertext;
  const contextMenuTemplate = () => {
    return (
      <div>
        <ContextMenuComponent
          id="contextmenu"
          ref={contextMenuRef}
          items={menuItems}
        />
        <ButtonComponent
          onClick={btnClick}
        >Click me</ButtonComponent>
      </div>
    );
  }
  const datePickerTemplate = () => {
    return <DatePickerComponent />;
  }
  const calendarTemplate = () => {

```

```

    return <CalendarComponent />;
  }
  headertext = [{ text: "ContextMenuComponent" }, { text:
  "DatePickerComponent" }, { text: "CalendarComponent" }];
  return (
    <div id="container">
      <TabComponent heightAdjustMode="Auto" id="tabelement">
        <TabItemsDirective>
          <TabItemDirective
            header={headertext[0]}
            content={conTextTemplate}
          />
          <TabItemDirective
            header={headertext[1]}
            content={datePickerTemplate}
          />
          <TabItemDirective
            header={headertext[2]}
            content={calenderTemplate}
          />
        </TabItemsDirective>
      </TabComponent>
    </div>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

## INDEX.TSX

```

import { useRef, useState } from "react";
import * as React from "react";
import * as ReactDOM from "react-dom";
import { enableRipple } from "@syncfusion/ej2-base";
import {
  TabComponent,
  TabItemDirective,
  TabItemsDirective,
  ContextMenuComponent
} from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from "@syncfusion/ej2-react-buttons";
import {
  DatePickerComponent,
  CalendarComponent
} from "@syncfusion/ej2-react-calendars";
enableRipple(true);
const App = () => {
  const conTextMenuRef = useRef<ContextMenuComponent>(null);
  const [menuItems] = useState([
    { text: "Cut" },
    { text: "Copy" },
    { text: "Paste" }
  ]);
  const btnClick = () => {
    conTextMenuRef.current.open(80, 20);
  }
  let headertext;
  const conTextTemplate = () => {
    return (

```

```

        <div>
            <ContextMenuComponent
                id="contextmenu"
                ref={conTextMenuRef}
                items={menuItems}
            />
            <ButtonComponent onClick={btnClick}>Click me</ButtonComponent>
        </div>
    );
}
const datePickerTemplate = () => {
    return <DatePickerComponent />;
}
const calenderTemplate = () => {
    return <CalendarComponent />;
}
headertext = [{ text: "ContextMenuComponent" }, { text:
"DatePickerComponent" }, { text: "CalendarComponent" }];
return (
    <div id="container">
        <TabComponent heightAdjustMode="Auto" id="tabelement">
            <TabItemsDirective>
                <TabItemDirective
                    header={headertext[0]}
                    content={conTextTemplate}
                />
                <TabItemDirective
                    header={headertext[1]}
                    content={datePickerTemplate}
                />
                <TabItemDirective
                    header={headertext[2]}
                    content={calenderTemplate}
                />
            </TabItemsDirective>
        </TabComponent>
    </div>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

## INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Syncfusion React Tab</title>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta name="description" content="Essential JS 2 for React Components" />
    <meta name="author" content="Syncfusion" />
    <link href="https://cdn.syncfusion.com/ej2/20.3.56/material.css"
rel="stylesheet" />

```

```

    <link
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
rel="stylesheet">
    <link href="index.css" rel="stylesheet" />
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
    <script src="systemjs.config.js"></script>
</head>
<body>
    <div id='container'>
        <div id='element'>
            <div id='loader'>Loading</div>
        </div>
    </div>
</body>
</html>

```

### Adding dynamic items with content reuse in React Tab component

You can add dynamic tabs by reusing the content using React **template**. Tabs can be added dynamically by passing array of items and index value to the [addTab](#) method.

Content reuse can be achieved by using the following steps:

1. Declare a template within the function returns jsx element. If the template does not need arguments no need to pass the properties.
2. Assign the function as value for the template property.
3. Provide separate template declaration for each react component and pass content by dynamically adding tab. It will reuse the content of react component.

Refer to the following sample.

#### INDEX.JSX

```

import { useRef } from 'react';
import * as React from "react";
import * as ReactDOM from 'react-dom/client';
import { TabComponent, TabItemDirective, TabItemsDirective } from
'@syncfusion/ej2-react-navigations';
import { DatePickerComponent } from '@syncfusion/ej2-react-calendars';
import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
const App = () => {
    const tabObj = useRef(null);
    const Dropdown = (props) => {
        return (
            <div>
                <DropDownListComponent
                    dataSource={props.data}
                    placeholder="Select a game"
                />
            </div>
        );
    };
    const addButtonClicked = (e) => {
        const newTabItem = [

```

```

        {
            header: { text: 'DynamicTabItem' },
            content: thirdDropdownTemplate
        }
    ];
    tabObj.current.addTab(newTabItem, 1);
}
const removeButtonClicked = (e) => {
    tabObj.current.removeTab(1);
}
const firstDropdownTemplate = () => {
    let sportsData = [
        'Badminton',
        'Basketball',
        'Cricket',
        'Golf',
        'Hockey',
        'Rugby'
    ];
    return <Dropdown data={sportsData} />;
}
const secondDropdownTemplate = () => {
    let sportsData = [
        'Cricket',
        'Golf',
        'Hockey',
        'Rugby',
        'Badminton',
        'Basketball'
    ];
    return <Dropdown data={sportsData} />;
}
const thirdDropdownTemplate = () => {
    let sportsData = [
        'Rugby',
        'Badminton',
        'Basketball',
        'Cricket',
        'Golf',
        'Hockey'
    ];
    return <Dropdown data={sportsData} />;
}
const datePickerTemplate = () => {
    return (
        <div>
            <DatePickerComponent />
        </div>
    );
}
let tabItemsHeaderText = [
    { text: 'DatePicker' },
    { text: 'Dropdown' },
    { text: 'Reused Dropdown' }
];
return (
    <div id='container'>

```



```

    <button
      id="add"
      className="e-btn"
      onClick={addButtonClicked}
    >
      Click to add
    </button>
    <button
      id="remove"
      className="e-btn"
      onClick={removeButtonClicked}
    >
      Click to remove
    </button>
    <TabComponent
      id="tabElement"
      ref={tabObj}
    >
      <TabItemsDirective>
        <TabItemDirective
          header={tabItemsHeaderText[0]}
          content={datePickerTemplate}
        />
        <TabItemDirective
          header={tabItemsHeaderText[1]}
          content={firstDropDownTemplate}
        />
        <TabItemDirective
          header={tabItemsHeaderText[2]}
          content={secondDropDownTemplate}
        />
      </TabItemsDirective>
    </TabComponent>
  </div>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

## INDEX.TSX

```

import { useRef } from 'react';
import * as React from "react";
import * as ReactDOM from 'react-dom';
import { TabComponent, TabItemDirective, TabItemsDirective } from
 '@syncfusion/ej2-react-navigations';
import { DatePickerComponent } from '@syncfusion/ej2-react-calendars';
import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
const App = () => {
  const tabObj = useRef<TabComponent>(null);
  const Dropdown = (props: any) => {
    return (
      <div>
        <DropDownListComponent
          dataSource={props.data}
          placeholder="Select a game"

```

```

        />
    </div>
    );
}
const addButtonClicked = (e: any): void => {
    const newTabItem = [
        {
            header: { text: 'DynamicTabItem' },
            content: thirdDropdownTemplate
        }
    ];
    tabObj.current.addTab(newTabItem as any, 1);
}
const removeButtonClicked = (e: any): void => {
    tabObj.current.removeTab(1);
}
const firstDropdownTemplate = () => {
    let sportsData = [
        'Badminton',
        'Basketball',
        'Cricket',
        'Golf',
        'Hockey',
        'Rugby'
    ];
    return <Dropdown data={sportsData} />;
}
const secondDropdownTemplate = () => {
    let sportsData = [
        'Cricket',
        'Golf',
        'Hockey',
        'Rugby',
        'Badminton',
        'Basketball'
    ];
    return <Dropdown data={sportsData} />;
}
const thirdDropdownTemplate = () => {
    let sportsData = [
        'Rugby',
        'Badminton',
        'Basketball',
        'Cricket',
        'Golf',
        'Hockey'
    ];
    return <Dropdown data={sportsData} />;
}
const datePickerTemplate = () => {
    return (
        <div>
            <DatePickerComponent />
        </div>
    );
}
let tabItemsHeaderText: any = [

```

```

    { text: 'DatePicker' },
    { text: 'Dropdown' },
    { text: 'Reused Dropdown' }
  ];
  return (
    <div id='container'>
      <button
        id="add"
        className="e-btn"
        onClick={addButtonClicked}
      >
        Click to add
      </button>
      <button
        id="remove"
        className="e-btn"
        onClick={removeButtonClicked}
      >
        Click to remove
      </button>
      <TabComponent
        id="tabElement"
        ref={tabObj}
      >
        <TabItemsDirective>
          <TabItemDirective
            header={tabItemsHeaderText[0]}
            content={datePickerTemplate}
          />
          <TabItemDirective
            header={tabItemsHeaderText[1]}
            content={firstDropdownTemplate}
          />
          <TabItemDirective
            header={tabItemsHeaderText[2]}
            content={secondDropdownTemplate}
          />
        </TabItemsDirective>
      </TabComponent>
    </div>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

### Reorder active tab in React Tab component

We can able to prevent the changing of the active tab item on resizing the browser when overflow mode is popup by using the [reorderActiveTab](#) property. By default, the active Tab should be reordered when we click the tab items from the popup. If we set `false` to [reorderActiveTab](#) property the active tab item from the popup will not be reordered and an active item is highlighted inside the popup. The following code example depicts to prevent the reorder active tab item inside the popup.

#### INDEX.JSX

```
import * as React from 'react';
```

```

import * as ReactDOM from 'react-dom';
import {
  TabComponent,
  TabItemDirective,
  TabItemsDirective,
} from '@syncfusion/ej2-react-navigations';
const App = () => {
  let headertext;
  // Mapping Tab items Header property
  headertext = [
    { text: 'HTML' },
    { text: 'C-Sharp(C#)' },
    { text: 'Java' },
    { text: 'VB.NET' },
    { text: 'Xamarin' },
    { text: 'ASP.NET' },
    { text: 'ASP.NET MVC' },
    { text: 'JavaScript' },
    { text: 'React' },
    { text: 'Angular' },
    { text: 'Vue' },
    { text: 'Typescript' },
  ];
  const htmlContent = () => {
    return (<div>
      HyperText Markup Language, commonly referred to as HTML, is the
      standard markup
      language used to create web pages. Along with CSS, and JavaScript, HTML
      is a cornerstone
      technology, used by most websites to create visually engaging web
      pages, user interfaces
      for web applications, and user interfaces for many mobile applications.
      Web browsers
      can read HTML files and render them into visible or audible web pages.
      HTML describes
      the structure of a website semantically along with cues for
      presentation, making it a
      markup language, rather than a programming language.
    </div>
  );
}
const csharpContent = () => {
  return (<div> C# is intended to be a simple, modern, general-purpose,
  object-oriented
  programming language. Its development team is led by Anders Hejlsberg.
  The most recent
  version is C# 5.0, which was released on August 15, 2012</div>
  );
}
const javaContent = () => {
  return (<div> Java is a set of computer software and specifications
  developed by Sun Microsystems,
  later acquired by Oracle Corporation, that provides a system for
  developing application
  software and deploying it in a cross-mobile phones to platform
  computing environment. Java

```

```

        is used in a wide variety of computing platforms from embedded devices
        and enterprise servers
        and supercomputers. While less common, Java applets run in secure,
        sandboxed environments to
        provide many features of native applications and can be embedded in
        HTML pages.</div>
    );
}
const vbnetContent = () => {
    return (<div> The command-line compiler, VBC.EXE, is installed as part of
the freeware .NET
    Framework SDK. Mono also includes a command-line VB.NET compiler. The
most recent version
    is VB 2012, which was released on August 15, 2012.
    </div>
    );
}
const xamarinContent = () => {
    return (<div>Xamarin is a San Francisco, California based software
company created in May
    2011 by the engineers that created Mono, Mono for Android and MonoTouch
that are
    cross-platform implementations of the Common Language Infrastructure
(CLI) and Common
    Language Specifications (often called Microsoft .NET). With a C#-shared
codebase, developers
    can use Xamarin tools to write native Android, iOS, and Windows apps
with native user interfaces
    and share code across multiple platforms. Xamarin has over 1 million
developers in more
    than 120 countries around the World as of May 2015.</div>
    );
}
const aspContent = () => {
    return (<div>
        ASP.NET is an open-source server-side web application framework
        designed for web
        development to produce dynamic web pages. It was developed by Microsoft
        to allow programmers
        to build dynamic web sites, web applications and web services. It was
        first released in January
        2002 with version 1.0 of the .NET Framework, and is the successor to
        Microsoft's Active Server
        Pages (ASP) technology. ASP.NET is built on the Common Language Runtime
        (CLR), allowing
        programmers to write ASP.NET code using any supported .NET language.
        The ASP.NET SOAP extension
        framework allows ASP.NET components to process SOAP messages.'
    </div>
    );
}
const mvcContent = () => {
    return (<div>The ASP.NET MVC is a web application framework developed by
Microsoft, which implements
    the model-view-controller (MVC) pattern. It is open-source software,
    apart from the ASP.NET Web

```

```

    Forms component which is proprietary. In the later versions of ASP.NET,
    ASP.NET MVC, ASP.NET Web
    API, and ASP.NET Web Pages (a platform using only Razor pages) will
    merge into a unified MVC 6.
    The project is called ASP.NET vNext.</div>
  );
}
const jsContent = () => {
  return (<div>
    JavaScript (JS) is an interpreted computer programming language. It was
    originally
    implemented as part of web browsers so that client-side scripts could
    interact with the
    user, control the browser, communicate asynchronously, and alter the
    document content that
    was displayed. More recently, however, it has become common in both
    game development
    and the creation of desktop applications.
  </div>);
}
const reactContent = () => {
  return (<div>
    React is a free and open-source front-end JavaScript library for
    building user interfaces based on UI components. It is maintained by Meta and
    a community of individual developers and companies.
  </div>);
}
const angularContent = () => {
  return (<div>
    Angular is a TypeScript-based free and open-source web application
    framework led by the Angular Team at Google and by a community of individuals
    and corporations. Angular is a complete rewrite from the same team that built
    AngularJS
  </div>);
}
const vueContent = () => {
  return (<div>
    Vue.js is an open-source model-view-viewmodel front end JavaScript
    framework for building user interfaces and single-page applications. It was
    created by Evan You, and is maintained by him and the rest of the active core
    team members.
  </div>);
}
const tsContent = () => {
  return (<div>
    TypeScript is a programming language developed and maintained by
    Microsoft. It is a strict syntactical superset of JavaScript and adds
    optional static typing to the language. TypeScript is designed for the
    development of large applications and transcompiles to JavaScript.
  </div>);
}
return (
  <TabComponent
    heightAdjustMode="Auto"
    overflowMode="Popup"
    reorderActiveTab={false}
  >

```

```

        <TabItemsDirective>
            <TabItemDirective header={headertext[0]} content={htmlContent} />
            <TabItemDirective header={headertext[1]} content={csharpContent} />
            <TabItemDirective header={headertext[2]} content={javaContent} />
            <TabItemDirective header={headertext[3]} content={vbnetContent} />
            <TabItemDirective header={headertext[4]} content={xamarinContent} />
            <TabItemDirective header={headertext[5]} content={aspContent} />
            <TabItemDirective header={headertext[6]} content={mvcContent} />
            <TabItemDirective header={headertext[7]} content={jsContent} />
            <TabItemDirective header={headertext[8]} content={reactContent} />
            <TabItemDirective header={headertext[9]} content={angularContent} />
            <TabItemDirective header={headertext[10]} content={vueContent} />
            <TabItemDirective header={headertext[11]} content={tsContent} />
        </TabItemsDirective>
    </TabComponent>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

## INDEX.TSX

```

import * as React from 'react';
import * as ReactDOM from 'react-dom';
import {
    TabComponent,
    TabItemDirective,
    TabItemsDirective,
} from '@syncfusion/ej2-react-navigations';
const App = () => {
    let headertext: any;
    // Mapping Tab items Header property
    headertext = [
        { text: 'HTML' },
        { text: 'C-Sharp(C#)' },
        { text: 'Java' },
        { text: 'VB.NET' },
        { text: 'Xamarin' },
        { text: 'ASP.NET' },
        { text: 'ASP.NET MVC' },
        { text: 'JavaScript' },
        { text: 'React' },
        { text: 'Angular' },
        { text: 'Vue' },
        { text: 'Typescript' },
    ];
    const htmlContent = () => {
        return (
            <div>
                HyperText Markup Language, commonly referred to as HTML, is the
                standard markup
                language used to create web pages. Along with CSS, and JavaScript, HTML
                is a cornerstone
                technology, used by most websites to create visually engaging web
                pages, user interfaces
                for web applications, and user interfaces for many mobile applications.
                Web browsers
            </div>
        );
    };

```

```

        can read HTML files and render them into visible or audible web pages.
HTML describes
        the structure of a website semantically along with cues for
presentation, making it a
        markup language, rather than a programming language.
    </div>
    );
}
    const csharpContent = () => {
        return (<div> C# is intended to be a simple, modern, general-purpose,
object-oriented
        programming language. Its development team is led by Anders Hejlsberg.
The most recent
        version is C# 5.0, which was released on August 15, 2012</div>
    );
}
    const javaContent = () => {
        return (<div> Java is a set of computer software and specifications
developed by Sun Microsystems,
        later acquired by Oracle Corporation, that provides a system for
developing application
        software and deploying it in a cross-mobile phones to platform
computing environment. Java
        is used in a wide variety of computing platforms from embedded devices
and enterprise servers
        and supercomputers. While less common, Java applets run in secure,
sandboxed environments to
        provide many features of native applications and can be embedded in
HTML pages.</div>
    );
}
    const vbnetContent = () => {
        return (<div> The command-line compiler, VBC.EXE, is installed as part of
the freeware .NET
        Framework SDK. Mono also includes a command-line VB.NET compiler. The
most recent version
        is VB 2012, which was released on August 15, 2012.
        </div>
    );
}
    const xamarinContent = () => {
        return (<div>Xamarin is a San Francisco, California based software
company created in May
        2011 by the engineers that created Mono, Mono for Android and MonoTouch
that are
        cross-platform implementations of the Common Language Infrastructure
(CLI) and Common
        Language Specifications (often called Microsoft .NET). With a C#-shared
codebase, developers
        can use Xamarin tools to write native Android, iOS, and Windows apps
with native user interfaces
        and share code across multiple platforms. Xamarin has over 1 million
developers in more
        than 120 countries around the World as of May 2015.</div>
    );
}
    const aspContent = () => {

```



```

    return (<div>
        ASP.NET is an open-source server-side web application framework
        designed for web
        development to produce dynamic web pages. It was developed by Microsoft
        to allow programmers
        to build dynamic web sites, web applications and web services. It was
        first released in January
        2002 with version 1.0 of the .NET Framework, and is the successor to
        Microsoft's Active Server
        Pages (ASP) technology. ASP.NET is built on the Common Language Runtime
        (CLR), allowing
        programmers to write ASP.NET code using any supported .NET language.
        The ASP.NET SOAP extension
        framework allows ASP.NET components to process SOAP messages.'
    </div>
    );
}
const mvcContent = () => {
    return (<div>The ASP.NET MVC is a web application framework developed by
    Microsoft, which implements
    the model-view-controller (MVC) pattern. It is open-source software,
    apart from the ASP.NET Web
    Forms component which is proprietary. In the later versions of ASP.NET,
    ASP.NET MVC, ASP.NET Web
    API, and ASP.NET Web Pages (a platform using only Razor pages) will
    merge into a unified MVC 6.
    The project is called ASP.NET vNext.</div>
    );
}
const jsContent = () => {
    return (<div>
        JavaScript (JS) is an interpreted computer programming language. It was
        originally
        implemented as part of web browsers so that client-side scripts could
        interact with the
        user, control the browser, communicate asynchronously, and alter the
        document content that
        was displayed. More recently, however, it has become common in both
        game development
        and the creation of desktop applications.
    </div>);
}
const reactContent = () => {
    return (<div>
        React is a free and open-source front-end JavaScript library for
        building user interfaces based on UI components. It is maintained by Meta and
        a community of individual developers and companies.
    </div>);
}
const angularContent = () => {
    return (<div>
        Angular is a TypeScript-based free and open-source web application
        framework led by the Angular Team at Google and by a community of individuals
        and corporations. Angular is a complete rewrite from the same team that built
        AngularJS
    </div>);
}

```

```

const vueContent = () => {
  return (<div>
    Vue.js is an open-source model-view-viewmodel front end JavaScript
    framework for building user interfaces and single-page applications. It was
    created by Evan You, and is maintained by him and the rest of the active core
    team members.
  </div>);
}
const tsContent = () => {
  return (<div>
    TypeScript is a programming language developed and maintained by
    Microsoft. It is a strict syntactical superset of JavaScript and adds
    optional static typing to the language. TypeScript is designed for the
    development of large applications and transcompiles to JavaScript.
  </div>);
}
return (
  <TabComponent
    heightAdjustMode="Auto"
    overflowMode="Popup"
    reorderActiveTab={false}
  >
    <TabItemsDirective>
      <TabItemDirective header={headertext[0]} content={htmlContent} />
      <TabItemDirective header={headertext[1]} content={csharpContent} />
      <TabItemDirective header={headertext[2]} content={javaContent} />
      <TabItemDirective header={headertext[3]} content={vbnetContent} />
      <TabItemDirective header={headertext[4]} content={xamarinContent} />
      <TabItemDirective header={headertext[5]} content={aspContent} />
      <TabItemDirective header={headertext[6]} content={mvcContent} />
      <TabItemDirective header={headertext[7]} content={jsContent} />
      <TabItemDirective header={headertext[8]} content={reactContent} />
      <TabItemDirective header={headertext[9]} content={angularContent} />
      <TabItemDirective header={headertext[10]} content={vueContent} />
      <TabItemDirective header={headertext[11]} content={tsContent} />
    </TabItemsDirective>
  </TabComponent>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

## INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Tab</title>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta name="description" content="Essential JS 2 for React Components" />
  <meta name="author" content="Syncfusion" />
  <link href="index.css" rel="stylesheet" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/material.css"
rel="stylesheet" />

```

```

    <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
    <script src="systemjs.config.js"></script>
</head>
<body>
    <div id='container'>
        <div id='element'>
            <div id='loader'>Loading</div>
        </div>
    </div>
</body>
</html>

```

### Display tool tip on tab header in React Tab component

You can display tooltip for the tab component header using [beforeRender](#) event of React Tooltip component which can be viewed as hint texts on mouse hovers over tab.

#### INDEX.JSX

```

import { TooltipComponent } from '@syncfusion/ej2-react-popups';
import { TabComponent, TabItemDirective, TabItemsDirective } from
 '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
const ReactApp = () => {
    const [content, setContent] = React.useState('');
    const onBeforeRender = (args) => {
        if (args.target) {
            setContent(`<div>` + args.target.innerText + `</div>`);
        }
    }
    const headerText = [{ text: "Twitter" }, { text: "Facebook" }, { text:
"WhatsApp" }];
    const twitterContent = () => {
        return <div>
            Twitter is an online social networking service that enables users to
            send and read short 140-character messages called "tweets". Registered users
            can read and post tweets, but those who are unregistered can only read them.
            Users access Twitter through the website interface, SMS or mobile device app
            Twitter Inc. is based in San Francisco and has more than 25 offices around
            the world. Twitter was created in March 2006 by Jack Dorsey, Evan Williams,
            Biz Stone, and Noah Glass and launched in July 2006. The service rapidly
            gained worldwide popularity, with more than 100 million users posting 340
            million tweets a day in 2012. The service also handled 1.6 billion search
            queries per day.
        </div>;
    }
    const faceBookcontent = () => {
        return <div>
            Facebook is an online social networking service headquartered in Menlo
            Park, California. Its website was launched on February 4, 2004, by Mark
            Zuckerberg with his Harvard College roommates and fellow students Eduardo
            Saverin, Andrew McCollum, Dustin Moskovitz and Chris Hughes. The founders had
            initially limited the website membership to Harvard students, but later
            expanded it to colleges in the Boston area, the Ivy League, and Stanford

```

```

University. It gradually added support for students at various other
universities and later to high-school students.
    </div>;
}
const whatsappcontent = () => {
    return <div>
        WhatsApp Messenger is a proprietary cross-platform instant messaging
        client for smartphones that operates under a subscription business model. It
        uses the Internet to send text messages, images, video, user location and
        audio media messages to other users using standard cellular mobile numbers.
        As of February 2016, WhatsApp had a user base of up to one billion, [10]
        making it the most globally popular messaging application. WhatsApp Inc.,
        based in Mountain View, California, was acquired by Facebook Inc. on February
        19, 2014, for approximately US$19.3 billion.
    </div>;
}
return (
    <TooltipComponent id="template-tootip" content={content} target=".e-
toolbar-item" beforeRender={onBeforeRender} position={'TopCenter'}>
        <TabComponent heightAdjustMode='Auto'>
            <TabItemsDirective>
                <TabItemDirective header={headerText[0]} content={twitterContent}
            />
                <TabItemDirective header={headerText[1]} content={faceBookcontent}
            />
                <TabItemDirective header={headerText[2]} content={whatsappcontent}
            />
            </TabItemsDirective>
        </TabComponent>
    </TooltipComponent>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

## INDEX.TSX

```

import { TooltipComponent, TooltipEventArgs } from '@syncfusion/ej2-react-
popups';
import { TabComponent, TabItemDirective, TabItemsDirective } from
 '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
const ReactApp = () => {
    const [content, setContent] = React.useState('');
    const onBeforeRender = (args: TooltipEventArgs) => {
        if (args.target) {
            setContent(`<div>` + args.target.innerText + `</div>`);
        }
    }
    const headerText: any = [{ text: "Twitter" }, { text: "Facebook" }, { text:
    "WhatsApp" }];
    const twitterContent = () => {
        return <div>
            Twitter is an online social networking service that enables users to
            send and read short 140-character messages called "tweets". Registered users

```

can read and post tweets, but those who are unregistered can only read them. Users access Twitter through the website **interface**, SMS or mobile device app Twitter Inc. **is** based **in** San Francisco and has more than **25** offices around the world. Twitter was created **in** March **2006** by Jack Dorsey, Evan Williams, Biz Stone, and Noah Glass and launched **in** July **2006**. The service rapidly gained worldwide popularity, with more than **100** million users posting **340** million tweets a day **in 2012**. The service also handled **1.6** billion search queries per day.

```

    </div>;
  }
  const faceBookcontent = () => {
    return <div>
      Facebook is an online social networking service headquartered in Menlo
      Park, California. Its website was launched on February 4, 2004, by Mark
      Zuckerberg with his Harvard College roommates and fellow students Eduardo
      Saverin, Andrew McCollum, Dustin Moskovitz and Chris Hughes. The founders had
      initially limited the website membership to Harvard students, but later
      expanded it to colleges in the Boston area, the Ivy League, and Stanford
      University. It gradually added support for students at various other
      universities and later to high-school students.
    </div>;
  }
  const whatsappcontent = () => {
    return <div>
      WhatsApp Messenger is a proprietary cross-platform instant messaging
      client for smartphones that operates under a subscription business model. It
      uses the Internet to send text messages, images, video, user location and
      audio media messages to other users using standard cellular mobile numbers.
      As of February 2016, WhatsApp had a user base of up to one billion, [10]
      making it the most globally popular messaging application. WhatsApp Inc.,
      based in Mountain View, California, was acquired by Facebook Inc. on February
      19, 2014, for approximately US$19.3 billion.
    </div>;
  }
  return (
    <TooltipComponent id="template-tootip" content={content} target=".e-
toolbar-item" beforeRender={onBeforeRender} position={'TopCenter'}>
      <TabComponent heightAdjustMode='Auto'>
        <TabItemsDirective>
          <TabItemDirective header={headerText[0]} content={twitterContent}
/>
          <TabItemDirective header={headerText[1]} content={faceBookcontent}
/>
          <TabItemDirective header={headerText[2]} content={whatsappcontent}
/>
        </TabItemsDirective>
      </TabComponent>
    </TooltipComponent>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

## INDEX.HTML

```
<!DOCTYPE html>
```

```

<html lang="en">
<head>
  <title>Syncfusion React Tab</title>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta name="description" content="Essential JS 2 for React Components" />
  <meta name="author" content="Syncfusion" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/material.css"
rel="stylesheet" />
  <link
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
rel="stylesheet">
  <link href="index.css" rel="stylesheet" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
  <script src="systemjs.config.js"></script>
</head>
<body>
  <div id='container'>
    <div id='element'>
      <div id='loader'>Loading</div>
    </div>
  </div>
</body>
</html>

```

### Tab selection in React Tab component

We can able to find the tab selection whether it is selected by user interaction or programmatically way in the [selecting](#) and [selected](#) event argument with the field of `isInteracted`. When the user changes the tab through click actions it will return true otherwise, it will return false. The following code example depicts to find the tab selecting the state in selecting and selected events.

### INDEX.JSX

```

import * as React from "react";
import { useRef } from 'react';
import * as ReactDOM from 'react-dom';
import { TabComponent, TabItemDirective, TabItemsDirective } from
'@syncfusion/ej2-react-navigations';
import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
const App = () => {
  const tabInstance = useRef(null);
  const eventLogRef = useRef(null)
  let dropdownData = [
    { text: 'HTML', Id: 0 },
    { text: 'C-Sharp(C#)', Id: 1 },
    { text: 'Java', Id: 2 },
    { text: 'VB.NET', Id: 3 },
    { text: 'Xamarin', Id: 4 },
    { text: 'ASP.NET', Id: 5 },
    { text: 'ASP.NET MVC', Id: 6 },
    { text: 'JavaScript', Id: 7 }
  ];
  let fields = { text: 'text', value: 'Id' };

```

```

const htmlContent = () => {
    return (<div>
        HyperText Markup Language, commonly referred to as HTML, is the
standard markup
        language used to create web pages. Along with CSS, and JavaScript, HTML
is a cornerstone
        technology, used by most websites to create visually engaging web
pages, user interfaces
        for web applications, and user interfaces for many mobile applications.
Web browsers
        can read HTML files and render them into visible or audible web pages.
HTML describes
        the structure of a website semantically along with cues for
presentation, making it a
        markup language, rather than a programming language.
    </div>
    );
}
const csharpContent = () => {
    return (<div> C# is intended to be a simple, modern, general-purpose,
object-oriented
        programming language. Its development team is led by Anders Hejlsberg.
The most recent
        version is C# 5.0, which was released on August 15, 2012</div>
    );
}
const javaContent = () => {
    return (<div> Java is a set of computer software and specifications
developed by Sun Microsystems,
        later acquired by Oracle Corporation, that provides a system for
developing application
        software and deploying it in a cross-mobile phones to platform
computing environment. Java
        is used in a wide variety of computing platforms from embedded devices
and enterprise servers
        and supercomputers. While less common, Java applets run in secure,
sandboxed environments to
        provide many features of native applications and can be embedded in
HTML pages.</div>
    );
}
const vbnetContent = () => {
    return (<div> The command-line compiler, VBC.EXE, is installed as part of
the freeware .NET
        Framework SDK. Mono also includes a command-line VB.NET compiler. The
most recent version
        is VB 2012, which was released on August 15, 2012.
    </div>
    );
}
const xamarinContent = () => {
    return (<div>Xamarin is a San Francisco, California based software
company created in May
        2011 by the engineers that created Mono, Mono for Android and MonoTouch
that are
        cross-platform implementations of the Common Language Infrastructure
(CLI) and Common

```

```

        Language Specifications (often called Microsoft .NET). With a C#-shared
        codebase, developers
        can use Xamarin tools to write native Android, iOS, and Windows apps
        with native user interfaces
        and share code across multiple platforms. Xamarin has over 1 million
        developers in more
        than 120 countries around the World as of May 2015.is an open-source server-side web application framework
        designed for web
        development to produce dynamic web pages. It was developed by Microsoft
        to allow programmers
        to build dynamic web sites, web applications and web services. It was
        first released in January
        2002 with version 1.0 of the .NET Framework, and is the successor to
        Microsoft's Active Server
        Pages (ASP) technology. ASP.NET is built on the Common Language Runtime
        (CLR), allowing
        programmers to write ASP.NET code using any supported .NET language.
        The ASP.NET SOAP extension
        framework allows ASP.NET components to process SOAP messages.'
    </div>
    );
}
const mvcContent = () => {
    return (<div>The ASP.NET MVC is a web application framework developed by
    Microsoft, which implements
    the model-view-controller (MVC) pattern. It is open-source software,
    apart from the ASP.NET Web
    Forms component which is proprietary. In the later versions of ASP.NET,
    ASP.NET MVC, ASP.NET Web
    API, and ASP.NET Web Pages (a platform using only Razor pages) will
    merge into a unified MVC 6.
    The project is called ASP.NET vNext.</div>
    );
}
const jsContent = () => {
    return (<div>
        JavaScript (JS) is an interpreted computer programming language. It was
        originally
        implemented as part of web browsers so that client-side scripts could
        interact with the
        user, control the browser, communicate asynchronously, and alter the
        document content that
        was displayed. More recently, however, it has become common in both
        game development
        and the creation of desktop applications.
    </div>);
}
const dropdownChange = (args) => {
    tabInstance.current.select(args.value)
}
const selecting = (args) => {
    getInteractionDetail(args.isInteracted);
}

```



```

    }
    const selected = (args) => {
        getInteractionDetail(args.isInteracted);
    }
    const getInteractionDetail = (interact) => {
        let eventlog = interact
            ? 'Tab Item selected by user interaction'
            : 'Tab Item selected by programmatically';
        eventLogRef.current.innerHTML = eventLogRef.current.innerHTML = '<b>' +
        eventlog + '</b></span>';
    }
    let headertext = [
        { text: 'HTML' },
        { text: 'C-Sharp(C#)' },
        { text: 'Java' },
        { text: 'VB.NET' },
        { text: 'Xamarin' },
        { text: 'ASP.NET' },
        { text: 'ASP.NET MVC' },
        { text: 'JavaScript' }
    ];
    return (
        <div>
            <div className="EventLog" ref={eventLogRef} ></div>
            <DropDownListComponent index={0} change={dropdownChange}
            dataSource={dropdownData} fields={fields} floatLabelType="Always"
            placeholder="Select Tab Item using dropdown" />
            <TabComponent heightAdjustMode='Auto' selecting={selecting}
            ref={tabInstance} selected={selected} >
                <TabItemsDirective>
                    <TabItemDirective header={headertext[0]} content={htmlContent} />
                    <TabItemDirective header={headertext[1]} content={csharpContent} />
                    <TabItemDirective header={headertext[2]} content={javaContent} />
                    <TabItemDirective header={headertext[3]} content={vbnetContent} />
                    <TabItemDirective header={headertext[4]} content={xamarinContent}
                />
                    <TabItemDirective header={headertext[5]} content={aspContent} />
                    <TabItemDirective header={headertext[6]} content={mvcContent} />
                    <TabItemDirective header={headertext[7]} content={jsContent} />
                </TabItemsDirective>
            </TabComponent>
        </div>
    );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

## INDEX.TSX

```

import { useRef } from 'react';
import * as React from "react";
import * as ReactDOM from 'react-dom';
import { TabComponent, TabItemDirective, TabItemsDirective,
SelectingEventArgs, SelectEventArgs } from '@syncfusion/ej2-react-
navigations';

```

```

import { ChangeEventArgs, DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
const App = () => {
  const tabInstance = useRef<TabComponent>(null);
  const eventLogRef = useRef(null)
  let dropdownData: { [key: string]: Object; }[] = [
    { text: 'HTML', Id: 0 },
    { text: 'C-Sharp(C#)', Id: 1 },
    { text: 'Java', Id: 2 },
    { text: 'VB.NET', Id: 3 },
    { text: 'Xamarin', Id: 4 },
    { text: 'ASP.NET', Id: 5 },
    { text: 'ASP.NET MVC', Id: 6 },
    { text: 'JavaScript', Id: 7 }
  ];
  let fields: Object = { text: 'text', value: 'Id' };
  const htmlContent = () => {
    return (<div>
      HyperText Markup Language, commonly referred to as HTML, is the
standard markup
      language used to create web pages. Along with CSS, and JavaScript, HTML
is a cornerstone
      technology, used by most websites to create visually engaging web
pages, user interfaces
      for web applications, and user interfaces for many mobile applications.
Web browsers
      can read HTML files and render them into visible or audible web pages.
HTML describes
      the structure of a website semantically along with cues for
presentation, making it a
      markup language, rather than a programming language.
    </div>
  );
}
  const csharpContent = () => {
    return (<div> C# is intended to be a simple, modern, general-purpose,
object-oriented
      programming language. Its development team is led by Anders Hejlsberg.
The most recent
      version is C# 5.0, which was released on August 15, 2012</div>
  );
}
  const javaContent = () => {
    return (<div> Java is a set of computer software and specifications
developed by Sun Microsystems,
      later acquired by Oracle Corporation, that provides a system for
developing application
      software and deploying it in a cross-mobile phones to platform
computing environment. Java
      is used in a wide variety of computing platforms from embedded devices
and enterprise servers
      and supercomputers. While less common, Java applets run in secure,
sandboxed environments to
      provide many features of native applications and can be embedded in
HTML pages.</div>
  );
}
}

```

```

const vbnetContent = () => {
    return (<div> The command-line compiler, VBC.EXE, is installed as part of
the freeware .NET
    Framework SDK. Mono also includes a command-line VB.NET compiler. The
most recent version
    is VB 2012, which was released on August 15, 2012.
    </div>
    );
}
const xamarinContent = () => {
    return (<div>Xamarin is a San Francisco, California based software
company created in May
    2011 by the engineers that created Mono, Mono for Android and MonoTouch
that are
    cross-platform implementations of the Common Language Infrastructure
(CLI) and Common
    Language Specifications (often called Microsoft .NET). With a C#-shared
codebase, developers
    can use Xamarin tools to write native Android, iOS, and Windows apps
with native user interfaces
    and share code across multiple platforms. Xamarin has over 1 million
developers in more
    than 120 countries around the World as of May 2015.</div>
    );
}
const aspContent = () => {
    return (<div>
    ASP.NET is an open-source server-side web application framework
designed for web
    development to produce dynamic web pages. It was developed by Microsoft
to allow programmers
    to build dynamic web sites, web applications and web services. It was
first released in January
    2002 with version 1.0 of the .NET Framework, and is the successor to
Microsoft's Active Server
    Pages (ASP) technology. ASP.NET is built on the Common Language Runtime
(CLR), allowing
    programmers to write ASP.NET code using any supported .NET language.
The ASP.NET SOAP extension
    framework allows ASP.NET components to process SOAP messages.'
    </div>
    );
}
const mvcContent = () => {
    return (<div>The ASP.NET MVC is a web application framework developed by
Microsoft, which implements
    the model-view-controller (MVC) pattern. It is open-source software,
apart from the ASP.NET Web
    Forms component which is proprietary. In the later versions of ASP.NET,
ASP.NET MVC, ASP.NET Web
    API, and ASP.NET Web Pages (a platform using only Razor pages) will
merge into a unified MVC 6.
    The project is called ASP.NET vNext.</div>
    );
}
const jsContent = () => {
    return (<div>

```

```

        JavaScript (JS) is an interpreted computer programming language. It was
        originally
        implemented as part of web browsers so that client-side scripts could
        interact with the
        user, control the browser, communicate asynchronously, and alter the
        document content that
        was displayed. More recently, however, it has become common in both
        game development
        and the creation of desktop applications.
    </div>);
}
const dropdownChange = (args: ChangeEventArgs): void => {
    tabInstance.current.select(args.value as number)
}
const selecting = (args: SelectingEventArgs) => {
    getInteractionDetail(args.isInteracted as boolean);
}
const selected = (args: SelectEventArgs) => {
    getInteractionDetail(args.isInteracted as boolean);
}
const getInteractionDetail = (interact: boolean): void => {
    let eventlog = interact
        ? 'Tab Item selected by user interaction'
        : 'Tab Item selected by programmatically';
    eventLogRef.current.innerHTML = eventLogRef.current.innerHTML = '<b>' +
eventlog + '</b></span>';
}
let headertext: any = [
    { text: 'HTML' },
    { text: 'C-Sharp(C#)' },
    { text: 'Java' },
    { text: 'VB.NET' },
    { text: 'Xamarin' },
    { text: 'ASP.NET' },
    { text: 'ASP.NET MVC' },
    { text: 'JavaScript' }
];
return (
    <div>
        <div className="EventLog" ref={eventLogRef} ></div>
        <DropDownListComponent index={0} change={dropdownChange}
dataSource={dropdownData} fields={fields} floatLabelType="Always"
placeholder="Select Tab Item using dropdown" />
        <TabComponent heightAdjustMode='Auto' selecting={selecting}
            ref={tabInstance} selected={selected} >
            <TabItemsDirective>
                <TabItemDirective header={headertext[0]} content={htmlContent} />
                <TabItemDirective header={headertext[1]} content={csharpContent} />
                <TabItemDirective header={headertext[2]} content={javaContent} />
                <TabItemDirective header={headertext[3]} content={vbnetContent} />
                <TabItemDirective header={headertext[4]} content={xamarinContent}
/>
                <TabItemDirective header={headertext[5]} content={aspContent} />
                <TabItemDirective header={headertext[6]} content={mvcContent} />
                <TabItemDirective header={headertext[7]} content={jsContent} />
            </TabItemsDirective>
        </TabComponent>
    </div>
);

```

```

    </div>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

## INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Tab</title>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta name="description" content="Essential JS 2 for React Components" />
  <meta name="author" content="Syncfusion" />
  <link href="index.css" rel="stylesheet" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/material.css"
rel="stylesheet" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
  <script src="systemjs.config.js"></script>
</head>
<body>
  <div id='container'>
    <div id='element'>
      <div id='loader'>Loading</div>
    </div>
  </div>
</body>
</html>

```

## React hooks tab in React Tab component

You can prevent the React JSX component reload issue on state changes by rendering the React JSX component as children of Tab component. In the below example, we have prevented the re-rendering of textbox component on state changes in Tab component.

## INDEX.JSX

```

import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { useState } from 'react';
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import { TabComponent } from '@syncfusion/ej2-react-navigations';
const App = () => {
  const [name, setName] = useState('');
  return (<div>
    <div className="control_wrapper" id="control_wrapper">
      <TabComponent>
        <div className="e-tab-header">
          <div> TextBox </div>
        </div>
        <div className="e-content">
          <div>

```

```

        <TextBoxComponent placeholder="Enter Name" value={name}
input={ (e) => setName(e.value) }></TextBoxComponent>
        <div>My name is {name}</div>
    </div>
</div>
</TabComponent>
</div>
</div>);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

## INDEX.TSX

```

import { useState } from 'react';
import * as React from 'react';
import * as ReactDOM from "react-dom";
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import { TabComponent } from '@syncfusion/ej2-react-navigations';
const App = () => {
    const [name, setName] = useState('');
    return (
        <div>
            <div className="control_wrapper" id="control_wrapper">
                <TabComponent>
                    <div className="e-tab-header">
                        <div> TextBox </div>
                    </div>
                    <div className="e-content">
                        <div>
                            <TextBoxComponent placeholder="Enter Name" value={name}
input={ (e) => setName(e.value) } ></TextBoxComponent>
                            <div>My name is {name}</div>
                        </div>
                    </div>
                </TabComponent>
            </div>
        </div>
    );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

## Tab key navigation in React Tab component

The [tabIndex](#) property of a Tab item is used to enable tab key navigation for that particular item. When a positive value is assigned to the [tabIndex](#) property, it allows the user to switch focus to the next or previous tab item using the Tab or Shift+Tab keys. By default, the user can only switch between tab items using the arrow keys.

If the [tabIndex](#) value is set to 0 for all tab items, the tab will switch based on the order of the elements on the page. This means that if the tab items are listed in a specific order on the page, the user will be able to navigate through them using the Tab key in that same order.

To use the [tabIndex](#) property, you can assign a positive value to the property of each tab item that you want to enable tab key navigation. For example:

### INDEX.JSX

```
import { TabComponent, TabItemDirective, TabItemsDirective } from
 '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
const ReactApp = () => {
  const headerText = [{ text: "HTML" }, { text: "C Sharp(C#)" }, { text:
    "Java" }, { text: "VB.Net" }, { text: "Xamarin" }, { text: "ASP.NET" }, {
    text: "ASP.NET MVC" }, { text: "JavaScript" }];
  const htmlContent = () => {
    return (<div>
      HyperText Markup Language, commonly referred to as HTML, is the
      standard markup
      language used to create web pages. Along with CSS, and JavaScript, HTML
      is a cornerstone
      technology, used by most websites to create visually engaging web
      pages, user interfaces
      for web applications, and user interfaces for many mobile applications.
      Web browsers
      can read HTML files and render them into visible or audible web pages.
      HTML describes
      the structure of a website semantically along with cues for
      presentation, making it a
      markup language, rather than a programming language.
    </div>
    );
  }
  const csharpContent = () => {
    return (<div> C# is intended to be a simple, modern, general-purpose,
    object-oriented
    programming language. Its development team is led by Anders Hejlsberg.
    The most recent
    version is C# 5.0, which was released on August 15, 2012</div>
    );
  }
  const javaContent = () => {
    return (<div> Java is a set of computer software and specifications
    developed by Sun Microsystems,
    later acquired by Oracle Corporation, that provides a system for
    developing application
    software and deploying it in a cross-mobile phones to platform
    computing environment. Java
    is used in a wide variety of computing platforms from embedded devices
    and enterprise servers
    and supercomputers. While less common, Java applets run in secure,
    sandboxed environments to
    provide many features of native applications and can be embedded in
    HTML pages.</div>
    );
  }
  const vbnetContent = () => {
    return (<div> The command-line compiler, VBC.EXE, is installed as part of
    the freeware .NET
```

```

Framework SDK. Mono also includes a command-line VB.NET compiler. The
most recent version
    is VB 2012, which was released on August 15, 2012.
</div>
);
}
const xamarinContent = () => {
    return (<div>Xamarin is a San Francisco, California based software
company created in May
    2011 by the engineers that created Mono, Mono for Android and MonoTouch
that are
    cross-platform implementations of the Common Language Infrastructure
(CLI) and Common
    Language Specifications (often called Microsoft .NET). With a C#-shared
codebase, developers
    can use Xamarin tools to write native Android, iOS, and Windows apps
with native user interfaces
    and share code across multiple platforms. Xamarin has over 1 million
developers in more
    than 120 countries around the World as of May 2015.</div>
);
}
const aspContent = () => {
    return (<div>
        ASP.NET is an open-source server-side web application framework
designed for web
        development to produce dynamic web pages. It was developed by Microsoft
to allow programmers
        to build dynamic web sites, web applications and web services. It was
first released in January
        2002 with version 1.0 of the .NET Framework, and is the successor to
Microsoft's Active Server
        Pages (ASP) technology. ASP.NET is built on the Common Language Runtime
(CLR), allowing
        programmers to write ASP.NET code using any supported .NET language.
The ASP.NET SOAP extension
        framework allows ASP.NET components to process SOAP messages.'
    </div>
);
}
const mvcContent = () => {
    return (<div>The ASP.NET MVC is a web application framework developed by
Microsoft, which implements
        the model-view-controller (MVC) pattern. It is open-source software,
apart from the ASP.NET Web
        Forms component which is proprietary. In the later versions of ASP.NET,
ASP.NET MVC, ASP.NET Web
        API, and ASP.NET Web Pages (a platform using only Razor pages) will
merge into a unified MVC 6.
        The project is called ASP.NET vNext.</div>
);
}
const jsContent = () => {
    return (<div>
        JavaScript (JS) is an interpreted computer programming language. It was
originally

```



```

        implemented as part of web browsers so that client-side scripts could
        interact with the
        user, control the browser, communicate asynchronously, and alter the
        document content that
        was displayed. More recently, however, it has become common in both
        game development
        and the creation of desktop applications.
    </div>);
}
return (
    <TabComponent heightAdjustMode='Auto' overflowMode='Scrollable'>
        <TabItemsDirective>
            <TabItemDirective header={headerText[0]} content={htmlContent}
tabIndex={0} />
            <TabItemDirective header={headerText[1]} content={csharpContent}
tabIndex={0} />
            <TabItemDirective header={headerText[2]} content={javaContent}
tabIndex={0} />
            <TabItemDirective header={headerText[3]} content={vbnetContent}
tabIndex={0} />
            <TabItemDirective header={headerText[4]} content={xamarinContent}
tabIndex={0} />
            <TabItemDirective header={headerText[5]} content={aspContent}
tabIndex={0} />
            <TabItemDirective header={headerText[6]} content={mvcContent}
tabIndex={0} />
            <TabItemDirective header={headerText[7]} content={jsContent}
tabIndex={0} />
        </TabItemsDirective>
    </TabComponent>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

## INDEX.TSX

```

import { TabComponent, TabItemDirective, TabItemsDirective } from
'@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
const ReactApp = () => {
    const headerText: any = [{ text: "HTML" }, { text: "C Sharp(C#)" }, { text:
"Java" }, { text: "VB.Net" }, { text: "Xamarin" }, { text: "ASP.NET" }, {
text: "ASP.NET MVC" }, { text: "JavaScript" }];
    const htmlContent = () => {
        return (<div>
            HyperText Markup Language, commonly referred to as HTML, is the
            standard markup
            language used to create web pages. Along with CSS, and JavaScript, HTML
is a cornerstone
            technology, used by most websites to create visually engaging web
            pages, user interfaces
            for web applications, and user interfaces for many mobile applications.
            Web browsers

```

```

        can read HTML files and render them into visible or audible web pages.
HTML describes
        the structure of a website semantically along with cues for
presentation, making it a
        markup language, rather than a programming language.
    </div>
    );
}
    const csharpContent = () => {
        return (<div> C# is intended to be a simple, modern, general-purpose,
object-oriented
        programming language. Its development team is led by Anders Hejlsberg.
The most recent
        version is C# 5.0, which was released on August 15, 2012</div>
    );
}
    const javaContent = () => {
        return (<div> Java is a set of computer software and specifications
developed by Sun Microsystems,
        later acquired by Oracle Corporation, that provides a system for
developing application
        software and deploying it in a cross-mobile phones to platform
computing environment. Java
        is used in a wide variety of computing platforms from embedded devices
and enterprise servers
        and supercomputers. While less common, Java applets run in secure,
sandboxed environments to
        provide many features of native applications and can be embedded in
HTML pages.</div>
    );
}
    const vbnetContent = () => {
        return (<div> The command-line compiler, VBC.EXE, is installed as part of
the freeware .NET
        Framework SDK. Mono also includes a command-line VB.NET compiler. The
most recent version
        is VB 2012, which was released on August 15, 2012.
        </div>
    );
}
    const xamarinContent = () => {
        return (<div>Xamarin is a San Francisco, California based software
company created in May
        2011 by the engineers that created Mono, Mono for Android and MonoTouch
that are
        cross-platform implementations of the Common Language Infrastructure
(CLI) and Common
        Language Specifications (often called Microsoft .NET). With a C#-shared
codebase, developers
        can use Xamarin tools to write native Android, iOS, and Windows apps
with native user interfaces
        and share code across multiple platforms. Xamarin has over 1 million
developers in more
        than 120 countries around the World as of May 2015.</div>
    );
}
    const aspContent = () => {

```

```

    return (<div>
        ASP.NET is an open-source server-side web application framework
        designed for web
        development to produce dynamic web pages. It was developed by Microsoft
        to allow programmers
        to build dynamic web sites, web applications and web services. It was
        first released in January
        2002 with version 1.0 of the .NET Framework, and is the successor to
        Microsoft's Active Server
        Pages (ASP) technology. ASP.NET is built on the Common Language Runtime
        (CLR), allowing
        programmers to write ASP.NET code using any supported .NET language.
        The ASP.NET SOAP extension
        framework allows ASP.NET components to process SOAP messages.'
    </div>
    );
}
const mvcContent = () => {
    return (<div>The ASP.NET MVC is a web application framework developed by
    Microsoft, which implements
    the model-view-controller (MVC) pattern. It is open-source software,
    apart from the ASP.NET Web
    Forms component which is proprietary. In the later versions of ASP.NET,
    ASP.NET MVC, ASP.NET Web
    API, and ASP.NET Web Pages (a platform using only Razor pages) will
    merge into a unified MVC 6.
    The project is called ASP.NET vNext.</div>
    );
}
const jsContent = () => {
    return (<div>
        JavaScript (JS) is an interpreted computer programming language. It was
        originally
        implemented as part of web browsers so that client-side scripts could
        interact with the
        user, control the browser, communicate asynchronously, and alter the
        document content that
        was displayed. More recently, however, it has become common in both
        game development
        and the creation of desktop applications.
    </div>);
}
return (
    <TabComponent heightAdjustMode='Auto' overflowMode='Scrollable'>
        <TabItemsDirective>
            <TabItemDirective header={headerText[0]} content={htmlContent}
            tabIndex={0} />
            <TabItemDirective header={headerText[1]} content={csharpContent}
            tabIndex={0} />
            <TabItemDirective header={headerText[2]} content={javaContent}
            tabIndex={0} />
            <TabItemDirective header={headerText[3]} content={vbnetContent}
            tabIndex={0} />
            <TabItemDirective header={headerText[4]} content={xamarinContent}
            tabIndex={0} />
            <TabItemDirective header={headerText[5]} content={aspContent}
            tabIndex={0} />

```

```

        <TabItemDirective header={headerText[6]} content={mvcContent}
        tabIndex={0} />
        <TabItemDirective header={headerText[7]} content={jsContent}
        tabIndex={0} />
      </TabItemsDirective>
    </TabComponent>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

## INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Tab</title>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta name="description" content="Essential JS 2 for React Components" />
  <meta name="author" content="Syncfusion" />
  <link href="index.css" rel="stylesheet" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/material.css"
  rel="stylesheet" />
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
  ipt>
  <script src="systemjs.config.js"></script>
</head>
<body>
  <div id='container'>
    <div id='element'>
      <div id='loader'>Loading</div>
    </div>
  </div>
</body>
</html>

```

With this code, the user will be able to switch between the tab items using the Tab and Shift+Tab keys, in the order specified by the [tabIndex](#) values.

It's important to note that the [tabIndex](#) property only affects the ability to navigate between tab items using the Tab key. The user will still be able to use the arrow keys to switch between tab items, regardless of the value of the [tabIndex](#) property.

## Ej1 api migration in React Tab component

This article describes the API migration process of Tab component from Essential JS 1 to Essential JS 2.

### Accessibility and Localization

<!-- markdownlint-disable MD033 -->

Behavior	Property in Essential JS 1	Property in Essential JS 2

--	--	--

| Keyboard Navigation | **Property** : allowKeyboardNavigation<br/> <EJ.Tab id="Tab"  
allowKeyboardNavigation={false} ></EJ.Tab> | <b>Not Applicable</b> |

| Localization | <b>Not Applicable</b> | **Property** : locale<br/> <TabComponent id="Tab"  
locale="fr-BE"></TabComponent > |

| Right to left | **Property**: enableRTL<br/> <EJ.Tab id="tab" enableRTL={true}></EJ.Tab> | **Property**:  
enableRTL<br/> <TabComponent id='tab' enableRTL={true}> </TabComponent > |

### AjaxSettings

<!-- markdownlint-disable MD033 -->

| **Behavior** | **Property in Essential JS 1** | **Property in Essential JS 2** |

| ----- | ----- | ----- |

| Default | **Property** : ajaxSettings<br/> <EJ.Tab id="tab" ajaxSettings.type="GET"></EJ.Tab> |  
<b>Not Applicable</b> |

| Asynchronous | **Property** : ajaxSettings.async <br/> <EJ.Tab id="tab"  
ajaxSettings.async={true}></EJ.Tab> | <b>Not Applicable</b> |

| Browser Cache | **Property**: ajaxSettings.cache<br/> <EJ.Tab id="tab"  
ajaxSettings.cache={false}></EJ.Tab> | <b>Not Applicable</b> |

| Request type | **Property** : ajaxSettings.contentType<br/> <EJ.Tab id="tab"  
ajaxSettings.contentType="html "></EJ.Tab> | <b>Not Applicable</b> |

| Data | **Property** : ajaxSettings.data<br/> <EJ.Tab id="tab" ajaxSettings.data={ }></EJ.Tab> |  
<b>Not Applicable</b> |

| Response type | **Property** : ajaxSettings.dataType <br/> <EJ.Tab id="tab"  
ajaxSettings.dataType="html"></EJ.Tab> | <b>Not Applicable</b> |

| HTTP request type | **Property**: ajaxSettings.type<br/> <EJ.Tab id="tab"  
ajaxSettings.type="GET"></EJ.Tab> | <b>Not Applicable</b> |

| AjaxBeforeLoad | **Event**: ajaxBeforeLoad<br/> <EJ.Tab id='tab'  
ajaxBeforeLoad={this.onajaxBeforeLoad}></EJ.Tab><br/><br/> onajaxBeforeLoad (event){ } |  
<b>Not Applicable</b> |

| AjaxError | **Event**: AjaxError<br/> <EJ.Tab id='tab' ajaxError={this.onajaxError}></EJ.Tab><br/>  
<br/> onajaxError (event){ } | <b>Not Applicable</b> |

| AjaxLoad | **Event**: ajaxLoad<br/> <EJ.Tab id='tab' ajaxLoad={this.onajaxLoad}></EJ.Tab><br/>  
<br/> onajaxLoad (event){ } | <b>Not Applicable</b> |

| AjaxSuccess | **Event**: ajaxSuccess<br/> <EJ.Tab id='tab' ajaxSuccess={this.onajaxSuccess}  
></EJ.Tab><br/> <br/> onajaxSuccess (event){ } | <b>Not Applicable</b> |

### Animation

<!-- markdownlint-disable MD033 -->

| **Behavior** | **Property in Essential JS 1** | **Property in Essential JS 2** |

| ----- | ----- | ----- |

| Default | **<b>Not Applicable</b>** | **Property** :animation<br/> <TabComponent id="tab" animation={this.animation}></TabComponent ><br/> <br/>constructor(props: {}) {<br/> this.animation = { prev: { }, next: { } };} |

| EnableAnimation | **Property** :animation<br/> <EJ.Tab id="tab" enableAnimation={false}></EJ.Tab><br/> | **<b>Not Applicable</b>**|

| Previous animation | **<b>Not Applicable</b>** | **Property** :animation.prev<br/> <TabComponent id='tab' animation={this.animation}></TabComponent ><br/><br/>constructor(props: {}) {<br/> this.animation = { prev: { duration: 400,easing: 'ease-in', effect: 'SlideLeft' } };} |

| Next animation | **<b>Not Applicable</b>** | **Property** :animation.next<br/> <TabComponent id='tab' animation={this.animation}></TabComponent ><br/><br/> constructor(props: {}) {<br/> this.animation = { next: { duration: 400, easing: 'ease-in', effect: 'SlideLeft' } };} |

| Duration [prev / next] | **<b>Not Applicable</b>** | **Property** :animation.next.duration<br/> <TabComponent id='tab' animation={this.animation}></TabComponent ><br/><br/>constructor(props: {}) {<br/> this.animation = { prev: { duration: 400 } };} |

| Easing [expand / collapse] | **<b>Not Applicable</b>** | **Property** :animation.next.easing<br/> <TabComponent id='tab' animation={this.animation}></TabComponent ><br/><br/>constructor(props: {}) {<br/> this.animation = { prev: { easing: 'ease-in' } };} |

| Effect [expand / collapse] | **<b>Not Applicable</b>** | **Property** :animation.next.effect<br/> <TabComponent id='tab' animation={this.animation}></TabComponent ><br/><br/>constructor(props: {}) {<br/> this.animation = { prev: { effect: 'SlideLeft' } };} |

## Header

<!-- markdownlint-disable MD033 -->

| **Behavior** | **Property in Essential JS 1** | **Property in Essential JS 2** |

| ----- | ----- | ----- |

| Header position | **Property** : headerPosition<br/> <EJ.Tab id="Tab" headerPosition="Bottom" ></EJ.Tab> | **Property** : headerPlacement<br/> <TabComponent id="Tab" headerPlacement="Bottom" ></TabComponent > |

| Header size | **Property** : headerSize<br/> <EJ.Tab id="Tab" headerSize="100px" ></EJ.Tab> | **<b>Not Applicable</b>** |

| OverflowModes | **<b>Not Applicable</b>** | **Property:** overflowMode<br/> <TabComponent id='tab' overflowMode='Popup' > </TabComponent > |

| TabScroll | **Property:** enableTabScroll<br/> <EJ.Tab id="Tab" enableTabScroll={false} ></EJ.Tab> | **<b>Not Applicable</b>** |

## Items

<!-- markdownlint-disable MD033 -->

| **Behavior** | **Property in Essential JS 1** | **Property in Essential JS 2** |

| ----- | ----- | ----- |

| Default | **<b>Not Applicable</b>** | **Property** : items<br/> <TabComponent id="tab"  
items={this.items}> </TabComponent > |

| Content | **<b>Not Applicable</b>** | **Property** : items[0].content <br/> <TabComponent id="tab"  
items={this.items}> </TabComponent ><br/> <br/> constructor(props: {}) {<br/> this.items =  
{content: 'Contents'}}|

| Custom class | **<b>Not Applicable</b>** | **Property** : items[0].cssClass <br/> <TabComponent  
id="tab" items={this.items}> </TabComponent ><br/> <br/> constructor(props: {}) {<br/> this.items  
= { cssClass: 'customClass'}}|

| Header | **<b>Not Applicable</b>** | **Property** : items[0].Header <br/> <TabComponent id="tab"  
items={this.items}> </TabComponent ><br/> <br/> constructor(props: {}) {<br/> this.items = {  
header: { } }|

| Icon class | **<b>Not Applicable</b>** | **Property** : items[0].header.iconCss <br/> <TabComponent  
id="tab" items={this.items}> </TabComponent ><br/> <br/> constructor(props: {}) {<br/> this.items  
= { header: { iconCss: 'e-icon' } }|

| Icon position | **<b>Not Applicable</b>** | **Property** : items[0].header.iconPosition <br/>  
<TabComponent id="tab" items={this.items}> </TabComponent ><br/> <br/> constructor(props:  
{}) {<br/> this.items = { header: { iconPosition: 'Left' } }|

| Header text | **<b>Not Applicable</b>** | **Property** : items[0].header.text <br/> <TabComponent  
id="tab" items={this.items}> </TabComponent ><br/> <br/> constructor(props: {}) {<br/> this.items  
= { header: { text: 'Tab1' } }|

| Get items length | **Method** : getItemCount() <br/> <EJ.Tab id="Tab"></EJ.Tab><br/> <br/> var  
obj=\$('#Tab').ejTab('instance')<br/>obj.getItemCount(); | **<b>Not Applicable</b>** |

| Add Items | **Method** : addItem(url, displayLabel, index, cssClass, id) <br/> <EJ.Tab  
id="Tab"></EJ.Tab><br/> <br/> var obj=\$('#Tab').ejTab('instance')<br/>obj.addItem("#new", "New  
Item", 3, "myClass", "newItem"); | **Method** : addTab(items, index) <br/> <TabComponent id="tab" ref  
= {(scope) => {this.TabObj = scope}}> </TabComponent ><br/> <br/> constructor(props: {}) {<br/>  
this.TabObj.addTab([{header: { text: 'Tab1' },content: 'contents' }], 1 ); }|

| BeforeAdd | **<b>Not Applicable</b>** | **Event**: adding<br/> <TabComponent id="tab"  
adding={onadding.bind(this)}> </TabComponent ><br/> <br/> onadding(event){ } |

| AfterAdd | **Event**: itemAdd<br/> <EJ.Tab id='tab' itemAdd={this.onitemAdd}></EJ.Tab><br/> <br/>  
onitemAdd(event){ } | **Event**: added <br/> <TabComponent id="tab"  
added={onadded.bind(this)}> </TabComponent ><br/> <br/> onadded(event){ } |

| Remove Item | **Method** : removeItem(index) <br/> <EJ.Tab id="Tab" ></EJ.Tab><br/> <br/> var  
obj=\$('#Tab').ejTab('instance')<br/>obj.removeItem(0); } | **Method** : addItem(items, index) <br/>  
<TabComponent id="tab" ref = {(scope) => {this.TabObj = scope}}> </TabComponent ><br/>  
<br/> constructor(props: {}) {<br/> this.TabObj.addItem([ header: 'App', content: 'text' ], 0);}|

| BeforeRemove | **Event**: beforeItemRemove <br/> <EJ.Tab id='tab'  
beforeItemRemove={this.onbeforeItemRemove}></EJ.Tab><br/> <br/>  
onbeforeItemRemove(event){ } | **Event**: removing <br/> <TabComponent id="tab"  
removing={onremoving.bind(this)}> </TabComponent ><br/> <br/> onremoving(event){ } |

|AfterRemove | **Event:** afterRemove <br/> <EJ.Tab id='tab'  
 itemRemove={this.onitemRemove}></EJ.Tab><br/> <br/> onitemRemove(event){ } | **Event:**  
 removed <br/> <TabComponent id="tab" removed={onremoved.bind(this)} ></TabComponent  
 ><br/> <br/> onremoved(event){ } |

|Select item | <b>Not Applicable</b> | **Method** :select(index)<br/> <TabComponent id="tab" ref =  
 {(scope) => {this.TabObj = scope}} > </TabComponent ><br/> <br/> constructor(props: {}) {<br/>  
 this.TabObj.select(1);}

| SelectedItemIndex | **Property** : selectedItemIndex <br/> <EJ.Tab id="tab" selectedItemIndex="1" >  
 </EJ.Tab> | **Property** : selectedItem <br/> <TabComponent id="tab" selectedItem="1" >  
 </TabComponent > |

|BeforeActive | **Event:** beforeActive<br/> <TabComponent id="tab"  
 beforeActive={this.onbeforeActive}></TabComponent ><br/> <br/> onbeforeActive(event){ } |  
**Event:** selecting<br/> <TabComponent id="tab"  
 selecting={onselecting.bind(this)}></TabComponent ><br/> <br/> onselecting(event){ } |

|AfterActive | **Event:** itemActive<br/> <TabComponent id="tab"  
 itemActive={this.onitemActive}></TabComponent ><br/> <br/> onitemActive(event){ } | **Event:**  
 selected<br/> <TabComponent id="tab" selected={onselected.bind(this)}></TabComponent  
 ><br/> <br/> onselected(event){ } |

|Disable items | **Property** : disabledItemIndex <br/> <EJ.Tab id="tab" disabledItemIndex="[1,2]" >  
 </EJ.Tab> | <b>Not Applicable</b> |

|Enable items | **Property** : enabledItemIndex <br/> <EJ.Tab id="tab" enabledItemIndex="[1,2]" >  
 </EJ.Tab> | <b>Not Applicable</b> |

|Enable/Disable item | <b>Not Applicable</b> | **Property** : items[0].disabled <br/> <TabComponent  
 id="tab" items={this.items} > </TabComponent ><br/> <br/> this.items = { disabled: true }; |

|Hide items | **Property** : hiddenItemIndex <br/> <EJ.Tab id="tab" hiddenItemIndex="[1,2]" >  
 </EJ.Tab> | <b>Not Applicable</b> |

|Hide item | **Method** : hideItem(index) <br/> <EJ.Tab id="Tab"></EJ.Tab><br/> <br/>var  
 obj=\$('#Tab').ejTab('instance')<br/>obj.hideItem(1); | **Method** :hideTab(index, true) <br/>  
 <TabComponent id="tab" ref = {(scope) => {this.TabObj = scope}}> </TabComponent ><br/>  
 <br/> constructor(props: {}) {<br/> this.TabObj.hideTab(1, true); }|

|Show item | **Method** : showItem(index)<br/> <EJ.Tab id="Tab" ></EJ.Tab><br/> <br/>var  
 obj=\$('#Tab').ejTab('instance')<br/>obj.showItem(1); | **Method** : hideTab(index, false) <br/>  
 <TabComponent id="tab" ref = {(scope) => {this.TabObj = scope}}> </TabComponent ><br/>  
 <br/> constructor(props: {}) {<br/> this.TabObj.hideTab(1, false); }|

### Common

<!-- markdownlint-disable MD033 -->

| **Behavior** | **Property in Essential JS 1** | **Property in Essential JS 2** |

| ----- | ----- | ----- |



| Collapse active item | **Property** : collapsible <br/> <EJ.Tab id="tab" collapsible={true}> </EJ.Tab> |  
 <b>Not Applicable</b> |

| Custom class | **Property** : cssClass <br/> <EJ.Tab id="tab" cssClass="customClass" > </EJ.Tab> |  
**Property** : cssClass <br/> <TabComponent id="tab" cssClass="customClass" > </TabComponent >  
 |

| Enabled | **Property** : enabled <br/> <EJ.Tab id="tab" enabled={false}> </EJ.Tab> | **Method** :  
 disable(false)<br/> <EJ.Tab id="Tab"></EJ.Tab><br/> <br/> constructor(props: {}) {<br/>  
 this.TabObj.disable(false); |

| Persistence | **Property** : enablePersistence <br/> <EJ.Tab id="tab" enablePersistence={false}> </EJ.Tab> | **Property** : enablePersistence <br/> <TabComponent id="tab"  
 enablePersistence={false}> </TabComponent > |

| Events | **Property** : events <br/> <EJ.Tab id="tab" events="click" > </EJ.Tab> | <b>Not  
 Applicable</b> |

| Height | **Property** : height <br/> <EJ.Tab id="Tab" height="100%" > </EJ.Tab> | **Property** : height  
 <br/> <TabComponent id="Tab" height="100%" > </TabComponent > |

| HeightAdjustMode | **Property** : heightAdjustMode <br/> <EJ.Tab id="Tab"  
 heightAdjustMode="Content" > </EJ.Tab> | **Property** : heightAdjustMode <br/> <TabComponent  
 id="Tab" heightAdjustMode="Content" > </TabComponent > |

| HtmlAttributes | **Property** : htmlAttributes <br/> <EJ.Tab id="Tab" [htmlAttributes]="attributes" >  
 </EJ.Tab><br/> <br/> this.attributes = {class: "my-class"}; | <b>Not Applicable</b> |

| ID prefix | **Property** : idPrefix <br/> <EJ.Tab id="Tab" [idPrefix]="EJ.Tab-" > </EJ.Tab> | <b>Not  
 Applicable</b> |

| ShowCloseButton | **Property** : showCloseButton <br/> <EJ.Tab id="Tab" showCloseButton={true}> </EJ.Tab> | **Property** : showCloseButton <br/> <TabComponent id="Tab"  
 {showCloseButton}="true" > </TabComponent > |

| showReloadIcon | **Property** : showReloadIcon <br/> <EJ.Tab id="Tab" showReloadIcon={true}> </EJ.Tab> | <b>Not Applicable</b> |

| ShowRoundedCorner | **Property** : showRoundedCorner <br/> <EJ.Tab id="Tab"  
 showRoundedCorner={true}> </EJ.Tab> | <b>Not Applicable</b> |

| Destroy | **Method** : destroy() <br/> <EJ.Tab id="Tab"></EJ.Tab><br/> <br/> var  
 obj=\$('#Tab').ejTab('instance')<br/>obj.destroy(); | **Method** : destroy() <br/> <TabComponent  
 id="tab"></TabComponent ><br/> <br/> constructor(props: {}) {<br/> this.TabObj.destroy(); |

| Disable Tab | **Method** : disable() <br/> <EJ.Tab id="Tab"></EJ.Tab><br/> <br/> var  
 obj=\$('#Tab').ejTab('instance')<br/>obj.disable(); | **Method** : disable(true) <br/> <TabComponent  
 id="Tab"></TabComponent ><br/> <br/> constructor(props: {}) {<br/> this.TabObj.disable(true); |

| Enable Tab | **Method** : enable() <br/> <EJ.Tab id="Tab"></EJ.Tab><br/> <br/> var  
 obj=\$('#Tab').ejTab('instance')<br/>obj.enable(); | **Method** : disable(false) <br/> <TabComponent  
 id="Tab"></TabComponent ><br/> <br/> constructor(props: {}) {<br/> this.TabObj.disable(false); |

```
| Show Tab | Method : show() <br/> <EJ.Tab id="Tab"></EJ.Tab><br/> <br/>var
obj=$('#Tab').ejTab('instance')<br/>obj.show(); | <b>Not Applicable</b> |

| Hide Tab | Method : hide() <br/> <EJ.Tab id="Tab"></EJ.Tab><br/> <br/>var
obj=$('#Tab').ejTab('instance')<br/>obj.hide(); | <b>Not Applicable</b> |

| Refresh | <b>Not Applicable</b> | Method : refresh() <br/> <TabComponent
id="tab"></TabComponent ><br/> <br/>constructor(props: {}) {<br/> this.TabObj.refresh(); }|

| Created | Event: create<br/> <EJ.Tab id="tab" create={this.onCreate}></EJ.Tab><br/> <br/>
oncreate(event) { } | Event: created<br/> <TabComponent id="tab"
created={oncreated.bind(this)}></TabComponent ><br/> <br/> oncreated(event) { } |

| Destroyed | Event: destroy<br/> <EJ.Tab id="tab" destroy={this.ondestroy}></EJ.Tab><br/> <br/>
ondestroy(event) { } | Event: destroyed<br/> <TabComponent id="tab"
destroyed={ondestroyed.bind(this)} ></TabComponent ><br/> <br/> ondestroyed(event) { } |
```

## TextArea

### Getting Started with React TextArea Component

This section briefly explains how to create a simple TextArea and demonstrate the basic functionalities of the TextArea component in an React environment.

#### Dependencies

The following list of dependencies are required to use the TextArea component in your application.

```
`javascript
|-- @syncfusion/ej2-react-inputs
|-- @syncfusion/ej2-react-base
|-- @syncfusion/ej2-inputs
|-- @syncfusion/ej2-base
`,`
```

#### Installation and configuration

- You can use [create-react-app](#) to setup the applications.

To install `create-react-app` run the following command.

```
`bash
npm install -g create-react-app
`,`
```

- Start a new project using create-react-app command as follows.

```
`bash
create-react-app quickstart --scripts-version=react-scripts-ts
```

cd quickstart

,

- To install TextArea component, use the following command.

```
`bash
```

```
npm install @syncfusion/ej2-react-inputs --save
```

,

- The above package installs [Input dependencies](#) which are required to render the TextArea component in React environment.
- The TextArea CSS files are available in the `ej2-react-inputs` package folder. This can be referenced in your application using the following code.

```
[src/App.css]
```

```
`css
```

```
@import "../node_modules/@syncfusion/ej2-base/styles/material.css";
```

```
@import "../node_modules/@syncfusion/ej2-react-inputs/styles/material.css";
```

,

The [Custom Resource Generator \(CRG\)](#) is an online web tool, which can be used to generate the custom script and styles for a set of specific components.

This web tool is useful to combine the required component scripts and styles in a single file.

#### Adding TextArea to the application

To include the TextArea component in your application import the `TextAreaComponent` from `ej2-react-inputs` package in `App.tsx`.

Add the TextArea component in application as shown in below code example.

```
`ts
```

```
{ / Import the TextArea. / }
```

```
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
```

```
import * as React from 'react';
```

```
import './App.css';
```

```
{ / To render TextArea. / }
```

```
function App() {
```

```
  return (
```

```
    <TextAreaComponent id='default'></TextAreaComponent>
```

```
  );
```

```
}
```

```
export default App;
```

,

### Running the application

Run the application in the browser using the following command:

,

```
npm start
```

,

The following example shows a basic TextArea component.

#### INDEX.JSX

```
// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {
    return (<div className='wrap'>
        <TextAreaComponent id='default'></TextAreaComponent>
    </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));
```

#### INDEX.TSX

```
// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {
    return (
        <div className='wrap'>
            <TextAreaComponent id='default'></TextAreaComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));
```

### Getting and setting values

To set the initial value of the TextArea component, you can utilize the [value](#) property. Here's how you can achieve it:

```
`ts
```

```
{/ Import the TextArea. /}
```

```
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
```

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{/ To render TextArea./}
function App() {
  return (
    <div className='wrap'>
      <TextAreaComponent id='default' value='Comments'></TextAreaComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />,document.getElementById('input-container'));
`
```

- Alternatively, you can set the value of the TextArea using a state variable.

```
`ts
{/ Import the TextArea. /}
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import { useState } from 'react';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{/ To render TextArea./}
function App() {
  // Declare state variable textValue and a function setTextValue to update it.
  const [textValue, setTextValue] = useState("comments");
  return (
    <div className='wrap'>
      {/ Render TextArea with value property set to textValue state variable. /}
      <TextAreaComponent id='default' value={textValue}></TextAreaComponent>
    </div>
  );
}
export default App;
```

```
ReactDOM.render(<App />,document.getElementById('input-container'));
```

- You can dynamically retrieve the value of the TextArea component using the state variable assigned to the `value` property of the TextArea component.

`ts

```
{/ Import the TextArea. /}
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import { useState } from 'react';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{/ To render TextArea./}
function App() {
  const [textValue, setTextValue] = useState("comments");
  function onButtonClick () {
    // Get the value of the TextArea using state variable.
    let value = textValue;
  }
  return (
    <div className='wrap'>
      <TextAreaComponent id='rating' value={textValue}></TextAreaComponent>
      <button id="valuebtn" onclick={onButtonClick.bind(this)}>Get Value</button>
    </div>
  );
}
export default App;
ReactDOM.render(<App />,document.getElementById('input-container'));
```

- You can retrieve the value of the TextArea by accessing it as an argument from the [change](#) event.

`ts

```
{/ Import the TextArea. /}
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
```

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{/ To render TextArea./}
function App() {
  // triggers when the content of TextArea has changed and gets focus-out.
  function change(args) {
    // Get the value of the TextArea from event arguments.
    let value = args.value;
  }
  return (
    <div className='wrap'>
      <TextAreaComponent id='default' value='Comments' change={change}></TextAreaComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />,document.getElementById('input-container'));
```

## Floating Label in React TextArea Component

The floating label functionality in the TextArea component allows the placeholder text to float above the TextArea while the user interacts with it, providing a more intuitive user experience. This feature can be achieved using the [floatLabelType](#) API, which offers various options for defining the floating behavior:

Type	Description
--	--
Auto	The label floats above the TextArea when it receives focus or input, returning to its initial position when the TextArea loses focus and contains no value.
Always	The label always remains floating above the TextArea, regardless of user interaction.
Never	The label never floats; it remains in its default position within the TextArea.

## INDEX.JSX

```
// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {
  return (
    <div className='wrap'>
      <TextAreaComponent id='default' placeholder='Enter your comments'
        floatLabelType='Auto'></TextAreaComponent>
    </div>
  );
}
```

```

        </div>);
    }
    export default App;
    ReactDOM.render(<App />, document.getElementById('input-container'));

```

### INDEX.TSX

```

// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {
    return (
        <div className='wrap'>
            <TextAreaComponent id='default' placeholder='Enter your comments'
floatLabelType='Auto'></TextAreaComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));

```

### Placeholder with localization

Localization library allows to localize the placeholder text of the TextArea to different cultures using the `locale` property.

### INDEX.JSX

```

// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {
    return (
        <div className='wrap'>
            <TextAreaComponent id="default" placeholder="veuillez inscrire
vos commentaires" locale="fr-BE" floatLabelType="Auto"></TextAreaComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));

```

### INDEX.TSX

```

// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {

```



```

    return (
      <div className='wrap'>
        <TextAreaComponent id="default" placeholder="veuillez inscrire vos commentaires" locale="fr-BE" floatLabelType="Auto"></TextAreaComponent>
      </div>
    );
  }
  export default App;
  ReactDOM.render(<App />, document.getElementById('input-container'));

```

To load translation object in an application use `load` function of `L10n` class.

In the below sample, `German` culture is loaded to the TextArea placeholder text.

#### INDEX.JSX

```

// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import { L10n } from '@syncfusion/ej2-base';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// Load culture for textarea
L10n.load({
  'de-DE': {
    'textarea': { 'placeholder': "Geben Sie Ihre Kommentare ein" }
  }
});
// To render TextArea.
function App() {
  return (
    <div className='wrap'>
      <TextAreaComponent id="default" locale="de-DE" floatLabelType="Auto"></TextAreaComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));

```

#### INDEX.TSX

```

// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import { L10n } from '@syncfusion/ej2-base';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// Load culture for textarea
L10n.load({
  'de-DE': {
    'textarea': { 'placeholder': "Geben Sie Ihre Kommentare ein" }
  }
});
// To render TextArea.
function App() {
  return (

```

```

        <div className='wrap'>
            <TextAreaComponent id="default" locale="de-DE"
floatLabelType="Auto"></TextAreaComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));
```

## Rows and Columns in React TextArea Component

Two essential attributes, **rows** and **columns**, play a pivotal role in customizing the TextArea's appearance and layout.

The **rows** attribute determines the initial visible number of lines within the TextArea, controlling its vertical size. Conversely, the **columns** attribute specifies the visible width of the TextArea in characters per line, determining its initial width.

- You can customize the TextArea component by setting the number of rows using the [rowCount](#) property and the number of columns using the [columnsCount](#) property. These properties allow precise control over the dimensions of the TextArea, ensuring it fits seamlessly within the layout of the application.

### INDEX.JSX

```

// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {
    return (
        <div className='wrap'>
            <div>
                <TextAreaComponent id="default1" placeholder="Enter your
comments" floatLabelType="Auto" rowCount="3"
columnsCount="35"></TextAreaComponent>
            </div>
            <div>
                <TextAreaComponent id="default2" placeholder="Enter your
comments" floatLabelType="Auto" rowCount="5"
columnsCount="40"></TextAreaComponent>
            </div>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));
```

### INDEX.TSX

```

// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
```

```
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {

    return (
        <div className='wrap'>
            <div>
                <TextAreaComponent id="default1" placeholder="Enter your
comments" floatLabelType="Auto" rowCount="3"
columnsCount="35"></TextAreaComponent>
            </div>
            <div>
                <TextAreaComponent id="default2" placeholder="Enter your
comments" floatLabelType="Auto" rowCount="5"
columnsCount="40"></TextAreaComponent>
            </div>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));
```

## Resize in React TextArea Component

The TextArea allows users to input and edit large amounts of text. Resizing this control effectively can enhance the user experience and accommodate varying content needs. This resizing behavior can be enabled and configured using the [resizeMode](#) API, which offers several options for resizing the TextArea:

Type	Description
Vertical	Allows users to adjust the height of the TextArea vertically. It is suitable when users want to resize the TextArea only along the vertical axis, accommodating varying amounts of text input.
Horizontal	Users can adjust the width of the TextArea horizontally. This option is helpful for accommodating longer lines of text without altering the height of the control.
Both	Allows users to adjust both the height and width of the TextArea, offering maximum flexibility in resizing. It is ideal for situations where users need precise control over the dimensions of the TextArea.
None	Disables the resizing in the TextArea. This option is ideal for situations where maintaining a consistent layout is critical, and resizing by users is unnecessary.

### INDEX.JSX

```
// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {

    return (
        <div className='wrap'>
            <TextAreaComponent id="default" placeholder="Enter your comments"
resizeMode="Both"></TextAreaComponent>
        </div>
    );
}
```

```

        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));

```

## INDEX.TSX

```

{% raw %}
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from "react-dom";
export default class App extends React.Component<{}, {}> {
    public textareaObj: any;
    public render() {
        return (
            <div className="multiline">
                <TextBoxComponent multiline={true} value= 'Mr. Dodsworth Dodsworth,
System Analyst, Studio 103, The Business Center' input={this.onInput =
this.onInput.bind(this)} created={this.onCreate = this.onCreate.bind(this)}
placeholder='Enter your address' floatLabelType='Auto' ref = {scope =>
{this.textareaObj = scope }}/>
            </div>
        );
    }
    private onCreate(): void {
        this.textareaObj.addAttributes({rows: 1});
        this.textareaObj.respectiveElement.style.height = "auto";
        this.textareaObj.respectiveElement.style.height =
(this.textareaObj.respectiveElement.scrollHeight)+"px";
    }
    private onInput(): void {
        this.textareaObj.respectiveElement.style.height = "auto";
        this.textareaObj.respectiveElement.style.height =
(this.textareaObj.respectiveElement.scrollHeight)+"px";
    }
}
ReactDOM.render(<App />, document.getElementById('default'));
{% endraw %}

```

## Width of react TextArea component

You can easily customize the width of the TextArea using the [width](#) property. This property allows precise adjustment of the TextArea's width according to the specific layout requirements of the application.

## INDEX.JSX

```

// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {
    return (
        <div className='wrap'>

```

```

        <TextAreaComponent id="default" placeholder="Enter your comments"
        resizeMode="Both" width="500"></TextAreaComponent>
      </div>
    );
  }
  export default App;
  ReactDOM.render(<App />, document.getElementById('input-container'));

```

## INDEX.TSX

```

{% raw %}
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from "react-dom";
function App() {
  let textareaObj: any;
  return (
    <div className="multiline">
      <TextBoxComponent multiline={true} value= 'Mr. Dodsworth Dodsworth,
System Analyst, Studio 103, The Business Center' input={onInput =
onInput.bind(this)} created={onCreate = onCreate.bind(this)}
placeholder='Enter your address' floatLabelType='Auto' ref = {scope =>
{textareaObj = scope }}/>
    </div>
  );
  function onCreate(): void {
    textareaObj.addAttributes({rows: 1});
    textareaObj.respectiveElement.style.height = "auto";
    textareaObj.respectiveElement.style.height =
(textareaObj.respectiveElement.scrollHeight)+"px";
  }
  function onInput(): void {
    textareaObj.respectiveElement.style.height = "auto";
    textareaObj.respectiveElement.style.height =
(textareaObj.respectiveElement.scrollHeight)+"px";
  }
}
ReactDOM.render(<App />, document.getElementById('default'));
{% endraw %}

```

## Maximum Length in React TextArea Component

You can enforce a maximum length limit for the text input in the TextArea using the [maxLength](#) property. This property allows to define the maximum number of characters that users can input into the TextArea.

- By setting the **maxLength** property, you can control the length of text input, preventing users from exceeding a specified character limit.

## INDEX.JSX

```

// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';

```

```
// To render TextArea.
function App() {
  return (
    <div className='wrap'>
      <TextAreaComponent id="default" placeholder="Enter your comments"
        maxLength="20"></TextAreaComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));
```

## INDEX.TSX

```
// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {

  return (
    <div className='wrap'>
      <TextAreaComponent id="default" placeholder="Enter your comments"
        maxLength="20"></TextAreaComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));
```

When the user reaches the specified limit, the TextArea prevents further input, ensuring compliance with the defined character limit. This feature helps maintain data integrity and provides users with clear feedback on the allowed input length.

## Form Support in React TextArea Component

The TextArea component seamlessly integrates with HTML forms, enabling efficient submission of longer text data. By including TextArea inputs within HTML forms, users can conveniently input multiline text content and submit it as part of form submissions.

This integration enhances the usability of forms, allowing users to provide detailed feedback, enter lengthy descriptions, or input other multiline text data seamlessly.

## INDEX.JSX

```
// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {
  return (
    <div className='wrap'>
      <br />
    </div>
  );
}
```

```

        <TextAreaComponent id='default' name="myTextarea"
placeholder='Enter your comments' floatLabelType='Auto' required =
{true}></TextAreaComponent>
      </div>
    );
  }
  export default App;
  ReactDOM.render(<App />, document.getElementById('input-container'));

```

## INDEX.TSX

```

// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {
  return (
    <div className='wrap'>
      <br />
      <TextAreaComponent id='default' name="myTextarea"
placeholder='Enter your comments' floatLabelType='Auto' required =
{true}></TextAreaComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));

```

## Integration of react TextArea component with FormValidator component

TextArea component seamlessly integrates with the **FormValidator** component, allowing users to incorporate textarea inputs into form validation processes efficiently.

By integrating TextArea components with the **FormValidator** component, users can enforce validation rules specific to text inputs, such as required fields, minimum and maximum length constraints, pattern matching, and more. This ensures that user-submitted text data meets specified criteria and maintains data integrity.

## INDEX.JSX

```

// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import { FormValidator } from '@syncfusion/ej2-inputs';
import { useEffect } from 'react';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
let formObject;
// To render TextArea.
function App() {
  useEffect(() => {
    const options = {
      // validation rules
      rules: {
        email: {

```

```

        required: [true, '* Please enter valid email'],
      },
      comments: {
        required: [true, '* Please enter your comments'],
      }
    },
  };
  // Initialize the form validator
  formObject = new FormValidator('#form1', options);
}, []);
return (
  <div>
    <div className="control_wrapper" id="control_wrapper">
      <h3 className="form-title">Feedback</h3>
      <div className="control_wrapper textarea-form">
        <form id="form1" method="post">
          <div className="form-group">
            <div className="e-float-input">
              <label>Email</label>
              <input type="email" id="email" name="Email"
data-email-message="Please enter valid email address."
data-required-message="Required field"
required data-msg-containerid="emailError"/>
              <span className="e-float-line"></span>
            </div>
            <div id="emailError"></div>
          </div>
          <div className="form-group">
            <div>
              <label>Comments</label>
              <TextAreaComponent id='default'
name="myTextArea" data-msg-containerid="commentError" placeholder='Enter your
comments' floatLabelType='Auto' required = {true}></TextAreaComponent>
            </div>
            <div id="commentError"></div>
          </div>
          <div className="row">
            <div style={{ float: 'left' }}>
              <button className="btn"
type="submit">Submit</button>
            </div>
            <div style={{ float: 'left' }}>
              <button className="btn"
type="reset">Reset</button>
            </div>
          </div>
        </form>
      </div>
    </div>
    <br />
    <br />
  </div>
</div>
);
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));
```



# INDEX.TSX

```
// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import { FormValidator } from '@syncfusion/ej2-inputs';
import { useEffect } from 'react';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
let formObject;
// To render TextArea.
function App() {
  useEffect(() => {
    const options = {
      // validation rules
      rules: {
        email: {
          required: [true, '* Please enter valid email'],
        },
        comments: {
          required: [true, '* Please enter your comments']
        }
      },
    };
    // Initialize the form validator
    formObject = new FormValidator('#form1', options);
  }, []);

  return (
    <div>
      <div className="control_wrapper" id="control_wrapper">
        <h3 className="form-title">Feedback</h3>
        <div className="control_wrapper textarea-form">
          <form id="form1" method="post">
            <div className="form-group">
              <div className="e-float-input">
                <label>Email</label>
                <input type="email" id="email" name="email"
data-email-message="Please enter valid email address."
data-required-message="Required field"
required data-msg-containerid="emailError"/>
                <span className="e-float-line"></span>
              </div>
              <div id="emailError"></div>
            </div>
            <div className="form-group">
              <div>
                <label>Comments</label>
                <TextAreaComponent id='default'
name="comments" data-msg-containerid="commentError" placeholder='Enter your
comments' floatLabelType='Auto' required = {true}></TextAreaComponent>
              </div>
              <div id="commentError"></div>
            </div>
            <div className="row">
              <div style={{ float: 'left' }}>

```

```

                                <button className="btn"
type="submit">Submit</button>
                                </div>
                                <div style={{ float: 'left' }}>
                                <button className="btn"
type="reset">Reset</button>
                                </div>
                            </div>
                        </form>
                    </div>
                <br />
                <br />
            </div>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));

```

## Sizing in React TextArea Component

you can adjust the size of the TextArea by applying specific classes:

| Property | Description |

| -- | -- |

| Small | Add the **e-small** class to the input element or its container to render a smaller-sized TextArea.

| Bigger | Add the **e-bigger** class to the input element or its container to render a larger-sized TextArea.

By applying these classes, users can easily customize the appearance of the TextArea to better fit their application's design requirements.

### INDEX.JSX

```

import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {
    return (
        <div className='wrap'>
            <h4> Small Size </h4>
            <div className="e-input-group e-small">
                <textarea className="e-input" placeholder="Enter your
comments" onFocus={floatFocus} onBlur={floatBlur}></textarea>
            </div>
            <h4> Bigger Size </h4>
            <div className="e-input-group e-bigger">
                <textarea className="e-input" placeholder="Enter your
comments" onFocus={floatFocus} onBlur={floatBlur}></textarea>
            </div>
        </div>
    );
}
function floatFocus(args) {

```

```

    args.target.parentElement.classList.add("e-input-focus");
  }
  function floatBlur(args) {
    args.target.parentElement.classList.remove('e-input-focus');
  }
  export default App;
  ReactDOM.render(<App />, document.getElementById('input-container'));

```

## INDEX.TSX

```

import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {

  return (
    <div className='wrap'>
      <h4> Small Size </h4>
      <div className="e-input-group e-small">
        <textarea className="e-input" placeholder="Enter your
comments" onFocus={floatFocus} onBlur={floatBlur}></textarea>
      </div>
      <h4> Bigger Size </h4>
      <div className="e-input-group e-bigger">
        <textarea className="e-input" placeholder="Enter your
comments" onFocus={floatFocus} onBlur={floatBlur}></textarea>
      </div>
    </div>
  );
}
function floatFocus(args: React.FocusEvent) {
  args.target.parentElement.classList.add("e-input-focus");
}
function floatBlur(args: React.FocusEvent) {
  args.target.parentElement.classList.remove('e-input-focus');
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));

```

## Filled and outline mode

The Filled and Outline modes can be enabled in the TextArea component by adding the `e-outline` or `e-filled` class to the `cssClass` API.

By adding these classes, users can choose between a filled or outline appearance for the TextArea component, aligning with the design aesthetics of their application.

## INDEX.JSX

```

// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {
  return (

```

```

        <div className='wrap'>
          <label className="custom-input-label"> Filled TextArea </label>
          <div id='container'>
            <TextAreaComponent id='filled' placeholder="Filled"
floatLabelType="Auto" cssClass="e-filled"></TextAreaComponent>
          </div>
          <label className="custom-input-label"> Outlined TextArea </label>
          <div id='container1'>
            <TextAreaComponent id='outlined' placeholder="Outlined"
floatLabelType="Auto" cssClass="e-outline"></TextAreaComponent>
          </div>
        </div>
      );
    }
    export default App;
    ReactDOM.render(<App />, document.getElementById('input-container'));
```

## INDEX.TSX

```

// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {

  return (
    <div className='wrap'>
      <label className="custom-input-label"> Filled TextArea </label>
      <div id='container'>
        <TextAreaComponent id='filled' placeholder="Filled"
floatLabelType="Auto" cssClass="e-filled"></TextAreaComponent>
      </div>
      <label className="custom-input-label"> Outlined TextArea </label>
      <div id='container1'>
        <TextAreaComponent id='outlined' placeholder="Outlined"
floatLabelType="Auto" cssClass="e-outline"></TextAreaComponent>
      </div>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));
```

Note: Filled and Outline theme customization are available only with Material themes.

### Custom styling with cssClass API in TextArea

The `cssClass` API provides a powerful way to apply custom styling to the TextArea component, allowing users to customize its appearance and layout according to their design requirements.

By utilizing the `cssClass` API, users can apply custom CSS classes to the TextArea component's container, enabling control over its styling properties such as color, padding, margins, borders, and more.

## INDEX.JSX

```
// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {
    return (
        <div className='wrap'>
            <TextAreaComponent id='default' placeholder="Enter your comments"
floatLabelType="Auto" cssClass="custom-textarea"></TextAreaComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));
```

## INDEX.TSX

```
// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {

    return (
        <div className='wrap'>
            <TextAreaComponent id='default' placeholder="Enter your comments"
floatLabelType="Auto" cssClass="custom-textarea"></TextAreaComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));
```

## Setting the disabled state in TextArea

To disable the TextArea, you can utilize the [enabled](#) property. When set to **false**, the TextArea becomes disabled, preventing user interaction.

`ts

*{/ Import the TextArea. /}*

import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';

import \* as React from 'react';

import \* as ReactDOM from 'react-dom';

*{/ To render TextArea./}*

function App() {

return (

<div className='wrap'>

```
<TextAreaComponent id="default" enabled={false}></TextAreaComponent>
</div>
);
}
export default App;
ReactDOM.render(<App />,document.getElementById('input-container'));
`ts
```

### Set the readonly TextArea

To make the TextArea read-only , you can use the [readonly](#) property. When set to `true`, it prevents users from editing the content of the TextArea.

```
`ts
{/ Import the TextArea. /}
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{/ To render TextArea./}
function App() {
  return (
    <div className='wrap'>
      <TextAreaComponent id="default" readonly={true}></TextAreaComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />,document.getElementById('input-container'));
`
```

### Set the rounded corner in TextArea

Render the TextArea with `rounded corner` by adding the `e-corner` class to the input parent element.

This rounded corner is visible only in box model input component

```
`ts
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{/ To render TextArea./}
function App() {
```

```
return (
  <div class="e-input-group e-corner">
    <textarea class="e-input" placeholder="Enter your comments"></textarea>
  </div>
);
}

export default App;

ReactDOM.render(<App />,document.getElementById('input-container'));
```

### Static clear button in TextArea

To display a static clear button in the TextArea component, you can add the `e-static-clear` class to the `cssClass` property. This class ensures that the clear button remains visible at all times, providing users with the ability to easily clear the TextArea content without needing to focus on the control.

#### INDEX.JSX

```
// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {
  return (
    <div className='wrap'>
      <TextAreaComponent id='default' placeholder="Enter your comments"
        cssClass="e-static-clear" showClearButton={true}></TextAreaComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));
```

#### INDEX.TSX

```
// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {
  return (
    <div className='wrap'>
      <TextAreaComponent id='default' placeholder="Enter your comments"
        cssClass="e-static-clear" showClearButton={true}></TextAreaComponent>
    </div>
  );
}
export default App;
```

```
ReactDOM.render(<App />, document.getElementById('input-container'));
```

### Customize the TextArea background color and text color

You can customize the TextArea styles such as background-color, text-color and border-color by overriding its default styles to achieve the desired appearance for the TextArea.

#### INDEX.JSX

```
// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {
    return (
        <div className='wrap'>
            <TextAreaComponent id='default' placeholder="Enter your comments"
floatLabelType="Auto"></TextAreaComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));
```

#### INDEX.TSX

```
// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {
    return (
        <div className='wrap'>
            <TextAreaComponent id='default' placeholder="Enter your comments"
floatLabelType="Auto"></TextAreaComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));
```

### Change the floating label color of the TextArea

You can change the floating label color of the TextArea for both **success** and **warning** validation states by applying the following CSS styles.

```
`css
```

```
/ For Success state /
```

```
.e-float-input.e-success label.e-float-text,
```

```
.e-float-input.e-success input:focus ~ label.e-float-text,
```



```
.e-float-input.e-success input:valid ~ label.e-float-text {
color: #22b24b;
}
/ For Warning state /
.e-float-input.e-warning label.e-float-text,
.e-float-input.e-warning input:focus ~ label.e-float-text,
.e-float-input.e-warning input:valid ~ label.e-float-text {
color: #ffca1c;
}
`
```

### INDEX.JSX

```
// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {
  return (
    <div className='wrap'>
      <div>
        <TextAreaComponent id='default1' placeholder="Success"
floatLabelType="Auto" cssClass="e-success"></TextAreaComponent>
      </div>
      <div>
        <TextAreaComponent id='default2' placeholder="Warning"
floatLabelType="Auto" cssClass="e-warning"></TextAreaComponent>
      </div>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));
```

### INDEX.TSX

```
// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {
  return (
    <div className='wrap'>
      <div>
        <TextAreaComponent id='default1' placeholder="Success"
floatLabelType="Auto" cssClass="e-success"></TextAreaComponent>
      </div>
    </div>
  );
}
```

```

        <div>
          <TextAreaComponent id='default2' placeholder="Warning"
floatLabelType="Auto" cssClass="e-warning"></TextAreaComponent>
        </div>
      </div>
    );
  }
  export default App;
  ReactDOM.render(<App />, document.getElementById('input-container'));

```

### Adding mandatory asterisk to placeholder

To add a mandatory asterisk (\*) to the placeholder in the TextArea component, you can utilize CSS to append the asterisk after the placeholder text.

`css

*/ To add asterick to float label in textarea /*

*.e-float-input.e-control-wrapper .e-float-text::after {*

*content: '\*/ Add asterisk after the placeholder \*/*

*color: red; / Customize asterisk color /*

*}*

*,*

### INDEX.JSX

```

// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {
  return (
    <div className='wrap'>
      <TextAreaComponent id='default' placeholder="Enter your comments"
floatLabelType="Auto"></TextAreaComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));

```

### INDEX.TSX

```

// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {
  return (
    <div className='wrap'>

```

```

        <TextAreaComponent id='default' placeholder="Enter your comments"
floatLabelType="Auto"></TextAreaComponent>
      </div>
    );
  }
  export default App;
  ReactDOM.render(<App />, document.getElementById('input-container'));

```

## Events in React TextArea Component

This section describes the TextArea events that will be triggered when appropriate actions are performed. The following events are available in the TextArea component.

### Created event

The TextArea component triggers the [created](#) event when the TextArea component is created. This event provides users with an opportunity to perform actions immediately after the TextArea has been created and initialized.

```

`ts
{ / Import the TextArea. /}

import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';

{ / To render TextArea./}

function App() {
  function created(){
    //Your required action here
  }
  return (
    <div className='wrap'>
    <TextAreaComponent id="default" created={ created }></TextAreaComponent>
    </div>
  );
}

export default App;

ReactDOM.render(<App />,document.getElementById('input-container'));
`

```

### Input event

The TextArea component triggers the [input](#) each time when the value of TextArea has changed. This event provides users with an opportunity to perform actions in response to real-time changes in the TextArea's content.

The [InputEventArgs](#) passed as an event argument provides the details about the input event in the TextArea.

```
`ts
{/ Import the TextArea. /}
import { TextAreaComponent, InputEventArgs } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{/ To render TextArea./}
function App() {
function inputHandler(args: InputEventArgs){
//Your required action here
}
return (
<div className='wrap'>
<TextAreaComponent id="default" input={ inputHandler }></TextAreaComponent>
</div>
);
}
export default App;
ReactDOM.render(<App />,document.getElementById('input-container'));
```

### Change event

The TextArea component triggers the [change](#) event when the content of TextArea has changed and gets focus-out. This event provides users with an opportunity to execute specific actions in response to changes made by the user.

The [ChangedEventArgs](#) passed as an event argument provides the details about the changes in the TextArea's value.

```
`ts
{/ Import the TextArea. /}
import { TextAreaComponent, ChangedEventArgs } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{/ To render TextArea./}
function App() {
function changeHandler(args: ChangedEventArgs){
```

```
//Your required action here
}
return (
<div className='wrap'>
<TextAreaComponent id="default" change={ changeHandler }></TextAreaComponent>
</div>
);
}
export default App;
ReactDOM.render(<App />,document.getElementById('input-container'));
```

### Focus event

The TextArea component triggers the [focus](#) when the TextArea gains focus. This event allows developers to execute specific actions when the user interacts with the TextArea by focusing on it.

The [FocusInEventArgs](#) passed as an argument provides details about the focus event in the TextArea.

```
`ts
{/ Import the TextArea. /}
import { TextAreaComponent, FocusInEventArgs } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{/ To render TextArea./}
function App() {
function focusHandler(args: FocusInEventArgs){
//Your required action here
}
return (
<div className='wrap'>
<TextAreaComponent id="default" focus={ focusHandler }></TextAreaComponent>
</div>
);
}
export default App;
ReactDOM.render(<App />,document.getElementById('input-container'));
```

### Blur event

The TextArea component triggers the [blur](#) when the TextArea loses focus. This event allows users to execute specific actions when the user interacts with the TextArea by moving focus away from it.

The [FocusOutEventArgs](#) passed as an argument provides details about the blur event in the TextArea.

```
`ts
{/ Import the TextArea. /}
import { TextAreaComponent, FocusOutEventArgs } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{/ To render TextArea./}
function App() {
  function blurHandler(args: FocusOutEventArgs){
    //Your required action here
  }
  return (
    <div className='wrap'>
      <TextAreaComponent id="default" blur={ blurHandler }></TextAreaComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />,document.getElementById('input-container'));
`
```

### Destroyed event

The TextArea component triggers the [destroyed](#) when the TextArea component is destroyed.

```
`ts
{/ Import the TextArea. /}
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
{/ To render TextArea./}
function App() {
  function destroyed(){
    //Your required action here
  }
}
```

```

}
return (
  <div className='wrap'>
    <TextAreaComponent id="default" destroyed={ destroyed }></TextAreaComponent>
  </div>
);
}
export default App;
ReactDOM.render(<App />,document.getElementById('input-container'));
`

```

## Methods in React TextArea Component

This section outlines the methods available for interacting with the TextArea component.

### FocusIn method

The [focusIn](#) method in the TextArea, is used to set focus to the textarea element, enabling user interaction.

By calling the `focusIn` method, you can programmatically set focus to the TextArea component, allowing users to interact with it via keyboard input or other means.

### INDEX.JSX

```

// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {
  let textareaInstance;
  function onClick() {
    // use focusIn method to focus TextArea
    textareaInstance.focusIn();
  }
  return (
    <div className='wrap'>
      <TextAreaComponent id='default' ref={(scope) => {
        textareaInstance = scope }}></TextAreaComponent>
      <br/>
      <button id="button" onClick={onClick}>Focus-in</button>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));

```

### INDEX.TSX

```

// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';

```

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {
  let textareaInstance;
  function onClick() {
    // use focusIn method to focus TextArea
    textareaInstance.focusIn();
  }
  return (
    <div className='wrap'>
      <TextAreaComponent id='default' ref={(scope) => {
        textareaInstance = scope }}></TextAreaComponent>
      <br/>
      <button id="button" onClick={onClick}>Focus-in</button>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));
```

### FocusOut method

The [focusOut](#) method in the TextArea component is used to remove focus from the textarea element, ending user interaction.

This method is beneficial for scenarios where user need to programmatically remove focus from the TextArea component, such as after completing a specific task or when navigating to another element in the application.

### INDEX.JSX

```
// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {
  let textareaInstance;
  function onClick() {
    // use focusOut method to remove focus from the TextArea
    textareaInstance.focusOut();
  }
  return (
    <div className='wrap'>
      <TextAreaComponent id='default' ref={(scope) => {
        textareaInstance = scope }}></TextAreaComponent>
      <br/>
      <button id="button" onClick={onClick}>Focus-Out</button>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));
```

### INDEX.TSX



```
// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {
  let textareaInstance;
  function onClick() {
    // use focusOut method to remove focus from the TextArea
    textareaInstance.focusOut();
  }
  return (
    <div className='wrap'>
      <TextAreaComponent id='default' ref={ (scope) => {
        textareaInstance = scope } }></TextAreaComponent>
      <br/>
      <button id="button" onClick={onClick}>Focus-Out</button>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));
```

### GetPersistData method

The [getPersistData](#) method in the TextArea component retrieves the properties that need to be maintained in the persisted state.

This method returns an object containing the properties to be persisted, which can include various configuration options and state information of the TextArea component.

### INDEX.JSX

```
// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {
  let textareaInstance;
  function onClick() {
    // use getPersistData method in TextArea
    textareaInstance.getPersistData();
  }
  return (
    <div className='wrap'>
      <TextAreaComponent id='default' ref={ (scope) => {
        textareaInstance = scope } }></TextAreaComponent>
      <br/>
      <button id="button" onClick={onClick}>Get Persist-
        data</button>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));
```

**INDEX.TSX**

```
// Import the TextArea.
import { TextAreaComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render TextArea.
function App() {
    let textareaInstance;
    function onButtonClick() {
        // use getPersistData method in TextArea
        textareaInstance.getPersistData();
    }
    return (
        <div className='wrap'>
            <TextAreaComponent id='default' ref={(scope) => {
                textareaInstance = scope }}></TextAreaComponent>
            <br/>
            <button id="button" onClick={onButtonClick}>Get Persist-
            data</button>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('input-container'));
```

## TextBox

### Getting Started

This section briefly explains about how to create a simple TextBox through CSS classes using **Create React App**.

#### Dependencies

The following list of dependencies are required to use the TextBox component in your application.

```
`javascript
|-- @syncfusion/ej2-react-inputs
|-- @syncfusion/ej2-react-base
|-- @syncfusion/ej2-inputs
|-- @syncfusion/ej2-base
\`
```

#### Installation and Configuration

- You can use [create-react-app](#) to setup the applications.

To install **create-react-app** run the following command.

```
`bash
npm install -g create-react-app
```

`

- Start a new project using create-react-app command as follows.

```
`bash
create-react-app quickstart --scripts-version=react-scripts-ts
cd quickstart
```

`

- To install TextBox component, use the following command.

```
`bash
npm install @syncfusion/ej2-react-inputs --save
```

`

- The above package installs [Input dependencies](#) which are required to render the TextBox component in React environment.
- The TextBox CSS files are available in the `ej2-react-inputs` package folder. This can be referenced in your application using the following code.

[src/App.css]

```
`css
@import "../node_modules/@syncfusion/ej2-base/styles/material.css";
@import "../node_modules/@syncfusion/ej2-react-inputs/styles/material.css";
```

`

The [Custom Resource Generator \(CRG\)](#) is an online web tool, which can be used to generate the custom script and styles for a set of specific components.

This web tool is useful to combine the required component scripts and styles in a single file.

#### [Adding TextBox to the application](#)

Return the `HTML input` element with `e-input` class within `render` method in `src/App.tsx` file to render the TextBox component as like below code.

[Class-component]

```
`ts
import * as React from "react";
import './App.css';
export default class App extends React.Component<{}, {}> {
  public render() {
    return (
```

```
// element which is going to render the TextBox
<input className="e-input" type="text" placeholder="Enter Name" />
);
}
};
`
```

#### [Functional-component]

```
`ts
import * as React from "react";
import './App.css';
function App() {
  return (
    // element which is going to render the TextBox
    <input className="e-input" type="text" placeholder="Enter Name" />
  );
};
ReactDOM.render(<App />, document.getElementById('input-container'));
`
```

#### Adding icons to the TextBox

You can create a TextBox with icon as a group by creating the parent div element with the class `e-input-group` and

add the icon element as span with the class `e-input-group-icon`. For detailed information, refer to the [Groups](#) section.

#### [Class-component]

```
`ts
import * as React from "react";
import * as ReactDOM from "react-dom";
export default class Default extends React.Component {
  public render() {
    return (
      <div className="e-input-group">
        <input className="e-input" name="input" type="text" onFocus = {this.onInputFocus} onBlur =
        {this.onInputBlur} placeholder="Enter Date"/>
      </div>
    );
  }
}
```

```

<span className="e-input-group-icon e-input-popup-date" onMouseDown = {this.onIconMouseDown}
onMouseUp = {this.onIconMouseUp}/>
</div>
);
}
public onInputFocus(args: React.FocusEvent) {
((args.target as HTMLElement).parentElement as HTMLElement).classList.add('e-input-focus');
}
public onInputBlur(args: React.FocusEvent) {
((args.target as HTMLElement).parentElement as HTMLElement).classList.remove('e-input-focus');
}
public onIconMouseDown(args: React.MouseEvent) {
args.persist();
setTimeout(
() => {
(args.target as HTMLElement).classList.add('e-input-btn-ripple');
},
300);
}
public onIconMouseUp(args: React.MouseEvent) {
(args.target as HTMLElement).classList.remove('e-input-btn-ripple');
}
}
ReactDOM.render(<Default />, document.getElementById('input-container'));
`

```

### [Functional-component]

```

`ts
import * as React from "react";
import * as ReactDOM from "react-dom";
function Default() {
return (
<div className="e-input-group">
<input className="e-input" name="input" type="text" onFocus = {onInputFocus} onBlur = {onInputBlur}
placeholder="Enter Date"/>

```

```

<span className="e-input-group-icon e-input-popup-date" onMouseDown = {onIconMouseDown}
onMouseUp = {onIconMouseUp}/>
</div>
);
};
function onInputFocus(args: React.FocusEvent) {
((args.target as HTMLElement).parentElement as HTMLElement).classList.add('e-input-focus');
}
function onInputBlur(args: React.FocusEvent) {
((args.target as HTMLElement).parentElement as HTMLElement).classList.remove('e-input-focus');
}
function onIconMouseDown(args: React.MouseEvent) {
args.persist();
setTimeout(
() => {
(args.target as HTMLElement).classList.add('e-input-btn-ripple');
},
300);
}
function onIconMouseUp(args: React.MouseEvent) {
(args.target as HTMLElement).classList.remove('e-input-btn-ripple');
}
ReactDOM.render(<Default />, document.getElementById('input-container'));
、
、

.e-input-group .e-input-group-icon.e-input-popup-date {
font-size:16px;
}
.e-input-group-icon.e-input-popup-date:before {
content: "\e901";
}
、

```

- Run the application in the browser using the following command.

npm start

Output will be as follows:

[Class-component]

### INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
export default class Default extends React.Component {
  render() {
    return (
      <div className="e-input-group">
        <input className="e-input" name="input" type="text"
onFocus={this.onInputFocus} onBlur={this.onInputBlur} placeholder="Enter
Date"/>
        <span className="e-icons e-input-group-icon e-input-popup-date"
onMouseDown={this.onIconMouseDown} onMouseUp={this.onIconMouseUp}/>
      </div>
    );
  }
  onInputFocus(args) {
    args.target.parentElement.classList.add('e-input-focus');
  }
  onInputBlur(args) {
    args.target.parentElement.classList.remove('e-input-focus');
  }
  onIconMouseDown(args) {
    args.persist();
    setTimeout(() => {
      args.target.classList.add('e-input-btn-ripple');
    }, 300);
  }
  onIconMouseUp(args) {
    args.target.classList.remove('e-input-btn-ripple');
  }
}
ReactDOM.render(<Default />, document.getElementById('input-container'));
```

### INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
export default class Default extends React.Component {
  public render() {
    return (
      <div className="e-input-group">
        <input className="e-input" name="input" type="text" onFocus =
{this.onInputFocus} onBlur = {this.onInputBlur} placeholder="Enter Date"/>
        <span className="e-icons e-input-group-icon e-input-popup-date"
onMouseDown = {this.onIconMouseDown} onMouseUp = {this.onIconMouseUp}/>
      </div>
    );
  }
  public onInputFocus(args: React.FocusEvent) {
```

```

        ((args.target as HTMLElement).parentElement as
HTMLElement).classList.add('e-input-focus');
    }
    public onInputBlur(args: React.FocusEvent) {
        ((args.target as HTMLElement).parentElement as
HTMLElement).classList.remove('e-input-focus');
    }
    public onIconMouseDown(args: React.MouseEvent) {
        args.persist();
        setTimeout(
            () => {
                (args.target as HTMLElement).classList.add('e-input-btn-ripple');
            },
            300);
    }
    public onIconMouseUp(args: React.MouseEvent) {
        (args.target as HTMLElement).classList.remove('e-input-btn-ripple');
    }
}
ReactDOM.render(<Default />, document.getElementById('input-container'));

```

## [Functional-component]

### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function Default() {
    return (<div className="e-input-group">
        <input className="e-input" name="input" type="text"
onFocus={onInputFocus} onBlur={onInputBlur} placeholder="Enter Date"/>
        <span className="e-icons e-input-group-icon e-input-popup-date"
onMouseDown={onIconMouseDown} onMouseUp={onIconMouseUp}/>
    </div>);
}
function onInputFocus(args) {
    args.target.parentElement.classList.add('e-input-focus');
}
function onInputBlur(args) {
    args.target.parentElement.classList.remove('e-input-focus');
}
function onIconMouseDown(args) {
    args.persist();
    setTimeout(() => {
        args.target.classList.add('e-input-btn-ripple');
    }, 300);
}
function onIconMouseUp(args) {
    args.target.classList.remove('e-input-btn-ripple');
}
ReactDOM.render(<Default />, document.getElementById('input-container'));

```

### INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";

```



```
function Default() {
  return (
    <div className="e-input-group">
      <input className="e-input" name="input" type="text" onFocus =
{onInputFocus} onBlur = {onInputBlur} placeholder="Enter Date"/>
      <span className="e-icons e-input-group-icon e-input-popup-date"
onMouseDown = {onIconMouseDown} onMouseUp = {onIconMouseUp}/>
    </div>
  );
}
function onInputFocus(args: React.FocusEvent) {
  ((args.target as HTMLElement).parentElement as
HTMLElement).classList.add('e-input-focus');
}
function onInputBlur(args: React.FocusEvent) {
  ((args.target as HTMLElement).parentElement as
HTMLElement).classList.remove('e-input-focus');
}
function onIconMouseDown(args: React.MouseEvent) {
  args.persist();
  setTimeout(
    () => {
      (args.target as HTMLElement).classList.add('e-input-btn-ripple');
    },
    300);
}
function onIconMouseUp(args: React.MouseEvent) {
  (args.target as HTMLElement).classList.remove('e-input-btn-ripple');
}
ReactDOM.render(<Default />, document.getElementById('input-container'));
```

## Floating Label

The floating label TextBox floats the label above the TextBox after focusing, or filled with value in the TextBox.

You can create the floating label TextBox by using the `floatLabelType` API.

[Class-component]

## INDEX.JSX

```
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from "react";
import * as ReactDOM from "react-dom";
export default class App extends React.Component {
  render() {
    return (
      <div className="App">
        <div className="textboxes">
          <h4>FloatLabelType as Auto</h4>
          <TextBoxComponent placeholder="First Name"
floatLabelType="Auto"/>
        </div>
        <div className="textboxes">
          <h4>FloatLabelType as Always</h4>
          <TextBoxComponent placeholder="Last Name"
floatLabelType="Always"/>
        </div>
      </div>
    );
  }
}
```

```

        </div>
    </div>);
}
;
ReactDOM.render(<App />, document.getElementById('input-container'));

```

**INDEX.TSX**

```

import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from "react";
import * as ReactDOM from "react-dom";
export default class App extends React.Component<{}, {}> {
    public render() {
        return (
            <div className="App">
                <div className="textboxes">
                    <h4>FloatLabelType as Auto</h4>
                    <TextBoxComponent placeholder="First Name"
floatLabelType="Auto"/>
                </div>
                <div className="textboxes">
                    <h4>FloatLabelType as Always</h4>
                    <TextBoxComponent placeholder="Last Name"
floatLabelType="Always"/>
                </div>
            </div>
        )
    }
};
ReactDOM.render(<App />, document.getElementById('input-container'));

```

**[Functional-component]****INDEX.JSX**

```

import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<div className="App">
        <div className="textboxes">
            <h4>FloatLabelType as Auto</h4>
            <TextBoxComponent placeholder="First Name"
floatLabelType="Auto"/>
        </div>
        <div className="textboxes">
            <h4>FloatLabelType as Always</h4>
            <TextBoxComponent placeholder="Last Name"
floatLabelType="Always"/>
        </div>
    </div>);
}
;
ReactDOM.render(<App />, document.getElementById('input-container'));

```

## INDEX.TSX

```
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from "react";
import * as ReactDOM from "react-dom";

function App() {
  return (
    <div className="App">
      <div className="textboxes">
        <h4>FloatLabelType as Auto</h4>
        <TextBoxComponent placeholder="First Name"
floatLabelType="Auto"/>
      </div>
      <div className="textboxes">
        <h4>FloatLabelType as Always</h4>
        <TextBoxComponent placeholder="Last Name"
floatLabelType="Always"/>
      </div>
    </div>
  );
};

ReactDOM.render(<App />, document.getElementById('input-container'));
```

You can refer to our [React TextBox](#) feature tour page for its groundbreaking feature representations. You can also explore our [React TextBox example](#) to know how to render and configure the textbox.

See Also

- [How to render TextBox programmatically](#)
- [How to add floating label to TextBox programmatically](#)

## Groups in React Textbox component

The following section explains you the steps required to create TextBox with **icon** and **floating label**.

TextBox:

- Create a parent div element with the class **e-input-group**
- Place input element with the class **e-input** inside the parent div element.

[Class-component]

`ts

```
import * as React from "react";
import './App.css';
export default class App extends React.Component<{}, {}> {
  public render() {
    return (
      // element which is going to render the TextBox
      <input className="e-input" type="text" placeholder="Enter Name" />
    );
  }
};
```

```
);
}
};
`
```

#### [Functional-component]

```
`ts
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
  return (
    // element which is going to render the TextBox
    <input className="e-input" type="text" placeholder="Enter Name" />
  );
}
ReactDOM.render(<App />, document.getElementById('input-container'));
```

Floating label:

- Add the `e-float-input` class to the parent div element.
- Remove the `e-input` class and add `required` attribute to the input element.
- Place the span element with class `e-float-line` after the input element.
- Place the label element with class `e-float-text` after the above created span element.

When you focus or filled with value in the TextBox, the label floats above the TextBox.

Creating the Floating label TextBox, you have to set the `required` attribute to the Input element to achieve the floating label functionality which is used for validating the value existence in TextBox.

If you want to render the Floating label TextBox without `required` attribute then refer to the [Floating Label without required attribute](#) section.

#### [Class-component]

```
`ts
import * as React from "react";
import './App.css';
export default class App extends React.Component<{}, {}> {
  public render() {
    return (
```

```
// element which is going to render the Floating TextBox
<div className="e-float-input e-input-group">
  <input type="text" required={true}/>
  <span className="e-float-line"/>
  <label className="e-float-text">Enter Name </label>
</div>
);
}
};
`
```

### [Functional-component]

```
`ts
import * as React from "react";
import * as ReactDOM from 'react-dom';
import './App.css';
function App() {
  return (
    // element which is going to render the Floating TextBox
    <div className="e-float-input e-input-group">
      <input type="text" required={true}/>
      <span className="e-float-line"/>
      <label className="e-float-text">Enter Name </label>
    </div>
  );
}

ReactDOM.render(<App />, document.getElementById('input-container'));
```

And refer to the following sections to add the icons to the TextBox.

### With icon and floating label

Create an icon element as a span with the class `e-input-group-icon`, and the user can place the icon in either side of TextBox by adding the created icon element before/after the input.

For the floating label enabled TextBox add the icon element as first or last element inside the TextBox wrapper, and based on the element position it will act as prefix or suffix icon.

### [Class-component]

**INDEX.JSX**

```

import * as React from "react";
import * as ReactDOM from "react-dom";
export default class Default extends React.Component {
  render() {
    return (
      <div>
        <h4> TextBox with icons </h4>
        <div className="e-input-group ">
          <input className="e-input" type="text" onFocus={this.onInputFocus}
onBlur={this.onInputBlur} placeholder="Enter Date"/>
          <span className="e-input-group-icon e-input-popup-date"
onMouseDown={this.onIconMouseDown} onMouseUp={this.onIconMouseUp}/>
        </div>
        <div className="e-input-group e-float-icon-left">
          <span className="e-input-group-icon e-input-date"
onMouseDown={this.onIconMouseDown} onMouseUp={this.onIconMouseUp}/>
          <div className="e-input-in-wrap">
            <input className="e-input" type="text"
onFocus={this.onInputFocus} onBlur={this.onInputBlur} placeholder="Enter
Date"/>
          </div>
        </div>
        <div className="e-input-group e-float-icon-left">
          <span className="e-input-group-icon e-input-date"
onMouseDown={this.onIconMouseDown} onMouseUp={this.onIconMouseUp}/>
          <div className="e-input-in-wrap">
            <input className="e-input" type="text"
onFocus={this.onInputFocus} onBlur={this.onInputBlur} placeholder="Enter
Date"/>
            <span className="e-input-group-icon e-input-down"/>
          </div>
        </div>
        <h4> Floating label with icons </h4>
        <div className="e-float-input e-input-group">
          <input required={true} type="text" onFocus={this.onInputFocus}
onBlur={this.onInputBlur}/>
          <span className="e-float-line"/>
          <label className="e-float-text"> Enter Date </label>
          <span className="e-input-group-icon e-input-popup-date"
onMouseDown={this.onIconMouseDown} onMouseUp={this.onIconMouseUp}/>
        </div>
        <div className="e-float-input e-input-group e-float-icon-left">
          <span className="e-input-group-icon e-input-date"
onMouseDown={this.onIconMouseDown} onMouseUp={this.onIconMouseUp}/>
          <div className="e-input-in-wrap">
            <input required={true} type="text" onFocus={this.onInputFocus}
onBlur={this.onInputBlur}/>
            <span className="e-float-line"/>
            <label className="e-float-text"> Enter Date </label>
          </div>
        </div>
        <div className="e-float-input e-input-group e-float-icon-left">
          <span className="e-input-group-icon e-input-date"
onMouseDown={this.onIconMouseDown} onMouseUp={this.onIconMouseUp}/>
          <div className="e-input-in-wrap">

```

```

        <input required={true} type="text" onFocus={this.onInputFocus}
onBlur={this.onInputBlur}/>
        <span className="e-float-line"/>
        <label className="e-float-text"> Enter Date </label>
        <span className="e-input-group-icon e-input-down"/>
    </div>
</div>
</div>);
}
onInputFocus(args) {
    if (!args.target.parentElement.classList.contains('e-input-in-wrap'))
    {
        args.target.parentElement.classList.add('e-input-focus');
    }
    else {
        args.target.parentElement.parentElement.classList.add('e-input-
focus');
    }
}
onInputBlur(args) {
    if (!args.target.parentElement.classList.contains('e-input-in-wrap'))
    {
        args.target.parentElement.classList.remove('e-input-focus');
    }
    else {
        args.target.parentElement.parentElement.classList.remove('e-
input-focus');
    }
}
onIconMouseDown(args) {
    args.persist();
    setTimeout(() => {
        args.target.classList.add('e-input-btn-ripple');
    }, 300);
}
onIconMouseUp(args) {
    args.target.classList.remove('e-input-btn-ripple');
}
}
ReactDOM.render(<Default />, document.getElementById('input-container'));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
export default class Default extends React.Component {
    public render() {
        return (
            <div>
                <h4> TextBox with icons </h4>
                <div className="e-input-group ">
                    <input className="e-input" type="text" onFocus = {this.onInputFocus}
onBlur = {this.onInputBlur} placeholder="Enter Date"/>
                    <span className="e-input-group-icon e-input-popup-date" onMouseDown =
{this.onIconMouseDown} onMouseUp = {this.onIconMouseUp}/>
                </div>
            </div>

```

```

        <div className="e-input-group e-float-icon-left">
            <span className="e-input-group-icon e-input-date" onMouseDown =
{this.onIconMouseDown} onMouseUp = {this.onIconMouseUp}/>
            <div className="e-input-in-wrap">
                <input className="e-input" type="text" onFocus =
{this.onInputFocus} onBlur = {this.onInputBlur} placeholder="Enter Date"/>
            </div>
        </div>
        <div className="e-input-group e-float-icon-left">
            <span className="e-input-group-icon e-input-date" onMouseDown =
{this.onIconMouseDown} onMouseUp = {this.onIconMouseUp}/>
            <div className="e-input-in-wrap">
                <input className="e-input" type="text" onFocus =
{this.onInputFocus} onBlur = {this.onInputBlur} placeholder="Enter Date"/>
                <span className="e-input-group-icon e-input-down"/>
            </div>
        </div>
        <h4> Floating label with icons </h4>
        <div className="e-float-input e-input-group">
            <input required = {true} type="text" onFocus = {this.onInputFocus}
onBlur = {this.onInputBlur} />
            <span className="e-float-line"/>
            <label className="e-float-text"> Enter Date </label>
            <span className="e-input-group-icon e-input-popup-date" onMouseDown =
{this.onIconMouseDown} onMouseUp = {this.onIconMouseUp}/>
        </div>
        <div className="e-float-input e-input-group e-float-icon-left">
            <span className="e-input-group-icon e-input-date" onMouseDown =
{this.onIconMouseDown} onMouseUp = {this.onIconMouseUp}/>
            <div className="e-input-in-wrap">
                <input required = {true} type="text" onFocus =
{this.onInputFocus} onBlur = {this.onInputBlur} />
                <span className="e-float-line"/>
                <label className="e-float-text"> Enter Date </label>
            </div>
        </div>
        <div className="e-float-input e-input-group e-float-icon-left">
            <span className="e-input-group-icon e-input-date" onMouseDown =
{this.onIconMouseDown} onMouseUp = {this.onIconMouseUp}/>
            <div className="e-input-in-wrap">
                <input required = {true} type="text" onFocus =
{this.onInputFocus} onBlur = {this.onInputBlur} />
                <span className="e-float-line"/>
                <label className="e-float-text"> Enter Date </label>
                <span className="e-input-group-icon e-input-down"/>
            </div>
        </div>
    </div>
);
}

public onInputFocus(args: React.FocusEvent) {
    if (!((args.target as HTMLElement).parentElement as
HTMLElement).classList.contains('e-input-in-wrap')) {
        ((args.target as HTMLElement).parentElement as
HTMLElement).classList.add('e-input-focus');
    } else {

```



```

        (((args.target as HTMLElement).parentElement as
HTMLElement).parentElement as HTMLElement).classList.add('e-input-focus')
    }
}
public onInputBlur(args: React.FocusEvent) {
    if (!((args.target as HTMLElement).parentElement as
HTMLElement).classList.contains('e-input-in-wrap')) {
        ((args.target as HTMLElement).parentElement as
HTMLElement).classList.remove('e-input-focus');
    } else {
        ((args.target as HTMLElement).parentElement as
HTMLElement).parentElement as HTMLElement).classList.remove('e-input-focus');
    }
}
public onIconMouseDown(args: React.MouseEvent) {
    args.persist();
    setTimeout(
        () => {
            (args.target as HTMLElement).classList.add('e-input-btn-ripple');
        },
        300);
}
public onIconMouseUp(args: React.MouseEvent) {
    (args.target as HTMLElement).classList.remove('e-input-btn-ripple');
}
}
ReactDOM.render(<Default />, document.getElementById('input-container'));

```

## [Functional-component]

### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function Default() {
    return <div>
        <h4> TextBox with icons </h4>
        <div className="e-input-group ">
            <input className="e-input" type="text" onFocus={onInputFocus}
onBlur={onInputBlur} placeholder="Enter Date"/>
            <span className="e-input-group-icon e-input-popup-date"
onMouseDown={onIconMouseDown} onMouseUp={onIconMouseUp}/>
        </div>
        <div className="e-input-group e-float-icon-left">
            <span className="e-input-group-icon e-input-date"
onMouseDown={onIconMouseDown} onMouseUp={onIconMouseUp}/>
            <div className="e-input-in-wrap">
                <input className="e-input" type="text"
onFocus={onInputFocus} onBlur={onInputBlur} placeholder="Enter Date"/>
            </div>
        </div>
        <div className="e-input-group e-float-icon-left">
            <span className="e-input-group-icon e-input-date"
onMouseDown={onIconMouseDown} onMouseUp={onIconMouseUp}/>
            <div className="e-input-in-wrap">

```

```

        <input className="e-input" type="text"
onFocus={onInputFocus} onBlur={onInputBlur} placeholder="Enter Date"/>
        <span className="e-input-group-icon e-input-down"/>
      </div>
    </div>
    <h4> Floating label with icons </h4>
    <div className="e-float-input e-input-group">
      <input required={true} type="text" onFocus={onInputFocus}
onBlur={onInputBlur}/>
      <span className="e-float-line"/>
      <label className="e-float-text"> Enter Date </label>
      <span className="e-input-group-icon e-input-popup-date"
onMouseDown={onIconMouseDown} onMouseUp={onIconMouseUp}/>
    </div>
    <div className="e-float-input e-input-group e-float-icon-left">
      <span className="e-input-group-icon e-input-date"
onMouseDown={onIconMouseDown} onMouseUp={onIconMouseUp}/>
      <div className="e-input-in-wrap">
        <input required={true} type="text" onFocus={onInputFocus}
onBlur={onInputBlur}/>
        <span className="e-float-line"/>
        <label className="e-float-text"> Enter Date </label>
      </div>
    </div>
    <div className="e-float-input e-input-group e-float-icon-left">
      <span className="e-input-group-icon e-input-date"
onMouseDown={onIconMouseDown} onMouseUp={onIconMouseUp}/>
      <div className="e-input-in-wrap">
        <input required={true} type="text" onFocus={onInputFocus}
onBlur={onInputBlur}/>
        <span className="e-float-line"/>
        <label className="e-float-text"> Enter Date </label>
        <span className="e-input-group-icon e-input-down"/>
      </div>
    </div>
  </div>);
  function onInputFocus(args) {
    if (!args.target.parentElement.classList.contains('e-input-in-wrap'))
    {
      args.target.parentElement.classList.add('e-input-focus');
    }
    else {
      args.target.parentElement.parentElement.classList.add('e-input-
focus');
    }
  }
  function onInputBlur(args) {
    if (!args.target.parentElement.classList.contains('e-input-in-wrap'))
    {
      args.target.parentElement.classList.remove('e-input-focus');
    }
    else {
      args.target.parentElement.parentElement.classList.remove('e-
input-focus');
    }
  }
  function onIconMouseDown(args) {

```

```

        args.persist();
        setTimeout(() => {
            args.target.classList.add('e-input-btn-ripple');
        }, 300);
    }
    function onIconMouseUp(args) {
        args.target.classList.remove('e-input-btn-ripple');
    }
}
ReactDOM.render(<Default />, document.getElementById('input-container'));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function Default() {
    return (
        <div>
            <h4> TextBox with icons </h4>
            <div className="e-input-group ">
                <input className="e-input" type="text" onFocus =
{onInputFocus} onBlur = {onInputBlur} placeholder="Enter Date"/>
                <span className="e-input-group-icon e-input-popup-date"
onMouseDown = {onIconMouseDown} onMouseUp = {onIconMouseUp}/>
            </div>
            <div className="e-input-group e-float-icon-left">
                <span className="e-input-group-icon e-input-date" onMouseDown
= {onIconMouseDown} onMouseUp = {onIconMouseUp}/>
                <div className="e-input-in-wrap">
                    <input className="e-input" type="text" onFocus =
{onInputFocus} onBlur = {onInputBlur} placeholder="Enter Date"/>
                </div>
            </div>
            <div className="e-input-group e-float-icon-left">
                <span className="e-input-group-icon e-input-date" onMouseDown
= {onIconMouseDown} onMouseUp = {onIconMouseUp}/>
                <div className="e-input-in-wrap">
                    <input className="e-input" type="text" onFocus =
{onInputFocus} onBlur = {onInputBlur} placeholder="Enter Date"/>
                    <span className="e-input-group-icon e-input-down"/>
                </div>
            </div>
            <h4> Floating label with icons </h4>
            <div className="e-float-input e-input-group">
                <input required = {true} type="text" onFocus = {onInputFocus}
onBlur = {onInputBlur} />
                <span className="e-float-line"/>
                <label className="e-float-text"> Enter Date </label>
                <span className="e-input-group-icon e-input-popup-date"
onMouseDown = {onIconMouseDown} onMouseUp = {onIconMouseUp}/>
            </div>
            <div className="e-float-input e-input-group e-float-icon-left">
                <span className="e-input-group-icon e-input-date" onMouseDown
= {onIconMouseDown} onMouseUp = {onIconMouseUp}/>
                <div className="e-input-in-wrap">

```

```

        <input required = {true} type="text" onFocus =
{onInputFocus} onBlur = {onInputBlur} />
        <span className="e-float-line"/>
        <label className="e-float-text"> Enter Date </label>
      </div>
    </div>
    <div className="e-float-input e-input-group e-float-icon-left">
      <span className="e-input-group-icon e-input-date" onMouseDown
= {onIconMouseDown} onMouseUp = {onIconMouseUp}/>
      <div className="e-input-in-wrap">
        <input required = {true} type="text" onFocus =
{onInputFocus} onBlur = {onInputBlur} />
        <span className="e-float-line"/>
        <label className="e-float-text"> Enter Date </label>
        <span className="e-input-group-icon e-input-down"/>
      </div>
    </div>
  </div>
);
function onInputFocus(args: React.FocusEvent) {
  if (!((args.target as HTMLElement).parentElement as
HTMLElement).classList.contains('e-input-in-wrap')) {
    ((args.target as HTMLElement).parentElement as
HTMLElement).classList.add('e-input-focus');
  } else {
    ((args.target as HTMLElement).parentElement as
HTMLElement).parentElement as HTMLElement).classList.add('e-input-focus')
  }
}
function onInputBlur(args: React.FocusEvent) {
  if (!((args.target as HTMLElement).parentElement as
HTMLElement).classList.contains('e-input-in-wrap')) {
    ((args.target as HTMLElement).parentElement as
HTMLElement).classList.remove('e-input-focus');
  } else {
    ((args.target as HTMLElement).parentElement as
HTMLElement).parentElement as HTMLElement).classList.remove('e-input-focus');
  }
}
function onIconMouseDown(args: React.MouseEvent) {
  args.persist();
  setTimeout(
    () => {
      (args.target as HTMLElement).classList.add('e-input-btn-
ripple');
    },
    300);
}
function onIconMouseUp(args: React.MouseEvent) {
  (args.target as HTMLElement).classList.remove('e-input-btn-ripple');
}
}
ReactDOM.render(<Default />, document.getElementById('input-container'));

```

To place the icon on left side of the TextBox, create a div element with the class `e-input-in-wrap` as wrapper to the input element and place the floating line, floating text, and right side icon element within it. Add the `e-float-icon-left` class to the TextBox container element.

#### With clear button and floating label

The clear button is added to the input for clearing the value given in the TextBox. It is shown only when the input field has a value, otherwise not shown.

You can add the clear button to the TextBox by using `showClearButton` API in textbox.

#### [Class-component]

##### INDEX.JSX

```
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from "react";
import * as ReactDOM from "react-dom";
export default class App extends React.Component {
  render() {
    return (
      <div className="App">
        <div className="textboxes">
          <h4>Textbox with clear button</h4>
          <TextBoxComponent placeholder="First Name"
            showClearButton={true} floatLabelType="Never"/>
        </div>
        <div className="textboxes">
          <h4>Floating textbox with clear button</h4>
          <TextBoxComponent placeholder="Last Name"
            showClearButton={true} floatLabelType="Auto"/>
        </div>
      </div>);
  }
}
;
ReactDOM.render(<App />, document.getElementById('input-container'));
```

##### INDEX.TSX

```
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from "react";
import * as ReactDOM from "react-dom";
export default class App extends React.Component<{}, {}> {
  public render() {
    return (
      <div className="App">
        <div className="textboxes">
          <h4>Textbox with clear button</h4>
          <TextBoxComponent placeholder="First Name"
            showClearButton={true} floatLabelType="Never"/>
        </div>
        <div className="textboxes">
          <h4>Floating textbox with clear button</h4>
          <TextBoxComponent placeholder="Last Name"
            showClearButton={true} floatLabelType="Auto"/>
        </div>
      </div>
    );
  }
}
```

```

    )
  }
};
ReactDOM.render(<App />, document.getElementById('input-container'));

```

## [Functional-component]

### INDEX.JSX

```

import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (
    <div className="App">
      <div className="textboxes">
        <h4>Textbox with clear button</h4>
        <TextBoxComponent placeholder="First Name"
showClearButton={true} floatLabelType="Never"/>
      </div>
      <div className="textboxes">
        <h4>Floating textbox with clear button</h4>
        <TextBoxComponent placeholder="Last Name"
showClearButton={true} floatLabelType="Auto"/>
      </div>
    </div>);
}
ReactDOM.render(<App />, document.getElementById('input-container'));

```

### INDEX.TSX

```

import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (
    <div className="App">
      <div className="textboxes">
        <h4>Textbox with clear button</h4>
        <TextBoxComponent placeholder="First Name" showClearButton=
{true} floatLabelType="Never"/>
      </div>
      <div className="textboxes">
        <h4>Floating textbox with clear button</h4>
        <TextBoxComponent placeholder="Last Name" showClearButton=
{true} floatLabelType="Auto"/>
      </div>
    </div>
  )
}
ReactDOM.render(<App />, document.getElementById('input-container'));

```

## Floating Label without required attribute

You can render the Floating label TextBox without required attribute by manually float the label above of the TextBox using input events.

You can manually float the label above of the TextBox by adding the below list of classes to the floating label element. The classes are:

Class Name | Description

e-label-top | Floats the label above of the TextBox.

e-label-bottom | Label to be placed as placeholder for the TextBox.

[Class-component]

### INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
export default class App extends React.Component {
  textboxInstance;
  constructor(props) {
    super(props);
    this.onFocusOut = this.onFocusOut.bind(this);
    this.onFocusIn = this.onFocusIn.bind(this);
    this.onInputEvt = this.onInputEvt.bind(this);
  }
  onFocusOut(args) {
    /* Update the label position based on Input value */
    this.updateLabelState(args.target.value,
    args.target.parentElement.querySelector('.e-float-text'));
  }
  onFocusIn(args) {
    const label = args.target.parentElement.querySelector('.e-float-
    text');
    label.classList.add('e-label-bottom');
    label.classList.remove('e-label-top');
  }
  onInputEvt(args) {
    /* Update the label position based on Input value */
    this.updateLabelState(args.target.value,
    args.target.parentElement.querySelector('.e-float-text'));
  }
  /* Update the label position based on Input value */
  updateLabelState(value, label) {
    /* e-label-top - Float the label above of the Input */
    /* e-label-bottom - Move the label to the Input */
    if (value) {
      label.classList.add('e-label-top');
      label.classList.remove('e-label-bottom');
    }
    else {
      label.classList.add('e-label-bottom');
      label.classList.remove('e-label-top');
    }
  }
  render() {
    return (<div className="inner-container">
      <h4> Floating label without required attribute </h4>
      <div className="e-float-input">
```

```

        <input id='inpt1' type="text" onInput={this.onInputEvt}
onFocus={this.onFocusIn} onBlur={this.onFocusOut} ref={e =>
this.textboxInstance = e}/>
        <span className="e-float-line"/>
        <label className="e-float-text">First Name</label>
    </div>
</div>);
    }
    componentDidMount() {
        /* Update the label position based on initial input value */
        this.updateLabelState(this.textboxInstance.value,
this.textboxInstance.parentElement.querySelector('.e-float-text'));
    }
}
;
ReactDOM.render(<App />, document.getElementById('input-container'));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
export default class App extends React.Component<{}, {}> {
    public textboxInstance: any;
    constructor(props: any) {
        super(props);
        this.onFocusOut = this.onFocusOut.bind(this);
        this.onFocusIn = this.onFocusIn.bind(this);
        this.onInputEvt = this.onInputEvt.bind(this);
    }
    public onFocusOut(args: React.FocusEvent) {
        /* Update the label position based on Input value */
        this.updateLabelState((args.target as HTMLInputElement).value,
((args.target as HTMLInputElement).parentElement as
HTMLInputElement).querySelector('.e-float-text') as HTMLInputElement);
    }
    public onFocusIn(args: React.FocusEvent) {
        const label = ((args.target as HTMLInputElement).parentElement as
HTMLInputElement).querySelector('.e-float-text') as HTMLInputElement;
        label.classList.add('e-label-bottom');
        label.classList.remove('e-label-top');
    }
    public onInputEvt(args: React.FormEvent) {
        /* Update the label position based on Input value */
        this.updateLabelState((args.target as HTMLInputElement).value,
((args.target as HTMLInputElement).parentElement as
HTMLInputElement).querySelector('.e-float-text') as HTMLInputElement);
    }
    /* Update the label position based on Input value */
    public updateLabelState(value: string, label: HTMLInputElement) {
        /* e-label-top - Float the label above of the Input */
        /* e-label-bottom - Move the label to the Input */
        if (value) {
            label.classList.add('e-label-top');
            label.classList.remove('e-label-bottom');
        } else {
            label.classList.add('e-label-bottom');
        }
    }
}

```



```

        label.classList.remove('e-label-top');
    }
}
public render() {
    return (
        <div className="inner-container">
            <h4> Floating label without required attribute </h4>
            <div className="e-float-input">
                <input id='inpt1' type="text" onInput= {this.onInputEvt}
onFocus={this.onFocusIn} onBlur={ this.onFocusOut } ref = {e =>
this.textboxInstance = e!} />
                <span className="e-float-line"/>
                <label className="e-float-text">First Name</label>
            </div>
        </div>
    )
}
public componentDidMount() {
    /* Update the label position based on initial input value */
    this.updateLabelState(this.textboxInstance.value,
this.textboxInstance.parentElement.querySelector('.e-float-text'));
}
};
ReactDOM.render(<App />, document.getElementById('input-container'));

```

## [Functional-component]

### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    let textboxInstance;
    function onFocusOut(args) {
        /* Update the label position based on Input value */
        updateLabelState(args.target.value,
args.target.parentElement.querySelector('.e-float-text'));
    }
    function onFocusIn(args) {
        const label = args.target.parentElement.querySelector('.e-float-
text');
        label.classList.add('e-label-bottom');
        label.classList.remove('e-label-top');
    }
    function onInputEvt(args) {
        /* Update the label position based on Input value */
        updateLabelState(args.target.value,
args.target.parentElement.querySelector('.e-float-text'));
    }
    /* Update the label position based on Input value */
    function updateLabelState(value, label) {
        /* e-label-top - Float the label above of the Input */
        /* e-label-bottom - Move the label to the Input */
        if (value) {
            label.classList.add('e-label-top');
            label.classList.remove('e-label-bottom');
        }
    }
}

```

```

    }
    else {
        label.classList.add('e-label-bottom');
        label.classList.remove('e-label-top');
    }
}
return (<div className="inner-container">
    <h4> Floating label without required attribute </h4>
    <div className="e-float-input">
        <input id='inpt1' type="text" onInput={onInputEvt}
onFocus={onFocusIn} onBlur={onFocusOut} ref={e => e.textboxInstance = e}/>
        <span className="e-float-line"/>
        <label className="e-float-text">First Name</label>
    </div>
</div>);
React.useEffect(() => {
    /* Update the label position based on initial input value */
    updateLabelState(textboxInstance.value,
textboxInstance.parentElement.querySelector('.e-float-text'));
    [];
});
}
;
ReactDOM.render(<App />, document.getElementById('input-container'));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
export default class App extends React.Component<{}, {}> {
    public textboxInstance: any;
    constructor(props: any) {
        super(props);
        this.onFocusOut = this.onFocusOut.bind(this);
        this.onFocusIn = this.onFocusIn.bind(this);
        this.onInputEvt = this.onInputEvt.bind(this);
    }
    public componentDidMount() {
        /* Update the label position based on initial input value */
        this.updateLabelState(this.textboxInstance.value,
this.textboxInstance.parentElement.querySelector('.e-float-text'));
    }
    public onFocusOut(args: React.FocusEvent) {
        /* Update the label position based on Input value */
        this.updateLabelState((args.target as HTMLInputElement).value,
((args.target as HTMLInputElement).parentElement as
HTMLInputElement).querySelector('.e-float-text') as HTMLInputElement);
        public onFocusOut(args: React.FocusEvent) {
            /* Update the label position based on Input value */
            this.updateLabelState((args.target as HTMLInputElement).value,
((args.target as HTMLInputElement).parentElement as
HTMLInputElement).querySelector('.e-float-text') as HTMLInputElement);
        }
        public onFocusIn(args: React.FocusEvent) {
            const label = ((args.target as HTMLInputElement).parentElement as
HTMLInputElement).querySelector('.e-float-text') as HTMLInputElement;

```

```

        label.classList.add('e-label-bottom');
        label.classList.remove('e-label-top');
    }
    public onInputEvt(args: React.FormEvent) {
        /* Update the label position based on Input value */
        this.updateLabelState((args.target as HTMLInputElement).value,
            ((args.target as HTMLInputElement).parentElement as
            HTMLInputElement).querySelector('.e-float-text') as HTMLInputElement);
    }
    /* Update the label position based on Input value */
    public updateLabelState(value: string, label: HTMLInputElement) {
        /* e-label-top - Float the label above of the Input */
        /* e-label-bottom - Move the label to the Input */
        if (value) {
            label.classList.add('e-label-top');
            label.classList.remove('e-label-bottom');
        } else {
            label.classList.add('e-label-bottom');
            label.classList.remove('e-label-top');
        }
    }
    public render() {
        return (
            <div className="inner-container">
                <h4> Floating label without required attribute </h4>
                <div className="e-float-input">
                    <input id='inpt1' type="text" onInput= {this.onInputEvt}
onFocus={this.onFocusIn} onBlur={this.onFocusOut} ref = {e =>
this.textboxInstance = e!} />
                    <span className="e-float-line"/>
                    <label className="e-float-text">First Name</label>
                </div>
            </div>
        )
    }
};
ReactDOM.render(<App />, document.getElementById('input-container'));

```

### Multi-line input with floating label

Add the HTML textarea element with the `e-input` class to create default multi-line input.

Add the floating label support to the `multi-line input` by creating the floating label structure as defined in the initial section.

### [Class-component]

#### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
export default class App extends React.Component {
    render() {
        return (
            <div>
                <textarea className="e-input" placeholder="Address"/>
                <div className="e-float-input">
                    <textarea required={true}/>
                </div>
            </div>
        )
    }
}

```

```

        <span className="e-float-line"/>
        <label className="e-float-text"> Address</label>
      </div>
    </div>);
  }
}
;
ReactDOM.render(<App />, document.getElementById('input-container'));

```

**INDEX.TSX**

```

import * as React from "react";
import * as ReactDOM from "react-dom";
export default class App extends React.Component<{}, {}> {
  public render() {
    return (
      <div>
        <textarea className="e-input" placeholder="Address"/>
        <div className="e-float-input">
          <textarea required={true}/>
          <span className="e-float-line"/>
          <label className="e-float-text"> Address</label>
        </div>
      </div>
    )
  }
};
ReactDOM.render(<App />, document.getElementById('input-container'));

```

**[Functional-component]****INDEX.JSX**

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (
    <div>
      <textarea className="e-input" placeholder="Address"/>
      <div className="e-float-input">
        <textarea required={true}/>
        <span className="e-float-line"/>
        <label className="e-float-text"> Address</label>
      </div>
    </div>
  );
}
ReactDOM.render(<App />, document.getElementById('input-container'));

```

**INDEX.TSX**

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (
    <div>
      <textarea className="e-input" placeholder="Address"/>

```

```

        <div className="e-float-input">
          <textarea required={true}/>
          <span className="e-float-line"/>
          <label className="e-float-text"> Address</label>
        </div>
      </div>
    )
  }
ReactDOM.render(<App />, document.getElementById('input-container'));

```

See Also

- [How to add floating label to TextBox programmatically](#)
- [Change the floating label color of the TextBox](#)
- [Change the color of the TextBox based on its value](#)

## Sizing in React Textbox component

You can render the TextBox in two different sizes.

Property | Description

Normal | By default, the TextBox is rendered with normal size.

Small | You need to add `e-small` class to the input element, or else add to the input container.

[Class-component]

### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
export default class Default extends React.Component {
  render() {
    return (
      <div>
        <h4> Normal Size </h4>
        <div className="e-input-group">
          <input className="e-input" type="text" placeholder="Enter Name"
onFocus={this.onInputFocus} onBlur={this.onInputBlur}/>
        </div>
        <div className="e-float-input">
          <input type='text' required={true} onFocus={this.onInputFocus}
onBlur={this.onInputBlur}/>
          <span className="e-float-line"/>
          <label className="e-float-text">Enter Name</label>
        </div>
        <div className="e-input-group">
          <input className="e-input" type="text" placeholder="Enter Date"
onFocus={this.onInputFocus} onBlur={this.onInputBlur}/>
          <span className="e-input-group-icon e-input-popup-date"
onMouseDown={this.onIconMouseDown} onMouseUp={this.onIconMouseUp}/>
        </div>
        <h4> Small Size </h4>
        <div className="e-input-group e-small">
          <input className="e-input" type="text" placeholder="Enter Name"
onFocus={this.onInputFocus} onBlur={this.onInputBlur}/>
        </div>
      </div>
    );
  }
}

```

```

    <div className="e-float-input e-small">
      <input type='text' required={true} onFocus={this.onInputFocus}
onBlur={this.onInputBlur}/>
      <span className="e-float-line"/>
      <label className="e-float-text">Enter Name</label>
    </div>
    <div className="e-input-group e-small">
      <input className="e-input" type="text" placeholder="Enter Date"
onFocus={this.onInputFocus} onBlur={this.onInputBlur}/>
      <span className="e-input-group-icon e-input-popup-date"
onMouseDown={this.onIconMouseDown} onMouseUp={this.onIconMouseUp}/>
    </div>
  </div>;
}
onInputFocus(args) {
  if (!args.target.parentElement.classList.contains('e-input-in-wrap'))
  {
    args.target.parentElement.classList.add('e-input-focus');
  }
  else {
    args.target.parentElement.parentElement.classList.add('e-input-
focus');
  }
}
onInputBlur(args) {
  if (!args.target.parentElement.classList.contains('e-input-in-wrap'))
  {
    args.target.parentElement.classList.remove('e-input-focus');
  }
  else {
    args.target.parentElement.parentElement.classList.remove('e-
input-focus');
  }
}
onIconMouseDown(args) {
  args.persist();
  setTimeout(() => {
    args.target.classList.add('e-input-btn-ripple');
  }, 300);
}
onIconMouseUp(args) {
  args.target.classList.remove('e-input-btn-ripple');
}
}
ReactDOM.render(<Default />, document.getElementById('input-container'));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
export default class Default extends React.Component {
  public render() {
    return (
      <div>
        <h4> Normal Size </h4>
        <div className="e-input-group">

```

```

        <input className="e-input" type="text" placeholder="Enter Name" onFocus
= {this.onInputFocus} onBlur = {this.onInputBlur} />
    </div>
    <div className="e-float-input">
        <input type='text' required = {true} onFocus = {this.onInputFocus}
onBlur = {this.onInputBlur}/>
        <span className="e-float-line"/>
        <label className="e-float-text">Enter Name</label>
    </div>
    <div className="e-input-group">
        <input className="e-input" type="text" placeholder="Enter Date" onFocus
= {this.onInputFocus} onBlur = {this.onInputBlur}/>
        <span className="e-input-group-icon e-input-popup-date" onMouseDown =
{this.onIconMouseDown} onMouseUp = {this.onIconMouseUp}/>
    </div>
    <h4> Small Size </h4>
    <div className="e-input-group e-small">
        <input className="e-input" type="text" placeholder="Enter Name" onFocus
= {this.onInputFocus} onBlur = {this.onInputBlur}/>
    </div>
    <div className="e-float-input e-small">
        <input type='text' required = {true} onFocus = {this.onInputFocus}
onBlur = {this.onInputBlur} />
        <span className="e-float-line"/>
        <label className="e-float-text">Enter Name</label>
    </div>
    <div className="e-input-group e-small">
        <input className="e-input" type="text" placeholder="Enter Date" onFocus
= {this.onInputFocus} onBlur = {this.onInputBlur} />
        <span className="e-input-group-icon e-input-popup-date" onMouseDown =
{this.onIconMouseDown} onMouseUp = {this.onIconMouseUp}/>
    </div>
</div>
);
}
public onInputFocus(args: React.FocusEvent) {
    if (!((args.target as HTMLElement).parentElement as
HTMLElement).classList.contains('e-input-in-wrap')) {
        ((args.target as HTMLElement).parentElement as
HTMLElement).classList.add('e-input-focus');
    } else {
        ((args.target as HTMLElement).parentElement as
HTMLElement).parentElement as HTMLElement).classList.add('e-input-focus')
    }
}
public onInputBlur(args: React.FocusEvent) {
    if (!((args.target as HTMLElement).parentElement as
HTMLElement).classList.contains('e-input-in-wrap')) {
        ((args.target as HTMLElement).parentElement as
HTMLElement).classList.remove('e-input-focus');
    } else {
        ((args.target as HTMLElement).parentElement as
HTMLElement).parentElement as HTMLElement).classList.remove('e-input-focus');
    }
}
public onIconMouseDown(args: React.MouseEvent) {
    args.persist();

```

```

        setTimeout(
            () => {
                (args.target as HTMLElement).classList.add('e-input-btn-ripple');
            },
            300);
    }
    public onIconMouseUp(args: React.MouseEvent) {
        (args.target as HTMLElement).classList.remove('e-input-btn-ripple');
    }
}
ReactDOM.render(<Default />, document.getElementById('input-container'));

```

## [Functional-component]

### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function Default() {
    return (<div>
        <h4> Normal Size </h4>
        <div className="e-input-group">
            <input className="e-input" type="text" placeholder="Enter Name"
onFocus={onInputFocus} onBlur={onInputBlur}/>
        </div>
        <div className="e-float-input">
            <input type='text' required={true} onFocus={onInputFocus}
onBlur={onInputBlur}/>
            <span className="e-float-line"/>
            <label className="e-float-text">Enter Name</label>
        </div>
        <div className="e-input-group">
            <input className="e-input" type="text" placeholder="Enter Date"
onFocus={onInputFocus} onBlur={onInputBlur}/>
            <span className="e-input-group-icon e-input-popup-date"
onMouseDown={onIconMouseDown} onMouseUp={onIconMouseUp}/>
        </div>
        <h4> Small Size </h4>
        <div className="e-input-group e-small">
            <input className="e-input" type="text" placeholder="Enter Name"
onFocus={onInputFocus} onBlur={onInputBlur}/>
        </div>
        <div className="e-float-input e-small">
            <input type='text' required={true} onFocus={onInputFocus}
onBlur={onInputBlur}/>
            <span className="e-float-line"/>
            <label className="e-float-text">Enter Name</label>
        </div>
        <div className="e-input-group e-small">
            <input className="e-input" type="text" placeholder="Enter Date"
onFocus={onInputFocus} onBlur={onInputBlur}/>
            <span className="e-input-group-icon e-input-popup-date"
onMouseDown={onIconMouseDown} onMouseUp={onIconMouseUp}/>
        </div>
    </div>);
}

```



```
function onInputFocus(args) {
    if (!args.target.parentElement.classList.contains('e-input-in-wrap'))
    {
        args.target.parentElement.classList.add('e-input-focus');
    }
    else {
        args.target.parentElement.parentElement.classList.add('e-input-
focus');
    }
}
function onInputBlur(args) {
    if (!args.target.parentElement.classList.contains('e-input-in-wrap'))
    {
        args.target.parentElement.classList.remove('e-input-focus');
    }
    else {
        args.target.parentElement.parentElement.classList.remove('e-
input-focus');
    }
}
function onIconMouseDown(args) {
    args.persist();
    setTimeout(() => {
        args.target.classList.add('e-input-btn-ripple');
    }, 300);
}
function onIconMouseUp(args) {
    args.target.classList.remove('e-input-btn-ripple');
}
}
ReactDOM.render(<Default />, document.getElementById('input-container'));
```

## INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
function Default() {
    return (
        <div>
            <h4> Normal Size </h4>
            <div className="e-input-group">
                <input className="e-input" type="text" placeholder="Enter Name" onFocus
= {onInputFocus} onBlur = {onInputBlur} />
            </div>
            <div className="e-float-input">
                <input type='text' required = {true} onFocus = {onInputFocus} onBlur =
{onInputBlur}/>
                <span className="e-float-line"/>
                <label className="e-float-text">Enter Name</label>
            </div>
            <div className="e-input-group">
                <input className="e-input" type="text" placeholder="Enter Date" onFocus
= {onInputFocus} onBlur = {onInputBlur}/>
                <span className="e-input-group-icon e-input-popup-date" onMouseDown =
{onIconMouseDown} onMouseUp = {onIconMouseUp}/>
            </div>
        </div>
    );
}
```

```

<h4> Small Size </h4>
<div className="e-input-group e-small">
  <input className="e-input" type="text" placeholder="Enter Name" onFocus
= {onInputFocus} onBlur = {onInputBlur}/>
</div>
<div className="e-float-input e-small">
  <input type='text' required = {true} onFocus = {onInputFocus} onBlur =
{onInputBlur} />
  <span className="e-float-line"/>
  <label className="e-float-text">Enter Name</label>
</div>
<div className="e-input-group e-small">
  <input className="e-input" type="text" placeholder="Enter Date" onFocus
= {onInputFocus} onBlur = {onInputBlur} />
  <span className="e-input-group-icon e-input-popup-date" onMouseDown =
{onIconMouseDown} onMouseUp = {onIconMouseUp}/>
</div>
</div>
);
function onInputFocus(args: React.FocusEvent) {
  if (!((args.target as HTMLElement).parentElement as
HTMLElement).classList.contains('e-input-in-wrap')) {
    ((args.target as HTMLElement).parentElement as
HTMLElement).classList.add('e-input-focus');
  } else {
    ((args.target as HTMLElement).parentElement as
HTMLElement).parentElement as HTMLElement).classList.add('e-input-focus')
  }
}
function onInputBlur(args: React.FocusEvent) {
  if (!((args.target as HTMLElement).parentElement as
HTMLElement).classList.contains('e-input-in-wrap')) {
    ((args.target as HTMLElement).parentElement as
HTMLElement).classList.remove('e-input-focus');
  } else {
    ((args.target as HTMLElement).parentElement as
HTMLElement).parentElement as HTMLElement).classList.remove('e-input-focus');
  }
}
function onIconMouseDown(args: React.MouseEvent) {
  args.persist();
  setTimeout(
    () => {
      (args.target as HTMLElement).classList.add('e-input-btn-
ripple');
    },
    300);
}
function onIconMouseUp(args: React.MouseEvent) {
  (args.target as HTMLElement).classList.remove('e-input-btn-ripple');
}
}
ReactDOM.render(<Default />, document.getElementById('input-container'));

```

## Validation in React Textbox component

The TextBox supports three types of validation styles namely `error`, `warning`, and `success`. These states are enabled by adding corresponding classes `.e-error`, `.e-warning`, or `.e-success` to the input parent element.

### [Class-component]

#### INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
export default class Default extends React.Component {
  render() {
    return (
      <div>
        <div className="e-input-group e-warning">
          <input className="e-input" type="text" placeholder="Input with warning" onFocus={this.onInputFocus} onBlur={this.onInputBlur}/>
        </div>
        <div className="e-input-group e-error">
          <input className="e-input" type="text" placeholder="Input with error" onFocus={this.onInputFocus} onBlur={this.onInputBlur}/>
        </div>
        <div className="e-input-group e-success">
          <input className="e-input" type="text" placeholder="Input with success" onFocus={this.onInputFocus} onBlur={this.onInputBlur}/>
        </div>
      </div>
    );
  }
  onInputFocus(args) {
    if (!args.target.parentElement.classList.contains('e-input-in-wrap')) {
      args.target.parentElement.classList.add('e-input-focus');
    } else {
      args.target.parentElement.parentElement.classList.add('e-input-focus');
    }
  }
  onInputBlur(args) {
    if (!args.target.parentElement.classList.contains('e-input-in-wrap')) {
      args.target.parentElement.classList.remove('e-input-focus');
    } else {
      args.target.parentElement.parentElement.classList.remove('e-input-focus');
    }
  }
}
ReactDOM.render(<Default />, document.getElementById('input-container'));
```

#### INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
export default class Default extends React.Component {
```

```

public render() {
    return (
        <div>
            <div className="e-input-group e-warning">
                <input className="e-input" type="text" placeholder="Input with
warning" onFocus = {this.onInputFocus} onBlur = {this.onInputBlur} />
            </div>
            <div className="e-input-group e-error">
                <input className="e-input" type="text" placeholder="Input with error"
onFocus = {this.onInputFocus} onBlur = {this.onInputBlur} />
            </div>
            <div className="e-input-group e-success">
                <input className="e-input" type="text" placeholder="Input with
success" onFocus = {this.onInputFocus} onBlur = {this.onInputBlur}/>
            </div>
        </div>
    );
}

public onInputFocus(args: React.FocusEvent) {
    if (!((args.target as HTMLElement).parentElement as
HTMLElement).classList.contains('e-input-in-wrap')) {
        ((args.target as HTMLElement).parentElement as
HTMLElement).classList.add('e-input-focus');
    } else {
        ((args.target as HTMLElement).parentElement as
HTMLElement).parentElement as HTMLElement).classList.add('e-input-focus')
    }
}

public onInputBlur(args: React.FocusEvent) {
    if (!((args.target as HTMLElement).parentElement as
HTMLElement).classList.contains('e-input-in-wrap')) {
        ((args.target as HTMLElement).parentElement as
HTMLElement).classList.remove('e-input-focus');
    } else {
        ((args.target as HTMLElement).parentElement as
HTMLElement).parentElement as HTMLElement).classList.remove('e-input-focus');
    }
}

ReactDOM.render(<Default />, document.getElementById('input-container'));

```

## [Functional-component]

### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function Default() {
    return (<div>
        <div className="e-input-group e-warning">
            <input className="e-input" type="text" placeholder="Input with
warning" onFocus={onInputFocus} onBlur={onInputBlur}/>
        </div>
        <div className="e-input-group e-error">
            <input className="e-input" type="text" placeholder="Input with error"
onFocus={onInputFocus} onBlur={onInputBlur}/>
        </div>
    </div>
);
}

```

```

        </div>
        <div className="e-input-group e-success">
          <input className="e-input" type="text" placeholder="Input with
success" onFocus={onInputFocus} onBlur={onInputBlur}/>
        </div>
      </div>);
      function onInputFocus(args) {
        if (!args.target.parentElement.classList.contains('e-input-in-wrap'))
        {
          args.target.parentElement.classList.add('e-input-focus');
        }
        else {
          args.target.parentElement.parentElement.classList.add('e-input-
focus');
        }
      }
      function onInputBlur(args) {
        if (!args.target.parentElement.classList.contains('e-input-in-wrap'))
        {
          args.target.parentElement.classList.remove('e-input-focus');
        }
        else {
          args.target.parentElement.parentElement.classList.remove('e-
input-focus');
        }
      }
    }
    ReactDOM.render(<Default />, document.getElementById('input-container'));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function Default() {
  return (
    <div>
      <div className="e-input-group e-warning">
        <input className="e-input" type="text" placeholder="Input with
warning" onFocus = {onInputFocus} onBlur = {onInputBlur} />
      </div>
      <div className="e-input-group e-error">
        <input className="e-input" type="text" placeholder="Input with error"
onFocus = {onInputFocus} onBlur = {onInputBlur} />
      </div>
      <div className="e-input-group e-success">
        <input className="e-input" type="text" placeholder="Input with
success" onFocus = {onInputFocus} onBlur = {onInputBlur}/>
      </div>
    </div>
  );
  function onInputFocus(args: React.FocusEvent) {
    if (!((args.target as HTMLElement).parentElement as
HTMLElement).classList.contains('e-input-in-wrap')) {
      ((args.target as HTMLElement).parentElement as
HTMLElement).classList.add('e-input-focus');
    } else {

```

```

        (((args.target as HTMLElement).parentElement as
HTMLElement).parentElement as HTMLElement).classList.add('e-input-focus')
    }
}
function onInputBlur(args: React.FocusEvent) {
    if (!((args.target as HTMLElement).parentElement as
HTMLElement).classList.contains('e-input-in-wrap')) {
        ((args.target as HTMLElement).parentElement as
HTMLElement).classList.remove('e-input-focus');
    } else {
        ((args.target as HTMLElement).parentElement as
HTMLElement).parentElement as HTMLElement).classList.remove('e-input-focus');
    }
}
}
ReactDOM.render(<Default />, document.getElementById('input-container'));

```

## Multiline in React Textbox component

This feature allows the textbox to accept one or more lines of text like address, description, comments, and more.

### Create multiline textbox

You can convert the default textbox into the multiline textbox by setting the [multiline](#) API value as true or pass HTML5 textarea as element to the textbox.

The multiline textbox allows you to resize it in vertical direction alone.

### [Class-component]

#### INDEX.JSX

```

import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from "react-dom";
export default class App extends React.Component {
    render() {
        return (<div className="multiline">
            <TextBoxComponent multiline={true} placeholder='Enter your
address' value='Mr. Dodsworth Dodsworth, System Analyst, Studio 103, The
Business Center' />
        </div>);
    }
}
ReactDOM.render(<App />, document.getElementById('default'));

```

#### INDEX.TSX

```

import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from "react-dom";
export default class App extends React.Component<{}, {}> {
    public render() {
        return (
            <div className="multiline">

```

```

        <TextBoxComponent multiline={true} placeholder='Enter your
address' value= 'Mr. Dodsworth Dodsworth, System Analyst, Studio 103, The
Business Center' />
    </div>
    );
}
}
ReactDOM.render(<App />, document.getElementById('default'));

```

## [Functional-component]

### INDEX.JSX

```

import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from "react-dom";
function App() {
    return (<div className="multiline">
        <TextBoxComponent multiline={true} placeholder='Enter your
address' value='Mr. Dodsworth Dodsworth, System Analyst, Studio 103, The
Business Center' />
    </div>);
}
ReactDOM.render(<App />, document.getElementById('default'));

```

### INDEX.TSX

```

import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from "react-dom";
function App() {
    return (
        <div className="multiline">
            <TextBoxComponent multiline={true} placeholder='Enter your
address' value= 'Mr. Dodsworth Dodsworth, System Analyst, Studio 103, The
Business Center' />
        </div>
    );
}
ReactDOM.render(<App />, document.getElementById('default'));

```

## Implementing floating label

You can achieve the floating label behavior in the multiline textbox by setting `floatLabelType` as 'Auto'. The placeholder text act as floating label to the multiline textbox. You can provide the placeholder text to the multiline textbox either by using the `placeholder` property or placeholder attribute.

## [Class-component]

### INDEX.JSX

```

import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from "react-dom";
export default class App extends React.Component {
    render() {

```

```

        return (<div className="multiline">
            <label className='label'>Float label type Auto</label>
            <TextBoxComponent multiline={true} placeholder='Enter your
address' floatLabelType='Auto' />
            <label className='label'>Float label type Always</label>
            <TextBoxComponent multiline={true} placeholder='Enter your
address' floatLabelType='Always' />
            <label className='label'>Float label type Never</label>
            <TextBoxComponent multiline={true} placeholder='Enter your
address' floatLabelType='Never' />
        </div>);
    }
}
ReactDOM.render(<App />, document.getElementById('default'));

```

## INDEX.TSX

```

import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}, {}> {
    public render() {
        return (
            <div className="multiline">
                <label className='label'>Float label type Auto</label>
                <TextBoxComponent multiline={true} placeholder='Enter your
address' floatLabelType='Auto' />
                <label className='label'>Float label type Always</label>
                <TextBoxComponent multiline={true} placeholder='Enter your
address' floatLabelType='Always' />
                <label className='label'>Float label type Never</label>
                <TextBoxComponent multiline={true} placeholder='Enter your
address' floatLabelType='Never' />
            </div>
        );
    }
}
ReactDOM.render(<App />, document.getElementById('default'));

```

## [Functional-component]

## INDEX.JSX

```

import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    return (<div className="multiline">
        <label className='label'>Float label type Auto</label>
        <TextBoxComponent multiline={true} placeholder='Enter your
address' floatLabelType='Auto' />
        <label className='label'>Float label type Always</label>
        <TextBoxComponent multiline={true} placeholder='Enter your
address' floatLabelType='Always' />
        <label className='label'>Float label type Never</label>
    </div>);
}

```



```

        <TextBoxComponent multiline={true} placeholder='Enter your
address' floatLabelType='Never' />
    </div>);
}
ReactDOM.render(<App />, document.getElementById('default'));

```

## INDEX.TSX

```

import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from "react-dom";
function App() {
    return (
        <div className="multiline">
            <label className='label'>Float label type Auto</label>
            <TextBoxComponent multiline={true} placeholder='Enter your
address' floatLabelType='Auto' />
            <label className='label'>Float label type Always</label>
            <TextBoxComponent multiline={true} placeholder='Enter your
address' floatLabelType='Always' />
            <label className='label'>Float label type Never</label>
            <TextBoxComponent multiline={true} placeholder='Enter your
address' floatLabelType='Never' />
        </div>
    );
}
ReactDOM.render(<App />, document.getElementById('default'));

```

## Auto resizing

By default, you can manually resize the multiline textbox. But you can also create an **auto resizing** multiline textbox with both the initial and dynamic value change. It can be done by calculating the height of the textarea in the created event for initial value update and in the input event for dynamic value update of the auto resize multiline textbox, as explained in the following code sample.

## [Class-component]

## INDEX.JSX

```

{% raw %}
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from "react-dom";
export default class App extends React.Component {
    textareaObj;
    render() {
        return (<div className="multiline">
            <TextBoxComponent multiline={true} value='Mr. Dodsworth Dodsworth,
System Analyst, Studio 103, The Business Center' input={this.onInput =
this.onInput.bind(this)} created={this.onCreate = this.onCreate.bind(this)}
placeholder='Enter your address' floatLabelType='Auto' ref={scope => {
this.textareaObj = scope; }} />
        </div>);
    }
    onCreate() {
        this.textareaObj.addAttributes({ rows: 1 });
    }
}

```

```

        this.textareaObj.respectiveElement.style.height = "auto";
        this.textareaObj.respectiveElement.style.height =
        (this.textareaObj.respectiveElement.scrollHeight) + "px";
    }
    onInput() {
        this.textareaObj.respectiveElement.style.height = "auto";
        this.textareaObj.respectiveElement.style.height =
        (this.textareaObj.respectiveElement.scrollHeight) + "px";
    }
}
ReactDOM.render(<App />, document.getElementById('default'));
{% endraw %}

```

## INDEX.TSX

```

{% raw %}
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from "react-dom";
export default class App extends React.Component<{}, {}> {
    public textareaObj: any;
    public render() {
        return (
            <div className="multiline">
                <TextBoxComponent multiline={true} value= 'Mr. Dodsworth Dodsworth,
System Analyst, Studio 103, The Business Center' input={this.onInput =
this.onInput.bind(this)} created={this.onCreate = this.onCreate.bind(this)}
placeholder='Enter your address' floatLabelType='Auto' ref = {scope =>
{this.textareaObj = scope }}/>
            </div>
        );
    }
    private onCreate(): void {
        this.textareaObj.addAttributes({rows: 1});
        this.textareaObj.respectiveElement.style.height = "auto";
        this.textareaObj.respectiveElement.style.height =
        (this.textareaObj.respectiveElement .scrollHeight)+"px";
    }
    private onInput(): void {
        this.textareaObj.respectiveElement.style.height = "auto";
        this.textareaObj.respectiveElement.style.height =
        (this.textareaObj.respectiveElement .scrollHeight)+"px";
    }
}
ReactDOM.render(<App />, document.getElementById('default'));
{% endraw %}

```

## [Functional-component]

## INDEX.JSX

```

{% raw %}
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from "react-dom";
function App() {

```

```

let textareaObj;
return (<div className="multiline">
  <TextBoxComponent multiline={true} value='Mr. Dodsworth Dodsworth,
System Analyst, Studio 103, The Business Center' input={onInput =
onInput.bind(this)} created={onCreate = onCreate.bind(this)}
placeholder='Enter your address' floatLabelType='Auto' ref={scope => {
textareaObj = scope; }}/>
</div>);
function onCreate() {
  textareaObj.addAttributes({ rows: 1 });
  textareaObj.respectiveElement.style.height = "auto";
  textareaObj.respectiveElement.style.height =
(textareaObj.respectiveElement.scrollHeight) + "px";
}
function onInput() {
  textareaObj.respectiveElement.style.height = "auto";
  textareaObj.respectiveElement.style.height =
(textareaObj.respectiveElement.scrollHeight) + "px";
}
}
ReactDOM.render(<App />, document.getElementById('default'));
{% endraw %}

```

## INDEX.TSX

```

{% raw %}
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from "react-dom";
function App() {
  let textareaObj: any;
  return (
    <div className="multiline">
      <TextBoxComponent multiline={true} value= 'Mr. Dodsworth Dodsworth,
System Analyst, Studio 103, The Business Center' input={onInput =
onInput.bind(this)} created={onCreate = onCreate.bind(this)}
placeholder='Enter your address' floatLabelType='Auto' ref = {scope =>
{textareaObj = scope } }/>
    </div>
  );
  function onCreate(): void {
    textareaObj.addAttributes({rows: 1});
    textareaObj.respectiveElement.style.height = "auto";
    textareaObj.respectiveElement.style.height =
(textareaObj.respectiveElement .scrollHeight)+"px";
  }
  function onInput(): void {
    textareaObj.respectiveElement.style.height = "auto";
    textareaObj.respectiveElement.style.height =
(textareaObj.respectiveElement .scrollHeight)+"px";
  }
}
ReactDOM.render(<App />, document.getElementById('default'));
{% endraw %}

```

### Disable resizing

By default, the multiline textbox is rendered with resizable. You can disable the resize of the multiline textbox by applying the following CSS styles.

```
`css
textarea.e-input,
.e-float-input textarea,
.e-float-input.e-control-wrapper textarea,
.e-input-group textarea,
.e-input-group.e-control-wrapper textarea {
resize: none;
}
```

### [Class-component]

#### INDEX.JSX

```
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from "react-dom";
export default class App extends React.Component {
  render() {
    return (<div className="multiline">
      <TextBoxComponent name="disable" multiline={true}
placeholder='Enter your address' floatLabelType='Auto' cssClass="sample"/>
    </div>);
  }
}
ReactDOM.render(<App />, document.getElementById('default'));
```

#### INDEX.TSX

```
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from "react-dom";
export default class App extends React.Component<{}, {}> {
  public render() {
    return (
      <div className="multiline">
        <TextBoxComponent name="disable" multiline={true}
placeholder='Enter your address' floatLabelType='Auto' cssClass="sample"/>
      </div>
    );
  }
}
ReactDOM.render(<App />, document.getElementById('default'));
```

### [Functional-component]

## INDEX.JSX

```
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from "react-dom";
function App() {
    return (<div className="multiline">
        <TextBoxComponent name="disable" multiline={true}
placeholder='Enter your address' floatLabelType='Auto' cssClass="sample"/>
        </div>);
}
ReactDOM.render(<App />, document.getElementById('default'));
```

## INDEX.TSX

```
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from "react-dom";
function App() {
    return (
        <div className="multiline">
            <TextBoxComponent name="disable" multiline={true}
placeholder='Enter your address' floatLabelType='Auto' cssClass="sample"/>
            </div>
    );
}
ReactDOM.render(<App />, document.getElementById('default'));
```

## Limit the text length

By default, the text length of the multiline textbox is unlimited. You can limit the text length by setting the `maxLength` attribute using the `addAttribute` method.

## [Class-component]

## INDEX.JSX

```
{% raw %}
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from "react-dom";
export default class App extends React.Component {
    textareaObj;
    render() {
        return (<div className="multiline">
            <label className="label">Add maxlength attribute through
inline</label>
            <TextBoxComponent multiline={true} maxLength={15}
placeholder='Enter your address' floatLabelType='Auto' />
            <label className="label">Add maxlength attribute through
addAttributes method</label>
            <TextBoxComponent multiline={true} ref={scope => {
this.textareaObj = scope; }} placeholder='Enter your address'
floatLabelType='Auto' />
            <button className='e-control e-btn' id='targetButton1'
role='button' onClick={this.handleClick = this.handleClick.bind(this)}>Add
max length</button>
        </div>);
    }
}
```

```

        </div>);
    }
    handleClick() {
        this.textareaObj.addAttributes({ maxLength: 15 });
    }
}
ReactDOM.render(<App />, document.getElementById('default'));
{% endraw %}

```

## INDEX.TSX

```

{% raw %}
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from "react-dom";
export default class App extends React.Component<{}, {}> {
    public textareaObj: any;
    public render() {
        return (
            <div className="multiline">
                <label className="label">Add maxlength attribute through
inline</label>
                <TextBoxComponent multiline={true} maxLength={'15'}
placeholder='Enter your address' floatLabelType='Auto' />
                <label className="label">Add maxlength attribute through
addAttributes method</label>
                <TextBoxComponent multiline={true} ref = {scope =>
{this.textareaObj = scope }} placeholder='Enter your address'
floatLabelType='Auto' />
                <button className='e-control e-btn' id='targetButton1'
role='button' onClick={this.handleClick = this.handleClick.bind(this)} >Add
max length</button>
            </div>
        );
    }
    private handleClick(): void {
        this.textareaObj.addAttributes({maxlength: 15});
    }
}
ReactDOM.render(<App />, document.getElementById('default'));
{% endraw %}

```

## [Functional-component]

## INDEX.JSX

```

{% raw %}
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from "react-dom";
function App() {
    let textareaObj;
    return (<div className="multiline">
        <label className="label">Add maxlength attribute through
inline</label>

```

```

        <TextBoxComponent multiline={true} maxLength={'15'}
placeholder='Enter your address' floatLabelType='Auto' />
        <label className="label">Add maxlength attribute through
addAttributes method</label>
        <TextBoxComponent multiline={true} ref={scope => { textareaObj =
scope; }} placeholder='Enter your address' floatLabelType='Auto' />
        <button className='e-control e-btn' id='targetButton1'
role='button' onClick={handleClick = handleClick.bind(this)}>Add max
length</button>
    </div>;
    function handleClick() {
        textareaObj.addAttributes({ maxLength: 15 });
    }
}
ReactDOM.render(<App />, document.getElementById('default'));
{% endraw %}

```

## INDEX.TSX

```

{% raw %}
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from "react-dom";
function App(){
    let textareaObj: any;
    return (
        <div className="multiline">
            <label className="label">Add maxlength attribute through
inline</label>
            <TextBoxComponent multiline={true} maxLength={'15'}
placeholder='Enter your address' floatLabelType='Auto' />
            <label className="label">Add maxlength attribute through
addAttributes method</label>
            <TextBoxComponent multiline={true} ref = {scope => {textareaObj =
scope }} placeholder='Enter your address' floatLabelType='Auto' />
            <button className='e-control e-btn' id='targetButton1'
role='button' onClick={handleClick = handleClick.bind(this)} >Add max
length</button>
        </div>
    );
    function handleClick(): void {
        textareaObj.addAttributes({maxLength: 15});
    }
}
ReactDOM.render(<App />, document.getElementById('default'));
{% endraw %}

```

## Count characters

You can show the number of characters entered inside the textarea by calculating the character count in the input event of multiline textbox. The character count is updated while entering or deleting any character inside the textarea. The character count shows how many characters can be entered or left to be entered.

[Class-component]

## INDEX.JSX

```
{% raw %}
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from "react-dom";
export default class App extends React.Component {
  textareaObj;
  render() {
    return (<div className="multiline">
      <TextBoxComponent multiline={true} input={this.onInput =
this.onInput.bind(this)} placeholder='Enter your address'
floatLabelType='Auto' maxLength='25' ref={scope => { this.textareaObj =
scope; }}/>
      <span id='numbercount' />
    </div>);
  }
  onInput() {
    let word;
    let addressCount;
    word = this.textareaObj.respectiveElement.value;
    addressCount = word.length;
    document.getElementById('numbercount').textContent = addressCount +
"/25";
  }
}
ReactDOM.render(<App />, document.getElementById('default'));
{% endraw %}
```

## INDEX.TSX

```
{% raw %}
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from "react-dom";
export default class App extends React.Component<{}, {}> {
  public textareaObj: any;
  public render() {
    return (
      <div className="multiline">
        <TextBoxComponent multiline={true} input={this.onInput =
this.onInput.bind(this)} placeholder='Enter your address'
floatLabelType='Auto' maxLength='25' ref = {scope => {this.textareaObj =
scope }}/>
        <span id='numbercount' />
      </div>
    );
  }
  private onInput(): void {
    let word: string;
    let addressCount: number;
    word = this.textareaObj.respectiveElement.value;
    addressCount = word.length;
    (document.getElementById('numbercount') as HTMLElement).textContent =
addressCount+"/25";
  }
}
```



```
}
ReactDOM.render(<App />, document.getElementById('default'));
{% endraw %}
```

## [Functional-component]

### INDEX.JSX

```
{% raw %}
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from "react-dom";
function App() {
  let textareaObj;
  return (<div className="multiline">
    <TextBoxComponent multiline={true} input={onInput =
onInput.bind(this)} placeholder='Enter your address' floatLabelType='Auto'
maxLength='25' ref={scope => { textareaObj = scope; }}/>
    <span id='numbercount' />
  </div>);
  function onInput() {
    let word;
    let addressCount;
    word = textareaObj.respectiveElement.value;
    addressCount = word.length;
    document.getElementById('numbercount').textContent = addressCount +
"/25";
  }
}
ReactDOM.render(<App />, document.getElementById('default'));
{% endraw %}
```

### INDEX.TSX

```
{% raw %}
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
import * as React from 'react';
import * as ReactDOM from "react-dom";
function App() {
  let textareaObj: any;
  return (
    <div className="multiline">
      <TextBoxComponent multiline={true} input={onInput =
onInput.bind(this)} placeholder='Enter your address' floatLabelType='Auto'
maxLength='25' ref = {scope => {textareaObj = scope; }}/>
      <span id='numbercount' />
    </div>
  );
  function onInput(): void {
    let word: string;
    let addressCount: number;
    word = textareaObj.respectiveElement.value;
    addressCount = word.length;
    (document.getElementById('numbercount') as HTMLElement).textContent =
addressCount+"/25";
  }
}
```

```
}
ReactDOM.render(<App />, document.getElementById('default'));
{% endraw %}
```

## Accessibility in React Textbox component

The Textbox component followed the accessibility guidelines and standards, including [ADA](#), [Section 508](#), [WCAG 2.2](#) standards, and [WCAG roles](#) that are commonly used to evaluate accessibility.

The accessibility compliance for the Textbox component is outlined below.

| Accessibility Criteria | Compatibility |

| -- | -- |

| [WCAG 2.2 Support](#) |  |

| [Section 508 Support](#) |  |

| [Screen Reader Support](#) |  |

| [Right-To-Left Support](#) |  |

| [Color Contrast](#) |  |

| [Mobile Device Support](#) |  |

| [Keyboard Navigation Support](#) |  |

| [Accessibility Checker Validation](#) |  |

| [Axe-core Accessibility Validation](#) |  |

<style>

.post .post-content img {

display: inline-block;

margin: 0.5em 0;

}

</style>

<div> - All features of the component meet the requirement.</div>

<div> - Some features of the component do not meet the requirement.</div>

<div> - The component does not meet the requirement.</div>

#### WAI-ARIA attributes

The Textbox is characterized with complete ARIA Accessibility support that helps to access through the on-screen readers and other assistive technology devices. This component is designed with the reference of the guidelines document given in [WAI ARAI Accessibility practices](#).

The Textbox uses the `textbox` role and following ARIA properties for its element based on its state.

Property	Functionality
aria-placeholder	The <code>aria-placeholder</code> is a short hint to help the users with data entry when the Textbox has no value.
aria-labelledby	The <code>aria-labelledby</code> property indicates the floating label element of the Textbox.

#### Ensuring accessibility

The Textbox component's accessibility levels are ensured through an [accessibility-checker](#) and [axe-core](#) software tools during automated testing.

The accessibility compliance of the Textbox component is shown in the following sample. Open the [sample](#) in a new window to evaluate the accessibility of the Textbox component with accessibility tools.

See also

- [Accessibility in Syncfusion React components](#)

#### React functional component in React Textbox component

This section explains how to render the textbox component in the functional component with react hooks methods. Please find the list of basic hook methods in the following table.

Hooks	Description
useState	useState is a Hook that allows you to define the state in the functional components. If you pass the initial state to this function, then it will return a state variable with value and function to update this state value. Here, useState is used to store the value of description field.
useEffect	useEffect is a Hook that allows you to execute code after rendering and re-rendering the component. Here, defined the set of rules for the name and email fields and initiated the form validation to those fields.
useRef	useRef is a Hook function that allows to directly create a reference to the DOM element in the functional component. Here, assigned the name field reference for focusing the field on initial loading.
useReducer	useReducer is a Hook function that accepts a reducer function and an initial state as argument. It returns a state value and another function to dispatch an action. Here, dispatched the update action to update the name and email state values.

The following example shows how to render a simple textbox component with react hooks.

**APP.JSX**

```

import { useState, useEffect, useRef, useReducer } from 'react';
import * as React from 'react';
import { FormValidator } from '@syncfusion/ej2-inputs';
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
let formObject;
function App() {
    const [description, setDescription] = useState('');
    const initialState = { name: '', email: '' };
    const userNameRef = useRef(null);
    const reducer = (state, action) => {
        switch (action.type) {
            case 'update':
                return { ...state, [action.field]: action.value };
            default:
                return initialState;
        }
    };
    const [state, dispatch] = useReducer(reducer, initialState);
    const changeHandler = (event) => {
        setDescription(event.value);
    };
    const update = (field) => (event) => {
        //update action is dispatched and triggered state changes
        dispatch({ type: 'update', field, value: event.value });
    };
    useEffect(() => {
        userNameRef.current.focusIn();
        const options = {
            // validation rules
            rules: {
                name: {
                    required: [true, '* Please enter your name'],
                },
                email: {
                    required: [true, '* Please enter your email'],
                },
                message: {
                    required: [true, '* Please enter your message'],
                },
            },
        };
        // Initialize the form validator
        formObject = new FormValidator('#form1', options);
    }, []);
    return (<div>
        <div className="control_wrapper" id="control_wrapper">
            <h3 className="form-title">User Detail</h3>
            <div className="control_wrapper textbox-form">
                <form id="form1" method="post">
                    <div className="form-group">
                        <TextBoxComponent ref={userNameRef} name="name"
value={state.name} change={update('name')} placeholder="Name"
floatLabelType="Auto" data-msg-containerid="errroForName"/>
                        <div id="errroForName"/>
                    </div>
                </form>
            </div>
        </div>
    </div>

```

```

        <div className="form-group">
            <TextBoxComponent type="email" name="email" value={state.email}
change={update('email')} placeholder="Email" floatLabelType="Auto" data-msg-
containerid="errroForEmail"/>
            <div id="errroForEmail"/>
        </div>
        <div className="form-group">
            <TextBoxComponent name="message" value={description}
change={changeHandler} multiline="true" placeholder="Description"
floatLabelType="Auto" data-msg-containerid="errroForMessage"/>
            <div id="errroForMessage"/>
        </div>
    </form>
</div>
<br />
<br />
</div>
</div>);
}
export default App;

```

## APP.TSX

```

import { useState, useEffect, useRef, useReducer } from 'react';
import * as React from 'react';
import { FormValidator } from '@syncfusion/ej2-inputs';
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
let formObject;
function App() {
    const [description, setDescription] = useState('');
    const initialState = { name: '', email: '' };
    const userNameRef = useRef(null);
    const reducer = (state, action) => {
        switch (action.type) {
            case 'update':
                return { ...state, [action.field]: action.value };
            default:
                return initialState;
        }
    };
    const [state, dispatch] = useReducer(reducer, initialState);
    const changeHandler = (event) => {
        setDescription(event.value);
    };
    const update = (field) => (event) => {
        //update action is dispatched and triggered state changes
        dispatch({ type: 'update', field, value: event.value });
    };
    useEffect(() => {
        userNameRef.current.focusIn();
        const options = {
            // validation rules
            rules: {
                name: {
                    required: [true, '* Please enter your name'],
                },
            },
        };
    });
}

```

```

    email: {
      required: [true, '* Please enter your email'],
    },
    message: {
      required: [true, '* Please enter your message'],
    },
  },
};
// Initialize the form validator
formObject = new FormValidator('#form1', options);
}, []);
return (
  <div>
    <div className="control_wrapper" id="control_wrapper">
      <h3 className="form-title">User Detail</h3>
      <div className="control_wrapper textbox-form">
        <form id="form1" method="post">
          <div className="form-group">
            <TextBoxComponent
              ref={userNameRef}
              name="name"
              value={state.name}
              change={update('name')}
              placeholder="Name"
              floatLabelType="Auto"
              data-msg-containerid="erroForName"
            />
            <div id="erroForName" />
          </div>
          <div className="form-group">
            <TextBoxComponent
              type="email"
              name="email"
              value={state.email}
              change={update('email')}
              placeholder="Email"
              floatLabelType="Auto"
              data-msg-containerid="erroForEmail"
            />
            <div id="erroForEmail" />
          </div>
          <div className="form-group">
            <TextBoxComponent
              name="message"
              value={description}
              change={changeHandler}
              multiline="true"
              placeholder="Description"
              floatLabelType="Auto"
              data-msg-containerid="erroForMessage"
            />
            <div id="erroForMessage" />
          </div>
        </form>
      </div>
    </div>
    <br />
    <br />
  </div>
);

```

```

    </div>
  </div>
);
}
export default App;

```

## Style appearance in React Textbox component

The following content provides the exact CSS structure that can be used to modify the control's appearance based on the user preference.

### Customizing the appearance of TextBox wrapper element

Use the following CSS to customize the appearance of wrapper element.

```
`css
```

*/ To specify height and font size /*

```
.e-input:not(:valid), .e-input:valid, .e-float-input.e-control-wrapper input:not(:valid), .e-float-input.e-control-wrapper input:valid, .e-float-input input:not(:valid), .e-float-input input:valid, .e-input-group input:not(:valid), .e-input-group input:valid, .e-input-group.e-control-wrapper input:not(:valid), .e-input-group.e-control-wrapper input:valid, .e-float-input.e-control-wrapper textarea:not(:valid), .e-float-input.e-control-wrapper textarea:valid, .e-float-input textarea:not(:valid), .e-float-input textarea:valid, .e-input-group.e-control-wrapper textarea:not(:valid), .e-input-group.e-control-wrapper textarea:valid, .e-input-group textarea:not(:valid), .e-input-group textarea:valid {
```

```
font-size: 30px;
```

```
height: 40px;
```

```
}
```

```
,
```

### Customizing the TextBox placeholder

Use the following CSS to customize the TextBox placeholder

```
`css
```

*/ To specify font size and color /*

```
.e-float-input.e-control-wrapper:not(.e-error) input:valid ~ label.e-float-text, .e-float-input.e-control-wrapper:not(.e-error) input ~ label.e-label-top.e-float-text {
```

```
color: pink;
```

```
font-size: 15px;
```

```
}
```

```
,
```

## How To

### Set the rounded corner in React Textbox component

Render the TextBox with **rounded corner** by adding the **e-corner** class to the input parent element.

### INDEX.JSX

```
import * as React from "react";
```

```
import * as ReactDOM from "react-dom";
export default class Default extends React.Component {
  render() {
    return (<div>
      <div className="e-input-group e-corner">
        <input className="e-input" type="text" placeholder="Enter Date"
onFocus={this.onInputFocus} onBlur={this.onInputBlur}/>
        <span className="e-input-group-icon e-input-popup-date"
onMouseDown={this.onIconMouseDown} onMouseUp={this.onIconMouseUp}/>
      </div>
      <div className="e-float-input e-input-group e-corner">
        <input type='text' required={true} onFocus={this.onInputFocus}
onBlur={this.onInputBlur}/>
        <span className="e-float-line"/>
        <label className="e-float-text">Enter Date </label>
        <span className="e-input-group-icon e-input-popup-date"
onMouseDown={this.onIconMouseDown} onMouseUp={this.onIconMouseUp}/>
      </div>
    </div>);
  }
  onInputFocus(args) {
    if (!args.target.parentElement.classList.contains('e-input-in-wrap'))
    {
      args.target.parentElement.classList.add('e-input-focus');
    }
    else {
      args.target.parentElement.parentElement.classList.add('e-input-
focus');
    }
  }
  onInputBlur(args) {
    if (!args.target.parentElement.classList.contains('e-input-in-wrap'))
    {
      args.target.parentElement.classList.remove('e-input-focus');
    }
    else {
      args.target.parentElement.parentElement.classList.remove('e-
input-focus');
    }
  }
  onIconMouseDown(args) {
    args.persist();
    setTimeout(() => {
      args.target.classList.add('e-input-btn-ripple');
    }, 300);
  }
  onIconMouseUp(args) {
    args.target.classList.remove('e-input-btn-ripple');
  }
}
ReactDOM.render(<Default />, document.getElementById('input-container'));
```

## INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
```



```

export default class Default extends React.Component {
  public render() {
    return (
      <div>
        <div className="e-input-group e-corner">
          <input className="e-input" type="text" placeholder="Enter Date"
onFocus = {this.onInputFocus} onBlur = {this.onInputBlur} />
          <span className="e-input-group-icon e-input-popup-date"
onMouseDown = { this.onIconMouseDown } onMouseUp = {this.onIconMouseUp}/>
        </div>
        <div className="e-float-input e-input-group e-corner">
          <input type='text' required = {true} onFocus =
{this.onInputFocus} onBlur = {this.onInputBlur} />
          <span className="e-float-line"/>
          <label className="e-float-text">Enter Date </label>
          <span className="e-input-group-icon e-input-popup-date"
onMouseDown = { this.onIconMouseDown } onMouseUp = {this.onIconMouseUp}/>
        </div>
      </div>
    );
  }
  public onInputFocus(args: React.FocusEvent) {
    if (!((args.target as HTMLElement).parentElement as
HTMLElement).classList.contains('e-input-in-wrap')) {
      ((args.target as HTMLElement).parentElement as
HTMLElement).classList.add('e-input-focus');
    } else {
      ((args.target as HTMLElement).parentElement as
HTMLElement).parentElement as HTMLElement).classList.add('e-input-focus')
    }
  }
  public onInputBlur(args: React.FocusEvent) {
    if (!((args.target as HTMLElement).parentElement as
HTMLElement).classList.contains('e-input-in-wrap')) {
      ((args.target as HTMLElement).parentElement as
HTMLElement).classList.remove('e-input-focus');
    } else {
      ((args.target as HTMLElement).parentElement as
HTMLElement).parentElement as HTMLElement).classList.remove('e-input-focus');
    }
  }
  public onIconMouseDown(args: React.MouseEvent) {
    args.persist();
    setTimeout(
      () => {
        (args.target as HTMLElement).classList.add('e-input-btn-ripple');
      },
      300);
  }
  public onIconMouseUp(args: React.MouseEvent) {
    (args.target as HTMLElement).classList.remove('e-input-btn-ripple');
  }
}
ReactDOM.render(<Default />, document.getElementById('input-container'));

```

### Set the disabled state in React Textbox component

Disable the TextBox by adding the `e-disabled` to the input parent element and set `disabled` attribute to the input element.

#### INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
export default class App extends React.Component {
  render() {
    return (
      <div>
        <input className="e-input" type="text" placeholder="Enter Name" disabled={true}/>
        <div className="e-float-input e-disabled">
          <input type='text' required={true} disabled={true}/>
          <span className="e-float-line"/>
          <label className="e-float-text">Enter Name</label>
        </div>
      </div>);
  }
};
ReactDOM.render(<App />, document.getElementById('input-container'));
```

#### INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
export default class App extends React.Component<{}, {}> {
  public render() {
    return (
      <div>
        <input className="e-input" type="text" placeholder="Enter Name" disabled={true}/>
        <div className="e-float-input e-disabled">
          <input type='text' required = {true} disabled = {true}/>
          <span className="e-float-line"/>
          <label className="e-float-text">Enter Name</label>
        </div>
      </div>
    );
  }
};
ReactDOM.render(<App />, document.getElementById('input-container'));
```

### Set the read only textbox in React Textbox component

You can make the TextBox as `read-only` by setting the `readonly` attribute to the input element.

#### INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
export default class Default extends React.Component {
  render() {
    return (
      <div>
```

```

        <input className="e-input" type="text" placeholder="Enter Name"
value="John" readOnly={true}/>
        <div className="e-float-input">
            <input type='text' required={true} readOnly={true} value="John"
onFocus={this.onInputFocus} onBlur={this.onInputBlur}/>
            <span className="e-float-line"/>
            <label className="e-float-text e-label-top">Enter Name</label>
        </div>
    </div>;
    }
    onInputFocus(args) {
        if (!args.target.parentElement.classList.contains('e-input-in-wrap'))
        {
            args.target.parentElement.classList.add('e-input-focus');
        }
        else {
            args.target.parentElement.parentElement.classList.add('e-input-
focus');
        }
    }
    onInputBlur(args) {
        if (!args.target.parentElement.classList.contains('e-input-in-wrap'))
        {
            args.target.parentElement.classList.remove('e-input-focus');
        }
        else {
            args.target.parentElement.parentElement.classList.remove('e-
input-focus');
        }
    }
    }
    ReactDOM.render(<Default />, document.getElementById('input-container'));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
export default class Default extends React.Component {
    public render() {
        return (
            <div>
                <input className="e-input" type="text" placeholder="Enter Name" value="John"
readOnly= {true}/>
                <div className="e-float-input">
                    <input type='text' required = {true} readOnly = {true} value="John"
onFocus = {this.onInputFocus} onBlur = {this.onInputBlur}/>
                    <span className="e-float-line"/>
                    <label className="e-float-text e-label-top">Enter Name</label>
                </div>
            </div>
        );
    }
    public onInputFocus(args: React.FocusEvent) {
        if (!((args.target as HTMLElement).parentElement as
HTMLElement).classList.contains('e-input-in-wrap')) {

```

```

        ((args.target as HTMLElement).parentElement as
HTMLElement).classList.add('e-input-focus');
    } else {
        ((args.target as HTMLElement).parentElement as
HTMLElement).parentElement as HTMLElement).classList.add('e-input-focus')
    }
}
public onInputBlur(args: React.FocusEvent) {
    if (!((args.target as HTMLElement).parentElement as
HTMLElement).classList.contains('e-input-in-wrap')) {
        ((args.target as HTMLElement).parentElement as
HTMLElement).classList.remove('e-input-focus');
    } else {
        ((args.target as HTMLElement).parentElement as
HTMLElement).parentElement as HTMLElement).classList.remove('e-input-focus');
    }
}
}
ReactDOM.render(<Default />, document.getElementById('input-container'));

```

### Add textbox programmatically in React Textbox component

- Create a TypeScript file and import the `Input` modules from `ej2-inputs` library as shown below.

`.ts`

```
import {Input} from '@syncfusion/ej2-inputs';
```

,

- Pass the `HTML Input` element as parameter to the `createInput` method.
- You can also add the `icons` on the input by passing `buttons` property value with the class name `e-input-group-icon` as parameter to the `createInput` method.

### INDEX.JSX

```

import { Input } from '@syncfusion/ej2-inputs';
import * as React from "react";
import * as ReactDOM from "react-dom";
export default class Default extends React.Component {
    input1;
    input2;
    render() {
        return (<div className="inner-container">
            <input id="input-01" type="text" ref={e1 => this.input1 =
e1}/>
            <input id="input-02" type="text" ref={e2 => this.input2 =
e2}/>
        </div>);
    }
    componentDidMount() {
        Input.createInput({
            element: this.input1,
            properties: {

```

```

        placeholder: 'Enter Name'
    }
    });
    Input.createInput({
        buttons: ['e-input-group-icon e-input-down'],
        element: this.input2,
        properties: {
            placeholder: 'Enter Value'
        }
    });
}
ReactDOM.render(<Default />, document.getElementById('input-container'));

```

## INDEX.TSX

```

import { Input } from '@syncfusion/ej2-inputs';
import * as React from "react";
import * as ReactDOM from "react-dom";
export default class Default extends React.Component {
    public input1: any;
    public input2: any;
    public render() {
        return (
            <div className="inner-container">
                <input id="input-01" type="text" ref = { e1 => this.input1 =
e1!} />
                <input id="input-02" type="text" ref = { e2 => this.input2 =
e2!} />
            </div>
        )
    }
    public componentDidMount() {
        Input.createInput({
            element: this.input1,
            properties:{
                placeholder:'Enter Name'
            }
        });
        Input.createInput({
            buttons: ['e-input-group-icon e-input-down'],
            element: this.input2,
            properties:{
                placeholder:'Enter Value'
            }
        });
    }
}
ReactDOM.render(<Default />, document.getElementById('input-container'));

```

Add floating label to textbox programmatically in React Textbox component

The **Floating Label TextBox** floats label above the TextBox after focusing, or entering a value in the TextBox.

Floating label supports the types of actions as given below.

Type | Description

Auto | The floating label will float above the input after focusing, or entering a value in the input.

Always | The floating label will always float above the input.

Never | By default, never float the label in the input when the placeholder is available.

- Create a TypeScript file and import the `Input` modules from `ej2-inputs` library as shown below.

`.ts`

```
import {Input} from '@syncfusion/ej2-inputs';
```

,

- Pass the `HTML Input` element and `floatLabelType` property as `Auto` to the `createInput` method.
- Set the `placeholder` value to the input element via `element attribute` or pass the parameter to the `createInput` method.

The watermark label will be updated based on the specified `placeholder` value of the `Floating Label TextBox`.

- You can add the `icons` on the input by passing `buttons` property value with the class name `e-input-group-icon` as parameter to the `createInput` method.

### INDEX.JSX

```
import { Input } from '@syncfusion/ej2-inputs';
import * as React from "react";
import * as ReactDOM from "react-dom";
export default class Default extends React.Component {
  input1;
  input2;
  input3;
  input4;
  render() {
    return (
      <div className="inner-container">
        <h4> FloatLabelType as Auto </h4>
        <input id="input-01" type="text" ref={ele1 => this.input1 =
ele1}/>
        <h4> FloatLabelType as Always </h4>
        <input id="input-02" type="text" ref={ele2 => this.input2 =
ele2}/>
        <h4> FloatLabelType as Never </h4>
        <input id="input-03" type="text" ref={ele3 => this.input3 =
ele3}/>
        <h4> Float label input with icons </h4>
        <input id="input-04" type="text" ref={ele4 => this.input4 =
ele4}/>
      </div>
    );
  }
  componentDidMount() {
```

```

        Input.createInput({
            element: this.input1,
            floatLabelType: "Auto",
            properties: {
                placeholder: 'Enter Name'
            }
        });
        Input.createInput({
            element: this.input2,
            floatLabelType: "Always",
            properties: {
                placeholder: 'Enter Name'
            }
        });
        Input.createInput({
            element: this.input3,
            floatLabelType: "Never",
            properties: {
                placeholder: 'Enter Name'
            }
        });
        // Render Float Label TextBox with Icon
        Input.createInput({
            buttons: ['e-input-group-icon e-input-down'],
            element: this.input4,
            floatLabelType: "Auto",
            properties: {
                placeholder: 'Enter Value'
            }
        });
    }
}
ReactDOM.render(<Default />, document.getElementById('input-container'));

```

## INDEX.TSX

```

import { Input } from '@syncfusion/ej2-inputs';
import * as React from "react";
import * as ReactDOM from "react-dom";
export default class Default extends React.Component {
    public input1: any;
    public input2: any;
    public input3: any;
    public input4: any;
    public render() {
        return (
            <div className="inner-container">
                <h4> FloatLabelType as Auto </h4>
                <input id="input-01" type="text" ref = {ele1 => this.input1 =
ele1!} />
                <h4> FloatLabelType as Always </h4>
                <input id="input-02" type="text" ref = {ele2 => this.input2 =
ele2!} />
                <h4> FloatLabelType as Never </h4>
                <input id="input-03" type="text" ref = {ele3 => this.input3 =
ele3!} />
            </div>
        );
    }
}

```

```

        <h4> Float label input with icons </h4>
        <input id="input-04" type="text" ref = {ele4 => this.input4 =
ele4!} />
    </div>
    )
  }
  public componentDidMount() {
    Input.createInput({
      element: this.input1,
      floatLabelType: "Auto",
      properties:{
        placeholder:'Enter Name'
      }
    });
    Input.createInput({
      element: this.input2,
      floatLabelType: "Always",
      properties:{
        placeholder:'Enter Name'
      }
    });
    Input.createInput({
      element: this.input3,
      floatLabelType: "Never",
      properties:{
        placeholder:'Enter Name'
      }
    });
    // Render Float Label TextBox with Icon
    Input.createInput({
      buttons: ['e-input-group-icon e-input-down'],
      element: this.input4,
      floatLabelType: "Auto",
      properties:{
        placeholder:'Enter Value'
      }
    });
  }
}
ReactDOM.render(<Default />, document.getElementById('input-container'));

```

### Change the floating label color of the textbox in React Textbox component

You can change the floating label color of the TextBox for both **success** and **warning** validation states by applying the following CSS styles.

```

`css
/ For Success state /
.e-float-input.e-success label.e-float-text,
.e-float-input.e-success input:focus ~ label.e-float-text,
.e-float-input.e-success input:valid ~ label.e-float-text {
color: #22b24b;

```



```

}
/ For Warning state /
.e-float-input.e-warning label.e-float-text,
.e-float-input.e-warning input:focus ~ label.e-float-text,
.e-float-input.e-warning input:valid ~ label.e-float-text {
color: #ffca1c;
}
,

```

## INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
export default class Default extends React.Component {
  render() {
    return (
      <div>
        <div className="e-float-input e-input-group e-success">
          <input type='text' required={true} onFocus={this.onInputFocus}
onBlur={this.onInputBlur}/>
          <span className="e-float-line"/>
          <label className="e-float-text">Success</label>
        </div>
        <div className="e-float-input e-input-group e-warning">
          <input type='text' required={true} onFocus={this.onInputFocus}
onBlur={this.onInputBlur}/>
          <span className="e-float-line"/>
          <label className="e-float-text">Warning</label>
        </div>
      </div>);
  }
  onInputFocus(args) {
    if (!args.target.parentElement.classList.contains('e-input-in-wrap'))
    {
      args.target.parentElement.classList.add('e-input-focus');
    }
    else {
      args.target.parentElement.parentElement.classList.add('e-input-
focus');
    }
  }
  onInputBlur(args) {
    if (!args.target.parentElement.classList.contains('e-input-in-wrap'))
    {
      args.target.parentElement.classList.remove('e-input-focus');
    }
    else {
      args.target.parentElement.parentElement.classList.remove('e-
input-focus');
    }
  }
}
ReactDOM.render(<Default />, document.getElementById('input-container'));

```

## INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
export default class Default extends React.Component {
  public render() {
    return (
      <div>
        <div className="e-float-input e-input-group e-success">
          <input type='text' required = {true} onFocus =
{this.onInputFocus} onBlur = {this.onInputBlur}/>
          <span className="e-float-line"/>
          <label className="e-float-text">Success</label>
        </div>
        <div className="e-float-input e-input-group e-warning">
          <input type='text' required = {true} onFocus =
{this.onInputFocus} onBlur = {this.onInputBlur}/>
          <span className="e-float-line"/>
          <label className="e-float-text">Warning</label>
        </div>
      </div>
    );
  }
  public onInputFocus(args: React.FocusEvent) {
    if (!((args.target as HTMLElement).parentElement as
HTMLElement).classList.contains('e-input-in-wrap')) {
      ((args.target as HTMLElement).parentElement as
HTMLElement).classList.add('e-input-focus');
    } else {
      ((args.target as HTMLElement).parentElement as
HTMLElement).parentElement as HTMLElement).classList.add('e-input-focus')
    }
  }
  public onInputBlur(args: React.FocusEvent) {
    if (!((args.target as HTMLElement).parentElement as
HTMLElement).classList.contains('e-input-in-wrap')) {
      ((args.target as HTMLElement).parentElement as
HTMLElement).classList.remove('e-input-focus');
    } else {
      ((args.target as HTMLElement).parentElement as
HTMLElement).parentElement as HTMLElement).classList.remove('e-input-focus');
    }
  }
}
ReactDOM.render(<Default />, document.getElementById('input-container'));
```

Change the color of the textbox based on its value in React Textbox component

You can change the color of the TextBox by validating its value using regular expression in the **keyup** event for predicting the numeric values as demonstrated in the following code sample.

## INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
```

```

export default class App extends React.Component {
  constructor(props) {
    super(props);
    this.onKeyUp = this.onKeyUp.bind(this);
  }
  onKeyUp(e) {
    const str = e.target.value.match(/^[\d-]+$/);
    if (!((str && str.length > 0)) && e.target.value.length) {
      e.target.parentNode.classList.add('e-error');
    }
    else {
      e.target.parentNode.classList.remove('e-error');
    }
  }
  onInputFocus(args) {
    if (!args.target.parentNode.classList.contains('e-input-in-wrap'))
    {
      args.target.parentNode.classList.add('e-input-focus');
    }
    else {
      args.target.parentNode.parentNode.classList.add('e-input-
focus');
    }
  }
  onInputBlur(args) {
    if (!args.target.parentNode.classList.contains('e-input-in-wrap'))
    {
      args.target.parentNode.classList.remove('e-input-focus');
    }
    else {
      args.target.parentNode.parentNode.classList.remove('e-
input-focus');
    }
  }
  render() {
    return (<div className="wrap">
      <label> Normal Input </label>
      <div className="e-input-group">
        <input className="e-input" id="numericOnly" type="text"
placeholder="Enter numeric values" onKeyUp={this.onKeyUp}
onFocus={this.onInputFocus} onBlur={this.onInputBlur}/>
      </div>
      <label> Floating Input </label>
      <div className="e-float-input e-input-group">
        <input type="text" onKeyUp={this.onKeyUp} required={true}
onFocus={this.onInputFocus} onBlur={this.onInputBlur}/>
        <span className="e-float-line"/>
        <label className="e-float-text">Enter numeric
values</label>
      </div>
    </div>);
  }
}
;
ReactDOM.render(<App />, document.getElementById('input-container'));

```

# INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
export default class App extends React.Component<{}, {}> {
  constructor(props: any) {
    super(props);
    this.onKeyUp = this.onKeyUp.bind(this);
  }
  public onKeyUp(e: React.KeyboardEvent) {
    const str = (e.target as HTMLInputElement).value.match(/^[0-9]+$/);
    if (!((str && str.length > 0)) && (e.target as
HTMLInputElement).value.length) {
      ((e.target as HTMLInputElement).parentNode as
HTMLInputElement).classList.add('e-error');
    } else {
      ((e.target as HTMLInputElement).parentNode as
HTMLInputElement).classList.remove('e-error');
    }
  }
  public onInputFocus(args: React.FocusEvent) {
    if (!((args.target as HTMLInputElement).parentElement as
HTMLInputElement).classList.contains('e-input-in-wrap')) {
      ((args.target as HTMLInputElement).parentElement as
HTMLInputElement).classList.add('e-input-focus');
    } else {
      ((args.target as HTMLInputElement).parentElement as
HTMLInputElement).parentElement as HTMLInputElement).classList.add('e-input-focus')
    }
  }
  public onInputBlur(args: React.FocusEvent) {
    if (!((args.target as HTMLInputElement).parentElement as
HTMLInputElement).classList.contains('e-input-in-wrap')) {
      ((args.target as HTMLInputElement).parentElement as
HTMLInputElement).classList.remove('e-input-focus');
    } else {
      ((args.target as HTMLInputElement).parentElement as
HTMLInputElement).parentElement as HTMLInputElement).classList.remove('e-input-focus');
    }
  }
  public render() {
    return (
      <div className="wrap">
        <label> Normal Input </label>
        <div className="e-input-group">
          <input className="e-input" id="numericOnly" type="text"
placeholder="Enter numeric values" onKeyUp={this.onKeyUp} onFocus =
{this.onInputFocus} onBlur = {this.onInputBlur}/>
        </div>
        <label> Floating Input </label>
        <div className="e-float-input e-input-group">
          <input type="text" onKeyUp={this.onKeyUp} required =
{true} onFocus = {this.onInputFocus} onBlur = {this.onInputBlur}/>
          <span className="e-float-line"/>
          <label className="e-float-text">Enter numeric
values</label>
        </div>
      </div>
    );
  }
}
```

```

        </div>
    )
  }
};
ReactDOM.render(<App />, document.getElementById('input-container'));

```

### Add floating label to read only textbox in React Textbox component

You can achieve floating label for read-only textboxes by adding/removing `e-label-top` and `e-label-bottom` classes to the label element

In this sample, click the update value button to fill the read-only textbox with value and float a label.

#### INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
export default class Default extends React.Component {
  valueBtn;
  removeBtn;
  inputElement;
  render() {
    return (
      <div className="wrap">
        <div className="row">
          <div className="e-float-input">
            <input
              className="e-input myField"
              onFocus={this.floatFocus}
              onBlur={this.floatBlur}
              ref={v => this.inputElement = v}
              type="text"
              id="myText"
              readOnly={true}
            />
            <span className="e-float-line" />
            <label
              className="e-float-text"
              >Enter value</label>
          </div>
        </div>
        <div className="row">
          <button
            className="e-btn update_value"
            id="valuebtn"
            onClick={this.updateBtnClick}
            ref={btn => this.valueBtn = btn}
            >Set value</button>
          <button
            className="e-btn remove_value"
            id="removebtn"
            onClick={this.removeBtnClick}
            ref={removeBtn => this.removeBtn = removeBtn}
            >Remove value</button>
        </div>
      </div>
    );
  }
  updateBtnClick = () => {
    this.inputElement.value = '10';
    this.checkFloatingLabel('myText');
  };
  removeBtnClick = () => {
    this.inputElement.value = '';
    this.checkFloatingLabel('myText');
  };
  checkFloatingLabel(id) {
    const inputElement = document.getElementById(id);
    const labelElement = inputElement.parentElement.querySelector('.e-float-text');
    if (inputElement.value !== '') {
      labelElement.classList.remove('e-label-bottom');
      labelElement.classList.add('e-label-top');
    }
  }
}

```

```

    }
    else {
      labelElement.classList.remove('e-label-top');
      labelElement.classList.add('e-label-bottom');
    }
  }
  floatFocus(args) {
    args.target.parentElement.classList.add('e-input-focus');
  }
  floatBlur(args) {
    args.target.parentElement.classList.remove('e-input-focus');
  }
}
ReactDOM.render(<Default />, document.getElementById('input-container'));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
export default class Default extends React.Component {
  public valueBtn: HTMLButtonElement;
  public removeBtn: HTMLButtonElement;
  public inputElement: HTMLInputElement;
  public render() {
    return (
      <div className="wrap">
        <div className="row">
          <div className="e-float-input">
            <input className="e-input myField" onFocus= {this.floatFocus}
onBlur= {this.floatBlur} ref = { v => this.inputElement = v!} type="text"
id="myText" readOnly={true} />
            <span className="e-float-line" />
            <label className="e-float-text">Enter value</label>
          </div>
        </div>
        <div className="row">
          <button className="e-btn update_value" id="valuebtn" onClick = {
this.updateBtnClick} ref = { btn => this.valueBtn = btn! }>Set value</button>
          <button className="e-btn remove_value" id="removebtn" onClick = {
this.removeBtnClick} ref = { removeBtn => this.removeBtn = removeBtn!}>Remove
value</button>
        </div>
      </div>
    );
  }
  public updateBtnClick = (): void => {
    this.inputElement.value = '10';
    this.checkFloatingLabel('myText')
  }
  public removeBtnClick = (): void => {
    this.inputElement.value = '';
    this.checkFloatingLabel('myText')
  }
  public checkFloatingLabel(id: any): void {
    const inputElement: HTMLInputElement = document.getElementById(id) as
HTMLInputElement;

```

```
const labelElement: HTMLInputElement = (inputElement as
any).parentElement.querySelector('.e-float-text') as HTMLInputElement;
if (inputElement.value !== '') {
    labelElement.classList.remove('e-label-bottom');
    labelElement.classList.add('e-label-top');
} else {
    labelElement.classList.remove('e-label-top');
    labelElement.classList.add('e-label-bottom');
}
}
public floatFocus(args: any): void {
    args.target.parentElement.classList.add('e-input-focus');
}
public floatBlur(args: any): void {
    args.target.parentElement.classList.remove('e-input-focus');
}
}
ReactDOM.render(<Default />, document.getElementById('input-container'));
```

### Customize the textbox background color and text color in React Textbox component

You can customize the React textbox styles such as background-color, text-color and border-color by overriding its default styles.

To change the styles of the **floating label**, you must override the style to the input element.

#### INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
export default class App extends React.Component {
    constructor(props) {
        super(props);
    }
    render() {
        return (<div className="wrap">
            <label> Normal Input </label>
            <div className="e-input-group">
                <input className="e-input" onFocus={this.floatFocus}
onBlur={this.floatBlur} type="text" placeholder="First Name"/>
            </div>
            <label> Floating Input </label>
            <div className="e-float-input">
                <input type="text" required={true}/>
                <span className="e-float-line"/>
                <label className="e-float-text">Last Name</label>
            </div>
        </div>);
    }
    floatFocus(args) {
        args.target.parentElement.classList.add('e-input-focus');
    }
    floatBlur(args) {
        args.target.parentElement.classList.remove('e-input-focus');
    }
}
ReactDOM.render(<App />, document.getElementById('input-container'));
```

## INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
export default class App extends React.Component<{}, {}> {
  constructor(props: any) {
    super(props);
  }
  public render() {
    return (
      <div className="wrap">
        <label> Normal Input </label>
        <div className="e-input-group">
          <input className="e-input" onFocus= {this.floatFocus} onBlur=
{this.floatBlur} type="text" placeholder="First Name"/>
        </div>
        <label> Floating Input </label>
        <div className="e-float-input">
          <input type="text" required={true} />
          <span className="e-float-line"/>
          <label className="e-float-text">Last Name</label>
        </div>
      </div>
    );
  }
  private floatFocus(args: React.FocusEvent): void {
    ((args.target as HTMLElement).parentElement as
HTMLElement).classList.add('e-input-focus');
  }
  private floatBlur(args: React.FocusEvent): void {
    ((args.target as HTMLElement).parentElement as
HTMLElement).classList.remove('e-input-focus');
  }
}
ReactDOM.render(<App />, document.getElementById('input-container'));
```

## Migration from css textbox to react textbox in React Textbox component

From v16.3.21 version, the textbox is provided as React component to achieve the floating label textbox with minimal code. You can find the available textbox properties, methods, and events in the [API reference](#).

The following table describes the migration from CSS textbox to React textbox component.

### Normal textbox

<!-- markdownlint-disable MD038 -->

Rendering mode	CSS textbox	React textbox component
Default rendering	<div class='e-input-group'> <input class='e-input' type='text' placeholder='Enter Value'/> </div>	<TextBoxComponent id="default" placeholder="Enter Value" floatLabelType="Never" />



| Textbox with clear button | `<div class='e-input-group'><br/><input class='e-input' placeholder='Enter Value' required='true'><br/><span class='e-clear-icon'></span><br/></div><br/><br/>`Note: You have to write action for clear button. | `<TextBoxComponent id="clear-input" placeholder="Enter Value" showClearButton={true} floatLabelType="Never" /> |`

| Validation states | `<div class='e-input-group e-error'><br/><input class='e-input' type='text' placeholder='Enter Value' /><br/></div><br/><br/>`Note: Textbox component consists of three types of validation rules such as success, warning, and error. | `<TextBoxComponent id="validation" placeholder="Enter Value" cssClass="e-error" floatLabelType="Never" /> |`

### Floating label textbox

`<!-- markdownlint-disable MD038 -->`

#### | Rendering mode | CSS textbox | React textbox component |

| ----- | ----- | ----- |

| Default rendering | `<div class='e-float-input'><br/><input type='text' required /><br/><span class='e-float-line'></span><br/><label class='e-float-text'>Enter Value</label><br/></div> | <TextBoxComponent id="default" placeholder="Enter Value" floatLabelType="Auto" /> |`

| Textbox with clear button | `<div class='e-float-input e-input-group'><br/><input type='text' required value='Clear Input' /><br/><span class='e-float-line'></span><br/><label class='e-float-text'>Enter Value</label><br/><span class='e-clear-icon'></span><br/></div><br/><br/>`Note: You have to write action for clear button. | `<TextBoxComponent id="clear-input" placeholder="Enter Value" showClearButton={true} floatLabelType="Auto" /> |`

| Validation states | `<div class='e-float-input e-error'><br/><input type='text' required /><br/><span class='e-float-line'></span><br/><label class='e-float-text'>Enter Value</label><br/></div><br/><br/>`Note: Textbox component consists of three types of validation rules such as success, warning, and error. | `<TextBoxComponent id="validation" placeholder="Enter Value" cssClass="e-error" floatLabelType="Auto" /> |`

## Timeline

### Getting Started

This section explains how to create a simple Timeline and configure its available functionalities in the React environment.

### Dependencies

The following list of dependencies is required to use the Timeline component in your application.

``js`

|-- @syncfusion/ej2-react-layouts

|-- @syncfusion/ej2-base

|-- @syncfusion/ej2-layouts

|-- @syncfusion/ej2-react-base

,

### Installation and Configuration

You can use [Create-react-app](#) to setup the applications. To install `create-react-app` run the following command.

```
`bash
```

```
npm install -g create-react-app
```

,

Start a new project using create-react-app command as follows

```
<div class='tsx'>
```

```
`bash
```

```
create-react-app quickstart --scripts-version=react-scripts-ts
```

```
cd quickstart
```

,

```
</div>
```

```
<div class='jsx'>
```

```
`bash
```

```
create-react-app quickstart
```

```
cd quickstart
```

,

```
</div>
```

'react-scripts-ts' is used for creating React app with typescript.

### Adding Syncfusion packages

All the available Essential JS 2 packages are published in [npmjs.com](https://www.npmjs.com) public registry.

To install `Timeline` component, use the following command

```
`bash
```

```
npm install @syncfusion/ej2-react-layouts --save
```

,

### Adding CSS Reference

Import the Timeline component's required CSS references as follows in `src/App.css`.

```
`css
```

```
@import "../node_modules/@syncfusion/ej2-base/styles/material.css";
```

```
@import "../node_modules/@syncfusion/ej2-layouts/styles/material.css";
```

,

### Adding Timeline component to the application

To include the Timeline component in your application, import the `TimelineComponent` from `ej2-react-layouts` package in the `App.tsx`.

To define each item, you can use the `ItemDirective` tag inside the `ItemsDirective` tag as shown in below code example.

```
{% raw %}
`ts
import { TimelineComponent, ItemsDirective, ItemDirective } from '@syncfusion/ej2-react-layouts';
import * as React from "react";
import * as ReactDOM from "react-dom";
import "./App.css";
function App() {
  return (
    <div id='timeline' style={{ height: "350px" }}>
      <TimelineComponent>
        <ItemsDirective>
          <ItemDirective />
          <ItemDirective />
          <ItemDirective />
          <ItemDirective />
          <ItemDirective />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
const root = ReactDOM.createRoot(document.getElementById("element"));
root.render(<App />);
`
{% endraw %}
```

### Running the application

Run the application in the browser using the following command:

```
`
npm start
`
```

The following example shows a basic Timeline component.

#### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
'@syncfusion/ej2-react-layouts';
function App() {
    return (
        <div id='timeline' style={{ height: "350px" }}>
            <TimelineComponent>
                <ItemsDirective>
                    <ItemDirective />
                    <ItemDirective />
                    <ItemDirective />
                    <ItemDirective />
                </ItemsDirective>
            </TimelineComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

#### INDEX.TSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
'@syncfusion/ej2-react-layouts';
function App() {
    return (
        <div id='timeline' style={{ height: "320px" }}>
            <TimelineComponent>
                <ItemsDirective>
                    <ItemDirective />
                    <ItemDirective />
                    <ItemDirective />
                    <ItemDirective />
                </ItemsDirective>
            </TimelineComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

### Items in React Timeline component

The Timeline items can be added by using the `ItemDirective` tag. Each item can be configured with options such as `content`, `oppositeContent`, `dotCss`, `disabled` and `cssClass`.

### Adding content

You can define the item content using the [content](#) property.

#### String content

You can define string content for the Timeline items.

#### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
"@syncfusion/ej2-react-layouts";
function App() {
  return (
    <div id='timeline' style={{ height: "350px" }}>
      <TimelineComponent>
        <ItemsDirective>
          <ItemDirective content='Shipped' />
          <ItemDirective content='Departed' />
          <ItemDirective content='Arrived' />
          <ItemDirective content='Out for Delivery' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

#### INDEX.TSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
"@syncfusion/ej2-react-layouts";
function App() {
  return (
    <div id='timeline' style={{ height: "330px" }}>
      <TimelineComponent>
        <ItemsDirective>
          <ItemDirective content='Shipped' />
          <ItemDirective content='Departed' />
          <ItemDirective content='Arrived' />
          <ItemDirective content='Out for Delivery' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

**INDEX.CSS**

```
/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
```

## Adding opposite content

You can add additional information to each Timeline item, by using the [oppositeContent](#) property which is positioned opposite to the item content.

**INDEX.JSX**

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
'@syncfusion/ej2-react-layouts';
function App() {
  return (
    <div id='timeline' style={{ height: "350px" }}>
      <TimelineComponent>
        <ItemsDirective>
          <ItemDirective content='Breakfast' oppositeContent='8:00
AM' />
          <ItemDirective content='Lunch' oppositeContent='1:00 PM'
/>
          <ItemDirective content='Dinner' oppositeContent='8:00 PM'
/>
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

**INDEX.TSX**

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
'@syncfusion/ej2-react-layouts';
function App() {
  return (
    <div id='timeline' style={{ height: "330px" }}>
      <TimelineComponent>
        <ItemsDirective>
```

```

        <ItemDirective content='Breakfast' oppositeContent='8:00
AM' />
        <ItemDirective content='Lunch' oppositeContent='1:00 PM'
/>
        <ItemDirective content='Dinner' oppositeContent='8:00 PM'
/>
    </ItemsDirective>
  </TimelineComponent>
</div>
);
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

### Dot item

You can define CSS class to set icons, background colors, or images to personalize the appearance of dots associated with each Timeline item by using the [dotCss](#) property.

#### Adding icons

You can define the CSS class to show the icon for each item using the `dotCss` property.

#### Adding images

You can include images for the Timeline items using the `dotCss` property, by setting the CSS `background-image` property.

#### Adding text

You can display text for the Timeline items using the `dotCss` property, by adding text to the CSS `content` property.

### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
'@syncfusion/ej2-react-layouts';
function App() {
  return (
    <div id='timeline' style={{ height: "350px" }}>
      <TimelineComponent>
        <ItemsDirective>
          <ItemDirective content='Default' />
          <ItemDirective content='Icon' dotCss='e-icons e-check' />
          <ItemDirective content='Image' dotCss='custom-image' />
          <ItemDirective content='Text' dotCss='custom-text' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

**INDEX.TSX**

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
'@syncfusion/ej2-react-layouts';
function App() {
    return (
        <div id='timeline' style={{ height: "330px" }}>
            <TimelineComponent>
                <ItemsDirective>
                    <ItemDirective content='Default' />
                    <ItemDirective content='Icon' dotCss='e-icons e-check' />
                    <ItemDirective content='Image' dotCss='custom-image' />
                    <ItemDirective content='Text' dotCss='custom-text' />
                </ItemsDirective>
            </TimelineComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

**INDEX.CSS**

```
/* Represents the styles for loader */
#loader {
    color: #008cff;
    height: 40px;
    left: 45%;
    position: absolute;
    top: 45%;
    width: 30%;
}
.e-dot.custom-image {
    background-image: url('./dot-image.png');
}
.e-dot.custom-text::before {
    content: 'A';
}
```

## Disabling items

You can use the [disabled](#) property to disable an item when set to `true`. By default, the value is `false`.

**INDEX.JSX**

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
'@syncfusion/ej2-react-layouts';
function App() {
    return (
        <div id='timeline' style={{ height: "350px" }}>
```



```

        <TimelineComponent>
          <ItemsDirective>
            <ItemDirective content='Eat' />
            <ItemDirective content='Code' />
            <ItemDirective content='Repeat' disabled={true} />
          </ItemsDirective>
        </TimelineComponent>
      </div>
    );
  }
  export default App;
  ReactDOM.render(<App />, document.getElementById("element"));
  {% endraw %}

```

### INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
'@syncfusion/ej2-react-layouts';
function App() {
  return (
    <div id='timeline' style={{ height: "330px" }}>
      <TimelineComponent>
        <ItemsDirective>
          <ItemDirective content='Eat' />
          <ItemDirective content='Code' />
          <ItemDirective content='Repeat' disabled={true} />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

### Css class

You can customize the appearance of the Timeline item by specifying a custom CSS class using the [cssClass](#) property.

### Orientations in React Timeline component

The Timeline component supports the display of items in both horizontal and vertical direction by using the [orientation](#) property.

#### Vertical

You can display the items one below the other vertically by setting the [orientation](#) property to **Vertical**. By default, the items are displayed in vertical orientation.

### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";

```

```
import { TimelineComponent, ItemsDirective, ItemDirective } from
'@syncfusion/ej2-react-layouts';
function App() {
  return (
    <div id='timeline' style={{ height: "350px" }}>
      <TimelineComponent orientation='Vertical'>
        <ItemsDirective>
          <ItemDirective content='Day 1, 4:00 PM'
oppositeContent='Check-in and campsite visit' />
          <ItemDirective content='Day 1, 7:00 PM'
oppositeContent='Dinner with music' />
          <ItemDirective content='Day 2, 5:30 AM'
oppositeContent='Sunrise between mountains' />
          <ItemDirective content='Day 2, 8:00 AM'
oppositeContent='Breakfast and check-out' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

**INDEX.TSX**

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
'@syncfusion/ej2-react-layouts';
function App() {
  return (
    <div id='timeline' style={{ height: "330px" }}>
      <TimelineComponent orientation='Vertical'>
        <ItemsDirective>
          <ItemDirective content='Day 1, 4:00 PM'
oppositeContent='Check-in and campsite visit' />
          <ItemDirective content='Day 1, 7:00 PM'
oppositeContent='Dinner with music' />
          <ItemDirective content='Day 2, 5:30 AM'
oppositeContent='Sunrise between mountains' />
          <ItemDirective content='Day 2, 8:00 AM'
oppositeContent='Breakfast and check-out' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

## Horizontal

In horizontal orientation, the items are displayed in a side-by-side manner by setting the [orientation](#) property to `Horizontal`.

### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
'@syncfusion/ej2-react-layouts';
function App() {
  return (
    <div id='timeline'>
      <TimelineComponent orientation='Horizontal'>
        <ItemsDirective>
          <ItemDirective content='Day 1, 4:00 PM'
oppositeContent='Check-in and campsite visit' />
          <ItemDirective content='Day 1, 7:00 PM'
oppositeContent='Dinner with music' />
          <ItemDirective content='Day 2, 5:30 AM'
oppositeContent='Sunrise between mountains' />
          <ItemDirective content='Day 2, 8:00 AM'
oppositeContent='Breakfast and check-out' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

### INDEX.TSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
'@syncfusion/ej2-react-layouts';
function App() {
  return (
    <div id='timeline'>
      <TimelineComponent orientation='Horizontal'>
        <ItemsDirective>
          <ItemDirective content='Day 1, 4:00 PM'
oppositeContent='Check-in and campsite visit' />
          <ItemDirective content='Day 1, 7:00 PM'
oppositeContent='Dinner with music' />
          <ItemDirective content='Day 2, 5:30 AM'
oppositeContent='Sunrise between mountains' />
          <ItemDirective content='Day 2, 8:00 AM'
oppositeContent='Breakfast and check-out' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
```

```

    );
  }
  export default App;
  ReactDOM.render(<App />, document.getElementById("element"));
  {% endraw %}

```

## Alignment in React Timeline component

You can display the Timeline content **Before**, **After**, **Alternate** and **AlternateReverse** by using the [align](#) property. The oppositeContent will be displayed parallel to the content when configured in the **ItemDirective**.

### Before

In **Before** alignment, for **horizontal** orientation the item content is placed at the top and oppositeContent at the bottom whereas in **vertical**, the content to the left and oppositeContent to the right.

### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
"@syncfusion/ej2-react-layouts";
function App() {
  return (
    <div id='timeline' style={{ height: "350px" }}>
      <TimelineComponent align='Before'>
        <ItemsDirective>
          <ItemDirective content='ReactJs' oppositeContent='Owned
by Facebook' />
          <ItemDirective content='Angular' oppositeContent='Owned
by Google' />
          <ItemDirective content='VueJs' oppositeContent='Owned by
Evan you' />
          <ItemDirective content='Svelte' oppositeContent='Owned by
Rich Harris' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

### INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
"@syncfusion/ej2-react-layouts";
function App() {
  return (

```

```

        <div id='timeline' style={{ height: "330px" }}>
            <TimelineComponent align='Before'>
                <ItemsDirective>
                    <ItemDirective content='ReactJs' oppositeContent='Owned
by Facebook' />
                    <ItemDirective content='Angular' oppositeContent='Owned
by Google' />
                    <ItemDirective content='VueJs' oppositeContent='Owned by
Evan you' />
                    <ItemDirective content='Svelte' oppositeContent='Owned by
Rich Harris' />
                </ItemsDirective>
            </TimelineComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

### After

In [After](#) alignment, for **horizontal** orientation the item content is placed at the bottom and oppositeContent at the top whereas in **vertical**, the content to the right and oppositeContent to the left.

### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
'@syncfusion/ej2-react-layouts';
function App() {
    return (
        <div id='timeline' style={{ height: "350px" }}>
            <TimelineComponent align='After'>
                <ItemsDirective>
                    <ItemDirective content='ReactJs' oppositeContent='Owned
by Facebook' />
                    <ItemDirective content='Angular' oppositeContent='Owned
by Google' />
                    <ItemDirective content='VueJs' oppositeContent='Owned by
Evan you' />
                    <ItemDirective content='Svelte' oppositeContent='Owned by
Rich Harris' />
                </ItemsDirective>
            </TimelineComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

### INDEX.TSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
'@syncfusion/ej2-react-layouts';
function App() {
  return (
    <div id='timeline' style={{ height: "330px" }}>
      <TimelineComponent align='After'>
        <ItemsDirective>
          <ItemDirective content='ReactJs' oppositeContent='Owned
by Facebook' />
          <ItemDirective content='Angular' oppositeContent='Owned
by Google' />
          <ItemDirective content='VueJs' oppositeContent='Owned by
Evan you' />
          <ItemDirective content='Svelte' oppositeContent='Owned by
Rich Harris' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

### Alternate

In [Alternate](#) alignment, the item content are arranged alternatively regardless of the Timeline orientation.

### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
'@syncfusion/ej2-react-layouts';
function App() {
  return (
    <div id='timeline' style={{ height: "350px" }}>
      <TimelineComponent align='Alternate'>
        <ItemsDirective>
          <ItemDirective content='ReactJs' oppositeContent='Owned
by Facebook' />
          <ItemDirective content='Angular' oppositeContent='Owned
by Google' />
          <ItemDirective content='VueJs' oppositeContent='Owned by
Evan you' />
          <ItemDirective content='Svelte' oppositeContent='Owned by
Rich Harris' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
{% endraw %}
```

```
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

**INDEX.TSX**

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
"@syncfusion/ej2-react-layouts";
function App() {
  return (
    <div id='timeline' style={{ height: "330px" }}>
      <TimelineComponent align='Alternate'>
        <ItemsDirective>
          <ItemDirective content='ReactJs' oppositeContent='Owned
by Facebook' />
          <ItemDirective content='Angular' oppositeContent='Owned
by Google' />
          <ItemDirective content='VueJs' oppositeContent='Owned by
Evan you' />
          <ItemDirective content='Svelte' oppositeContent='Owned by
Rich Harris' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

**Alternate reverse**

In [AlternateReverse](#) alignment, the item content are arranged in reverse alternate regardless of the Timeline orientation.

**INDEX.JSX**

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
"@syncfusion/ej2-react-layouts";
function App() {
  return (
    <div id='timeline' style={{ height: "350px" }}>
      <TimelineComponent align='Alternatereverse'>
        <ItemsDirective>
          <ItemDirective content='ReactJs' oppositeContent='Owned
by Facebook' />
          <ItemDirective content='Angular' oppositeContent='Owned
by Google' />
          <ItemDirective content='VueJs' oppositeContent='Owned by
Evan you' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

```

                                <ItemDirective content='Svelte' oppositeContent='Owned by
Rich Harris' />
                                </ItemsDirective>
                            </TimelineComponent>
                        </div>
                    );
                }
    export default App;
    ReactDOM.render(<App />, document.getElementById("element"));
    {% endraw %}

```

## INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
'@syncfusion/ej2-react-layouts';
function App() {
    return (
        <div id='timeline' style={{ height: "330px" }}>
            <TimelineComponent align='Alternatereverse'>
                <ItemsDirective>
                    <ItemDirective content='ReactJs' oppositeContent='Owned
by Facebook' />
                    <ItemDirective content='Angular' oppositeContent='Owned
by Google' />
                    <ItemDirective content='VueJs' oppositeContent='Owned by
Evan you' />
                    <ItemDirective content='Svelte' oppositeContent='Owned by
Rich Harris' />
                </ItemsDirective>
            </TimelineComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

## Reverse in React Timeline component

You can display the Timeline items in reverse order, for different alignments by using the [reverse](#) property which provides adaptability and improves user interaction.

## INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
'@syncfusion/ej2-react-layouts';
function App() {
    const careerProgress = [
        "Graduated \n Bachelors in Computer Engineering",
        "Software Engineering Internship \n ABC Software and Technology",
    ];
}

```



```

        "Associate Software Engineer \n ABC Software and Technology",
        "Software Level 1 Engineer \n XYZ Solutions"
    ];
    return (
        <div id='timeline' style={{ height: "350px" }}>
            <TimelineComponent align='before' reverse={true}>
                <ItemsDirective>
                    <ItemDirective content='June 2022' oppositeContent =
{careerProgress[0]} />
                    <ItemDirective content='Aug 2022' oppositeContent =
{careerProgress[1]} />
                    <ItemDirective content='Feb 2023' oppositeContent =
{careerProgress[2]} />
                    <ItemDirective content='Mar 2024' oppositeContent =
{careerProgress[3]} />
                </ItemsDirective>
            </TimelineComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

## INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
'@syncfusion/ej2-react-layouts';
function App() {

    const careerProgress = [
        "Graduated \n Bachelors in Computer Engineering",
        "Software Engineering Internship \n ABC Software and Technology",
        "Associate Software Engineer \n ABC Software and Technology",
        "Software Level 1 Engineer \n XYZ Solutions"
    ];
    return (
        <div id='timeline' style={{ height: "330px" }}>
            <TimelineComponent align='before' reverse={true}>
                <ItemsDirective>
                    <ItemDirective content='June 2022' oppositeContent =
{careerProgress[0]} />
                    <ItemDirective content='Aug 2022' oppositeContent =
{careerProgress[1]} />
                    <ItemDirective content='Feb 2023' oppositeContent =
{careerProgress[2]} />
                    <ItemDirective content='Mar 2024' oppositeContent =
{careerProgress[3]} />
                </ItemsDirective>
            </TimelineComponent>
        </div>
    );
}
}

```

```
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

## Template in React Timeline component

The Timeline component allows you to customize the appearance for each item by using the [template](#) to modify the dot items, templated contents, progress bar styling and more.

The `template` context receives the following information:

| Type | Purpose |

| --- | --- |

| `item` | Indicates the current data of the Timeline item. |

| `itemIndex` | Indicates the current index of the Timeline item. |

### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
'@syncfusion/ej2-react-layouts';
function App() {
  const projectMilestones = [
    { content: 'Kickoff meeting' },
    { content: 'Content approved' },
    { content: 'Design approved' },
    { content: 'Product delivered' }
  ];
  const getTemplate = (props) => {
    return (
      <div className={`template-container item-${props.itemIndex}`>
        <div className="content-container">
          <div className="timeline-content"> {props.item.content}
        </div>
        <div className="content-connector"></div>
        <div className="progress-line">
          <span className="indicator"></span>
        </div>
      </div>
    );
  }
  return (
    <div id='timeline' style={{ height: "150px", width: "600px", margin:
'50px auto' }}>
      <TimelineComponent cssClass='custom-timeline'
        template={getTemplate} orientation='Horizontal' >
        <ItemsDirective>
          {projectMilestones.map((item, index) => {
            return <ItemDirective key={index}
              content={item.content} />
          })}
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
```

```

        </TimelineComponent>
      </div>
    );
  }
  export default App;
  ReactDOM.render(<App />, document.getElementById("element"));
  {% endraw %}

```

## INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective, TimelineItemModel } from '@syncfusion/ej2-react-layouts';
function App() {

  const projectMilestones: TimelineItemModel [] = [
    { content: 'Kickoff meeting'},
    { content: 'Content approved'},
    { content: 'Design approved'},
    { content: 'Product delivered'}
  ];

  const getTemplate = (props: any) => {
    return (
      <div className={`template-container item-${props.itemIndex}`}>
        <div className="content-container">
          <div className="timeline-content"> {props.item.content}
        </div>
        </div>
        <div className="content-connector"></div>
        <div className="progress-line">
          <span className="indicator"></span>
        </div>
      </div>
    );
  }

  return (
    <div id='timeline' style={{ height: "150px", width: "600px", margin: '50px auto' }}>
      <TimelineComponent cssClass='custom-timeline'
        template={getTemplate} orientation='Horizontal' >
        <ItemsDirective>
          {projectMilestones.map((item, index) => {
            return <ItemDirective key={index}
              content={item.content} />
          })}
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

**INDEX.CSS**

```

/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
.custom-timeline .e-timeline-item.e-item-template {
  align-items: flex-end;
}
.custom-timeline .e-timeline-items {
  justify-content: center;
}
.template-container .content-connector {
  position: absolute;
  left: 88%;
  width: 3px;
  height: 28px;
}
.template-container .content-container {
  padding: 8px;
  border-width: 1px;
  border-style: solid;
}
.content-container .timeline-content {
  font-size: 14px;
}
/* Color customizations - Progress line, connector line, dot border */
.item-0 .progress-line, .item-0 .content-connector { background-color:
rgb(233, 93, 93); }
.item-1 .progress-line, .item-1 .content-connector { background-color:
rgba(247, 179, 22, 0.907); }
.item-2 .progress-line, .item-2 .content-connector { background-color:
rgb(60, 184, 60); }
.item-3 .progress-line, .item-3 .content-connector { background-color:
rgb(153, 29, 230); }
.item-0 .progress-line .indicator, .item-0 .content-container { border-color:
rgb(233, 93, 93); }
.item-1 .progress-line .indicator, .item-1 .content-container { border-color:
rgba(247, 179, 22, 0.907); }
.item-2 .progress-line .indicator, .item-2 .content-container { border-color:
rgb(60, 184, 60); }
.item-3 .progress-line .indicator, .item-3 .content-container { border-color:
rgb(153, 29, 230); }
.item-0 .content-container { box-shadow: 2px 2px 8px rgb(233, 93, 93); }
.item-1 .content-container { box-shadow: 2px 2px 8px rgba(247, 179, 22,
0.907); }
.item-2 .content-container { box-shadow: 2px 2px 8px rgb(60, 184, 60); }
.item-3 .content-container { box-shadow: 2px 2px 8px rgb(153, 29, 230); }
/* START --- Customizing Dot and progress line */
.custom-timeline .template-container .indicator {

```

```

position: absolute;
width: 25px;
height: 25px;
border-radius: 50%;
background-color: #fff;
border-width: 6px;
border-style: solid;
left: 88%;
transform: translate(-50%, -40%);
cursor: pointer;
}
.progress-line {
position: absolute;
height: 10px;
width: 100%;
left: 0;
top: 50%;
}
/* END --- Customizing Icon and progress line */

```

## Customization in React Timeline component

You can customize the Timeline items dot size, connectors, dot borders, dot outer space and more to personalize its appearance. This section explains the different ways for styling the items.

### Connector styling

#### Common styling

You can define the styles applicable to the all the Timeline item connectors.

#### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
"@syncfusion/ej2-react-layouts";
function App() {
  return (
    <div id='timeline' style={{ height: "350px" }}>
      <TimelineComponent cssClass='custom-connector'>
        <ItemsDirective>
          <ItemDirective content='Eat' />
          <ItemDirective content='Code' />
          <ItemDirective content='Repeat' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

#### INDEX.TSX

```

{% raw %}

```

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
 '@syncfusion/ej2-react-layouts';
function App() {
    return (
        <div id='timeline' style={{ height: "330px" }}>
            <TimelineComponent cssClass='custom-connector'>
                <ItemsDirective>
                    <ItemDirective content='Eat' />
                    <ItemDirective content='Code' />
                    <ItemDirective content='Repeat' />
                </ItemsDirective>
            </TimelineComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

## INDEX.CSS

```
/* Represents the styles for loader */
#loader {
    color: #008cff;
    height: 40px;
    left: 45%;
    position: absolute;
    top: 45%;
    width: 30%;
}
.custom-connector .e-timeline-item.e-connector::after {
    border-color: #f7c867;
    border-width: 1.4px;
    animation: progress 2s;
}
@keyframes progress {
    from { height: 0%; }
    to { height: 100%; }
}
```

## Individual styling

You can also apply unique styles to individual connectors, to differentiate specific items within the timeline.

## INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
 '@syncfusion/ej2-react-layouts';
function App() {
    return (
```

```

    <div id='timeline' style={{ height: "350px" }}>
      <TimelineComponent cssClass='custom-connector'>
        <ItemsDirective>
          <ItemDirective content='Eat' cssClass='state-initial' />
          <ItemDirective content='Code' cssClass='state-
intermediate' />
          <ItemDirective content='Repeat' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

**INDEX.TSX**

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
'@syncfusion/ej2-react-layouts';
function App() {
  return (
    <div id='timeline' style={{ height: "330px" }}>
      <TimelineComponent cssClass='custom-connector'>
        <ItemsDirective>
          <ItemDirective content='Eat' cssClass='state-initial' />
          <ItemDirective content='Code' cssClass='state-
intermediate' />
          <ItemDirective content='Repeat' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

**INDEX.CSS**

```

/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
.custom-connector .state-initial.e-connector::after {
  border: 1.5px #f8c050 dashed;
  animation: progress 1.5s ease-in-out;
}

```

```
.custom-connector .state-intermediate.e-connector::after {
  border: 1.5px #4d85f5 dashed;
  animation: progress 1.5s ease-in-out;
}
@keyframes progress {
  from { height: 0%; }
  to { height: 100%; }
}
```

## Dot styling

### Dot color

You can modify the color of the dots to highlight the specific Timeline items.

### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
"@syncfusion/ej2-react-layouts";
function App() {
  return (
    <div id='timeline' style={{ height: "350px" }}>
      <TimelineComponent cssClass='dot-color'>
        <ItemsDirective>
          <ItemDirective content='Ordered' cssClass='state-
completed' />
          <ItemDirective content='Shipped' cssClass='state-
progress' />
          <ItemDirective content='Delivered' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

### INDEX.TSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
"@syncfusion/ej2-react-layouts";
function App() {
  return (
    <div id='timeline' style={{ height: "330px" }}>
      <TimelineComponent cssClass='dot-color'>
        <ItemsDirective>
          <ItemDirective content='Ordered' cssClass='state-
completed' />
          <ItemDirective content='Shipped' cssClass='state-
progress' />

```



```

        <ItemDirective content='Delivered' />
      </ItemsDirective>
    </TimelineComponent>
  </div>
);
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

## INDEX.CSS

```

/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
.dot-color .state-completed .e-dot {
  background: #ff9900;
  outline: 1px dashed #ff9900;
  border-color: #ff9900;
}
.dot-color .state-progress .e-dot {
  background: #33cc33;
  outline: 1px dashed #33cc33;
  border-color: #33cc33;
}

```

## Dot size

You can adjust the size of the dot to make it larger or smaller by using the `--dot-size` variable.

## INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
'@syncfusion/ej2-react-layouts';
function App() {
  return (
    <div id='timeline' style={{ height: "350px" }}>
      <TimelineComponent cssClass='dot-size'>
        <ItemsDirective>
          <ItemDirective content='Ordered' cssClass='x-small' />
          <ItemDirective content='Shipped' cssClass='small' />
          <ItemDirective content='Delivered' cssClass='medium' />
          <ItemDirective content='Delivered' cssClass='large' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}

```

```
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

### INDEX.TSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
'@syncfusion/ej2-react-layouts';
function App() {
  return (
    <div id='timeline' style={{ height: "330px" }}>
      <TimelineComponent cssClass='dot-size'>
        <ItemsDirective>
          <ItemDirective content='Ordered' cssClass='x-small' />
          <ItemDirective content='Shipped' cssClass='small' />
          <ItemDirective content='Delivered' cssClass='medium' />
          <ItemDirective content='Delivered' cssClass='large' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

### INDEX.CSS

```
/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
.dot-size .e-dot {
  background: #33cc33;
}
.dot-size .x-small .e-dot {
  --dot-size: 12px;
}
.dot-size .small .e-dot {
  --dot-size: 18px;
}
.dot-size .medium .e-dot {
  --dot-size: 24px;
}
.dot-size .large .e-dot {
  --dot-size: 30px;
}
```

### Dot shadow

You can add shadow effects to the Timeline dots to make it feel visually engaging by using the `--dot-outer-space` & `--dot-border` variables.

#### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
'@syncfusion/ej2-react-layouts';
function App() {
  return (
    <div id='timeline' style={{ height: "350px" }}>
      <TimelineComponent cssClass='dot-shadow'>
        <ItemsDirective>
          <ItemDirective content='Ordered' />
          <ItemDirective content='Shipped' />
          <ItemDirective content='Delivered' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

#### INDEX.TSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
'@syncfusion/ej2-react-layouts';
function App() {
  return (
    <div id='timeline' style={{ height: "330px" }}>
      <TimelineComponent cssClass='dot-shadow'>
        <ItemsDirective>
          <ItemDirective content='Ordered' />
          <ItemDirective content='Shipped' />
          <ItemDirective content='Delivered' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

#### INDEX.CSS

```

/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
.dot-shadow .e-dot {
  --dot-outer-space: 3px;
  --dot-border: 3px;
  --dot-size: 20px;
  outline-color: #dee2e6;
  border-color: #fff;
  box-shadow: 3px 3px 10px rgba(0, 0, 0, 0.5) ,
    2px -2px 4px rgba(255, 255, 255, 0.5) inset;
}

```

### Dot variant

You can achieve the desired dot variant by customizing the border, outline and background colors of the Timeline dots.

### INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
"@syncfusion/ej2-react-layouts";
function App() {
  return (
    <div id='timeline' style={{ height: "350px" }}>
      <TimelineComponent cssClass='dot-variant'>
        <ItemsDirective>
          <ItemDirective content='Filled' cssClass='dot-filled' />
          <ItemDirective content='Flat' cssClass='dot-flat' />
          <ItemDirective content='Outlined' cssClass='dot-outlined' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

### INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
"@syncfusion/ej2-react-layouts";

```

```
function App() {
  return (
    <div id='timeline' style={{ height: "330px" }}>
      <TimelineComponent cssClass='dot-variant'>
        <ItemsDirective>
          <ItemDirective content='Filled' cssClass='dot-filled' />
          <ItemDirective content='Flat' cssClass='dot-flat' />
          <ItemDirective content='Outlined' cssClass='dot-outlined' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

## INDEX.CSS

```
/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;
  width: 30%;
}
.dot-variant .dot-filled .e-dot::before { content: 'A'; color: #fff;}
.dot-variant .dot-flat .e-dot::before { content: 'B'; color: #fff;}
.dot-variant .dot-outlined .e-dot::before { content: 'C';}
.dot-variant .dot-filled .e-dot {
  background: #33cc33;
  --dot-outer-space: 3px;
  outline-color: #81ff05;
  --dot-size: 25px;
}
.dot-variant .dot-flat .e-dot {
  background: #33cc33;
  --dot-size: 25px;
  --dot-radius: 10%;
}
.dot-variant .dot-outlined .e-dot {
  outline-color: #33cc33;
  --dot-outer-space: 3px;
  background-color: unset;
  --dot-size: 25px;
}
```

### Dot outline

By adding the `e-outline` class to the Timeline `cssClass` property, it enables the dots to have an outline state.

## INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
'@syncfusion/ej2-react-layouts';
function App() {
  return (
    <div id='timeline' style={{ height: "350px" }}>
      <TimelineComponent cssClass='e-outline'>
        <ItemsDirective>
          <ItemDirective content='Shipped' />
          <ItemDirective content='Departed' />
          <ItemDirective content='Arrived' />
          <ItemDirective content='Out for Delivery' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

**INDEX.TSX**

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
'@syncfusion/ej2-react-layouts';
function App() {
  return (
    <div id='timeline' style={{ height: "330px" }}>
      <TimelineComponent cssClass='e-outline'>
        <ItemsDirective>
          <ItemDirective content='Shipped' />
          <ItemDirective content='Departed' />
          <ItemDirective content='Arrived' />
          <ItemDirective content='Out for Delivery' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

**INDEX.CSS**

```
/* Represents the styles for loader */
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
```

```
top: 45%;
width: 30%;
}
```

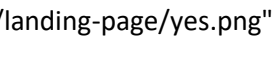
### Accessibility in React Timeline component

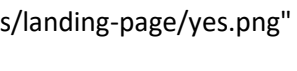
The Timeline component followed the accessibility guidelines and standards, including [ADA, Section 508](#), [WCAG 2.2](#) standards, and [WCAG roles](#) that are commonly used to evaluate accessibility.

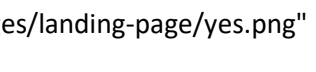
The accessibility compliance for the Timeline component is outlined below.

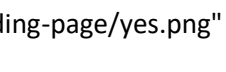
| Accessibility Criteria | Compatibility |

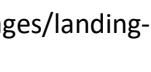
| -- | -- |

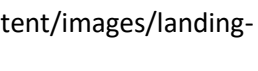
| [WCAG 2.2](#) Support |  |

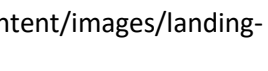
| [Section 508](#) Support |  |

| Right-To-Left Support |  |

| Color Contrast |  |

| Mobile Device Support |  |

| [Accessibility Checker](#) Validation |  |

| [Axe-core](#) Accessibility Validation |  |

<style>

.post .post-content img {

display: inline-block;

margin: 0.5em 0;

}

</style>

<div> - All features of the component meet the requirement.</div>

<div> - Some features of the component do not meet the requirement.</div>

<div> - The component does not meet the requirement.</div>

### WAI-ARIA attributes

The following ARIA attributes are used in the Timeline component:

| Attributes | Purpose |

| --- | --- |

| **role=navigation** | Specified its purpose as a navigational element. |

| **aria-label** | Provides an accessible name for an element when a visible label is not present |

### Ensuring accessibility

The Timeline component's accessibility levels are ensured through an [accessibility-checker](#) and [axe-core](#) software tools during automated testing.

See also

- [Accessibility in Syncfusion React components](#)

## Events in React Timeline component

This section describes the Timeline events that will be triggered when an appropriate actions are performed. The following events are available in the Timeline component.

### created

The Timeline component triggers the [created](#) event when the component rendering is completed.

### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
"@syncfusion/ej2-react-layouts";
function App() {
  return (
    <div id='timeline' style={{ height: "350px" }}>
      <TimelineComponent created={() => { /* Your required action here
*/ }}>
        <ItemsDirective>
          <ItemDirective content='Planning' />
          <ItemDirective content='Developing' />
          <ItemDirective content='Testing' />
          <ItemDirective content='Launch' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

### INDEX.TSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
"@syncfusion/ej2-react-layouts";
```



```
function App() {
  return (
    <div id='timeline' style={{ height: "330px" }}>
      <TimelineComponent created={() => { /* Your required action here
*/ }}>
        <ItemsDirective>
          <ItemDirective content='Planning' />
          <ItemDirective content='Developing' />
          <ItemDirective content='Testing' />
          <ItemDirective content='Launch' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

### beforeItemRender

The Timeline component triggers the [beforeItemRenderLink to the Video](#) event before rendering each item.

### INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { TimelineComponent, ItemsDirective, ItemDirective } from
'@syncfusion/ej2-react-layouts';
function App() {
  return (
    <div id='timeline' style={{ height: "350px" }}>
      <TimelineComponent beforeItemRender={(args) => { /* Your required
action here */ }}>
        <ItemsDirective>
          <ItemDirective content='Planning' />
          <ItemDirective content='Developing' />
          <ItemDirective content='Testing' />
          <ItemDirective content='Launch' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

### INDEX.TSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
```

```
import { TimelineComponent, ItemsDirective, ItemDirective,
TimelineRenderingEventArgs } from '@syncfusion/ej2-react-layouts';
function App() {
  return (
    <div id='timeline' style={{ height: "330px" }}>
      <TimelineComponent beforeItemRender={ (args:
TimelineRenderingEventArgs) => { /* Your required action here */ }}>
        <ItemsDirective>
          <ItemDirective content='Planning' />
          <ItemDirective content='Developing' />
          <ItemDirective content='Testing' />
          <ItemDirective content='Launch' />
        </ItemsDirective>
      </TimelineComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}
```

## TimePicker

### Getting started

This section explains you the steps required to create a simple TimePicker and demonstrate the basic usage of the TimePicker component.

To get start quickly with React TimePicker, you can check on this video:

### Dependencies

The below list of dependencies are required to use the TimePicker component in your application.

```
`javascript
|-- @syncfusion/ej2-react-calendars
|-- @syncfusion/ej2-react-base
|-- @syncfusion/ej2-base
|-- @syncfusion/ej2-data
|-- @syncfusion/ej2-calendars
|-- @syncfusion/ej2-inputs
|-- @syncfusion/ej2-splitbuttons
|-- @syncfusion/ej2-lists
|-- @syncfusion/ej2-popups
|-- @syncfusion/ej2-buttons
`
```

### Installation and configuration

You can use [create-react-app](#) to setup the applications. To install `create-react-app` run the following command.

`

```
npm install -g create-react-app
```

`

- To setup basic `React` sample use following commands.

`

```
create-react-app quickstart --scripts-version=react-scripts-ts
```

```
cd quickstart
```

`

### Adding Syncfusion packages

All the available Essential JS 2 packages are published in [npmjs.com](#) public registry. You can choose the component that you want to install. For this application, we are going to use `TimePicker` component.

To install `TimePicker` component, use the following command

```
`bash
```

```
npm install @syncfusion/ej2-react-calendars --save
```

`

### Adding Style sheet to the Application

To render the `TimePicker` component, need to import `TimePicker` and its dependent component's styles as given below in `App.css`.

```
`css
```

```
@import "../node_modules/@syncfusion/ej2-base/styles/material.css";
```

```
@import "../node_modules/@syncfusion/ej2-inputs/styles/material.css";
```

```
@import "../node_modules/@syncfusion/ej2-popups/styles/material.css";
```

```
@import "../node_modules/@syncfusion/ej2-lists/styles/material.css";
```

```
@import "../node_modules/@syncfusion/ej2-react-calendars/styles/material.css";
```

`

Note: If you want to refer the combined component styles, please make use of our [CRG](#) (Custom Resource Generator) in your application.

### Adding TimePicker component to the Application

- To include the `TimePicker` component in application import the `TimePickerComponent` from `ej2-react-calendars` package in `App.tsx`.
- Then add the `TimePicker` component as shown in below code example.

[src/App.tsx]

[Class-component]

```
`ts
import { enableRipple } from '@syncfusion/ej2-base';
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import './App.css';
// enable ripple effect
enableRipple(true);
export default class App extends React.Component<{}, {}> {
  public render() {
    return <TimePickerComponent id="time" />
  }
};
`
```

[Functional-component]

```
`ts
import { enableRipple } from '@syncfusion/ej2-base';
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import './App.css';
// enable ripple effect
enableRipple(true);
function App() {
  return <TimePickerComponent id="time" />
};
`
```

### Run the application

Now run the `npm start` command in the console, it will run your application and open the browser window.

```
npm start
```

The following examples shows the basic TimePicker component.

### [Class-component]

#### INDEX.JSX

```
// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
export default class App extends React.Component {
  render() {
    return <TimePickerComponent id="timepicker" placeholder="Select a
Time"/>;
  }
}
;
ReactDOM.render(<App />, document.getElementById('timer'));
```

#### INDEX.TSX

```
// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
export default class App extends React.Component<{}, {}> {
  public render() {
    return <TimePickerComponent id="timepicker" placeholder="Select a
Time" />
  }
};
ReactDOM.render(<App />, document.getElementById('timer'));
```

### [Functional-component]

#### INDEX.JSX

```
// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
function App() {
  return <TimePickerComponent id="timepicker" placeholder="Select a
Time"/>;
}
```

```
;
ReactDOM.render(<App />, document.getElementById('timer'));
```

### INDEX.TSX

```
// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
function App() {
    return <TimePickerComponent id="timepicker" placeholder="Select a Time" />
};
ReactDOM.render(<App />, document.getElementById('timer'));
```

Now, the TimePicker renders with default culture as **American English**('en-US'). For a different culture, refer to the

[Globalization](#) section.

### Setting the value, min, and max time

The following example demonstrates how to set the value, min and max time on initializing the TimePicker. Here the TimePicker allows you to select the time value within a range from **7:00 AM** to **4:00 PM**.

[Class-component]

### INDEX.JSX

```
// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
export default class App extends React.Component {
    // initialize the value, min and max time
    time = (new Date('8/3/2017 10:00 AM'));
    minTime = (new Date('8/3/2017 7:00 AM'));
    maxTime = (new Date('8/3/2017 4:00 PM'));
    render() {
        return <TimePickerComponent id="timepicker" value={this.time}
min={this.minTime} max={this.maxTime}/>;
    }
};
ReactDOM.render(<App />, document.getElementById('timer'));
```

### INDEX.TSX

```
// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
export default class App extends React.Component<{}, {}> {
    // initialize the value, min and max time
    private time: Date = (new Date('8/3/2017 10:00 AM'));
    private minTime: Date = (new Date('8/3/2017 7:00 AM'));
    private maxTime: Date = (new Date('8/3/2017 4:00 PM'));
    public render() {
        return <TimePickerComponent id="timepicker" value={this.time}
min={this.minTime} max={this.maxTime} />
    }
};
ReactDOM.render(<App />, document.getElementById('timer'));
```

## [Functional-component]

### INDEX.JSX

```
// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
function App() {
    // initialize the value, min and max time
    const time = (new Date('8/3/2017 10:00 AM'));
    const minTime = (new Date('8/3/2017 7:00 AM'));
    const maxTime = (new Date('8/3/2017 4:00 PM'));
    return <TimePickerComponent id="timepicker" value={time} min={minTime}
max={maxTime}/>;
}
;
ReactDOM.render(<App />, document.getElementById('timer'));
```

### INDEX.TSX

```
// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
function App() {
    // initialize the value, min and max time
    const time: Date = (new Date('8/3/2017 10:00 AM'));

```

```

const minTime: Date = (new Date('8/3/2017 7:00 AM'));
const maxTime: Date = (new Date('8/3/2017 4:00 PM'));
return <TimePickerComponent id="timepicker" value={time} min={minTime}
max={maxTime} />
};
ReactDOM.render(<App />, document.getElementById('timer'));

```

### Setting the time format

Time formats is a way of representing the time value in different string format in textbox and popup list. By default, the TimePicker's format is based on the culture. You can also customize the format by using the [format](#) property. To know more about the time format standards, refer to the [Date and Time Format](#) section.

The following example demonstrates the TimePicker component in 24 hours format with 60 minutes interval. The time interval is set to 60 minutes by using the [step](#) property.

#### [Class-component]

##### INDEX.JSX

```

// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
export default class App extends React.Component {
  time = (new Date('8/3/2017 10:00'));
  render() {
    return <TimePickerComponent id="timepicker" value={this.time}
step={60} format={'HH:mm'} />;
  }
}
;
ReactDOM.render(<App />, document.getElementById('timer'));

```

##### INDEX.TSX

```

// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
export default class App extends React.Component<{}, {}> {
  private time: Date = (new Date('8/3/2017 10:00'));
  public render() {
    return <TimePickerComponent id="timepicker" value={this.time}
step={60} format={'HH:mm'} />
  }
};

```



```
ReactDOM.render(<App />, document.getElementById('timer'));
```

### [Functional-component]

#### INDEX.JSX

```
// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
function App() {
    const time = (new Date('8/3/2017 10:00'));
    return <TimePickerComponent id="timepicker" value={time} step={60}
format={'HH:mm'} />;
}
;
ReactDOM.render(<App />, document.getElementById('timer'));
```

#### INDEX.TSX

```
// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
function App() {
    const time: Date = (new Date('8/3/2017 10:00'));
    return <TimePickerComponent id="timepicker" value={time} step={60}
format={'HH:mm'} />
};
ReactDOM.render(<App />, document.getElementById('timer'));
```

Once the time format property is defined, it will be applicable to all the cultures.

See Also

- [Render TimePicker with min and max time](#)
- [How to achieve validation with TimePicker](#)
- [Render TimePicker with specific culture](#)

### Time range in React Timepicker component

TimePicker provides an option to select a time value within a specified range by using the [min](#) and [max](#) properties. The min value should always be lesser than the max value.

When the min and max properties are configured and the selected time value is out-of-range or invalid, then the model value will be set to **out of range** time value or **null** respectively with highlighted **error** class to indicates the time is out of range or invalid.

The value property depends on the min/max with respect to [strictMode](#) property. The following example allows you to select a time value within a range of **9:00 AM** to **11:30 AM**.

#### [Class-component]

##### INDEX.JSX

```
// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
export default class App extends React.Component {
    // initialize the min and max time value
    minTime = (new Date('8/3/2017 9:00 AM'));
    maxTime = (new Date('8/3/2017 11:30 AM'));
    render() {
        return <TimePickerComponent id="timepicker" placeholder="Select a
Time" min={this.minTime} max={this.maxTime}/>;
    }
}
;
ReactDOM.render(<App />, document.getElementById('timer'));
```

##### INDEX.TSX

```
// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
export default class App extends React.Component<{}, {}> {
    // initialize the min and max time value
    private minTime: Date = (new Date('8/3/2017 9:00 AM'));
    private maxTime: Date = (new Date('8/3/2017 11:30 AM'));
    public render() {
        return <TimePickerComponent id="timepicker" placeholder="Select a
Time" min={this.minTime} max={this.maxTime} />
    }
};
ReactDOM.render(<App />, document.getElementById('timer'));
```

#### [Functional-component]

##### INDEX.JSX

```
// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
function App() {
    // initialize the min and max time value
    const minTime = (new Date('8/3/2017 9:00 AM'));
    const maxTime = (new Date('8/3/2017 11:30 AM'));
    return <TimePickerComponent id="timepicker" placeholder="Select a Time"
min={minTime} max={maxTime}/>;
}
;
ReactDOM.render(<App />, document.getElementById('timer'));
```

**INDEX.TSX**

```
// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
function App() {
    // initialize the min and max time value
    const minTime: Date = (new Date('8/3/2017 9:00 AM'));
    const maxTime: Date = (new Date('8/3/2017 11:30 AM'));
    return <TimePickerComponent id="timepicker" placeholder="Select a Time"
min={minTime} max={maxTime} />
};
ReactDOM.render(<App />, document.getElementById('timer'));
```

If the value of `min` or `max` property is changed through code behind you have to update the `value` property to set within the range.

**Time masking in React Timepicker component**

TimePicker has `enableMask` property that provides the option to enable the built-in date masking support. Also, you must inject the `MaskedDateTime` module to enable the masking support.

**[Class-component]****INDEX.JSX**

```
import { TimePickerComponent, Inject, MaskedDateTime } from '@syncfusion/ej2-react-calendars';
import * as ReactDOM from 'react-dom';
import * as React from 'react';
export default class App extends React.Component {
    render() {
```

```

        return <TimePickerComponent enableMask={true}><Inject
services={ [MaskedDateTime] }/></TimePickerComponent>;
    }
}
ReactDOM.render(<App />, document.getElementById('element'));

```

### INDEX.TSX

```

import { TimePickerComponent, Inject, MaskedDateTime } from '@syncfusion/ej2-react-calendars';
import * as ReactDOM from 'react-dom';
import * as React from 'react';
export default class App extends React.Component<{}, {}> {
    public render() {
        return <TimePickerComponent enableMask={true}><Inject
services={ [MaskedDateTime] } /></TimePickerComponent>
    }
}
ReactDOM.render(<App />, document.getElementById('element'));

```

### [Functional-component]

### INDEX.JSX

```

import { TimePickerComponent, Inject, MaskedDateTime } from '@syncfusion/ej2-react-calendars';
import * as ReactDOM from 'react-dom';
import * as React from 'react';
function App() {
    return <TimePickerComponent enableMask={true}><Inject
services={ [MaskedDateTime] }/></TimePickerComponent>;
}
ReactDOM.render(<App />, document.getElementById('element'));

```

### INDEX.TSX

```

import { TimePickerComponent, Inject, MaskedDateTime } from '@syncfusion/ej2-react-calendars';
import * as ReactDOM from 'react-dom';
import * as React from 'react';
function App() {
    return <TimePickerComponent enableMask={true}><Inject
services={ [MaskedDateTime] } /></TimePickerComponent>
}
ReactDOM.render(<App />, document.getElementById('element'));

```

The mask pattern is defined based on the provided time format to the component. If the format is not specified, the mask pattern is formed based on the default format of the current culture.

| Keys | Actions |

| --- | --- |

| Up / Down arrows | To increment and decrement the selected portion of the time. |

| Left / Right arrows and Tab | To navigate the selection from one portion to next portion |

The following example demonstrates default and custom format of TimePicker component with mask.

[Class-component]

#### INDEX.JSX

```
import { TimePickerComponent, Inject, MaskedDateTime } from '@syncfusion/ej2-react-calendars';
import * as ReactDOM from 'react-dom';
import * as React from 'react';
export default class App extends React.Component {
  render() {
    return (<div>
      /* specifies the masked TimePicker component without format */
      <TimePickerComponent enableMask={true}><Inject
services={ [MaskedDateTime] }/></TimePickerComponent>
      <br />
      <br />
      /* specifies the masked TimePicker component with format */
      <TimePickerComponent format='h:mm a'
enableMask={true}><Inject services={ [MaskedDateTime] }/></TimePickerComponent>
    </div>);
  }
}
ReactDOM.render(<App />, document.getElementById('element'));
```

#### INDEX.TSX

```
import { TimePickerComponent, Inject, MaskedDateTime } from '@syncfusion/ej2-react-calendars';
import * as ReactDOM from 'react-dom';
import * as React from 'react';
export default class App extends React.Component<{}, {}> {
  public render() {
    return (
      <div>
        /* specifies the masked TimePicker component without format */
        <TimePickerComponent enableMask={true}><Inject
services={ [MaskedDateTime] } /></TimePickerComponent>
        <br />
        <br />
        /* specifies the masked TimePicker component with format */
        <TimePickerComponent format='h:mm a'
enableMask={true}><Inject services={ [MaskedDateTime] }
/></TimePickerComponent>
      </div>
    );
  }
}
ReactDOM.render(<App />, document.getElementById('element'));
```

**[Functional-component]****INDEX.JSX**

```
import { TimePickerComponent, Inject, MaskedDateTime } from '@syncfusion/ej2-react-calendars';
import * as ReactDOM from 'react-dom';
import * as React from 'react';
function App() {
    return (<div>
        /* specifies the masked TimePicker component without format */
        <TimePickerComponent enableMask={true}><Inject
services={ [MaskedDateTime] }/></TimePickerComponent>
        <br />
        <br />
        /* specifies the masked TimePicker component with format */
        <TimePickerComponent format='h:mm a'
enableMask={true}><Inject services={ [MaskedDateTime] }/></TimePickerComponent>
    </div>);
}
ReactDOM.render(<App />, document.getElementById('element'));
```

**INDEX.TSX**

```
import { TimePickerComponent, Inject, MaskedDateTime } from '@syncfusion/ej2-react-calendars';
import * as ReactDOM from 'react-dom';
import * as React from 'react';
function App() {
    return (
        <div>
            /* specifies the masked TimePicker component without format */
            <TimePickerComponent enableMask={true}><Inject
services={ [MaskedDateTime] } /></TimePickerComponent>
            <br />
            <br />
            /* specifies the masked TimePicker component with format */
            <TimePickerComponent format='h:mm a'
enableMask={true}><Inject services={ [MaskedDateTime] }
/></TimePickerComponent>
        </div>
    );
}
ReactDOM.render(<App />, document.getElementById('element'));
```

**Configure Mask Placeholder**

You can change mask placeholder value through property **maskPlaceholder**. By default , it takes the full name of time co-ordinates such as **hour**, **minute** and **second**.

While changing to a culture other than **English**, ensure that locale text for the concerned culture is loaded through load method of L10n class for mask placeholder values like below.

```
`ts
//Load the L10n from ej2-base
import { L10n } from '@syncfusion/ej2-base';
//load the locale object to set the localized mask placeholder value
L10n.load({
  'de': {
    timepicker: { hour: 'Stunde', minute: 'Minute', second: 'Sekunde' }
  }
});
`
```

The following example demonstrates default and customized mask placeholder value.

#### [Class-component]

##### INDEX.JSX

```
import { TimePickerComponent, Inject, MaskedDateTime } from '@syncfusion/ej2-react-calendars';
import * as ReactDOM from 'react-dom';
import * as React from 'react';
export default class App extends React.Component {
  maskPlaceholderValue;
  constructor(props) {
    super(props);
    this.maskPlaceholderValue = { hour: 'h', minute: 'm', second: 's' };
  }
  render() {
    return (
      <div>
        /* specifies the masked TimePicker component without mask placeholder */
        <TimePickerComponent enableMask={true}><Inject
services={ [MaskedDateTime] }/></TimePickerComponent>
        <br />
        <br />
        /* specifies the masked TimePicker component with mask placeholder */
        <TimePickerComponent enableMask={true}
maskPlaceholder={this.maskPlaceholderValue}><Inject
services={ [MaskedDateTime] }/></TimePickerComponent>
        </div>
      );
  }
}
ReactDOM.render(<App />, document.getElementById('element'));
```

##### INDEX.TSX

```
import { TimePickerComponent, Inject, MaskedDateTime } from '@syncfusion/ej2-react-calendars';
import * as ReactDOM from 'react-dom';
import * as React from 'react';
```

```

export default class App extends React.Component<{}, {}> {
  private maskPlaceholderValue: object;
  constructor(props: {}) {
    super(props);
    this.maskPlaceholderValue = { hour: 'h', minute: 'm', second: 's' };
  }
  public render() {
    return (
      <div>
        /* specifies the masked TimePicker component without mask placeholder */
        <TimePickerComponent enableMask={true}><Inject
services=[MaskedDateTime] /></TimePickerComponent>
        <br />
        <br />
        /* specifies the masked TimePicker component with mask placeholder */
        <TimePickerComponent enableMask={true}
maskPlaceholder={this.maskPlaceholderValue}><Inject
services=[MaskedDateTime] /></TimePickerComponent>
      </div>
    );
  }
}
ReactDOM.render(<App />, document.getElementById('element'));

```

## [Functional-component]

### INDEX.JSX

```

import { TimePickerComponent, Inject, MaskedDateTime } from '@syncfusion/ej2-react-calendars';
import * as ReactDOM from 'react-dom';
import * as React from 'react';
function App() {
  let maskPlaceholderValue;
  maskPlaceholderValue = { hour: 'h', minute: 'm', second: 's' };
  return (<div>
    /* specifies the masked TimePicker component without mask placeholder */
    <TimePickerComponent enableMask={true}><Inject
services=[MaskedDateTime] /></TimePickerComponent>
    <br />
    <br />
    /* specifies the masked TimePicker component with mask placeholder */
    <TimePickerComponent enableMask={true}
maskPlaceholder={maskPlaceholderValue}><Inject
services=[MaskedDateTime] /></TimePickerComponent>
  </div>);
}
ReactDOM.render(<App />, document.getElementById('element'));

```

### INDEX.TSX



```

import { TimePickerComponent, Inject, MaskedDateTime } from '@syncfusion/ej2-react-calendars';
import * as ReactDOM from 'react-dom';
import * as React from 'react';
function App() {
    let maskPlaceholderValue: object;
    maskPlaceholderValue = { hour: 'h', minute: 'm', second: 's' };
    return (
        <div>
            /* specifies the masked TimePicker component without mask placeholder */
            <TimePickerComponent enableMask={true}><Inject
services={ [MaskedDateTime] } /></TimePickerComponent>
            <br />
            <br />
            /* specifies the masked TimePicker component with mask placeholder */
            <TimePickerComponent enableMask={true}
maskPlaceholder={maskPlaceholderValue}><Inject services={ [MaskedDateTime] }
/></TimePickerComponent>
        </div>
    );
}
ReactDOM.render(<App />, document.getElementById('element'));

```

### Strict mode in React Timepicker component

The [strictMode](#) is an act that allows you to enter only valid time value within the specified min/max range in the textbox. If the time value is invalid, the component value sets to the previous value.

If the time value is out of range, the component sets the time value to min/max value.

The following example demonstrates the TimePicker in `strictMode` with min/max range of 10:00 AM to 4:00 PM . It allows you to enter only valid time within the specified range. If you enter the out-of-range value like 8:00 PM, the value sets to the max time 4:00 PM as the value 8:00 PM is greater than max value of 4:00 PM. If you enter invalid time value like 9:00 tt, the value sets to the previous value.

[Class-component]

#### INDEX.JSX

```

// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
export default class App extends React.Component {
    // initialize the min and max time value
    minTime = new Date('8/3/2017 10:00 AM');
    maxTime = new Date('8/3/2017 4:00 PM');
    time = new Date('8/3/2017 8:00 PM');
    render() {

```

```

        return <TimePickerComponent id="timepicker" value={this.time}
placeholder="Select a Time" min={this.minTime} max={this.maxTime}
strictMode={true}/>;
    }
}
;
ReactDOM.render(<App />, document.getElementById('timer'));

```

**INDEX.TSX**

```

// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
export default class App extends React.Component<{}, {}> {
    // initialize the min and max time value
    private minTime: Date = new Date('8/3/2017 10:00 AM');
    private maxTime: Date = new Date('8/3/2017 4:00 PM');
    private time: Date = new Date('8/3/2017 8:00 PM');
    public render() {
        return <TimePickerComponent id="timepicker" value={this.time}
placeholder="Select a Time" min={this.minTime} max={this.maxTime}
strictMode={true} />
    }
};
ReactDOM.render(<App />, document.getElementById('timer'));

```

**[Functional-component]****INDEX.JSX**

```

// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
function App() {
    // initialize the min and max time value
    const minTime = new Date('8/3/2017 10:00 AM');
    const maxTime = new Date('8/3/2017 4:00 PM');
    const time = new Date('8/3/2017 8:00 PM');
    return <TimePickerComponent id="timepicker" value={time}
placeholder="Select a Time" min={minTime} max={maxTime} strictMode={true}/>;
}
;
ReactDOM.render(<App />, document.getElementById('timer'));

```

**INDEX.TSX**

```
// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
function App() {
    // initialize the min and max time value
    const minTime: Date = new Date('8/3/2017 10:00 AM');
    const maxTime: Date = new Date('8/3/2017 4:00 PM');
    const time: Date = new Date('8/3/2017 8:00 PM');
    return <TimePickerComponent id="timepicker" value={time}
placeholder="Select a Time" min={minTime} max={maxTime} strictMode={true} />
};
ReactDOM.render(<App />, document.getElementById('timer'));
```

By default, the TimePicker act in strictMode **false** state, that allows to enter the invalid or out-of-range time in textbox.

If the time is out-of-range or invalid, then the model value will be set to **out of range** time value or **null** respectively with highlighted **error** class to indicates the time is out of range or invalid.

The following example demonstrates the **strictMode** as **false**. Here, it allows to enter the valid or invalid value in textbox. If you are entering the out-of-range or invalid time value, then the model value will be set to **out of range** time value or **null** respectively with highlighted **error** class to indicates the time is out of range or invalid.

[Class-component]

#### INDEX.JSX

```
// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
export default class App extends React.Component {
    // initialize the min and max time value
    minTime = new Date('8/3/2017 10:00 AM');
    maxTime = new Date('8/3/2017 4:00 PM');
    time = new Date('8/3/2017 8:00 PM');
    render() {
        return <TimePickerComponent id="timepicker" value={this.time}
placeholder="Select a Time" min={this.minTime} max={this.maxTime}
strictMode={false}/>;
    }
}
;
ReactDOM.render(<App />, document.getElementById('timer'));
```

**INDEX.TSX**

```
// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
export default class App extends React.Component<{}, {}> {
    // initialize the min and max time value
    private minTime: Date = new Date('8/3/2017 10:00 AM');
    private maxTime: Date = new Date('8/3/2017 4:00 PM');
    private time: Date = new Date('8/3/2017 8:00 PM');
    public render() {
        return <TimePickerComponent id="timepicker" value={this.time}
placeholder="Select a Time" min={this.minTime} max={this.maxTime}
strictMode={false} />
    }
};
ReactDOM.render(<App />, document.getElementById('timer'));
```

**[Functional-component]****INDEX.JSX**

```
// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
function App() {
    // initialize the min and max time value
    const minTime = new Date('8/3/2017 10:00 AM');
    const maxTime = new Date('8/3/2017 4:00 PM');
    const time = new Date('8/3/2017 8:00 PM');
    return <TimePickerComponent id="timepicker" value={time}
placeholder="Select a Time" min={minTime} max={maxTime} strictMode={false}/>;
}
;
ReactDOM.render(<App />, document.getElementById('timer'));
```

**INDEX.TSX**

```
// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
```

```
function App() {
  // initialize the min and max time value
  const minTime: Date = new Date('8/3/2017 10:00 AM');
  const maxTime: Date = new Date('8/3/2017 4:00 PM');
  const time: Date = new Date('8/3/2017 8:00 PM');
  return <TimePickerComponent id="timepicker" value={time}
    placeholder="Select a Time" min={minTime} max={maxTime} strictMode={false} />
  };
ReactDOM.render(<App />, document.getElementById('timer'));
```

If the value of `min` or `max` property is changed through code behind, update the `value` property to set within the range.

### Globalization in React Timepicker component

Globalization is the combination of internationalization and localization. You can adapt the component to various languages by means of parsing and formatting the date or number [internationalization](#), and also add culture specific customization and translation to the text [localization](#).

By default, TimePicker time format and meridian names are specific to the **American English** culture. It utilizes the

[Essential JavaScript 2 Internationalization](#) package to parse and format the date object based on the culture by using the official [UNICODE CLDR](#) JSON data. It provides the `loadCldr` method to load culture specific CLDR JSON data. To go with the different culture other than **English**, follow the steps below.

- Install the **CLDR-Data** package by using the following command (it installs all the CLDR JSON data). To know more about CLDR-Data refer the [CLDR-Data](#) link.

```
npm install cldr-data --save
```

Once the package is installed, you can find the culture specific JSON data under the location `\node_modules\cldr-data`.

- Import the installed CLDR JSON data into the `app.ts` file.
- Use the [loadCldr](#) method to load the culture specific CLDR JSON data from the installed location to `app.ts` file.
- TimePicker displayed **Sunday** as the first day of week based on default culture ("en-US"). If you want to display the TimePicker with loaded culture's first day of week, you need to import `weekdata.json` file from the `cldr-data/supplemental` as given in the code example.

```
`ts
```

```
import { loadCldr } from '@syncfusion/ej2-base';
import { loadCldr } from "@syncfusion/ej2-base";
import * as gregorian from 'cldr-data/main/de/ca-gregorian.json';
import * as numbers from 'cldr-data/main/de/numbers.json';
```

```
import * as numberingSystems from 'cldr-data/supplemental/numberingSystems.json';
loadCldr(numberingSystems, gregorian, numbers);
`ts
```

if you are facing the error `/node_modules/cldr-data/main/de/*.json (1,1): unused expression, expected an assignment or function call` when you are adding the json files to render the culture sample, then add the below configuration in your `tslint.json` file

```
`ts
"linterOptions": {
"exclude": [
"*.json",
"/*.json"
]
}
`ts
```

- Before changing to a culture other than `English`, ensure that locale text for the concerned culture is loaded through `load` method of `L10n` class.

```
`ts
//Load the L10n, loadCldr from ej2-base
import { loadCldr, L10n } from '@syncfusion/ej2-base';
//load the locale object to set the localized placeholder value
L10n.load({
'de': {
'timepicker': { placeholder: 'Wählen Sie Zeit' }
}
});
`ts
```

- Set the culture by using the `locale` property. In the following code example, the `DateTimePicker` is initialized in `German` culture with corresponding localized text.

The following example demonstrates the TimePicker in `German` culture.

[Class-component]

#### **CA-GREGORIAN.JSON**

```
{
  "main": {
```

```

"de": {
  "identity": {
    "version": {
      "_number": "$Revision: 12879 $",
      "_cldrVersion": "30.0.3"
    },
    "language": "de"
  },
  "dates": {
    "calendars": {
      "gregorian": {
        "months": {
          "format": {
            "abbreviated": {
              "1": "Jan.",
              "2": "Feb.",
              "3": "März",
              "4": "Apr.",
              "5": "Mai",
              "6": "Juni",
              "7": "Juli",
              "8": "Aug.",
              "9": "Sep.",
              "10": "Okt.",
              "11": "Nov.",
              "12": "Dez."
            },
            "narrow": {
              "1": "J",
              "2": "F",
              "3": "M",
              "4": "A",
              "5": "M",
              "6": "J",
              "7": "J",
              "8": "A",
              "9": "S",
              "10": "O",
              "11": "N",
              "12": "D"
            },
            "wide": {
              "1": "Januar",
              "2": "Februar",
              "3": "März",
              "4": "April",
              "5": "Mai",
              "6": "Juni",
              "7": "Juli",
              "8": "August",
              "9": "September",
              "10": "Oktober",
              "11": "November",
              "12": "Dezember"
            }
          }
        },
        "stand-alone": {

```

```
    "abbreviated": {
      "1": "Jan",
      "2": "Feb",
      "3": "Mär",
      "4": "Apr",
      "5": "Mai",
      "6": "Jun",
      "7": "Jul",
      "8": "Aug",
      "9": "Sep",
      "10": "Okt",
      "11": "Nov",
      "12": "Dez"
    },
    "narrow": {
      "1": "J",
      "2": "F",
      "3": "M",
      "4": "A",
      "5": "M",
      "6": "J",
      "7": "J",
      "8": "A",
      "9": "S",
      "10": "O",
      "11": "N",
      "12": "D"
    },
    "wide": {
      "1": "Januar",
      "2": "Februar",
      "3": "März",
      "4": "April",
      "5": "Mai",
      "6": "Juni",
      "7": "Juli",
      "8": "August",
      "9": "September",
      "10": "Oktober",
      "11": "November",
      "12": "Dezember"
    }
  }
},
"days": {
  "format": {
    "abbreviated": {
      "sun": "So.",
      "mon": "Mo.",
      "tue": "Di.",
      "wed": "Mi.",
      "thu": "Do.",
      "fri": "Fr.",
      "sat": "Sa."
    },
    "narrow": {
      "sun": "S",
```



```
        "mon": "M",
        "tue": "D",
        "wed": "M",
        "thu": "D",
        "fri": "F",
        "sat": "S"
    },
    "short": {
        "sun": "So.",
        "mon": "Mo.",
        "tue": "Di.",
        "wed": "Mi.",
        "thu": "Do.",
        "fri": "Fr.",
        "sat": "Sa."
    },
    "wide": {
        "sun": "Sonntag",
        "mon": "Montag",
        "tue": "Dienstag",
        "wed": "Mittwoch",
        "thu": "Donnerstag",
        "fri": "Freitag",
        "sat": "Samstag"
    }
},
"stand-alone": {
    "abbreviated": {
        "sun": "So",
        "mon": "Mo",
        "tue": "Di",
        "wed": "Mi",
        "thu": "Do",
        "fri": "Fr",
        "sat": "Sa"
    },
    "narrow": {
        "sun": "S",
        "mon": "M",
        "tue": "D",
        "wed": "M",
        "thu": "D",
        "fri": "F",
        "sat": "S"
    },
    "short": {
        "sun": "So.",
        "mon": "Mo.",
        "tue": "Di.",
        "wed": "Mi.",
        "thu": "Do.",
        "fri": "Fr.",
        "sat": "Sa."
    },
    "wide": {
        "sun": "Sonntag",
        "mon": "Montag",
```

```
        "tue": "Dienstag",
        "wed": "Mittwoch",
        "thu": "Donnerstag",
        "fri": "Freitag",
        "sat": "Samstag"
      }
    },
    "quarters": {
      "format": {
        "abbreviated": {
          "1": "Q1",
          "2": "Q2",
          "3": "Q3",
          "4": "Q4"
        },
        "narrow": {
          "1": "1",
          "2": "2",
          "3": "3",
          "4": "4"
        },
        "wide": {
          "1": "1. Quartal",
          "2": "2. Quartal",
          "3": "3. Quartal",
          "4": "4. Quartal"
        }
      },
      "stand-alone": {
        "abbreviated": {
          "1": "Q1",
          "2": "Q2",
          "3": "Q3",
          "4": "Q4"
        },
        "narrow": {
          "1": "1",
          "2": "2",
          "3": "3",
          "4": "4"
        },
        "wide": {
          "1": "1. Quartal",
          "2": "2. Quartal",
          "3": "3. Quartal",
          "4": "4. Quartal"
        }
      }
    },
    "dayPeriods": {
      "format": {
        "abbreviated": {
          "midnight": "Mitternacht",
          "am": "vorm.",
          "pm": "nachm.",
          "morning1": "morgens",
```

```

        "morning2": "vormittags",
        "afternoon1": "mittags",
        "afternoon2": "nachmittags",
        "evening1": "abends",
        "night1": "nachts"
    },
    "narrow": {
        "midnight": "Mitternacht",
        "am": "vm.",
        "pm": "nm.",
        "morning1": "morgens",
        "morning2": "vormittags",
        "afternoon1": "mittags",
        "afternoon2": "nachmittags",
        "evening1": "abends",
        "night1": "nachts"
    },
    "wide": {
        "midnight": "Mitternacht",
        "am": "vorm.",
        "pm": "nachm.",
        "morning1": "morgens",
        "morning2": "vormittags",
        "afternoon1": "mittags",
        "afternoon2": "nachmittags",
        "evening1": "abends",
        "night1": "nachts"
    }
},
"stand-alone": {
    "abbreviated": {
        "midnight": "Mitternacht",
        "am": "vorm.",
        "pm": "nachm.",
        "morning1": "Morgen",
        "morning2": "Vormittag",
        "afternoon1": "Mittag",
        "afternoon2": "Nachmittag",
        "evening1": "Abend",
        "night1": "Nacht"
    },
    "narrow": {
        "midnight": "Mitternacht",
        "am": "vorm.",
        "pm": "nachm.",
        "morning1": "Morgen",
        "morning2": "Vormittag",
        "afternoon1": "Mittag",
        "afternoon2": "Nachmittag",
        "evening1": "Abend",
        "night1": "Nacht"
    },
    "wide": {
        "midnight": "Mitternacht",
        "am": "vorm.",
        "pm": "nachm.",
        "morning1": "Morgen",

```

```

        "morning2": "Vormittag",
        "afternoon1": "Mittag",
        "afternoon2": "Nachmittag",
        "evening1": "Abend",
        "night1": "Nacht"
    }
}
},
"eras": {
    "eraNames": {
        "0": "v. Chr.",
        "0-alt-variant": "vor unserer Zeitrechnung",
        "1": "n. Chr.",
        "1-alt-variant": "unserer Zeitrechnung"
    },
    "eraAbbr": {
        "0": "v. Chr.",
        "0-alt-variant": "v. u. Z.",
        "1": "n. Chr.",
        "1-alt-variant": "u. Z."
    },
    "eraNarrow": {
        "0": "v. Chr.",
        "0-alt-variant": "v. u. Z.",
        "1": "n. Chr.",
        "1-alt-variant": "u. Z."
    }
},
"dateFormats": {
    "full": "EEEE, d. MMMM y",
    "long": "d. MMMM y",
    "medium": "dd.MM.y",
    "short": "dd.MM.yy"
},
"timeFormats": {
    "full": "HH:mm:ss zzzz",
    "long": "HH:mm:ss z",
    "medium": "HH:mm:ss",
    "short": "HH:mm"
},
"dateTimeFormats": {
    "full": "{1} 'um' {0}",
    "long": "{1} 'um' {0}",
    "medium": "{1}, {0}",
    "short": "{1}, {0}",
    "availableFormats": {
        "d": "d",
        "E": "ccc",
        "Ed": "E, d.",
        "Ehm": "E h:mm a",
        "EHm": "E, HH:mm",
        "Ehms": "E, h:mm:ss a",
        "EHms": "E, HH:mm:ss",
        "Gy": "y G",
        "GyMMM": "MMM y G",
        "GyMMMd": "d. MMM y G",
        "GyMMMED": "E, d. MMM y G",
    }
}

```

```

    "h": "h 'Uhr' a",
    "H": "HH 'Uhr'",
    "hm": "h:mm a",
    "Hm": "HH:mm",
    "hms": "h:mm:ss a",
    "Hms": "HH:mm:ss",
    "hmsv": "h:mm:ss a v",
    "Hmsv": "HH:mm:ss v",
    "hmv": "h:mm a v",
    "Hmv": "HH:mm v",
    "M": "L",
    "Md": "d.M.",
    "MEd": "E, d.M.",
    "MMd": "d.MM.",
    "MMdd": "dd.MM.",
    "MMM": "LLL",
    "MMMd": "d. MMM",
    "MMMEd": "E, d. MMM",
    "MMMMd": "d. MMMM",
    "MMMMEd": "E, d. MMMM",
    "MMMMW": "'Woche' W 'im' MMM",
    "MMMMW": "'Woche' W 'im' MMM",
    "ms": "mm:ss",
    "y": "y",
    "yM": "M.y",
    "yMd": "d.M.y",
    "yMEd": "E, d.M.y",
    "yMM": "MM.y",
    "yMMdd": "dd.MM.y",
    "yMMM": "MMM y",
    "yMMMd": "d. MMM y",
    "yMMMEd": "E, d. MMM y",
    "yMMMM": "MMMM y",
    "yQQQ": "QQQ y",
    "yQQQQ": "QQQQ y",
    "yw": "'Woche' w 'des' 'Jahres' y",
    "yw": "'Woche' w 'des' 'Jahres' y"
  },
  "appendItems": {
    "Day": "{0} ({2}: {1})",
    "Day-Of-Week": "{0} {1}",
    "Era": "{1} {0}",
    "Hour": "{0} ({2}: {1})",
    "Minute": "{0} ({2}: {1})",
    "Month": "{0} ({2}: {1})",
    "Quarter": "{0} ({2}: {1})",
    "Second": "{0} ({2}: {1})",
    "Timezone": "{0} {1}",
    "Week": "{0} ({2}: {1})",
    "Year": "{1} {0}"
  },
  "intervalFormats": {
    "intervalFormatFallback": "{0} - {1}",
    "d": {
      "d": "d.-d."
    },
    "h": {

```

```

    "a": "h 'Uhr' a - h 'Uhr' a",
    "h": "h - h 'Uhr' a"
  },
  "H": {
    "H": "HH-HH 'Uhr'"
  },
  "hm": {
    "a": "h:mm a - h:mm a",
    "h": "h:mm-h:mm a",
    "m": "h:mm-h:mm a"
  },
  "Hm": {
    "H": "HH:mm-HH:mm 'Uhr'",
    "m": "HH:mm-HH:mm 'Uhr'"
  },
  "hmv": {
    "a": "h:mm a - h:mm a v",
    "h": "h:mm-h:mm a v",
    "m": "h:mm-h:mm a v"
  },
  "Hmv": {
    "H": "HH:mm-HH:mm 'Uhr' v",
    "m": "HH:mm-HH:mm 'Uhr' v"
  },
  "hv": {
    "a": "h a - h a v",
    "h": "h-h a v"
  },
  "Hv": {
    "H": "HH-HH 'Uhr' v"
  },
  "M": {
    "M": "M.-M."
  },
  "Md": {
    "d": "dd.MM. - dd.MM.",
    "M": "dd.MM. - dd.MM."
  },
  "MEd": {
    "d": "E, dd.MM. - E, dd.MM.",
    "M": "E, dd.MM. - E, dd.MM."
  },
  "MMM": {
    "M": "MMM-MMM"
  },
  "MMMd": {
    "d": "d.-d. MMM",
    "M": "d. MMM - d. MMM"
  },
  "MMMEd": {
    "d": "E, d. - E, d. MMM",
    "M": "E, d. MMM - E, d. MMM"
  },
  "MMMM": {
    "M": "LLLL-LLLL"
  },
  "Y": {

```

[illegible]

## CA-GREGORIAN.JSX

```
{
  "main";
  {
    "de";
    {
      "identity";
      {
        "version";
        {
          "_number";
          "$Revision: 12879 $",

```

```
        "_cldrVersion";
        "30.0.3";
    }
    "language";
    "de";
}
"dates";
{
    "calendars";
    {
        "gregorian";
        {
            "months";
            {
                "format";
                {
                    "abbreviated";
                    {
                        "1";
                        "Jan.",
                        "2";
                        "Feb.",
                        "3";
                        "März",
                        "4";
                        "Apr.",
                        "5";
                        "Mai",
                        "6";
                        "Juni",
                        "7";
                        "Juli",
                        "8";
                        "Aug.",
                        "9";
                        "Sep.",
                        "10";
                        "Okt.",
                        "11";
                        "Nov.",
                        "12";
                        "Dez.";
                    }
                }
            }
        }
    }
    "narrow";
    {
        "1";
        "J",
        "2";
        "F",
        "3";
        "M",
        "4";
        "A",
        "5";
        "M",
        "6";
        "J",
```



```
        "7";
        "J",
        "8";
        "A",
        "9";
        "S",
        "10";
        "O",
        "11";
        "N",
        "12";
        "D";
    }
    "wide";
    {
        "1";
        "Januar",
        "2";
        "Februar",
        "3";
        "März",
        "4";
        "April",
        "5";
        "Mai",
        "6";
        "Juni",
        "7";
        "Juli",
        "8";
        "August",
        "9";
        "September",
        "10";
        "Oktober",
        "11";
        "November",
        "12";
        "Dezember";
    }
}
"stand-alone";
{
    "abbreviated";
    {
        "1";
        "Jan",
        "2";
        "Feb",
        "3";
        "Mär",
        "4";
        "Apr",
        "5";
        "Mai",
        "6";
        "Jun",
```

```
        "7";
        "Jul",
        "8";
        "Aug",
        "9";
        "Sep",
        "10";
        "Okt",
        "11";
        "Nov",
        "12";
        "Dez";
    }
    "narrow";
    {
        "1";
        "J",
        "2";
        "F",
        "3";
        "M",
        "4";
        "A",
        "5";
        "M",
        "6";
        "J",
        "7";
        "J",
        "8";
        "A",
        "9";
        "S",
        "10";
        "O",
        "11";
        "N",
        "12";
        "D";
    }
    "wide";
    {
        "1";
        "Januar",
        "2";
        "Februar",
        "3";
        "März",
        "4";
        "April",
        "5";
        "Mai",
        "6";
        "Juni",
        "7";
        "Juli",
        "8";
```

```
        "August",
        "9";
        "September",
        "10";
        "Oktober",
        "11";
        "November",
        "12";
        "Dezember";
    }
}
"days";
{
    "format";
    {
        "abbreviated";
        {
            "sun";
            "So.",
            "mon";
            "Mo.",
            "tue";
            "Di.",
            "wed";
            "Mi.",
            "thu";
            "Do.",
            "fri";
            "Fr.",
            "sat";
            "Sa.";
        }
        "narrow";
        {
            "sun";
            "S",
            "mon";
            "M",
            "tue";
            "D",
            "wed";
            "M",
            "thu";
            "D",
            "fri";
            "F",
            "sat";
            "S";
        }
        "short";
        {
            "sun";
            "So.",
            "mon";
            "Mo.",
            "tue";
```

```
        "Di.",
        "wed";
        "Mi.",
        "thu";
        "Do.",
        "fri";
        "Fr.",
        "sat";
        "Sa.";
    }
    "wide";
    {
        "sun";
        "Sonntag",
        "mon";
        "Montag",
        "tue";
        "Dienstag",
        "wed";
        "Mittwoch",
        "thu";
        "Donnerstag",
        "fri";
        "Freitag",
        "sat";
        "Samstag";
    }
}
"stand-alone";
{
    "abbreviated";
    {
        "sun";
        "So",
        "mon";
        "Mo",
        "tue";
        "Di",
        "wed";
        "Mi",
        "thu";
        "Do",
        "fri";
        "Fr",
        "sat";
        "Sa";
    }
    "narrow";
    {
        "sun";
        "S",
        "mon";
        "M",
        "tue";
        "D",
        "wed";
        "M",
```

```
        "thu";
        "D",
        "fri";
        "F",
        "sat";
        "S";
    }
    "short";
    {
        "sun";
        "So.",
        "mon";
        "Mo.",
        "tue";
        "Di.",
        "wed";
        "Mi.",
        "thu";
        "Do.",
        "fri";
        "Fr.",
        "sat";
        "Sa.";
    }
    "wide";
    {
        "sun";
        "Sonntag",
        "mon";
        "Montag",
        "tue";
        "Dienstag",
        "wed";
        "Mittwoch",
        "thu";
        "Donnerstag",
        "fri";
        "Freitag",
        "sat";
        "Samstag";
    }
}
"quarters";
{
    "format";
    {
        "abbreviated";
        {
            "1";
            "Q1",
            "2";
            "Q2",
            "3";
            "Q3",
            "4";
            "Q4";
        }
    }
}
```

```
}
"narrow";
{
  "1";
  "1",
    "2";
  "2",
    "3";
  "3",
    "4";
  "4";
}
"wide";
{
  "1";
  "1. Quartal",
    "2";
  "2. Quartal",
    "3";
  "3. Quartal",
    "4";
  "4. Quartal";
}
}
"stand-alone";
{
  "abbreviated";
  {
    "1";
    "Q1",
      "2";
    "Q2",
      "3";
    "Q3",
      "4";
    "Q4";
  }
  "narrow";
  {
    "1";
    "1",
      "2";
    "2",
      "3";
    "3",
      "4";
    "4";
  }
  "wide";
  {
    "1";
    "1. Quartal",
      "2";
    "2. Quartal",
      "3";
    "3. Quartal",
      "4";
  }
}
```

```

        "4. Quartal";
    }
}
}
"dayPeriods";
{
    "format";
    {
        "abbreviated";
        {
            "midnight";
            "Mitternacht",
            "am";
            "vorm.",
            "pm";
            "nachm.",
            "morning1";
            "morgens",
            "morning2";
            "vormittags",
            "afternoon1";
            "mittags",
            "afternoon2";
            "nachmittags",
            "evening1";
            "abends",
            "night1";
            "nachts";
        }
        "narrow";
        {
            "midnight";
            "Mitternacht",
            "am";
            "vm.",
            "pm";
            "nm.",
            "morning1";
            "morgens",
            "morning2";
            "vormittags",
            "afternoon1";
            "mittags",
            "afternoon2";
            "nachmittags",
            "evening1";
            "abends",
            "night1";
            "nachts";
        }
        "wide";
        {
            "midnight";
            "Mitternacht",
            "am";
            "vorm.",
            "pm";

```

```
        "nachm.",
        "morning1";
    "morgens",
    "morning2";
    "vormittags",
    "afternoon1";
    "mittags",
    "afternoon2";
    "nachmittags",
    "evening1";
    "abends",
    "night1";
    "nachts";
    }
}
"stand-alone";
{
    "abbreviated";
    {
        "midnight";
        "Mitternacht",
        "am";
        "vorm.",
        "pm";
        "nachm.",
        "morning1";
        "Morgen",
        "morning2";
        "Vormittag",
        "afternoon1";
        "Mittag",
        "afternoon2";
        "Nachmittag",
        "evening1";
        "Abend",
        "night1";
        "Nacht";
    }
    "narrow";
    {
        "midnight";
        "Mitternacht",
        "am";
        "vorm.",
        "pm";
        "nachm.",
        "morning1";
        "Morgen",
        "morning2";
        "Vormittag",
        "afternoon1";
        "Mittag",
        "afternoon2";
        "Nachmittag",
        "evening1";
        "Abend",
        "night1";
    }
}
```



```

        "Nacht";
    }
    "wide";
    {
        "midnight";
        "Mitternacht",
            "am";
        "vorm.",
            "pm";
        "nachm.",
            "morning1";
        "Morgen",
            "morning2";
        "Vormittag",
            "afternoon1";
        "Mittag",
            "afternoon2";
        "Nachmittag",
            "evening1";
        "Abend",
            "night1";
        "Nacht";
    }
    }
}
"eras";
{
    "eraNames";
    {
        "0";
        "v. Chr.",
            "0-alt-variant";
        "vor unserer Zeitrechnung",
            "1";
        "n. Chr.",
            "1-alt-variant";
        "unserer Zeitrechnung";
    }
    "eraAbbr";
    {
        "0";
        "v. Chr.",
            "0-alt-variant";
        "v. u. Z.",
            "1";
        "n. Chr.",
            "1-alt-variant";
        "u. Z.";
    }
    "eraNarrow";
    {
        "0";
        "v. Chr.",
            "0-alt-variant";
        "v. u. Z.",
            "1";
        "n. Chr.",

```

```

        "1-alt-variant";
        "u. Z.";
    }
}
"dateFormats";
{
    "full";
    "EEEE, d. MMMM y",
    "long";
    "d. MMMM y",
    "medium";
    "dd.MM.y",
    "short";
    "dd.MM.yy";
}
"timeFormats";
{
    "full";
    "HH:mm:ss zzzz",
    "long";
    "HH:mm:ss z",
    "medium";
    "HH:mm:ss",
    "short";
    "HH:mm";
}
"dateTimeFormats";
{
    "full";
    "{1} 'um' {0}",
    "long";
    "{1} 'um' {0}",
    "medium";
    "{1}, {0}",
    "short";
    "{1}, {0}",
    "availableFormats";
    {
        "d";
        "d",
        "E";
        "ccc",
        "Ed";
        "E, d.",
        "Ehm";
        "E h:mm a",
        "EHm";
        "E, HH:mm",
        "Ehms";
        "E, h:mm:ss a",
        "EHms";
        "E, HH:mm:ss",
        "Gy";
        "y G",
        "GyMMM";
        "MMM y G",
        "GyMMMd";
    }
}

```

```
"d. MMM y G",
  "GyMMMED";
"E, d. MMM y G",
  "h";
"h 'Uhr' a",
  "H";
"HH 'Uhr' ",
  "hm";
"h:mm a",
  "Hm";
"HH:mm",
  "hms";
"h:mm:ss a",
  "Hms";
"HH:mm:ss",
  "hmsv";
"h:mm:ss a v",
  "Hmsv";
"HH:mm:ss v",
  "hmv";
"h:mm a v",
  "Hmv";
"HH:mm v",
  "M";
"L",
  "Md";
"d.M.",
  "MED";
"E, d.M.",
  "MMd";
"d.MM.",
  "MMdd";
"dd.MM.",
  "MMM";
"LLL",
  "MMMd";
"d. MMM",
  "MMMED";
"E, d. MMM",
  "MMMMd";
"d. MMMM",
  "MMMMEd";
"E, d. MMMM",
  "MMMMW";
"'Woche' W 'im' MMM",
  "MMMMW";
"'Woche' W 'im' MMM",
  "ms";
"mm:ss",
  "y";
"Y",
  "yM";
"M.y",
  "yMd";
"d.M.y",
  "yMED";
"E, d.M.y",
```

```

        "yMM";
        "MM.y",
        "yMMdd";
        "dd.MM.y",
        "yMMM";
        "MMM y",
        "yMMMd";
        "d. MMM y",
        "yMMMEd";
        "E, d. MMM y",
        "yMMMM";
        "MMMM y",
        "yQQQ";
        "QQQ y",
        "yQQQQ";
        "QQQQ y",
        "yw";
        "'Woche' w 'des' 'Jahres' y",
        "yw";
        "'Woche' w 'des' 'Jahres' y";
    }
    "appendItems";
    {
        "Day";
        "{0} ({2}: {1})",
        "Day-Of-Week";
        "{0} {1}",
        "Era";
        "{1} {0}",
        "Hour";
        "{0} ({2}: {1})",
        "Minute";
        "{0} ({2}: {1})",
        "Month";
        "{0} ({2}: {1})",
        "Quarter";
        "{0} ({2}: {1})",
        "Second";
        "{0} ({2}: {1})",
        "Timezone";
        "{0} {1}",
        "Week";
        "{0} ({2}: {1})",
        "Year";
        "{1} {0}";
    }
    "intervalFormats";
    {
        "intervalFormatFallback";
        "{0} - {1}",
        "d";
        {
            "d";
            "d.-d.";
        }
        "h";
        {

```

```
        "a";
        "h 'Uhr' a - h 'Uhr' a",
        "h";
        "h - h 'Uhr' a";
    }
    "H";
    {
        "H";
        "HH-HH 'Uhr'";
    }
    "hm";
    {
        "a";
        "h:mm a - h:mm a",
        "h";
        "h:mm-h:mm a",
        "m";
        "h:mm-h:mm a";
    }
    "Hm";
    {
        "H";
        "HH:mm-HH:mm 'Uhr'",
        "m";
        "HH:mm-HH:mm 'Uhr'";
    }
    "hmv";
    {
        "a";
        "h:mm a - h:mm a v",
        "h";
        "h:mm-h:mm a v",
        "m";
        "h:mm-h:mm a v";
    }
    "Hmv";
    {
        "H";
        "HH:mm-HH:mm 'Uhr' v",
        "m";
        "HH:mm-HH:mm 'Uhr' v";
    }
    "hv";
    {
        "a";
        "h a - h a v",
        "h";
        "h-h a v";
    }
    "Hv";
    {
        "H";
        "HH-HH 'Uhr' v";
    }
    "M";
    {
        "M";
```

```

        "M.-M.";
    }
    "Md";
    {
        "d";
        "dd.MM. - dd.MM.",
        "M";
        "dd.MM. - dd.MM.";
    }
    "MED";
    {
        "d";
        "E, dd.MM. - E, dd.MM.",
        "M";
        "E, dd.MM. - E, dd.MM.";
    }
    "MMM";
    {
        "M";
        "MMM-MMM";
    }
    "MMMd";
    {
        "d";
        "d.-d. MMM",
        "M";
        "d. MMM - d. MMM";
    }
    "MMMED";
    {
        "d";
        "E, d. - E, d. MMM",
        "M";
        "E, d. MMM - E, d. MMM";
    }
    "MMMM";
    {
        "M";
        "LLLL-LLLL";
    }
    "Y";
    {
        "Y";
        "Y-Y";
    }
    "YM";
    {
        "M";
        "MM.Y - MM.Y",
        "Y";
        "MM.Y - MM.Y";
    }
    "yMd";
    {
        "d";
        "dd.MM.Y - dd.MM.Y",
        "M";
    }

```

## INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
//Load the L10n, loadCldr from ej2-base
import { loadCldr, L10n } from '@syncfusion/ej2-base';
//import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
//import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
// enable ripple effect
enableRipple(true);
//load the CLDR data files.
// Here we have referred local json files for preview purpose
import * as numberingSystems from './numberingSystems.json';
import * as gregorian from './ca-gregorian.json';
import * as numbers from './numbers.json';
loadCldr(numberingSystems, gregorian, numbers);
L10n.load({
    'de': {
        'timepicker': { placeholder: 'Wählen Sie Zeit' }
    }
});
//import the timepickercomponent
class App extends React.Component {
    render() {
        return <TimePickerComponent id="time" locale='de'/>;
    }
}
;
ReactDOM.render(<App />, document.getElementById('timer'));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
//Load the L10n, loadCldr from ej2-base
import { loadCldr, L10n } from '@syncfusion/ej2-base';
//import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
//import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
// enable ripple effect
enableRipple(true);
//load the CLDR data files.
// Here we have referred local json files for preview purpose
import * as numberingSystems from './numberingSystems.json';
import * as gregorian from './ca-gregorian.json';
import * as numbers from './numbers.json';
loadCldr(numberingSystems, gregorian, numbers);
L10n.load({
    'de': {
        'timepicker': { placeholder: 'Wählen Sie Zeit' }
    }
});
//import the timepickercomponent
class App extends React.Component<{}, {}> {
    render() {

```



```

    return <TimePickerComponent id="time" locale='de' />
  }
};
ReactDOM.render(<App />, document.getElementById('timer'));

```

### NUMBERINGSYSTEMS.JSON

```

{
  "supplemental": {
    "version": {
      "_number": "$Revision: 12732 $",
      "_unicodeVersion": "9.0.0",
      "_cldrVersion": "31"
    },
    "numberingSystems": {
      "adlm": {
        "_digits": "ᲐᲑᲒᲓᲔᲕᲖᲗᲘᲙ",
        "_type": "numeric"
      },
      "ahom": {
        "_digits": "ᩌᩍᩎᩏᩐᩑᩒᩓᩔᩕ",
        "_type": "numeric"
      },
      "arab": {
        "_digits": "٠١٢٣٤٥٦٧٨٩",
        "_type": "numeric"
      },
      "arabext": {
        "_digits": "٠١٢٣٤٥٦٧٨٩",
        "_type": "numeric"
      },
      "armn": {
        "_rules": "armenian-upper",
        "_type": "algorithmic"
      },
      "armnlow": {
        "_rules": "armenian-lower",
        "_type": "algorithmic"
      },
      "bali": {
        "_digits": "ᬂᬃᬄᬅᬆᬇᬈᬉᬊᬋᬌᬍᬎᬏᬐᬑᬒᬓᬔᬕᬖᬗᬘᬙᬚᬛᬜᬝᬞᬟᬠᬡᬢᬣᬤᬥᬦᬧᬨᬩᬪᬫᬬᬭᬮᬯᬰᬱᬲᬳ᬴ᬵᬶᬷᬸᬹᬺᬻᬼᬽᬾᬿ",
        "_type": "numeric"
      },
      "beng": {
        "_digits": "০১২৩৪৫৬৭৮৯",
        "_type": "numeric"
      },
      "bhks": {
        "_digits": "ᩌᩍᩎᩏᩐᩑᩒᩓᩔᩕ",
        "_type": "numeric"
      },
      "brah": {
        "_digits": "᱀᱁᱂᱃᱄᱅᱆᱇᱈᱉᱊᱋᱌ᱍᱎᱏ᱐᱑᱒᱓᱔᱕᱖᱗᱘᱙ᱚᱛᱜᱝᱞᱟᱠᱡᱢᱣᱤᱥᱦᱧᱨᱩᱪᱫᱬᱭᱮᱹᱺᱻᱼᱽ᱾᱿",
        "_type": "numeric"
      }
    }
  }
}

```

```
"cakm": {
    "_digits": "᠐ᠸᠡᠭᠦᠨᠢᠯᠤᠰ",
    "_type": "numeric"
},
"cham": {
    "_digits": "᠊᠎ᠠᠵᠤ᠋ᠹᠤᠮᠤᠩᠴᠤᠳᠤᠨᠢ",
    "_type": "numeric"
},
"cyrl": {
    "_rules": "cyrillic-lower",
    "_type": "algorithmic"
},
"deva": {
    "_digits": "ॐ१२३४५६७८९",
    "_type": "numeric"
},
"ethi": {
    "_rules": "ethiopic",
    "_type": "algorithmic"
},
"fullwide": {
    "_digits": "0 1 2 3 4 5 6 7 8 9",
    "_type": "numeric"
},
"geor": {
    "_rules": "georgian",
    "_type": "algorithmic"
},
"grek": {
    "_rules": "greek-upper",
    "_type": "algorithmic"
},
"greklow": {
    "_rules": "greek-lower",
    "_type": "algorithmic"
},
"gujr": {
    "_digits": "ૐ૧૨૩૪૫૬૭૮૯",
    "_type": "numeric"
},
"guru": {
    "_digits": "੐ੑ੒੓੔੕੖੗੘ਖ਼ਗ਼ਜ਼",
    "_type": "numeric"
},
"hanidays": {
    "_rules": "zh/SpelloutRules/spellout-numbering-days",
    "_type": "algorithmic"
},
"hanidec": {
    "_digits": "〇一二三四五六七八九",
    "_type": "numeric"
},
"hans": {
    "_rules": "zh/SpelloutRules/spellout-cardinal",
    "_type": "algorithmic"
```

```

    },
    "hansfin": {
      "_rules": "zh/SpelloutRules/spellout-cardinal-financial",
      "_type": "algorithmic"
    },
    "hant": {
      "_rules": "zh_Hant/SpelloutRules/spellout-cardinal",
      "_type": "algorithmic"
    },
    "hantfin": {
      "_rules": "zh_Hant/SpelloutRules/spellout-cardinal-financial",
      "_type": "algorithmic"
    },
    "hebr": {
      "_rules": "hebrew",
      "_type": "algorithmic"
    },
    "hmng": {
      "_digits": "□□□□□□□□□□",
      "_type": "numeric"
    },
    "java": {
      "_digits": "᐀ᐁᐂᐃᐄᐅᐆᐇᐈᐉᐊᐋᐌᐍᐎᐏᐐᐑᐒᐓᐔᐕᐖᐗᐘᐙᐚᐛᐜᐝᐞᐟᐠᐡᐢᐣᐤᐥᐦᐧᐨᐩᐪᐫᐬᐭᐮᐯᐰᐱᐲᐳᐴᐵᐶᐷᐸᐹᐺᐻᐼᐽᐾᐿ",
      "_type": "numeric"
    },
    "jpan": {
      "_rules": "ja/SpelloutRules/spellout-cardinal",
      "_type": "algorithmic"
    },
    "jpanfin": {
      "_rules": "ja/SpelloutRules/spellout-cardinal-financial",
      "_type": "algorithmic"
    },
    "kali": {
      "_digits": "□□□□□□□□□□",
      "_type": "numeric"
    },
    "khmr": {
      "_digits": "០១២៣៤៥៦៧៨៩",
      "_type": "numeric"
    },
    "knda": {
      "_digits": "೦೧೨೩೪೫೬೭೮೯",
      "_type": "numeric"
    },
    "lana": {
      "_digits": "□□□□□□□□□□",
      "_type": "numeric"
    },
    "lanatham": {
      "_digits": "□□□□□□□□□□",
      "_type": "numeric"
    },
    "lao": {
      "_digits": "໐໑໒໓໔໕໖໗໘໙",

```

```

    "_type": "numeric"
  },
  "latn": {
    "_digits": "0123456789",
    "_type": "numeric"
  },
  "lepc": {
    "_digits": "□□□□□□□□□□",
    "_type": "numeric"
  },
  "limb": {
    "_digits": "□□□□□□□□□□",
    "_type": "numeric"
  },
  "mathbold": {
    "_digits": "0123456789",
    "_type": "numeric"
  },
  "mathdbl": {
    "_digits": "0123456789",
    "_type": "numeric"
  },
  "mathmono": {
    "_digits": "0123456789",
    "_type": "numeric"
  },
  "mathsanb": {
    "_digits": "0123456789",
    "_type": "numeric"
  },
  "mathsans": {
    "_digits": "0123456789",
    "_type": "numeric"
  },
  "mlym": {
    "_digits": "൦൧൨൩൪൫൬൭൮൯",
    "_type": "numeric"
  },
  "modi": {
    "_digits": "□□□□□□□□□□",
    "_type": "numeric"
  },
  "mong": {
    "_digits": "᠐᠑᠒᠓᠐ᠠᠨᠣᠯᠤᠰ",
    "_type": "numeric"
  },
  "mroo": {
    "_digits": "□□□□□□□□□□",
    "_type": "numeric"
  },
  "mtei": {
    "_digits": "᱀᱁᱂᱃᱄᱅᱆᱇᱈᱉",
    "_type": "numeric"
  },
  "mymr": {

```

```

    "_digits": "၀၁၂၃၄၅၆၇၈၉",
    "_type": "numeric"
  },
  "mymrshan": {
    "_digits": "၀၁၂၃၄၅၆၇၈၉",
    "_type": "numeric"
  },
  "mymrtlng": {
    "_digits": "၀၁၂၃၄၅၆၇၈၉",
    "_type": "numeric"
  },
  "newa": {
    "_digits": "၀၁၂၃၄၅၆၇၈၉",
    "_type": "numeric"
  },
  "nkoo": {
    "_digits": "၀၁၂၃၄၅၆၇၈၉",
    "_type": "numeric"
  },
  "olck": {
    "_digits": "၀၁၂၃၄၅၆၇၈၉",
    "_type": "numeric"
  },
  "orya": {
    "_digits": "၀၁၂၃၄၅၆၇၈၉",
    "_type": "numeric"
  },
  "osma": {
    "_digits": "၀၁၂၃၄၅၆၇၈၉",
    "_type": "numeric"
  },
  "roman": {
    "_rules": "roman-upper",
    "_type": "algorithmic"
  },
  "romanlow": {
    "_rules": "roman-lower",
    "_type": "algorithmic"
  },
  "saur": {
    "_digits": "၀၁၂၃၄၅၆၇၈၉",
    "_type": "numeric"
  },
  "shrd": {
    "_digits": "၀၁၂၃၄၅၆၇၈၉",
    "_type": "numeric"
  },
  "sind": {
    "_digits": "၀၁၂၃၄၅၆၇၈၉",
    "_type": "numeric"
  },
  "sinh": {
    "_digits": "၀၁၂၃၄၅၆၇၈၉",
    "_type": "numeric"
  }

```

```

    },
    "sora": {
      "_digits": "0ᑭᑭᑭᑭᑭᑭᑭᑭ",
      "_type": "numeric"
    },
    "sund": {
      "_digits": "□□□□□□□□□□",
      "_type": "numeric"
    },
    "takr": {
      "_digits": "□□□□□□□□□□",
      "_type": "numeric"
    },
    "talu": {
      "_digits": "၀၁၂၃၄၅၆၇၈၉",
      "_type": "numeric"
    },
    "taml": {
      "_rules": "tamil",
      "_type": "algorithmic"
    },
    "tamldec": {
      "_digits": "0௧௨௩௪௫௬௭௮௯",
      "_type": "numeric"
    },
    "telu": {
      "_digits": "౦౧౨౩౪౫౬౭౮౯",
      "_type": "numeric"
    },
    "thai": {
      "_digits": "๐๑๒๓๔๕๖๗๘๙",
      "_type": "numeric"
    },
    "tibt": {
      "_digits": "༠༡༢༣༤༥༦༧༨༩",
      "_type": "numeric"
    },
    "tirh": {
      "_digits": "□□□□□□□□□□",
      "_type": "numeric"
    },
    "vaih": {
      "_digits": "0ᐁᐂᐃᐄᐅᐆᐇᐈᐉ",
      "_type": "numeric"
    },
    "wara": {
      "_digits": "□□□□□□□□□□",
      "_type": "numeric"
    }
  }
}

```

## NUMBERINGSYSTEMS.JSX

```
{
  "supplemental";
  {
    "version";
    {
      "_number";
      "$Revision: 12732 $",
      "_unicodeVersion";
      "9.0.0",
      "_cldrVersion";
      "31";
    }
    "numberingSystems";
    {
      "adlm";
      {
        "_digits";
        "୧୨୩୪୫୬୭୮୯୦",
        "_type";
        "numeric";
      }
      "ahom";
      {
        "_digits";
        "᱀᱁᱂᱃᱄᱅᱆᱇᱈᱉",
        "_type";
        "numeric";
      }
      "arab";
      {
        "_digits";
        "٠١٢٣٤٥٦٧٨٩",
        "_type";
        "numeric";
      }
      "arabext";
      {
        "_digits";
        "٠١٢٣٤٥٦٧٨٩",
        "_type";
        "numeric";
      }
      "armn";
      {
        "_rules";
        "armenian-upper",
        "_type";
        "algorithmic";
      }
      "armnlow";
      {
        "_rules";
        "armenian-lower",
        "_type";
        "algorithmic";
      }
    }
  }
}
```

```
"bali";
{
  "_digits";
  "□□□□□□□□",
  "_type";
  "numeric";
}
"beng";
{
  "_digits";
  "০১২৩৪৫৬৭৮৯",
  "_type";
  "numeric";
}
"bhks";
{
  "_digits";
  "□□□□□□□□",
  "_type";
  "numeric";
}
"brah";
{
  "_digits";
  "·٠١٢٣٤٥٦٧٨٩",
  "_type";
  "numeric";
}
"cakm";
{
  "_digits";
  "ႤႤႬႬႭႭႬႬႬႬ",
  "_type";
  "numeric";
}
"cham";
{
  "_digits";
  "□□□□□□□□",
  "_type";
  "numeric";
}
"cyrl";
{
  "_rules";
  "cyrillic-lower",
  "_type";
  "algorithmic";
}
"deva";
{
  "_digits";
  "०१२३४५६७८९",
  "_type";
  "numeric";
}
```



```
"ethi";
{
  "_rules";
  "ethiopic",
  "_type";
  "algorithmic";
}
"fullwide";
{
  "_digits";
  "0 1 2 3 4 5 6 7 8 9",
  "_type";
  "numeric";
}
"geor";
{
  "_rules";
  "georgian",
  "_type";
  "algorithmic";
}
"grek";
{
  "_rules";
  "greek-upper",
  "_type";
  "algorithmic";
}
"greklow";
{
  "_rules";
  "greek-lower",
  "_type";
  "algorithmic";
}
"gujr";
{
  "_digits";
  "૦૧૨૩૪૫૬૭૮૯",
  "_type";
  "numeric";
}
"guru";
{
  "_digits";
  "੦੧੨੩੪੫੬੭੮੯",
  "_type";
  "numeric";
}
"hanidays";
{
  "_rules";
  "zh/SpelloutRules/spellout-numbering-days",
  "_type";
  "algorithmic";
}
```

```

"hanidec";
{
  "_digits";
  "〇一二三四五六七八九",
  "_type";
  "numeric";
}
"hans";
{
  "_rules";
  "zh/SpelloutRules/spellout-cardinal",
  "_type";
  "algorithmic";
}
"hansfin";
{
  "_rules";
  "zh/SpelloutRules/spellout-cardinal-financial",
  "_type";
  "algorithmic";
}
"hant";
{
  "_rules";
  "zh_Hant/SpelloutRules/spellout-cardinal",
  "_type";
  "algorithmic";
}
"hantfin";
{
  "_rules";
  "zh_Hant/SpelloutRules/spellout-cardinal-financial",
  "_type";
  "algorithmic";
}
"hebr";
{
  "_rules";
  "hebrew",
  "_type";
  "algorithmic";
}
"hmng";
{
  "_digits";
  "ᠠᠨᠠᠨᠠᠨᠠᠨᠠᠨᠠᠨᠠᠨ",
  "_type";
  "numeric";
}
"java";
{
  "_digits";
  "ᠠᠨᠠᠨᠠᠨᠠᠨᠠᠨᠠᠨᠠᠨ",
  "_type";
  "numeric";
}

```

```
}
"jpan";
{
  "_rules";
  "ja/SpelloutRules/spellout-cardinal",
  "_type";
  "algorithmic";
}
"jpanfin";
{
  "_rules";
  "ja/SpelloutRules/spellout-cardinal-financial",
  "_type";
  "algorithmic";
}
"kali";
{
  "_digits";
  "□□□□□□□□□□",
  "_type";
  "numeric";
}
"khmr";
{
  "_digits";
  "០១២៣៤៥៦៧៨៩",
  "_type";
  "numeric";
}
"knda";
{
  "_digits";
  "೦೧೨೩೪೫೬೭೮೯",
  "_type";
  "numeric";
}
"lana";
{
  "_digits";
  "□□□□□□□□□□",
  "_type";
  "numeric";
}
"lanatham";
{
  "_digits";
  "□□□□□□□□□□",
  "_type";
  "numeric";
}
"lao";
{
  "_digits";
  "໐໑໒໓໔໕໖໗໘໑",
  "_type";
  "numeric";
}
```

```
}
"latn";
{
  "_digits";
  "0123456789",
  "_type";
  "numeric";
}
"lepc";
{
  "_digits";
  "□□□□□□□□□□",
  "_type";
  "numeric";
}
"limb";
{
  "_digits";
  "□□□□□□□□□□",
  "_type";
  "numeric";
}
"mathbold";
{
  "_digits";
  "0123456789",
  "_type";
  "numeric";
}
"mathdbl";
{
  "_digits";
  "0123456789",
  "_type";
  "numeric";
}
"mathmono";
{
  "_digits";
  "0123456789",
  "_type";
  "numeric";
}
"mathsanb";
{
  "_digits";
  "0123456789",
  "_type";
  "numeric";
}
"mathsans";
{
  "_digits";
  "0123456789",
  "_type";
  "numeric";
}
```

```

    }
    "mlym";
    {
        "_digits";
        "൦൧൨൩൪൫൬൭൮൯",
        "_type";
        "numeric";
    }
    "modi";
    {
        "_digits";
        "൦൧൨൩൪൫൬൭൮൯",
        "_type";
        "numeric";
    }
    "mong";
    {
        "_digits";
        "0123456789",
        "_type";
        "numeric";
    }
    "mroo";
    {
        "_digits";
        "൦൧൨൩൪൫൬൭൮൯",
        "_type";
        "numeric";
    }
    "mtei";
    {
        "_digits";
        "0123456789",
        "_type";
        "numeric";
    }
    "mymr";
    {
        "_digits";
        "0123456789",
        "_type";
        "numeric";
    }
    "mymrshan";
    {
        "_digits";
        "0123456789",
        "_type";
        "numeric";
    }
    "mymrtlng";
    {
        "_digits";
        "0123456789",

```

```

        "_type";
        "numeric";
    }
    "newa";
    {
        "_digits";
        "□□□□□□□□□□",
        "_type";
        "numeric";
    }
    "nkoo";
    {
        "_digits";
        "ᲠᲡᲢᲣᲤᲦᲧᲨᲩ",
        "_type";
        "numeric";
    }
    "olck";
    {
        "_digits";
        "ᲚᲛᲜᲝᲞᲟᲠᲡᲢᲣᲤᲦᲧᲨᲩᲪᲫᲬᲭᲮᲯᲰᲱᲲᲳᲴᲵᲶᲷᲸᲹᲺ᲻᲼ᲽᲾᲿ",
        "_type";
        "numeric";
    }
    "orya";
    {
        "_digits";
        "ᲠᲡᲢᲣᲤᲦᲧᲨᲩᲪᲫᲬᲭᲮᲯᲰᲱᲲᲳᲴᲵᲶᲷᲸᲹᲺ᲻᲼ᲽᲾᲿ",
        "_type";
        "numeric";
    }
    "osma";
    {
        "_digits";
        "ᲠᲡᲢᲣᲤᲦᲧᲨᲩᲪᲫᲬᲭᲮᲯᲰᲱᲲᲳᲴᲵᲶᲷᲸᲹᲺ᲻᲼ᲽᲾᲿ",
        "_type";
        "numeric";
    }
    "roman";
    {
        "_rules";
        "roman-upper",
        "_type";
        "algorithmic";
    }
    "romanlow";
    {
        "_rules";
        "roman-lower",
        "_type";
        "algorithmic";
    }
    "saur";
    {
        "_digits";
        "□□□□□□□□□□",

```

```

        "_type";
        "numeric";
    }
    "shrd";
    {
        "_digits";
        "□□□□□□□□",
        "_type";
        "numeric";
    }
    "sind";
    {
        "_digits";
        "□□□□□□□□",
        "_type";
        "numeric";
    }
    "sinh";
    {
        "_digits";
        "௧௨௩௪௫௬௭௮",
        "_type";
        "numeric";
    }
    "sora";
    {
        "_digits";
        "0123456789",
        "_type";
        "numeric";
    }
    "sund";
    {
        "_digits";
        "□□□□□□□□",
        "_type";
        "numeric";
    }
    "takr";
    {
        "_digits";
        "□□□□□□□□",
        "_type";
        "numeric";
    }
    "talv";
    {
        "_digits";
        "0123456789",
        "_type";
        "numeric";
    }
    "taml";
    {
        "_rules";
        "tamil",

```

```

        "_type";
        "algorithmic";
    }
    "tamldec";
    {
        "_digits";
        "0௧௨௩௪௫௬௭௮௯",
        "_type";
        "numeric";
    }
    "telu";
    {
        "_digits";
        "0౧౨౩౪౫౬౭౮౯",
        "_type";
        "numeric";
    }
    "thai";
    {
        "_digits";
        "๐๑๒๓๔๕๖๗๘๙",
        "_type";
        "numeric";
    }
    "tibtb";
    {
        "_digits";
        "༠༡༢༣༤༥༦༧༨༩",
        "_type";
        "numeric";
    }
    "tirh";
    {
        "_digits";
        "୦୧୨୩୪୫୬୭୮୯",
        "_type";
        "numeric";
    }
    "vaii";
    {
        "_digits";
        "୦୧୨୩୪୫୬୭୮୯",
        "_type";
        "numeric";
    }
    "wara";
    {
        "_digits";
        "୦୧୨୩୪୫୬୭୮୯",
        "_type";
        "numeric";
    }
    }
}

```



**NUMBERS.JSON**

```
{
  "main": {
    "de": {
      "identity": {
        "version": {
          "_number": "$Revision: 13259 $",
          "_cldrVersion": "31"
        },
        "language": "de"
      },
      "numbers": {
        "defaultNumberingSystem": "latn",
        "otherNumberingSystems": {
          "native": "latn"
        },
        "minimumGroupingDigits": "1",
        "symbols-numberSystem-latn": {
          "decimal": ",",
          "group": ".",
          "list": ";",
          "percentSign": "%",
          "plusSign": "+",
          "minusSign": "-",
          "exponential": "E",
          "superscriptingExponent": "°",
          "perMille": "‰",
          "infinity": "∞",
          "nan": "NaN",
          "timeSeparator": ":"
        },
        "decimalFormats-numberSystem-latn": {
          "standard": "#,##0.###",
          "long": {
            "decimalFormat": {
              "1000-count-one": "0 Tausend",
              "1000-count-other": "0 Tausend",
              "10000-count-one": "00 Tausend",
              "10000-count-other": "00 Tausend",
              "100000-count-one": "000 Tausend",
              "100000-count-other": "000 Tausend",
              "1000000-count-one": "0 Million",
              "1000000-count-other": "0 Millionen",
              "10000000-count-one": "00 Millionen",
              "10000000-count-other": "00 Millionen",
              "100000000-count-one": "000 Millionen",
              "100000000-count-other": "000 Millionen",
              "1000000000-count-one": "0 Milliarden",
              "1000000000-count-other": "0 Milliarden",
              "10000000000-count-one": "00 Milliarden",
              "10000000000-count-other": "00 Milliarden",
              "100000000000-count-one": "000 Milliarden",
              "100000000000-count-other": "000 Milliarden",
              "1000000000000-count-one": "0 Billion",
              "1000000000000-count-other": "0 Billion"
```

```

        "1000000000000-count-other": "0 Billionen",
        "1000000000000-count-one": "00 Billionen",
        "1000000000000-count-other": "00 Billionen",
        "1000000000000-count-one": "000 Billionen",
        "1000000000000-count-other": "000 Billionen"
    }
},
"short": {
    "decimalFormat": {
        "1000-count-one": "0",
        "1000-count-other": "0",
        "10000-count-one": "0",
        "10000-count-other": "0",
        "100000-count-one": "0",
        "100000-count-other": "0",
        "1000000-count-one": "0 Mio'.'",
        "1000000-count-other": "0 Mio'.'",
        "10000000-count-one": "00 Mio'.'",
        "10000000-count-other": "00 Mio'.'",
        "100000000-count-one": "000 Mio'.'",
        "100000000-count-other": "000 Mio'.'",
        "1000000000-count-one": "0 Mrd'.'",
        "1000000000-count-other": "0 Mrd'.'",
        "10000000000-count-one": "00 Mrd'.'",
        "10000000000-count-other": "00 Mrd'.'",
        "100000000000-count-one": "000 Mrd'.'",
        "100000000000-count-other": "000 Mrd'.'",
        "1000000000000-count-one": "0 Bio'.'",
        "1000000000000-count-other": "0 Bio'.'",
        "10000000000000-count-one": "00 Bio'.'",
        "10000000000000-count-other": "00 Bio'.'",
        "100000000000000-count-one": "000 Bio'.'",
        "100000000000000-count-other": "000 Bio'.'"
    }
}
},
"scientificFormats-numberSystem-latn": {
    "standard": "#E0"
},
"percentFormats-numberSystem-latn": {
    "standard": "#,##0 %"
},
"currencyFormats-numberSystem-latn": {
    "currencySpacing": {
        "beforeCurrency": {
            "currencyMatch": "[:^S:]",
            "surroundingMatch": "[:digit:]",
            "insertBetween": " "
        },
        "afterCurrency": {
            "currencyMatch": "[:^S:]",
            "surroundingMatch": "[:digit:]",
            "insertBetween": " "
        }
    },
    "standard": "#,##0.00 ¤",
    "accounting": "#,##0.00 ¤",

```

```

    "short": {
      "standard": {
        "1000-count-one": "0 Tsd'.' ¤",
        "1000-count-other": "0 Tsd'.' ¤",
        "10000-count-one": "00 Tsd'.' ¤",
        "10000-count-other": "00 Tsd'.' ¤",
        "100000-count-one": "000 Tsd'.' ¤",
        "100000-count-other": "000 Tsd'.' ¤",
        "1000000-count-one": "0 Mio'.' ¤",
        "1000000-count-other": "0 Mio'.' ¤",
        "10000000-count-one": "00 Mio'.' ¤",
        "10000000-count-other": "00 Mio'.' ¤",
        "100000000-count-one": "000 Mio'.' ¤",
        "100000000-count-other": "000 Mio'.' ¤",
        "1000000000-count-one": "0 Mrd'.' ¤",
        "1000000000-count-other": "0 Mrd'.' ¤",
        "10000000000-count-one": "00 Mrd'.' ¤",
        "10000000000-count-other": "00 Mrd'.' ¤",
        "100000000000-count-one": "000 Mrd'.' ¤",
        "100000000000-count-other": "000 Mrd'.' ¤",
        "1000000000000-count-one": "0 Bio'.' ¤",
        "1000000000000-count-other": "0 Bio'.' ¤",
        "10000000000000-count-one": "00 Bio'.' ¤",
        "10000000000000-count-other": "00 Bio'.' ¤",
        "100000000000000-count-one": "000 Bio'.' ¤",
        "100000000000000-count-other": "000 Bio'.' ¤"
      }
    },
    "unitPattern-count-one": "{0} {1}",
    "unitPattern-count-other": "{0} {1}"
  },
  "miscPatterns-numberSystem-latn": {
    "atLeast": "{0}+",
    "range": "{0}-{1}"
  },
  "minimalPairs": {
    "pluralMinimalPairs": "{0} Tag",
    "pluralMinimalPairs": "{0} Tage",
    "other": "{0}. Abzweigung nach rechts nehmen"
  }
}
}
}
}
}

```

## **NUMBERS.JSX**

```

{
  "main";
  {
    "de";
    {
      "identity";
      {
        "version";
        {

```

```

        "_number";
        "$Revision: 13259 $",
        "_cldrVersion";
        "31";
    }
    "language";
    "de";
}
"numbers";
{
    "defaultNumberingSystem";
    "latn",
    "otherNumberingSystems";
    {
        "native";
        "latn";
    }
    "minimumGroupingDigits";
    "1",
    "symbols-numberSystem-latn";
    {
        "decimal";
        ",",
        "group";
        ".",
        "list";
        ";",
        "percentSign";
        "%",
        "plusSign";
        "+",
        "minusSign";
        "-",
        "exponential";
        "E",
        "superscriptingExponent";
        " .",
        "perMille";
        "‰",
        "infinity";
        "∞",
        "nan";
        "NaN",
        "timeSeparator";
        ":";
    }
    "decimalFormats-numberSystem-latn";
    {
        "standard";
        "#,##0.###",
        "long";
        {
            "decimalFormat";
            {
                "1000-count-one";
                "0 Tausend",
                "1000-count-other";
            }
        }
    }
}

```

```

        "0 Tausend",
        "10000-count-one";
    "00 Tausend",
        "10000-count-other";
    "00 Tausend",
        "100000-count-one";
    "000 Tausend",
        "100000-count-other";
    "000 Tausend",
        "1000000-count-one";
    "0 Million",
        "1000000-count-other";
    "0 Millionen",
        "10000000-count-one";
    "00 Millionen",
        "10000000-count-other";
    "00 Millionen",
        "100000000-count-one";
    "000 Millionen",
        "100000000-count-other";
    "000 Millionen",
        "1000000000-count-one";
    "0 Milliarde",
        "1000000000-count-other";
    "0 Milliarden",
        "10000000000-count-one";
    "00 Milliarden",
        "10000000000-count-other";
    "00 Milliarden",
        "100000000000-count-one";
    "000 Milliarden",
        "100000000000-count-other";
    "000 Milliarden",
        "1000000000000-count-one";
    "0 Billion",
        "1000000000000-count-other";
    "0 Billionen",
        "10000000000000-count-one";
    "00 Billionen",
        "10000000000000-count-other";
    "00 Billionen",
        "100000000000000-count-one";
    "000 Billionen",
        "100000000000000-count-other";
    "000 Billionen";
    }
}
"short";
{
    "decimalFormat";
    {
        "1000-count-one";
        "0",
        "1000-count-other";
        "0",
        "10000-count-one";
        "0",

```

```

        "10000-count-other";
        "0",
        "100000-count-one";
        "0",
        "100000-count-other";
        "0",
        "1000000-count-one";
        "0 Mio'.'",
        "1000000-count-other";
        "0 Mio'.'",
        "10000000-count-one";
        "00 Mio'.'",
        "10000000-count-other";
        "00 Mio'.'",
        "100000000-count-one";
        "000 Mio'.'",
        "100000000-count-other";
        "000 Mio'.'",
        "1000000000-count-one";
        "0 Mrd'.'",
        "1000000000-count-other";
        "0 Mrd'.'",
        "10000000000-count-one";
        "00 Mrd'.'",
        "10000000000-count-other";
        "00 Mrd'.'",
        "100000000000-count-one";
        "000 Mrd'.'",
        "100000000000-count-other";
        "000 Mrd'.'",
        "1000000000000-count-one";
        "0 Bio'.'",
        "1000000000000-count-other";
        "0 Bio'.'",
        "10000000000000-count-one";
        "00 Bio'.'",
        "10000000000000-count-other";
        "00 Bio'.'",
        "100000000000000-count-one";
        "000 Bio'.'",
        "1000000000000000-count-other";
        "000 Bio'.'";
    }
}
"scientificFormats-numberSystem-latn";
{
    "standard";
    "#E0";
}
"percentFormats-numberSystem-latn";
{
    "standard";
    "#,##0 %";
}
"currencyFormats-numberSystem-latn";
{

```

```

    "currencySpacing";
    {
      "beforeCurrency";
      {
        "currencyMatch";
        "[:^S:]",
        "surroundingMatch";
        "[:digit:]",
        "insertBetween";
        " ";
      }
      "afterCurrency";
      {
        "currencyMatch";
        "[:^S:]",
        "surroundingMatch";
        "[:digit:]",
        "insertBetween";
        " ";
      }
    }
    "standard";
    "#,##0.00 ¤",
    "accounting";
    "#,##0.00 ¤",
    "short";
    {
      "standard";
      {
        "1000-count-one";
        "0 Tsd'.' ¤",
        "1000-count-other";
        "0 Tsd'.' ¤",
        "10000-count-one";
        "00 Tsd'.' ¤",
        "10000-count-other";
        "00 Tsd'.' ¤",
        "100000-count-one";
        "000 Tsd'.' ¤",
        "100000-count-other";
        "000 Tsd'.' ¤",
        "1000000-count-one";
        "0 Mio'.' ¤",
        "1000000-count-other";
        "0 Mio'.' ¤",
        "10000000-count-one";
        "00 Mio'.' ¤",
        "10000000-count-other";
        "00 Mio'.' ¤",
        "100000000-count-one";
        "000 Mio'.' ¤",
        "100000000-count-other";
        "000 Mio'.' ¤",
        "1000000000-count-one";
        "0 Mrd'.' ¤",
        "1000000000-count-other";
        "0 Mrd'.' ¤",
      }
    }
  
```

```

        "10000000000-count-one";
    "00 Mrd'.' s",
    "10000000000-count-other";
    "00 Mrd'.' s",
    "10000000000-count-one";
    "000 Mrd'.' s",
    "10000000000-count-other";
    "000 Mrd'.' s",
    "10000000000-count-one";
    "0 Bio'.' s",
    "10000000000-count-other";
    "0 Bio'.' s",
    "10000000000-count-one";
    "00 Bio'.' s",
    "10000000000-count-other";
    "00 Bio'.' s",
    "10000000000-count-one";
    "000 Bio'.' s",
    "10000000000-count-other";
    "000 Bio'.' s";
}
}
unitPattern-count-one";
"{0} {1}",
    unitPattern-count-other";
"{0} {1}";
}
miscPatterns-numberSystem-latn";
{
    "atLeast";
    "{0}+",
    "range";
    "{0}-{1}";
}
minimalPairs";
{
    pluralMinimalPairs";
    "{0} Tag",
    pluralMinimalPairs";
    "{0} Tage",
    "other";
    "{0}. Abzweigung nach rechts nehmen";
}
}
}
}

```

[Functional-component]

## CA-GREGORIAN.JSON

```
{
  "main": {
    "de": {
      "identity": {
```



```
"version": {
  "_number": "$Revision: 12879 $",
  "_cldrVersion": "30.0.3"
},
"language": "de"
},
"dates": {
  "calendars": {
    "gregorian": {
      "months": {
        "format": {
          "abbreviated": {
            "1": "Jan.",
            "2": "Feb.",
            "3": "März",
            "4": "Apr.",
            "5": "Mai",
            "6": "Juni",
            "7": "Juli",
            "8": "Aug.",
            "9": "Sep.",
            "10": "Okt.",
            "11": "Nov.",
            "12": "Dez."
          },
          "narrow": {
            "1": "J",
            "2": "F",
            "3": "M",
            "4": "A",
            "5": "M",
            "6": "J",
            "7": "J",
            "8": "A",
            "9": "S",
            "10": "O",
            "11": "N",
            "12": "D"
          },
          "wide": {
            "1": "Januar",
            "2": "Februar",
            "3": "März",
            "4": "April",
            "5": "Mai",
            "6": "Juni",
            "7": "Juli",
            "8": "August",
            "9": "September",
            "10": "Oktober",
            "11": "November",
            "12": "Dezember"
          }
        }
      },
      "stand-alone": {
        "abbreviated": {
          "1": "Jan",
```

```
        "2": "Feb",
        "3": "Mär",
        "4": "Apr",
        "5": "Mai",
        "6": "Jun",
        "7": "Jul",
        "8": "Aug",
        "9": "Sep",
        "10": "Okt",
        "11": "Nov",
        "12": "Dez"
    },
    "narrow": {
        "1": "J",
        "2": "F",
        "3": "M",
        "4": "A",
        "5": "M",
        "6": "J",
        "7": "J",
        "8": "A",
        "9": "S",
        "10": "O",
        "11": "N",
        "12": "D"
    },
    "wide": {
        "1": "Januar",
        "2": "Februar",
        "3": "März",
        "4": "April",
        "5": "Mai",
        "6": "Juni",
        "7": "Juli",
        "8": "August",
        "9": "September",
        "10": "Oktober",
        "11": "November",
        "12": "Dezember"
    }
}
},
"days": {
    "format": {
        "abbreviated": {
            "sun": "So.",
            "mon": "Mo.",
            "tue": "Di.",
            "wed": "Mi.",
            "thu": "Do.",
            "fri": "Fr.",
            "sat": "Sa."
        },
        "narrow": {
            "sun": "S",
            "mon": "M",
            "tue": "D",
```

```
        "wed": "M",
        "thu": "D",
        "fri": "F",
        "sat": "S"
    },
    "short": {
        "sun": "So.",
        "mon": "Mo.",
        "tue": "Di.",
        "wed": "Mi.",
        "thu": "Do.",
        "fri": "Fr.",
        "sat": "Sa."
    },
    "wide": {
        "sun": "Sonntag",
        "mon": "Montag",
        "tue": "Dienstag",
        "wed": "Mittwoch",
        "thu": "Donnerstag",
        "fri": "Freitag",
        "sat": "Samstag"
    }
},
"stand-alone": {
    "abbreviated": {
        "sun": "So",
        "mon": "Mo",
        "tue": "Di",
        "wed": "Mi",
        "thu": "Do",
        "fri": "Fr",
        "sat": "Sa"
    },
    "narrow": {
        "sun": "S",
        "mon": "M",
        "tue": "D",
        "wed": "M",
        "thu": "D",
        "fri": "F",
        "sat": "S"
    },
    "short": {
        "sun": "So.",
        "mon": "Mo.",
        "tue": "Di.",
        "wed": "Mi.",
        "thu": "Do.",
        "fri": "Fr.",
        "sat": "Sa."
    },
    "wide": {
        "sun": "Sonntag",
        "mon": "Montag",
        "tue": "Dienstag",
        "wed": "Mittwoch",
```

```

        "thu": "Donnerstag",
        "fri": "Freitag",
        "sat": "Samstag"
    }
}
},
"quarters": {
    "format": {
        "abbreviated": {
            "1": "Q1",
            "2": "Q2",
            "3": "Q3",
            "4": "Q4"
        },
        "narrow": {
            "1": "1",
            "2": "2",
            "3": "3",
            "4": "4"
        },
        "wide": {
            "1": "1. Quartal",
            "2": "2. Quartal",
            "3": "3. Quartal",
            "4": "4. Quartal"
        }
    },
    "stand-alone": {
        "abbreviated": {
            "1": "Q1",
            "2": "Q2",
            "3": "Q3",
            "4": "Q4"
        },
        "narrow": {
            "1": "1",
            "2": "2",
            "3": "3",
            "4": "4"
        },
        "wide": {
            "1": "1. Quartal",
            "2": "2. Quartal",
            "3": "3. Quartal",
            "4": "4. Quartal"
        }
    }
},
"dayPeriods": {
    "format": {
        "abbreviated": {
            "midnight": "Mitternacht",
            "am": "vorm.",
            "pm": "nachm.",
            "morning1": "morgens",
            "morning2": "vormittags",
            "afternoon1": "mittags",

```

```

        "afternoon2": "nachmittags",
        "evening1": "abends",
        "night1": "nachts"
    },
    "narrow": {
        "midnight": "Mitternacht",
        "am": "vm.",
        "pm": "nm.",
        "morning1": "morgens",
        "morning2": "vormittags",
        "afternoon1": "mittags",
        "afternoon2": "nachmittags",
        "evening1": "abends",
        "night1": "nachts"
    },
    "wide": {
        "midnight": "Mitternacht",
        "am": "vorm.",
        "pm": "nachm.",
        "morning1": "morgens",
        "morning2": "vormittags",
        "afternoon1": "mittags",
        "afternoon2": "nachmittags",
        "evening1": "abends",
        "night1": "nachts"
    }
},
"stand-alone": {
    "abbreviated": {
        "midnight": "Mitternacht",
        "am": "vorm.",
        "pm": "nachm.",
        "morning1": "Morgen",
        "morning2": "Vormittag",
        "afternoon1": "Mittag",
        "afternoon2": "Nachmittag",
        "evening1": "Abend",
        "night1": "Nacht"
    },
    "narrow": {
        "midnight": "Mitternacht",
        "am": "vorm.",
        "pm": "nachm.",
        "morning1": "Morgen",
        "morning2": "Vormittag",
        "afternoon1": "Mittag",
        "afternoon2": "Nachmittag",
        "evening1": "Abend",
        "night1": "Nacht"
    },
    "wide": {
        "midnight": "Mitternacht",
        "am": "vorm.",
        "pm": "nachm.",
        "morning1": "Morgen",
        "morning2": "Vormittag",
        "afternoon1": "Mittag",

```

```

        "afternoon2": "Nachmittag",
        "evening1": "Abend",
        "night1": "Nacht"
    }
}
},
"eras": {
    "eraNames": {
        "0": "v. Chr.",
        "0-alt-variant": "vor unserer Zeitrechnung",
        "1": "n. Chr.",
        "1-alt-variant": "unserer Zeitrechnung"
    },
    "eraAbbr": {
        "0": "v. Chr.",
        "0-alt-variant": "v. u. Z.",
        "1": "n. Chr.",
        "1-alt-variant": "u. Z."
    },
    "eraNarrow": {
        "0": "v. Chr.",
        "0-alt-variant": "v. u. Z.",
        "1": "n. Chr.",
        "1-alt-variant": "u. Z."
    }
},
"dateFormats": {
    "full": "EEEE, d. MMMM y",
    "long": "d. MMMM y",
    "medium": "dd.MM.y",
    "short": "dd.MM.yy"
},
"timeFormats": {
    "full": "HH:mm:ss zzzz",
    "long": "HH:mm:ss z",
    "medium": "HH:mm:ss",
    "short": "HH:mm"
},
"dateTimeFormats": {
    "full": "{1} 'um' {0}",
    "long": "{1} 'um' {0}",
    "medium": "{1}, {0}",
    "short": "{1}, {0}",
    "availableFormats": {
        "d": "d",
        "E": "ccc",
        "Ed": "E, d.",
        "Ehm": "E h:mm a",
        "EHm": "E, HH:mm",
        "Ehms": "E, h:mm:ss a",
        "EHms": "E, HH:mm:ss",
        "Gy": "y G",
        "GyMMM": "MMM y G",
        "GyMMMd": "d. MMM y G",
        "GyMMMED": "E, d. MMM y G",
        "h": "h 'Uhr' a",
        "H": "HH 'Uhr'",
    }
}

```

```

    "hm": "h:mm a",
    "Hm": "HH:mm",
    "hms": "h:mm:ss a",
    "Hms": "HH:mm:ss",
    "hmsv": "h:mm:ss a v",
    "Hmsv": "HH:mm:ss v",
    "hmv": "h:mm a v",
    "Hmv": "HH:mm v",
    "M": "L",
    "Md": "d.M.",
    "MEd": "E, d.M.",
    "MMd": "d.MM.",
    "MMdd": "dd.MM.",
    "MMM": "LLL",
    "MMMd": "d. MMM",
    "MMMEd": "E, d. MMM",
    "MMMMd": "d. MMMM",
    "MMMMEd": "E, d. MMMM",
    "MMMMW": "'Woche' W 'im' MMM",
    "MMMMW": "'Woche' W 'im' MMM",
    "ms": "mm:ss",
    "y": "y",
    "yM": "M.y",
    "yMd": "d.M.y",
    "yMEd": "E, d.M.y",
    "yMM": "MM.y",
    "yMMdd": "dd.MM.y",
    "yMMM": "MMM y",
    "yMMMd": "d. MMM y",
    "yMMMEd": "E, d. MMM y",
    "yMMMM": "MMMM y",
    "yQQQ": "QQQ y",
    "yQQQQ": "QQQQ y",
    "yw": "'Woche' w 'des' 'Jahres' y",
    "yw": "'Woche' w 'des' 'Jahres' y"
  },
  "appendItems": {
    "Day": "{0} ({2}: {1})",
    "Day-Of-Week": "{0} {1}",
    "Era": "{1} {0}",
    "Hour": "{0} ({2}: {1})",
    "Minute": "{0} ({2}: {1})",
    "Month": "{0} ({2}: {1})",
    "Quarter": "{0} ({2}: {1})",
    "Second": "{0} ({2}: {1})",
    "Timezone": "{0} {1}",
    "Week": "{0} ({2}: {1})",
    "Year": "{1} {0}"
  },
  "intervalFormats": {
    "intervalFormatFallback": "{0} - {1}",
    "d": {
      "d": "d.-d."
    },
    "h": {
      "a": "h 'Uhr' a - h 'Uhr' a",
      "h": "h - h 'Uhr' a"
    }
  }
}

```

```

    },
    "H": {
      "H": "HH-HH 'Uhr'"
    },
    "hm": {
      "a": "h:mm a - h:mm a",
      "h": "h:mm-h:mm a",
      "m": "h:mm-h:mm a"
    },
    "Hm": {
      "H": "HH:mm-HH:mm 'Uhr'",
      "m": "HH:mm-HH:mm 'Uhr'"
    },
    "hmv": {
      "a": "h:mm a - h:mm a v",
      "h": "h:mm-h:mm a v",
      "m": "h:mm-h:mm a v"
    },
    "Hmv": {
      "H": "HH:mm-HH:mm 'Uhr' v",
      "m": "HH:mm-HH:mm 'Uhr' v"
    },
    "hv": {
      "a": "h a - h a v",
      "h": "h-h a v"
    },
    "Hv": {
      "H": "HH-HH 'Uhr' v"
    },
    "M": {
      "M": "M.-M."
    },
    "Md": {
      "d": "dd.MM. - dd.MM.",
      "M": "dd.MM. - dd.MM."
    },
    "MEd": {
      "d": "E, dd.MM. - E, dd.MM.",
      "M": "E, dd.MM. - E, dd.MM."
    },
    "MMM": {
      "M": "MMM-MMM"
    },
    "MMMd": {
      "d": "d.-d. MMM",
      "M": "d. MMM - d. MMM"
    },
    "MMMEd": {
      "d": "E, d. - E, d. MMM",
      "M": "E, d. MMM - E, d. MMM"
    },
    "MMMM": {
      "M": "LLLL-LLLL"
    },
    "Y": {
      "Y": "Y-Y"
    }
  },

```



```
"yM": {
    "M": "MM.y - MM.y",
    "y": "MM.y - MM.y"
},
"yMd": {
    "d": "dd.MM.y - dd.MM.y",
    "M": "dd.MM.y - dd.MM.y",
    "y": "dd.MM.y - dd.MM.y"
},
"yMEd": {
    "d": "E, dd.MM.y - E, dd.MM.y",
    "M": "E, dd.MM.y - E, dd.MM.y",
    "y": "E, dd.MM.y - E, dd.MM.y"
},
"yMMM": {
    "M": "MMM-MMM y",
    "y": "MMM y - MMM y"
},
"yMMMd": {
    "d": "d.-d. MMM y",
    "M": "d. MMM - d. MMM y",
    "y": "d. MMM y - d. MMM y"
},
"yMMMEd": {
    "d": "E, d. - E, d. MMM y",
    "M": "E, d. MMM - E, d. MMM y",
    "y": "E, d. MMM y - E, d. MMM y"
},
"yMMMM": {
    "M": "MMMM-MMMM y",
    "y": "MMMM y - MMMM y"
}
}
```

## CA-GREGORIAN.JSX

```
{
    "main";
    {
        "de";
        {
            "identity";
            {
                "version";
                {
                    "_number";
                    "$Revision: 12879 $",
                    "_cldrVersion";
                    "30.0.3";
```

```
    }  
    "language";  
    "de";  
  }  
  "dates";  
  {  
    "calendars";  
    {  
      "gregorian";  
      {  
        "months";  
        {  
          "format";  
          {  
            "abbreviated";  
            {  
              "1";  
              "Jan.",  
              "2";  
              "Feb.",  
              "3";  
              "März",  
              "4";  
              "Apr.",  
              "5";  
              "Mai",  
              "6";  
              "Juni",  
              "7";  
              "Juli",  
              "8";  
              "Aug.",  
              "9";  
              "Sep.",  
              "10";  
              "Okt.",  
              "11";  
              "Nov.",  
              "12";  
              "Dez.";  
            }  
            "narrow";  
            {  
              "1";  
              "J",  
              "2";  
              "F",  
              "3";  
              "M",  
              "4";  
              "A",  
              "5";  
              "M",  
              "6";  
              "J",  
              "7";  
              "J",  
            }  
          }  
        }  
      }  
    }  
  }  
}
```

```
        "8";
        "A",
        "9";
        "S",
        "10";
        "O",
        "11";
        "N",
        "12";
        "D";
    }
    "wide";
    {
        "1";
        "Januar",
        "2";
        "Februar",
        "3";
        "März",
        "4";
        "April",
        "5";
        "Mai",
        "6";
        "Juni",
        "7";
        "Juli",
        "8";
        "August",
        "9";
        "September",
        "10";
        "Oktober",
        "11";
        "November",
        "12";
        "Dezember";
    }
}
"stand-alone";
{
    "abbreviated";
    {
        "1";
        "Jan",
        "2";
        "Feb",
        "3";
        "Mär",
        "4";
        "Apr",
        "5";
        "Mai",
        "6";
        "Jun",
        "7";
        "Jul",
```

```
        "8";
        "Aug",
        "9";
        "Sep",
        "10";
        "Okt",
        "11";
        "Nov",
        "12";
        "Dez";
    }
    "narrow";
    {
        "1";
        "J",
        "2";
        "F",
        "3";
        "M",
        "4";
        "A",
        "5";
        "M",
        "6";
        "J",
        "7";
        "J",
        "8";
        "A",
        "9";
        "S",
        "10";
        "O",
        "11";
        "N",
        "12";
        "D";
    }
    "wide";
    {
        "1";
        "Januar",
        "2";
        "Februar",
        "3";
        "März",
        "4";
        "April",
        "5";
        "Mai",
        "6";
        "Juni",
        "7";
        "Juli",
        "8";
        "August",
        "9";
```

```
        "September",
        "10";
        "Oktober",
        "11";
        "November",
        "12";
        "Dezember";
    }
}
"days";
{
    "format";
    {
        "abbreviated";
        {
            "sun";
            "So.",
            "mon";
            "Mo.",
            "tue";
            "Di.",
            "wed";
            "Mi.",
            "thu";
            "Do.",
            "fri";
            "Fr.",
            "sat";
            "Sa.";
        }
        "narrow";
        {
            "sun";
            "S",
            "mon";
            "M",
            "tue";
            "D",
            "wed";
            "M",
            "thu";
            "D",
            "fri";
            "F",
            "sat";
            "S";
        }
        "short";
        {
            "sun";
            "So.",
            "mon";
            "Mo.",
            "tue";
            "Di.",
            "wed";
```

```
        "Mi.",
        "thu";
        "Do.",
        "fri";
        "Fr.",
        "sat";
        "Sa.";
    }
    "wide";
    {
        "sun";
        "Sonntag",
        "mon";
        "Montag",
        "tue";
        "Dienstag",
        "wed";
        "Mittwoch",
        "thu";
        "Donnerstag",
        "fri";
        "Freitag",
        "sat";
        "Samstag";
    }
}
"stand-alone";
{
    "abbreviated";
    {
        "sun";
        "So",
        "mon";
        "Mo",
        "tue";
        "Di",
        "wed";
        "Mi",
        "thu";
        "Do",
        "fri";
        "Fr",
        "sat";
        "Sa";
    }
    "narrow";
    {
        "sun";
        "S",
        "mon";
        "M",
        "tue";
        "D",
        "wed";
        "M",
        "thu";
        "D",
```

```
        "fri";
        "F",
        "sat";
        "S";
    }
    "short";
    {
        "sun";
        "So.",
        "mon";
        "Mo.",
        "tue";
        "Di.",
        "wed";
        "Mi.",
        "thu";
        "Do.",
        "fri";
        "Fr.",
        "sat";
        "Sa.";
    }
    "wide";
    {
        "sun";
        "Sonntag",
        "mon";
        "Montag",
        "tue";
        "Dienstag",
        "wed";
        "Mittwoch",
        "thu";
        "Donnerstag",
        "fri";
        "Freitag",
        "sat";
        "Samstag";
    }
    }
}
"quarters";
{
    "format";
    {
        "abbreviated";
        {
            "1";
            "Q1",
            "2";
            "Q2",
            "3";
            "Q3",
            "4";
            "Q4";
        }
    }
    "narrow";
```

```
{
  "1";
  "1",
    "2";
  "2",
    "3";
  "3",
    "4";
  "4";
}
"wide";
{
  "1";
  "1. Quartal",
    "2";
  "2. Quartal",
    "3";
  "3. Quartal",
    "4";
  "4. Quartal";
}
}
"stand-alone";
{
  "abbreviated";
  {
    "1";
    "Q1",
      "2";
    "Q2",
      "3";
    "Q3",
      "4";
    "Q4";
  }
  "narrow";
  {
    "1";
    "1",
      "2";
    "2",
      "3";
    "3",
      "4";
    "4";
  }
  "wide";
  {
    "1";
    "1. Quartal",
      "2";
    "2. Quartal",
      "3";
    "3. Quartal",
      "4";
    "4. Quartal";
  }
}
```



```
    }  
  }  
  "dayPeriods":  
  {  
    "format":  
    {  
      "abbreviated":  
      {  
        "midnight";  
        "Mitternacht",  
        "am";  
        "vorm.",  
        "pm";  
        "nachm.",  
        "morning1";  
        "morgens",  
        "morning2";  
        "vormittags",  
        "afternoon1";  
        "mittags",  
        "afternoon2";  
        "nachmittags",  
        "evening1";  
        "abends",  
        "night1";  
        "nachts";  
      }  
      "narrow":  
      {  
        "midnight";  
        "Mitternacht",  
        "am";  
        "vm.",  
        "pm";  
        "nm.",  
        "morning1";  
        "morgens",  
        "morning2";  
        "vormittags",  
        "afternoon1";  
        "mittags",  
        "afternoon2";  
        "nachmittags",  
        "evening1";  
        "abends",  
        "night1";  
        "nachts";  
      }  
      "wide":  
      {  
        "midnight";  
        "Mitternacht",  
        "am";  
        "vorm.",  
        "pm";  
        "nachm.",  
        "morning1";
```

```
        "morgens",
        "morning2";
    "vormittags",
        "afternoon1";
    "mittags",
        "afternoon2";
    "nachmittags",
        "evening1";
    "abends",
        "night1";
    "nachts";
    }
}
"stand-alone";
{
    "abbreviated";
    {
        "midnight";
        "Mitternacht",
            "am";
        "vorm.",
            "pm";
        "nachm.",
            "morning1";
        "Morgen",
            "morning2";
        "Vormittag",
            "afternoon1";
        "Mittag",
            "afternoon2";
        "Nachmittag",
            "evening1";
        "Abend",
            "night1";
        "Nacht";
    }
    "narrow";
    {
        "midnight";
        "Mitternacht",
            "am";
        "vorm.",
            "pm";
        "nachm.",
            "morning1";
        "Morgen",
            "morning2";
        "Vormittag",
            "afternoon1";
        "Mittag",
            "afternoon2";
        "Nachmittag",
            "evening1";
        "Abend",
            "night1";
        "Nacht";
    }
}
```

```
        "wide";
        {
            "midnight";
            "Mitternacht",
                "am";
            "vorm.",
                "pm";
            "nachm.",
                "morning1";
            "Morgen",
                "morning2";
            "Vormittag",
                "afternoon1";
            "Mittag",
                "afternoon2";
            "Nachmittag",
                "evening1";
            "Abend",
                "night1";
            "Nacht";
        }
    }
}
"eras";
{
    "eraNames";
    {
        "0";
        "v. Chr.",
            "0-alt-variant";
        "vor unserer Zeitrechnung",
            "1";
        "n. Chr.",
            "1-alt-variant";
        "unserer Zeitrechnung";
    }
    "eraAbbr";
    {
        "0";
        "v. Chr.",
            "0-alt-variant";
        "v. u. Z.",
            "1";
        "n. Chr.",
            "1-alt-variant";
        "u. Z.";
    }
    "eraNarrow";
    {
        "0";
        "v. Chr.",
            "0-alt-variant";
        "v. u. Z.",
            "1";
        "n. Chr.",
            "1-alt-variant";
        "u. Z.";
    }
}
```

```

    }
  }
  "dateFormats";
  {
    "full";
    "EEEE, d. MMMM y",
    "long";
    "d. MMMM y",
    "medium";
    "dd.MM.y",
    "short";
    "dd.MM.yy";
  }
  "timeFormats";
  {
    "full";
    "HH:mm:ss zzzz",
    "long";
    "HH:mm:ss z",
    "medium";
    "HH:mm:ss",
    "short";
    "HH:mm";
  }
  "dateTimeFormats";
  {
    "full";
    "{1} 'um' {0}",
    "long";
    "{1} 'um' {0}",
    "medium";
    "{1}, {0}",
    "short";
    "{1}, {0}",
    "availableFormats";
    {
      "d";
      "d",
      "E";
      "ccc",
      "Ed";
      "E, d.",
      "Ehm";
      "E h:mm a",
      "EHm";
      "E, HH:mm",
      "Ehms";
      "E, h:mm:ss a",
      "EHms";
      "E, HH:mm:ss",
      "Gy";
      "y G",
      "GyMMM";
      "MMM y G",
      "GyMMMd";
      "d. MMM y G",
      "GyMMMED";
    }
  }

```

```
"E, d. MMM y G",
  "h";
"h 'Uhr' a",
  "H";
"HH 'Uhr' ",
  "hm";
"h:mm a",
  "Hm";
"HH:mm",
  "hms";
"h:mm:ss a",
  "Hms";
"HH:mm:ss",
  "hmsv";
"h:mm:ss a v",
  "Hmsv";
"HH:mm:ss v",
  "hmv";
"h:mm a v",
  "Hmv";
"HH:mm v",
  "M";
"L",
  "Md";
"d.M.",
  "MEd";
"E, d.M.",
  "MMd";
"d.MM.",
  "MMdd";
"dd.MM.",
  "MMM";
"LLL",
  "MMMd";
"d. MMM",
  "MMMEd";
"E, d. MMM",
  "MMMMd";
"d. MMMM",
  "MMMMEd";
"E, d. MMMM",
  "MMMMW";
"'Woche' W 'im' MMM",
  "MMMMW";
"'Woche' W 'im' MMM",
  "ms";
"mm:ss",
  "Y";
"Y",
  "yM";
"M.Y",
  "yMd";
"d.M.Y",
  "yMEd";
"E, d.M.Y",
  "yMM";
"MM.Y",
```

```

        "yMMdd";
        "dd.MM.y",
        "yMMM";
        "MMM y",
        "yMMMd";
        "d. MMM y",
        "yMMMEd";
        "E, d. MMM y",
        "yMMMM";
        "MMMM y",
        "yQQQ";
        "QQQ y",
        "yQQQQ";
        "QQQQ y",
        "yw";
        "'Woche' w 'des' 'Jahres' y",
        "yw";
        "'Woche' w 'des' 'Jahres' y";
    }
    "appendItems";
    {
        "Day";
        "{0} ({2}: {1})",
        "Day-Of-Week";
        "{0} {1}",
        "Era";
        "{1} {0}",
        "Hour";
        "{0} ({2}: {1})",
        "Minute";
        "{0} ({2}: {1})",
        "Month";
        "{0} ({2}: {1})",
        "Quarter";
        "{0} ({2}: {1})",
        "Second";
        "{0} ({2}: {1})",
        "Timezone";
        "{0} {1}",
        "Week";
        "{0} ({2}: {1})",
        "Year";
        "{1} {0}";
    }
    "intervalFormats";
    {
        "intervalFormatFallback";
        "{0} - {1}",
        "d";
        {
            "d";
            "d.-d.";
        }
        "h";
        {
            "a";
            "h 'Uhr' a - h 'Uhr' a",

```

```
        "h";
        "h - h 'Uhr' a";
    }
    "H";
    {
        "H";
        "HH-HH 'Uhr' ";
    }
    "hm";
    {
        "a";
        "h:mm a - h:mm a",
        "h";
        "h:mm-h:mm a",
        "m";
        "h:mm-h:mm a";
    }
    "Hm";
    {
        "H";
        "HH:mm-HH:mm 'Uhr' ",
        "m";
        "HH:mm-HH:mm 'Uhr' ";
    }
    "hmv";
    {
        "a";
        "h:mm a - h:mm a v",
        "h";
        "h:mm-h:mm a v",
        "m";
        "h:mm-h:mm a v";
    }
    "Hmv";
    {
        "H";
        "HH:mm-HH:mm 'Uhr' v",
        "m";
        "HH:mm-HH:mm 'Uhr' v";
    }
    "hv";
    {
        "a";
        "h a - h a v",
        "h";
        "h-h a v";
    }
    "Hv";
    {
        "H";
        "HH-HH 'Uhr' v";
    }
    "M";
    {
        "M";
        "M.-M. ";
    }
}
```

```
"Md";
{
  "d";
  "dd.MM. - dd.MM.";
  "M";
  "dd.MM. - dd.MM.";
}
"MEd";
{
  "d";
  "E, dd.MM. - E, dd.MM.";
  "M";
  "E, dd.MM. - E, dd.MM.";
}
"MMM";
{
  "M";
  "MMM-MMM";
}
"MMMd";
{
  "d";
  "d.-d. MMM",
  "M";
  "d. MMM - d. MMM";
}
"MMMEd";
{
  "d";
  "E, d. - E, d. MMM",
  "M";
  "E, d. MMM - E, d. MMM";
}
"MMMM";
{
  "M";
  "LLLL-LLLL";
}
"Y";
{
  "Y";
  "Y-Y";
}
"yM";
{
  "M";
  "MM.y - MM.y",
  "Y";
  "MM.y - MM.y";
}
"yMd";
{
  "d";
  "dd.MM.y - dd.MM.y",
  "M";
  "dd.MM.y - dd.MM.y",
  "y";
}
```



## INDEX.JSX

Copyright © 2001 -2024 Syncfusion Inc.

```

//Load the L10n, loadCldr from ej2-base
import { loadCldr, L10n } from '@syncfusion/ej2-base';
//import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
//import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
// enable ripple effect
enableRipple(true);
//load the CLDR data files.
// Here we have referred local json files for preview purpose
import * as numberingSystems from './numberingSystems.json';
import * as gregorian from './ca-gregorian.json';
import * as numbers from './numbers.json';
loadCldr(numberingSystems, gregorian, numbers);
L10n.load({
    'de': {
        'timepicker': { placeholder: 'Wählen Sie Zeit' }
    }
});
//import the timepickercomponent
function App() {
    return <TimePickerComponent id="time" locale='de'/>;
}
;
ReactDOM.render(<App />, document.getElementById('timer'));

```

## INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
//Load the L10n, loadCldr from ej2-base
import { loadCldr, L10n } from '@syncfusion/ej2-base';
//import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
//import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
// enable ripple effect
enableRipple(true);
//load the CLDR data files.
// Here we have referred local json files for preview purpose
import * as numberingSystems from './numberingSystems.json';
import * as gregorian from './ca-gregorian.json';
import * as numbers from './numbers.json';
loadCldr(numberingSystems, gregorian, numbers);
L10n.load({
    'de': {
        'timepicker': { placeholder: 'Wählen Sie Zeit' }
    }
});
//import the timepickercomponent
function App() {
    return <TimePickerComponent id="time" locale='de'/>;
}
;
ReactDOM.render(<App />, document.getElementById('timer'));

```

**NUMBERINGSYSTEMS.JSON**

```
{
  "supplemental": {
    "version": {
      "_number": "$Revision: 12732 $",
      "_unicodeVersion": "9.0.0",
      "_cldrVersion": "31"
    },
    "numberingSystems": {
      "adlm": {
        "_digits": "ᲐᲑᲔᲕᲗᲘᲙᲚᲛᲞᲟᲠᲡᲢᲣᲤᲥᲦᲧᲨᲩᲪᲫᲬᲭᲮᲯᲰᲱᲲᲳᲴᲵᲶᲷᲸᲹᲺ᲻᲼ᲽᲾᲿ",
        "_type": "numeric"
      },
      "ahom": {
        "_digits": "ᩌᩍᩎᩏᩐᩑᩒᩓᩔᩕᩖᩗᩘᩙᩚᩛᩜᩝᩞ᩟᩠ᩡᩢᩣᩤᩥᩦᩧᩨᩩᩪᩫᩬᩭᩮᩯᩰᩱᩲᩳᩴ᩵᩶᩷᩸᩹᩺᩻᩼᩽᩾᩿",
        "_type": "numeric"
      },
      "arab": {
        "_digits": "٠١٢٣٤٥٦٧٨٩",
        "_type": "numeric"
      },
      "arabext": {
        "_digits": "٠١٢٣٤٥٦٧٨٩",
        "_type": "numeric"
      },
      "armn": {
        "_rules": "armenian-upper",
        "_type": "algorithmic"
      },
      "armnlow": {
        "_rules": "armenian-lower",
        "_type": "algorithmic"
      },
      "bali": {
        "_digits": "ᬀᬁᬂᬃᬄᬅᬆᬇᬈᬉᬊᬋᬌᬍᬎᬏᬐᬑᬒᬓᬔᬕᬖᬗᬘᬙᬚᬛᬜᬝᬞᬟᬠᬡᬢᬣᬤᬥᬦᬧᬨᬩᬪᬫᬬᬭᬮᬯᬰᬱᬲᬳ᬴ᬵᬶᬷᬸᬹᬺᬻᬼᬽᬾᬿ",
        "_type": "numeric"
      },
      "beng": {
        "_digits": "০১২৩৪৫৬৭৮৯",
        "_type": "numeric"
      },
      "bhks": {
        "_digits": "᱀᱁᱂᱃᱄᱅᱆᱇᱈᱉᱊᱋᱌ᱍᱎᱏ᱐᱑᱒᱓᱔᱕᱖᱗᱘᱙ᱚᱛᱜᱝᱞᱟᱠᱡᱢᱣᱤᱥᱦᱧᱨᱩᱪᱫᱬᱭᱮᱯᱰᱱᱲᱳᱴᱵᱶᱷᱸᱹᱺᱻᱼᱽ᱾᱿",
        "_type": "numeric"
      },
      "brah": {
        "_digits": "᱀᱁᱂᱃᱄᱅᱆᱇᱈᱉᱊᱋᱌ᱍᱎᱏ᱐᱑᱒᱓᱔᱕᱖᱗᱘᱙ᱚᱛᱜᱝᱞᱟᱠᱡᱢᱣᱤᱥᱦᱧᱨᱩᱪᱫᱬᱭᱮᱯᱰᱱᱲᱳᱴᱵᱶᱷᱸᱹᱺᱻᱼᱽ᱾᱿",
        "_type": "numeric"
      },
      "cakm": {
        "_digits": "ᬀᬁᬂᬃᬄᬅᬆᬇᬈᬉᬊᬋᬌᬍᬎᬏᬐᬑᬒᬓᬔᬕᬖᬗᬘᬙᬚᬛᬜᬝᬞᬟᬠᬡᬢᬣᬤᬥᬦᬧᬨᬩᬪᬫᬬᬭᬮᬯᬰᬱᬲᬳ᬴ᬵᬶᬷᬸᬹᬺᬻᬼᬽᬾᬿ",
        "_type": "numeric"
      },
      "cham": {
        "_digits": "ᬀᬁᬂᬃᬄᬅᬆᬇᬈᬉᬊᬋᬌᬍᬎᬏᬐᬑᬒᬓᬔᬕᬖᬗᬘᬙᬚᬛᬜᬝᬞᬟᬠᬡᬢᬣᬤᬥᬦᬧᬨᬩᬪᬫᬬᬭᬮᬯᬰᬱᬲᬳ᬴ᬵᬶᬷᬸᬹᬺᬻᬼᬽᬾᬿ",
        "_type": "numeric"
      }
    }
  }
}
```

```

    "_type": "numeric"
  },
  "cyrl": {
    "_rules": "cyrillic-lower",
    "_type": "algorithmic"
  },
  "deva": {
    "_digits": "୦୧୨୩୪୫୬୭୮",
    "_type": "numeric"
  },
  "ethi": {
    "_rules": "ethiopic",
    "_type": "algorithmic"
  },
  "fullwide": {
    "_digits": "0 1 2 3 4 5 6 7 8 9",
    "_type": "numeric"
  },
  "geor": {
    "_rules": "georgian",
    "_type": "algorithmic"
  },
  "grek": {
    "_rules": "greek-upper",
    "_type": "algorithmic"
  },
  "greklow": {
    "_rules": "greek-lower",
    "_type": "algorithmic"
  },
  "gujr": {
    "_digits": "૦૧૨૩૪૫૬૭૮૯",
    "_type": "numeric"
  },
  "guru": {
    "_digits": "੦੧੨੩੪੫੬੭੮੯",
    "_type": "numeric"
  },
  "hanidays": {
    "_rules": "zh/SpelloutRules/spellout-numbering-days",
    "_type": "algorithmic"
  },
  "hanidec": {
    "_digits": "〇一二三四五六七八九",
    "_type": "numeric"
  },
  "hans": {
    "_rules": "zh/SpelloutRules/spellout-cardinal",
    "_type": "algorithmic"
  },
  "hansfin": {
    "_rules": "zh/SpelloutRules/spellout-cardinal-financial",
    "_type": "algorithmic"
  },
  "hant": {
    "_rules": "zh_Hant/SpelloutRules/spellout-cardinal",

```

```

    "_type": "algorithmic"
  },
  "hantfin": {
    "_rules": "zh_Hant/SpelloutRules/spellout-cardinal-financial",
    "_type": "algorithmic"
  },
  "hebr": {
    "_rules": "hebrew",
    "_type": "algorithmic"
  },
  "hmng": {
    "_digits": "□□□□□□□□□□",
    "_type": "numeric"
  },
  "java": {
    "_digits": "᐀ᐁᐂᐃᐄᐅᐆᐇᐈᐉᐊᐋᐌᐍᐎᐏᐐᐑᐒᐓᐔᐕᐖᐗᐘᐙᐚᐛᐜᐝᐞᐟᐠᐡᐢᐣᐤᐥᐦᐧᐨᐩᐪᐫᐬᐭᐮᐯᐰᐱᐲᐳᐴᐵᐶᐷᐸᐹᐺᐻᐼᐽᐾᐿ",
    "_type": "numeric"
  },
  "jpan": {
    "_rules": "ja/SpelloutRules/spellout-cardinal",
    "_type": "algorithmic"
  },
  "jpanfin": {
    "_rules": "ja/SpelloutRules/spellout-cardinal-financial",
    "_type": "algorithmic"
  },
  "kali": {
    "_digits": "□□□□□□□□□□",
    "_type": "numeric"
  },
  "khmr": {
    "_digits": "០១២៣៤៥៦៧៨៩",
    "_type": "numeric"
  },
  "knda": {
    "_digits": "೦೧೨೩೪೫೬೭೮೯",
    "_type": "numeric"
  },
  "lana": {
    "_digits": "□□□□□□□□□□",
    "_type": "numeric"
  },
  "lanatham": {
    "_digits": "□□□□□□□□□□",
    "_type": "numeric"
  },
  "laoo": {
    "_digits": "໐໑໒໓໔໕໖໗໘໙",
    "_type": "numeric"
  },
  "latn": {
    "_digits": "0123456789",
    "_type": "numeric"
  },
  "lepc": {

```

```
    "_digits": "□□□□□□□□",
    "_type": "numeric"
  },
  "limb": {
    "_digits": "□□□□□□□□",
    "_type": "numeric"
  },
  "mathbold": {
    "_digits": "0123456789",
    "_type": "numeric"
  },
  "mathdbl": {
    "_digits": "0123456789",
    "_type": "numeric"
  },
  "mathmono": {
    "_digits": "0123456789",
    "_type": "numeric"
  },
  "mathsanb": {
    "_digits": "0123456789",
    "_type": "numeric"
  },
  "mathsans": {
    "_digits": "0123456789",
    "_type": "numeric"
  },
  "mlym": {
    "_digits": "൦൧൨൩൪൫൬൭൮൯",
    "_type": "numeric"
  },
  "modi": {
    "_digits": "□□□□□□□□",
    "_type": "numeric"
  },
  "mong": {
    "_digits": "᠐᠑᠒᠓᠐ᠠᠨᠨᠠᠭ",
    "_type": "numeric"
  },
  "mroo": {
    "_digits": "□□□□□□□□",
    "_type": "numeric"
  },
  "mtei": {
    "_digits": "ႤႤႬႬႬႬႬႬႬႬႬ",
    "_type": "numeric"
  },
  "mymr": {
    "_digits": "၀၁၂၃၄၅၆၇၈",
    "_type": "numeric"
  },
  "mymrshan": {
    "_digits": "0123456789",
    "_type": "numeric"
  }
```

```

    },
    "mymrtlng": {
      "_digits": "0୫୪୩୨୧୦୯୮୭୬୫",
      "_type": "numeric"
    },
    "newa": {
      "_digits": "୦୦୦୦୦୦୦୦୦୦",
      "_type": "numeric"
    },
    "nkoo": {
      "_digits": "୨୧୪୩୨୧୦",
      "_type": "numeric"
    },
    "olck": {
      "_digits": "୦୯୧୧୧୧୧୧୧୧",
      "_type": "numeric"
    },
    "orya": {
      "_digits": "୦୧୨୩୪୫୬୭୮୯",
      "_type": "numeric"
    },
    "osma": {
      "_digits": "୦୨୩୪୫୬୭୮୯",
      "_type": "numeric"
    },
    "roman": {
      "_rules": "roman-upper",
      "_type": "algorithmic"
    },
    "romanlow": {
      "_rules": "roman-lower",
      "_type": "algorithmic"
    },
    "saur": {
      "_digits": "୦୦୦୦୦୦୦୦୦୦",
      "_type": "numeric"
    },
    "shrd": {
      "_digits": "୦୦୦୦୦୦୦୦୦୦",
      "_type": "numeric"
    },
    "sind": {
      "_digits": "୦୦୦୦୦୦୦୦୦୦",
      "_type": "numeric"
    },
    "sinh": {
      "_digits": "୦୧୨୩୪୫୬୭୮୯",
      "_type": "numeric"
    },
    "sora": {
      "_digits": "୦୧୨୩୪୫୬୭୮୯",
      "_type": "numeric"
    },
    "sund": {
      "_digits": "୦୦୦୦୦୦୦୦୦୦",

```

```

    "_type": "numeric"
  },
  "takr": {
    "_digits": "□□□□□□□□□□",
    "_type": "numeric"
  },
  "talu": {
    "_digits": "၀၁၂၃၄၅၆၇၈၉",
    "_type": "numeric"
  },
  "taml": {
    "_rules": "tamil",
    "_type": "algorithmic"
  },
  "tamldec": {
    "_digits": "௦௧௨௩௪௫௬௭௮௯",
    "_type": "numeric"
  },
  "telu": {
    "_digits": "౦౧౨౩౪౫౬౭౮౯",
    "_type": "numeric"
  },
  "thai": {
    "_digits": "๐๑๒๓๔๕๖๗๘๙",
    "_type": "numeric"
  },
  "tibt": {
    "_digits": "༠༡༢༣༤༥༦༧༨༩",
    "_type": "numeric"
  },
  "tirh": {
    "_digits": "□□□□□□□□□□",
    "_type": "numeric"
  },
  "vaih": {
    "_digits": "᐀ᐁᐂᐃᐄᐅᐆᐇᐈᐉ",
    "_type": "numeric"
  },
  "wara": {
    "_digits": "□□□□□□□□□□",
    "_type": "numeric"
  }
}
}
}

```

### **NUMBERINGSYSTEMS.JSX**

```

{
  "supplemental";
  {
    "version";
    {
      "number";
      "$Revision: 12732 $",
    }
  }
}

```



```

        "_unicodeVersion";
        "9.0.0",
        "_cldrVersion";
        "31";
    }
    "numberingSystems";
    {
        "adlm";
        {
            "_digits";
            "୧୨୩୪୫୬୭୮୯୦",
            "_type";
            "numeric";
        }
        "ahom";
        {
            "_digits";
            "᱀᱁᱂᱃᱄᱅᱆᱇᱈᱉᱊᱋᱌ᱍᱎᱏ᱐᱑᱒᱓᱔᱕᱖᱗᱘᱙ᱚᱛᱜᱝᱞᱟᱠᱡᱢᱣᱤᱥᱦᱧᱨᱩᱪᱫᱬᱭᱮᱱᱲᱳᱴᱵᱶᱷᱸᱹ",
            "_type";
            "numeric";
        }
        "arab";
        {
            "_digits";
            "٠١٢٣٤٥٦٧٨٩",
            "_type";
            "numeric";
        }
        "arabext";
        {
            "_digits";
            "٠١٢٣٤٥٦٧٨٩",
            "_type";
            "numeric";
        }
        "armn";
        {
            "_rules";
            "armenian-upper",
            "_type";
            "algorithmic";
        }
        "armnlow";
        {
            "_rules";
            "armenian-lower",
            "_type";
            "algorithmic";
        }
        "bali";
        {
            "_digits";
            "᭐᭑᭒᭓᭔᭕᭖᭗᭘᭙᭚᭛᭜᭝᭞᭟᭠᭡᭢᭣᭤᭥᭦᭧᭨᭩᭪᭬᭫᭭᭮᭯᭰᭱᭲᭳᭴᭵᭶᭷᭸᭹",
            "_type";
            "numeric";
        }
    }

```

```
"beng";
{
  "_digits";
  "০১২৩৪৫৬৭৮৯",
  "_type";
  "numeric";
}
"bhks";
{
  "_digits";
  "□□□□□□□□□□",
  "_type";
  "numeric";
}
"brah";
{
  "_digits";
  "·٠١٢٣٤٥٦٧٨٩",
  "_type";
  "numeric";
}
"cakm";
{
  "_digits";
  "ႤႤႬႬႭႭႬႬႬႬႬႬႬႬ",
  "_type";
  "numeric";
}
"cham";
{
  "_digits";
  "□□□□□□□□□□",
  "_type";
  "numeric";
}
"cyrl";
{
  "_rules";
  "cyrillic-lower",
  "_type";
  "algorithmic";
}
"deva";
{
  "_digits";
  "०१२३४५६७८९",
  "_type";
  "numeric";
}
"ethi";
{
  "_rules";
  "ethiopic",
  "_type";
  "algorithmic";
}
```

```
"fullwide";
{
  "_digits";
  "0 1 2 3 4 5 6 7 8 9",
  "_type";
  "numeric";
}
"geor";
{
  "_rules";
  "georgian",
  "_type";
  "algorithmic";
}
"grek";
{
  "_rules";
  "greek-upper",
  "_type";
  "algorithmic";
}
"greklow";
{
  "_rules";
  "greek-lower",
  "_type";
  "algorithmic";
}
"gujr";
{
  "_digits";
  "૦ ૧ ૨ ૩ ૪ ૫ ૬ ૭ ૮ ૯",
  "_type";
  "numeric";
}
"guru";
{
  "_digits";
  "੦ ੧ ੨ ੩ ੪ ੫ ੬ ੭ ੮ ੯",
  "_type";
  "numeric";
}
"hanidays";
{
  "_rules";
  "zh/SpelloutRules/spellout-numbering-days",
  "_type";
  "algorithmic";
}
"hanidec";
{
  "_digits";
  "〇 一 二 三 四 五 六 七 八 九",
  "_type";
  "numeric";
}
```

```
"hans";
{
    "_rules";
    "zh/SpelloutRules/spellout-cardinal",
        "_type";
    "algorithmic";
}
"hansfin";
{
    "_rules";
    "zh/SpelloutRules/spellout-cardinal-financial",
        "_type";
    "algorithmic";
}
"hant";
{
    "_rules";
    "zh_Hant/SpelloutRules/spellout-cardinal",
        "_type";
    "algorithmic";
}
"hantfin";
{
    "_rules";
    "zh_Hant/SpelloutRules/spellout-cardinal-financial",
        "_type";
    "algorithmic";
}
"hebr";
{
    "_rules";
    "hebrew",
        "_type";
    "algorithmic";
}
"hmng";
{
    "_digits";
    "□□□□□□□□□□",
        "_type";
    "numeric";
}
"java";
{
    "_digits";
    "0000000000000000",
        "_type";
    "numeric";
}
"jpan";
{
    "_rules";
    "ja/SpelloutRules/spellout-cardinal",
        "_type";
    "algorithmic";
}
```

```
"jpanfin";
{
  "_rules";
  "ja/SpelloutRules/spellout-cardinal-financial",
  "_type";
  "algorithmic";
}
"kali";
{
  "_digits";
  "□□□□□□□□□□",
  "_type";
  "numeric";
}
"khmr";
{
  "_digits";
  "០១២៣៤៥៦៧៨៩",
  "_type";
  "numeric";
}
"knda";
{
  "_digits";
  "೦೧೨೩೪೫೬೭೮೯",
  "_type";
  "numeric";
}
"lana";
{
  "_digits";
  "□□□□□□□□□□",
  "_type";
  "numeric";
}
"lanatham";
{
  "_digits";
  "□□□□□□□□□□",
  "_type";
  "numeric";
}
"laoo";
{
  "_digits";
  "໐໑໒໓໔໕໖໗໘໑",
  "_type";
  "numeric";
}
"latn";
{
  "_digits";
  "0123456789",
  "_type";
  "numeric";
}
```

```

"lepc";
{
  "_digits";
  "□□□□□□□□",
  "_type";
  "numeric";
}
"limb";
{
  "_digits";
  "□□□□□□□□",
  "_type";
  "numeric";
}
"mathbold";
{
  "_digits";
  "0123456789",
  "_type";
  "numeric";
}
"mathdbl";
{
  "_digits";
  "0123456789",
  "_type";
  "numeric";
}
"mathmono";
{
  "_digits";
  "0123456789",
  "_type";
  "numeric";
}
"mathsanb";
{
  "_digits";
  "0123456789",
  "_type";
  "numeric";
}
"mathsans";
{
  "_digits";
  "0123456789",
  "_type";
  "numeric";
}
"mlym";
{
  "_digits";
  "ഘറ൩൪൫൬൭൮൯",
  "_type";
  "numeric";
}

```

```

"modi";
{
    "_digits";
    "□□□□□□□□□□",
    "_type";
    "numeric";
}
"mong";
{
    "_digits";
    "ᠮᠣᠩᠭ᠎ᠠ",
    "_type";
    "numeric";
}
"mroo";
{
    "_digits";
    "□□□□□□□□□□",
    "_type";
    "numeric";
}
"mtei";
{
    "_digits";
    "ᠮᠲᠡᠢ",
    "_type";
    "numeric";
}
"mymr";
{
    "_digits";
    "ᠮᠣᠮᠢᠷ",
    "_type";
    "numeric";
}
"mymrshan";
{
    "_digits";
    "ᠮᠣᠮᠢᠷᠰᠠᠨ",
    "_type";
    "numeric";
}
"mymrtlng";
{
    "_digits";
    "ᠮᠣᠮᠢᠷᠲᠠᠯᠠᠩ",
    "_type";
    "numeric";
}
"newa";
{
    "_digits";
    "□□□□□□□□□□",
    "_type";

```

```
"numeric";
}
"nkoo";
{
    "_digits";
    "ᠨᠠᠭᠤᠯᠠᠩᠭᠡᠢᠰᠦ᠋ᠨᠣ",
        "_type";
    "numeric";
}
"olck";
{
    "_digits";
    "ᠣᠯᠴᠻᠬᠠᠳᠤᠰᠤᠷ",
        "_type";
    "numeric";
}
"orya";
{
    "_digits";
    "ᠣᠷᠶᠠᠨᠵᠢᠰᠤᠷ",
        "_type";
    "numeric";
}
"osma";
{
    "_digits";
    "ᠣᠰᠤᠮᠠᠳᠤᠰᠤᠷ",
        "_type";
    "numeric";
}
"roman";
{
    "_rules";
    "roman-upper",
        "_type";
    "algorithmic";
}
"romanlow";
{
    "_rules";
    "roman-lower",
        "_type";
    "algorithmic";
}
"saur";
{
    "_digits";
    "ᠰᠠᠸᠠᠷᠠᠨᠵᠢᠰᠤᠷ",
        "_type";
    "numeric";
}
"shrd";
{
    "_digits";
    "ᠱᠠᠷᠳᠤᠰᠤᠷ",
        "type";
```



```

        "numeric";
    }
    "sind";
    {
        "_digits";
        "□□□□□□□□",
        "_type";
        "numeric";
    }
    "sinh";
    {
        "_digits";
        "௧௧௧௧௧௧௧௧௧௧௧௧",
        "_type";
        "numeric";
    }
    "sora";
    {
        "_digits";
        "0௧௨௩௪௫௬௭௮௯",
        "_type";
        "numeric";
    }
    "sund";
    {
        "_digits";
        "□□□□□□□□",
        "_type";
        "numeric";
    }
    "takr";
    {
        "_digits";
        "□□□□□□□□",
        "_type";
        "numeric";
    }
    "talv";
    {
        "_digits";
        "௦௧௨௩௪௫௬௭௮௯",
        "_type";
        "numeric";
    }
    "taml";
    {
        "_rules";
        "tamil",
        "_type";
        "algorithmic";
    }
    "tamldec";
    {
        "_digits";
        "0௧௨௩௪௫௬௭௮௯",
        "_type";
    }

```

```

        "numeric";
    }
    "telu";
    {
        "_digits";
        "୦୧୨୩୪୫୬୭୮",
        "_type";
        "numeric";
    }
    "thai";
    {
        "_digits";
        "๐๑๒๓๔๕๖๗๘๙",
        "_type";
        "numeric";
    }
    "tibtb";
    {
        "_digits";
        "༠༡༢༣༤༥༦༧༨༩",
        "_type";
        "numeric";
    }
    "tirh";
    {
        "_digits";
        "ଁଁଁଁଁଁଁଁଁଁଁଁ",
        "_type";
        "numeric";
    }
    "vaii";
    {
        "_digits";
        "᱀᱁᱂᱃᱄᱅᱆᱇᱈᱉᱊᱋᱌ᱍᱎᱏ᱐᱑᱒᱓᱔᱕᱖᱗᱘᱙ᱚᱛᱜᱝᱞᱟᱠᱡᱢᱣᱤᱥᱦᱧᱨᱩᱪᱫᱬᱭᱮᱹᱺᱻᱼᱽ᱾᱿",
        "_type";
        "numeric";
    }
    "wara";
    {
        "_digits";
        "ᳵᳶ᳷᳸᳹ᳺ᳻᳼᳽᳾᳿",
        "_type";
        "numeric";
    }
}

```

## NUMBERS.JSON

```
{
  "main": {
    "de": {
      "identity": {
        "version": {
```

```

    "_number": "$Revision: 13259 $",
    "_cldrVersion": "31"
  },
  "language": "de"
},
"numbers": {
  "defaultNumberingSystem": "latn",
  "otherNumberingSystems": {
    "native": "latn"
  },
  "minimumGroupingDigits": "1",
  "symbols-numberSystem-latn": {
    "decimal": ",",
    "group": ".",
    "list": ";",
    "percentSign": "%",
    "plusSign": "+",
    "minusSign": "-",
    "exponential": "E",
    "superscriptingExponent": "°",
    "perMille": "‰",
    "infinity": "∞",
    "nan": "NaN",
    "timeSeparator": ":"
  },
  "decimalFormats-numberSystem-latn": {
    "standard": "#,##0.###",
    "long": {
      "decimalFormat": {
        "1000-count-one": "0 Tausend",
        "1000-count-other": "0 Tausend",
        "10000-count-one": "00 Tausend",
        "10000-count-other": "00 Tausend",
        "100000-count-one": "000 Tausend",
        "100000-count-other": "000 Tausend",
        "1000000-count-one": "0 Million",
        "1000000-count-other": "0 Millionen",
        "10000000-count-one": "00 Millionen",
        "10000000-count-other": "00 Millionen",
        "100000000-count-one": "000 Millionen",
        "100000000-count-other": "000 Millionen",
        "1000000000-count-one": "0 Milliarden",
        "1000000000-count-other": "0 Milliarden",
        "10000000000-count-one": "00 Milliarden",
        "10000000000-count-other": "00 Milliarden",
        "100000000000-count-one": "000 Milliarden",
        "100000000000-count-other": "000 Milliarden",
        "1000000000000-count-one": "0 Billion",
        "1000000000000-count-other": "0 Billionen",
        "10000000000000-count-one": "00 Billionen",
        "10000000000000-count-other": "00 Billionen",
        "100000000000000-count-one": "000 Billionen",
        "100000000000000-count-other": "000 Billionen"
      }
    },
    "short": {
      "decimalFormat": {

```

```

        "1000-count-one": "0",
        "1000-count-other": "0",
        "10000-count-one": "0",
        "10000-count-other": "0",
        "100000-count-one": "0",
        "100000-count-other": "0",
        "1000000-count-one": "0 Mio'.'",
        "1000000-count-other": "0 Mio'.'",
        "10000000-count-one": "00 Mio'.'",
        "10000000-count-other": "00 Mio'.'",
        "100000000-count-one": "000 Mio'.'",
        "100000000-count-other": "000 Mio'.'",
        "1000000000-count-one": "0 Mrd'.'",
        "1000000000-count-other": "0 Mrd'.'",
        "10000000000-count-one": "00 Mrd'.'",
        "10000000000-count-other": "00 Mrd'.'",
        "100000000000-count-one": "000 Mrd'.'",
        "100000000000-count-other": "000 Mrd'.'",
        "1000000000000-count-one": "0 Bio'.'",
        "1000000000000-count-other": "0 Bio'.'",
        "10000000000000-count-one": "00 Bio'.'",
        "10000000000000-count-other": "00 Bio'.'",
        "100000000000000-count-one": "000 Bio'.'",
        "100000000000000-count-other": "000 Bio'.'"
    }
  },
  "scientificFormats-numberSystem-latn": {
    "standard": "#E0"
  },
  "percentFormats-numberSystem-latn": {
    "standard": "#,##0 %"
  },
  "currencyFormats-numberSystem-latn": {
    "currencySpacing": {
      "beforeCurrency": {
        "currencyMatch": "[:^S:]",
        "surroundingMatch": "[:digit:]",
        "insertBetween": " "
      },
      "afterCurrency": {
        "currencyMatch": "[:^S:]",
        "surroundingMatch": "[:digit:]",
        "insertBetween": " "
      }
    },
    "standard": "#,##0.00 ¤",
    "accounting": "#,##0.00 ¤",
    "short": {
      "standard": {
        "1000-count-one": "0 Tsd'.' ¤",
        "1000-count-other": "0 Tsd'.' ¤",
        "10000-count-one": "00 Tsd'.' ¤",
        "10000-count-other": "00 Tsd'.' ¤",
        "100000-count-one": "000 Tsd'.' ¤",
        "100000-count-other": "000 Tsd'.' ¤",
        "1000000-count-one": "0 Mio'.' ¤",

```

```
"1000000-count-other": "0 Mio'.' ¤",
"10000000-count-one": "00 Mio'.' ¤",
"10000000-count-other": "00 Mio'.' ¤",
"100000000-count-one": "000 Mio'.' ¤",
"100000000-count-other": "000 Mio'.' ¤",
"1000000000-count-one": "0 Mrd'.' ¤",
"1000000000-count-other": "0 Mrd'.' ¤",
"10000000000-count-one": "00 Mrd'.' ¤",
"10000000000-count-other": "00 Mrd'.' ¤",
"100000000000-count-one": "000 Mrd'.' ¤",
"100000000000-count-other": "000 Mrd'.' ¤",
"1000000000000-count-one": "0 Bio'.' ¤",
"1000000000000-count-other": "0 Bio'.' ¤",
"10000000000000-count-one": "00 Bio'.' ¤",
"10000000000000-count-other": "00 Bio'.' ¤",
"100000000000000-count-one": "000 Bio'.' ¤",
"100000000000000-count-other": "000 Bio'.' ¤"
    }
},
"unitPattern-count-one": "{0} {1}",
"unitPattern-count-other": "{0} {1}"
},
"miscPatterns-numberSystem-latn": {
    "atLeast": "{0}+",
    "range": "{0}-{1}"
},
"minimalPairs": {
    "pluralMinimalPairs": "{0} Tag",
    "pluralMinimalPairs": "{0} Tage",
    "other": "{0}. Abzweigung nach rechts nehmen"
}
}
}
}
```

## NUMBERS.JSX

```
{
  "main";
  {
    "de";
    {
      "identity";
      {
        "version";
        {
          "_number";
          "$Revision: 13259 $",
          "_cldrVersion";
          "31";
        }
        "language";
        "de";
      }
    }
  }
  "numbers";
}
```

```

    {
      "defaultNumberingSystem";
      "latn",
      "otherNumberingSystems";
      {
        "native";
        "latn";
      }
      "minimumGroupingDigits";
      "1",
      "symbols-numberSystem-latn";
      {
        "decimal";
        ",",
        "group";
        ".",
        "list";
        ";",
        "percentSign";
        "%",
        "plusSign";
        "+",
        "minusSign";
        "-",
        "exponential";
        "E",
        "superscriptingExponent";
        ". ",
        "perMille";
        "‰",
        "infinity";
        "∞",
        "nan";
        "NaN",
        "timeSeparator";
        ":";
      }
      "decimalFormats-numberSystem-latn";
      {
        "standard";
        "#,##0.###",
        "long";
        {
          "decimalFormat";
          {
            "1000-count-one";
            "0 Tausend",
            "1000-count-other";
            "0 Tausend",
            "10000-count-one";
            "00 Tausend",
            "10000-count-other";
            "00 Tausend",
            "100000-count-one";
            "000 Tausend",
            "100000-count-other";
            "000 Tausend",

```

```

        "1000000-count-one";
    "0 Million",
        "1000000-count-other";
    "0 Millionen",
        "10000000-count-one";
    "00 Millionen",
        "10000000-count-other";
    "00 Millionen",
        "100000000-count-one";
    "000 Millionen",
        "100000000-count-other";
    "000 Millionen",
        "1000000000-count-one";
    "0 Milliarde",
        "1000000000-count-other";
    "0 Milliarden",
        "10000000000-count-one";
    "00 Milliarden",
        "10000000000-count-other";
    "00 Milliarden",
        "100000000000-count-one";
    "000 Milliarden",
        "100000000000-count-other";
    "000 Milliarden",
        "1000000000000-count-one";
    "0 Billion",
        "1000000000000-count-other";
    "0 Billionen",
        "10000000000000-count-one";
    "00 Billionen",
        "10000000000000-count-other";
    "00 Billionen",
        "100000000000000-count-one";
    "000 Billionen",
        "100000000000000-count-other";
    "000 Billionen";
    }
}
"short";
{
    "decimalFormat";
    {
        "1000-count-one";
        "0",
            "1000-count-other";
        "0",
            "10000-count-one";
        "0",
            "10000-count-other";
        "0",
            "100000-count-one";
        "0",
            "100000-count-other";
        "0",
            "1000000-count-one";
        "0 Mio'.'",
            "1000000-count-other";
    }
}

```

```

        "0 Mio'.'",
        "10000000-count-one";
        "00 Mio'.'",
        "10000000-count-other";
        "00 Mio'.'",
        "100000000-count-one";
        "000 Mio'.'",
        "100000000-count-other";
        "000 Mio'.'",
        "1000000000-count-one";
        "0 Mrd'.'",
        "1000000000-count-other";
        "0 Mrd'.'",
        "10000000000-count-one";
        "00 Mrd'.'",
        "10000000000-count-other";
        "00 Mrd'.'",
        "100000000000-count-one";
        "000 Mrd'.'",
        "100000000000-count-other";
        "000 Mrd'.'",
        "1000000000000-count-one";
        "0 Bio'.'",
        "1000000000000-count-other";
        "0 Bio'.'",
        "10000000000000-count-one";
        "00 Bio'.'",
        "10000000000000-count-other";
        "00 Bio'.'",
        "100000000000000-count-one";
        "000 Bio'.'",
        "1000000000000000-count-other";
        "000 Bio'.'";
    }
}
}
"scientificFormats-numberSystem-latn";
{
    "standard";
    "#E0";
}
"percentFormats-numberSystem-latn";
{
    "standard";
    "#,##0 %";
}
"currencyFormats-numberSystem-latn";
{
    "currencySpacing";
    {
        "beforeCurrency";
        {
            "currencyMatch";
            "[^S:]",
            "surroundingMatch";
            "[:digit:]",
            "insertBetween";
        }
    }
}

```



```

        " ";
    }
    "afterCurrency";
    {
        "currencyMatch";
        "[:^S:]",
        "surroundingMatch";
        "[:digit:]",
        "insertBetween";
        " ";
    }
}
"standard";
"#,##0.00 ¤",
    "accounting";
"#,##0.00 ¤",
    "short";
{
    "standard";
    {
        "1000-count-one";
        "0 Tsd'.' ¤",
        "1000-count-other";
        "0 Tsd'.' ¤",
        "10000-count-one";
        "00 Tsd'.' ¤",
        "10000-count-other";
        "00 Tsd'.' ¤",
        "100000-count-one";
        "000 Tsd'.' ¤",
        "100000-count-other";
        "000 Tsd'.' ¤",
        "1000000-count-one";
        "0 Mio'.' ¤",
        "1000000-count-other";
        "0 Mio'.' ¤",
        "10000000-count-one";
        "00 Mio'.' ¤",
        "10000000-count-other";
        "00 Mio'.' ¤",
        "100000000-count-one";
        "000 Mio'.' ¤",
        "100000000-count-other";
        "000 Mio'.' ¤",
        "1000000000-count-one";
        "0 Mrd'.' ¤",
        "1000000000-count-other";
        "0 Mrd'.' ¤",
        "10000000000-count-one";
        "00 Mrd'.' ¤",
        "10000000000-count-other";
        "00 Mrd'.' ¤",
        "100000000000-count-one";
        "000 Mrd'.' ¤",
        "100000000000-count-other";
        "000 Mrd'.' ¤",
        "1000000000000-count-one";
    }
}

```

```

        "0 Bio'.' ' α",
        "1000000000000-count-other";
    "0 Bio'.' ' α",
        "1000000000000-count-one";
    "00 Bio'.' ' α",
        "1000000000000-count-other";
    "00 Bio'.' ' α",
        "1000000000000-count-one";
    "000 Bio'.' ' α",
        "1000000000000-count-other";
    "000 Bio'.' ' α";
}

}
"unitPattern-count-one";
"{0} {1}",
    "unitPattern-count-other";
"{0} {1}";
}
"miscPatterns-numberSystem-latn";
{
    "atLeast";
    "{0}+",
        "range";
    "{0}-{1}";
}
"minimalPairs";
{
    "pluralMinimalPairs";
    "{0} Tag",
        "pluralMinimalPairs";
    "{0} Tage",
        "other";
    "{0}. Abzweigung nach rechts nehmen";
}
}
}
}

```

## Right-To-Left

The TimePicker supports RTL (right-to-left) functionality for languages like Arabic and Hebrew to displays the text in the right-to-left direction. Use [enableRtl](#) property to set the RTL direction.

The code example demonstrates the TimePicker component in Arabic culture, also explains how to set the localized text to the placeholder using [L10n.load](#) method.

[Class-component]

## APP.COMPONENT.JSX

\_\_\_\_\_

## APP.COMPONENT.TS

**APP.MODULE.JSX**

```

var __decorate = (this && this.__decorate) || function (decorators, target,
key, desc) {
    var c = arguments.length, r = c < 3 ? target : desc === null ? desc =
Object.getOwnPropertyDescriptor(target, key) : desc, d;
    if (typeof Reflect === "object" && typeof Reflect.decorate ===
"function") r = Reflect.decorate(decorators, target, key, desc);
    else for (var i = decorators.length - 1; i >= 0; i--) if (d =
decorators[i]) r = (c < 3 ? d(r) : c > 3 ? d(target, key, r) : d(target,
key)) || r;
    return c > 3 && r && Object.defineProperty(target, key, r), r;
};
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { TimePickerModule } from '@syncfusion/ej2-ng-calendars';
import { AppComponent } from './app.component';
/**
 * Module
 */
let AppModule = class AppModule {
};
AppModule = __decorate([
    NgModule({
        imports: [
            BrowserModule,
            TimePickerModule
        ],
        declarations: [AppComponent],
        bootstrap: [AppComponent]
    })
], AppModule);
export { AppModule };

```

**APP.MODULE.TS**

```

import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { TimePickerModule } from '@syncfusion/ej2-ng-calendars';
import { AppComponent } from './app.component';
/**
 * Module
 */
@NgModule({
    imports: [
        BrowserModule,
        TimePickerModule
    ],
    declarations: [AppComponent],
    bootstrap: [AppComponent]
})
export class AppModule { }

```

**CA-GREGORIAN.JSON**

```

{
  "main": {
    "ar": {
      "identity": {
        "version": {
          "_number": "$Revision: 12879 $",
          "_cldrVersion": "30.0.3"
        },
        "language": "ar"
      },
      "dates": {
        "calendars": {
          "gregorian": {
            "months": {
              "format": {
                "abbreviated": {
                  "1": "يناير",
                  "2": "فبراير",
                  "3": "مارس",
                  "4": "أبريل",
                  "5": "مايو",
                  "6": "يونيو",
                  "7": "يوليو",
                  "8": "أغسطس",
                  "9": "سبتمبر",
                  "10": "أكتوبر",
                  "11": "نوفمبر",
                  "12": "ديسمبر"
                },
                "narrow": {
                  "1": "ي",
                  "2": "ف",
                  "3": "م",
                  "4": "أ",
                  "5": "و",
                  "6": "ن",
                  "7": "ل",
                  "8": "غ",
                  "9": "س",
                  "10": "ك",
                  "11": "ب",
                  "12": "د"
                },
                "wide": {
                  "1": "يناير",
                  "2": "فبراير",
                  "3": "مارس",
                  "4": "أبريل",
                  "5": "مايو",
                  "6": "يونيو",
                  "7": "يوليو",
                  "8": "أغسطس",
                  "9": "سبتمبر",
                  "10": "أكتوبر",
                  "11": "نوفمبر",

```

```

        "12": "ديسمبر"
    },
    },
    "stand-alone": {
        "abbreviated": {
            "1": "يناير",
            "2": "فبراير",
            "3": "مارس",
            "4": "أبريل",
            "5": "مايو",
            "6": "يونيو",
            "7": "يوليو",
            "8": "أغسطس",
            "9": "سبتمبر",
            "10": "أكتوبر",
            "11": "نوفمبر",
            "12": "ديسمبر"
        },
        "narrow": {
            "1": "ي",
            "2": "ف",
            "3": "م",
            "4": "أ",
            "5": "و",
            "6": "ن",
            "7": "ل",
            "8": "غ",
            "9": "س",
            "10": "ك",
            "11": "ب",
            "12": "د"
        },
        "wide": {
            "1": "يناير",
            "2": "فبراير",
            "3": "مارس",
            "4": "أبريل",
            "5": "مايو",
            "6": "يونيو",
            "7": "يوليو",
            "8": "أغسطس",
            "9": "سبتمبر",
            "10": "أكتوبر",
            "11": "نوفمبر",
            "12": "ديسمبر"
        }
    },
    },
    "days": {
        "format": {
            "abbreviated": {
                "sun": "الأحد",
                "mon": "الاثنين",
                "tue": "الثلاثاء",
                "wed": "الأربعاء",
                "thu": "الخميس",
                "fri": "الجمعة",
            }
        }
    }
}

```

```

        "sat": "السبت"
    },
    "narrow": {
        "sun": "ح",
        "mon": "ن",
        "tue": "ث",
        "wed": "ر",
        "thu": "خ",
        "fri": "ج",
        "sat": "س"
    },
    "short": {
        "sun": "الأحد",
        "mon": "الاثنين",
        "tue": "الثلاثاء",
        "wed": "الأربعاء",
        "thu": "الخميس",
        "fri": "الجمعة",
        "sat": "السبت"
    },
    "wide": {
        "sun": "الأحد",
        "mon": "الاثنين",
        "tue": "الثلاثاء",
        "wed": "الأربعاء",
        "thu": "الخميس",
        "fri": "الجمعة",
        "sat": "السبت"
    }
},
"stand-alone": {
    "abbreviated": {
        "sun": "الأحد",
        "mon": "الاثنين",
        "tue": "الثلاثاء",
        "wed": "الأربعاء",
        "thu": "الخميس",
        "fri": "الجمعة",
        "sat": "السبت"
    },
    "narrow": {
        "sun": "ح",
        "mon": "ن",
        "tue": "ث",
        "wed": "ر",
        "thu": "خ",
        "fri": "ج",
        "sat": "س"
    },
    "short": {
        "sun": "الأحد",
        "mon": "الاثنين",
        "tue": "الثلاثاء",
        "wed": "الأربعاء",
        "thu": "الخميس",
        "fri": "الجمعة",
        "sat": "السبت"
    }
}

```

```

    },
    "wide": {
      "sun": "الأحد",
      "mon": "الاثنين",
      "tue": "الثلاثاء",
      "wed": "الأربعاء",
      "thu": "الخميس",
      "fri": "الجمعة",
      "sat": "السبت"
    }
  },
  "quarters": {
    "format": {
      "abbreviated": {
        "1": "الربع الأول",
        "2": "الربع الثاني",
        "3": "الربع الثالث",
        "4": "الربع الرابع"
      },
      "narrow": {
        "1": "١",
        "2": "٢",
        "3": "٣",
        "4": "٤"
      },
      "wide": {
        "1": "الربع الأول",
        "2": "الربع الثاني",
        "3": "الربع الثالث",
        "4": "الربع الرابع"
      }
    },
    "stand-alone": {
      "abbreviated": {
        "1": "الربع الأول",
        "2": "الربع الثاني",
        "3": "الربع الثالث",
        "4": "الربع الرابع"
      },
      "narrow": {
        "1": "١",
        "2": "٢",
        "3": "٣",
        "4": "٤"
      },
      "wide": {
        "1": "الربع الأول",
        "2": "الربع الثاني",
        "3": "الربع الثالث",
        "4": "الربع الرابع"
      }
    }
  },
  "dayPeriods": {
    "format": {
      "abbreviated": {

```

```

    "am": "ص",
    "pm": "م",
    "morning1": "فجرا",
    "morning2": "ص",
    "afternoon1": "ظهرًا",
    "afternoon2": "بعد الظهر",
    "evening1": "مساءً",
    "night1": "منتصف الليل",
    "night2": "ل",
  },
  "narrow": {
    "am": "ص",
    "pm": "م",
    "morning1": "فجراً",
    "morning2": "صباحًا",
    "afternoon1": "ظهرًا",
    "afternoon2": "بعد الظهر",
    "evening1": "مساءً",
    "night1": "منتصف الليل",
    "night2": "ليلاً"
  },
  "wide": {
    "am": "ص",
    "pm": "م",
    "morning1": "فجراً",
    "morning2": "صباحًا",
    "afternoon1": "ظهرًا",
    "afternoon2": "بعد الظهر",
    "evening1": "مساءً",
    "night1": "منتصف الليل",
    "night2": "ليلاً"
  }
},
"stand-alone": {
  "abbreviated": {
    "am": "ص",
    "pm": "م",
    "morning1": "فجرا",
    "morning2": "ص",
    "afternoon1": "ظهرًا",
    "afternoon2": "بعد الظهر",
    "evening1": "مساءً",
    "night1": "منتصف الليل",
    "night2": "ليلاً"
  },
  "narrow": {
    "am": "ص",
    "pm": "م",
    "morning1": "فجراً",
    "morning2": "صباحًا",
    "afternoon1": "ظهرًا",
    "afternoon2": "بعد الظهر",
    "evening1": "مساءً",
    "night1": "منتصف الليل",
    "night2": "ليلاً"
  },
  "wide": {

```



```

        "am": "صباحًا",
        "pm": "مساءً",
        "morning1": "فجرًا",
        "morning2": "صباحًا",
        "afternoon1": "ظهرًا",
        "afternoon2": "بعد الظهر",
        "evening1": "مساءً",
        "night1": "منتصف الليل",
        "night2": "ليلاً"
    }
}
},
"eras": {
    "eraNames": {
        "0": "قبل الميلاد",
        "0-alt-variant": "BCE",
        "1": "ميلادي",
        "1-alt-variant": "بعد الميلاد"
    },
    "eraAbbr": {
        "0": "ق.م",
        "0-alt-variant": "BCE",
        "1": "م",
        "1-alt-variant": "ب.م"
    },
    "eraNarrow": {
        "0": "ق.م",
        "0-alt-variant": "BCE",
        "1": "م",
        "1-alt-variant": "ب.م"
    }
},
"dateFormats": {
    "full": "EEEE، d MMMM، y",
    "long": "d MMMM، y",
    "medium": "dd/MM/y",
    "short": "d/M/y"
},
"timeFormats": {
    "full": "h:mm:ss a zzzz",
    "long": "h:mm:ss a z",
    "medium": "h:mm:ss a",
    "short": "h:mm a"
},
"dateTimeFormats": {
    "full": "{1} {0}",
    "long": "{1} {0}",
    "medium": "{1} {0}",
    "short": "{1} {0}",
    "availableFormats": {
        "d": "d",
        "E": "ccc",
        "Ed": "E، d",
        "Ehm": "E h:mm a",
        "EHm": "E HH:mm",
        "Ehms": "E h:mm:ss a",
        "EHms": "E HH:mm:ss",
    }
}

```

```

"Gy": "y G",
"GyMMM": "MMM y G",
"GyMMMd": "d MMM, y G",
"GyMMMED": "E, d MMM, y G",
"h": "h a",
"H": "HH",
"hm": "h:mm a",
"Hm": "HH:mm",
"hms": "h:mm:ss a",
"Hms": "HH:mm:ss",
"hmsv": "h:mm:ss a v",
"Hmsv": "HH:mm:ss v",
"hm": "h:mm a",
"Hm": "HH:mm v",
"M": "L",
Md": "d/M",
MEd": "E, d/M",
MMdd": "dd/MM",
MMM": "LLL",
MMMd": "d MMM",
MMMED": "E, d MMM",
MMMMd": "d MMMM",
MMMMEd": "E, d MMMM",
MMMMMW": "الأسبوع W من MMM",
MMMMMW": "الأسبوع W من MMM",
MMMMMW": "الأسبوع W من MMM",
MMMMMW": "الأسبوع W من MMM",
MMMMMW": "الأسبوع W من MMM",
MMMMMW": "الأسبوع W من MMM",
ms": "mm:ss",
y": "y",
yM": "M/y",
yMd": "d/M/y",
yMEd": "E, d/M/y",
yMM": "MM/y",
yMMM": "MMM y",
yMMMd": "d MMM, y",
yMMMED": "E, d MMM, y",
yMMMM": "MMMM y",
yQQQ": "QQQ y",
yQQQQ": "QQQQ y",
yw": "الأسبوع w من سنة y",
yw": "الأسبوع w من سنة y",
yw": "الأسبوع w من سنة y",
yw": "الأسبوع w من سنة y",
yw": "الأسبوع w من سنة y",
yw": "الأسبوع w من سنة y",
},
"appendItems": {
  "Day": "{0} ({2}: {1})",
  "Day-Of-Week": "{0} {1}",
  "Era": "{1} {0}",
  "Hour": "{0} ({2}: {1})",
  "Minute": "{0} ({2}: {1})",
  "Month": "{0} ({2}: {1})",
  "Quarter": "{0} ({2}: {1})",
  "Second": "{0} ({2}: {1})",

```

```

    "Timezone": "{0} {1}",
    "Week": "{0} ({2}: {1})",
    "Year": "{1} {0}"
  },
  "intervalFormats": {
    "intervalFormatFallback": "{0} - {1}",
    "d": {
      "d": "d-d"
    },
    "h": {
      "a": "h a - h a",
      "h": "h-h a"
    },
    "H": {
      "H": "HH-HH"
    },
    "hm": {
      "a": "h:mm a - h:mm a",
      "h": "h:mm-h:mm a",
      "m": "h:mm-h:mm a"
    },
    "Hm": {
      "H": "HH:mm-HH:mm",
      "m": "HH:mm-HH:mm"
    },
    "hmv": {
      "a": "h:mm a - h:mm a v",
      "h": "h:mm-h:mm a v",
      "m": "h:mm-h:mm a v"
    },
    "Hmv": {
      "H": "HH:mm-HH:mm v",
      "m": "HH:mm-HH:mm v"
    },
    "hv": {
      "a": "h a - h a v",
      "h": "h-h a v"
    },
    "Hv": {
      "H": "HH-HH v"
    },
    "M": {
      "M": "M-M"
    },
    "Md": {
      "d": "M/d - M/d",
      "M": "M/d - M/d"
    },
    "MEd": {
      "d": "E, d/M - E, d/M",
      "M": "E, d/M - E, d/M"
    },
    "MMM": {
      "M": "MMM-MMM"
    },
    "MMMd": {
      "d": "d-d MMM",

```

[illegible]

## CA-GREGORIAN.JSX

{

```
"main";
{
  "ar";
  {
    "identity";
    {
      "version";
      {
        "_number";
        "$Revision: 12879 $",
        "_cldrVersion";
        "30.0.3";
      }
      "language";
      "ar";
    }
    "dates";
    {
      "calendars";
      {
        "gregorian";
        {
          "months";
          {
            "format";
            {
              "abbreviated";
              {
                "1";
                "يناير",
                "2";
                "فبراير",
                "3";
                "مارس",
                "4";
                "أبريل",
                "5";
                "مايو",
                "6";
                "يونيو",
                "7";
                "يوليو",
                "8";
                "أغسطس",
                "9";
                "سبتمبر",
                "10";
                "أكتوبر",
                "11";
                "نوفمبر",
                "12";
                "ديسمبر";
              }
            }
            "narrow";
            {
              "1";
              "ي",
```

```

        "2";
        "ف",
        "3";
        "م",
        "4";
        "أ",
        "5";
        "و",
        "6";
        "ن",
        "7";
        "ل",
        "8";
        "غ",
        "9";
        "س",
        "10";
        "ك",
        "11";
        "ب",
        "12";
        "د";
    }
    "wide";
    {
        "1";
        "يناير",
        "2";
        "فبراير",
        "3";
        "مارس",
        "4";
        "أبريل",
        "5";
        "مايو",
        "6";
        "يونيو",
        "7";
        "يوليو",
        "8";
        "أغسطس",
        "9";
        "سبتمبر",
        "10";
        "أكتوبر",
        "11";
        "نوفمبر",
        "12";
        "ديسمبر";
    }
}
"stand-alone";
{
    "abbreviated";
    {
        "1";
        "يناير",

```

```

        "2";
        "فبراير",
        "3";
        "مارس",
        "4";
        "أبريل",
        "5";
        "مايو",
        "6";
        "يونيو",
        "7";
        "يوليو",
        "8";
        "أغسطس",
        "9";
        "سبتمبر",
        "10";
        "أكتوبر",
        "11";
        "نوفمبر",
        "12";
        "ديسمبر";
    }
    "narrow";
    {
        "1";
        "ي",
        "2";
        "ف",
        "3";
        "م",
        "4";
        "أ",
        "5";
        "و",
        "6";
        "ن",
        "7";
        "ل",
        "8";
        "غ",
        "9";
        "س",
        "10";
        "ك",
        "11";
        "ب",
        "12";
        "د";
    }
    "wide";
    {
        "1";
        "يناير",
        "2";
        "فبراير",
        "3";

```

```

        "مارس",
        "4";
        "أبريل",
        "5";
        "مايو",
        "6";
        "يونيو",
        "7";
        "يوليو",
        "8";
        "أغسطس",
        "9";
        "سبتمبر",
        "10";
        "أكتوبر",
        "11";
        "نوفمبر",
        "12";
        "ديسمبر";
    }
}
}
"days";
{
    "format";
    {
        "abbreviated";
        {
            "sun";
            "الأحد",
            "mon";
            "الاثنين",
            "tue";
            "الثلاثاء",
            "wed";
            "الأربعاء",
            "thu";
            "الخميس",
            "fri";
            "الجمعة",
            "sat";
            "السبت";
        }
        "narrow";
        {
            "sun";
            "ح",
            "mon";
            "ن",
            "tue";
            "ث",
            "wed";
            "ر",
            "thu";
            "خ",
            "fri";
            "ج";
        }
    }
}

```



```
        "sat";
        "س";
    }
    "short";
    {
        "sun";
        "الأحد",
        "mon";
        "الاثنين",
        "tue";
        "الثلاثاء",
        "wed";
        "الأربعاء",
        "thu";
        "الخميس",
        "fri";
        "الجمعة",
        "sat";
        "السبت";
    }
    "wide";
    {
        "sun";
        "الأحد",
        "mon";
        "الاثنين",
        "tue";
        "الثلاثاء",
        "wed";
        "الأربعاء",
        "thu";
        "الخميس",
        "fri";
        "الجمعة",
        "sat";
        "السبت";
    }
}
"stand-alone";
{
    "abbreviated";
    {
        "sun";
        "الأحد",
        "mon";
        "الاثنين",
        "tue";
        "الثلاثاء",
        "wed";
        "الأربعاء",
        "thu";
        "الخميس",
        "fri";
        "الجمعة",
        "sat";
        "السبت";
    }
}
```

```
        "narrow";
        {
            "sun";
            "ح",
            "mon";
            "ن",
            "tue";
            "ث",
            "wed";
            "ر",
            "thu";
            "خ",
            "fri";
            "ج",
            "sat";
            "س";
        }
        "short";
        {
            "sun";
            "الأحد",
            "mon";
            "الاثنين",
            "tue";
            "الثلاثاء",
            "wed";
            "الأربعاء",
            "thu";
            "الخميس",
            "fri";
            "الجمعة",
            "sat";
            "السبت";
        }
        "wide";
        {
            "sun";
            "الأحد",
            "mon";
            "الاثنين",
            "tue";
            "الثلاثاء",
            "wed";
            "الأربعاء",
            "thu";
            "الخميس",
            "fri";
            "الجمعة",
            "sat";
            "السبت";
        }
    }
    "quarters";
    {
        "format";
        {
```

```
"abbreviated";
{
  "1";
  "الربع الأول",
  "2";
  "الربع الثاني",
  "3";
  "الربع الثالث",
  "4";
  "الربع الرابع";
}
"narrow";
{
  "1";
  "\",
  "2";
  "٢",
  "3";
  "٣",
  "4";
  "٤";
}
"wide";
{
  "1";
  "الربع الأول",
  "2";
  "الربع الثاني",
  "3";
  "الربع الثالث",
  "4";
  "الربع الرابع";
}
}
"stand-alone";
{
  "abbreviated";
  {
    "1";
    "الربع الأول",
    "2";
    "الربع الثاني",
    "3";
    "الربع الثالث",
    "4";
    "الربع الرابع";
  }
  "narrow";
  {
    "1";
    "\",
    "2";
    "٢",
    "3";
    "٣",
    "4";
    "٤";
  }
}
```

```

    }
    "wide";
    {
        "1";
        "الربع الأول",
        "2";
        "الربع الثاني",
        "3";
        "الربع الثالث",
        "4";
        "الربع الرابع";
    }
}
"dayPeriods";
{
    "format";
    {
        "abbreviated";
        {
            "am";
            "ص",
            "pm";
            "م",
            "morning1";
            "فجراً",
            "morning2";
            "ص",
            "afternoon1";
            "ظهراً",
            "afternoon2";
            "بعد الظهر",
            "evening1";
            "مساءً",
            "night1";
            "منتصف الليل",
            "night2";
            "ل";
        }
        "narrow";
        {
            "am";
            "ص",
            "pm";
            "م",
            "morning1";
            "فجراً",
            "morning2";
            "صباحاً",
            "afternoon1";
            "ظهراً",
            "afternoon2";
            "بعد الظهر",
            "evening1";
            "مساءً",
            "night1";
            "منتصف الليل",

```

```

        "night2";
        "ليلاً";
    }
    "wide";
    {
        "am";
        "ص",
        "pm";
        "م",
        "morning1";
        "فجرًا",
        "morning2";
        "صباحًا",
        "afternoon1";
        "ظهرًا",
        "afternoon2";
        "بعد الظهر",
        "evening1";
        "مساءً",
        "night1";
        "منتصف الليل",
        "night2";
        "ليلاً";
    }
}
"stand-alone";
{
    "abbreviated";
    {
        "am";
        "ص",
        "pm";
        "م",
        "morning1";
        "فجرًا",
        "morning2";
        "ص",
        "afternoon1";
        "ظهرًا",
        "afternoon2";
        "بعد الظهر",
        "evening1";
        "مساءً",
        "night1";
        "منتصف الليل",
        "night2";
        "ليلاً";
    }
    "narrow";
    {
        "am";
        "ص",
        "pm";
        "م",
        "morning1";
        "فجرًا",
        "morning2";
    }
}

```

```

        "صباحًا",
        "afternoon1";
        "ظهرًا",
        "afternoon2";
        "بعد الظهر",
        "evening1";
        "مساءً",
        "night1";
        "منتصف الليل",
        "night2";
        "ليلاً";
    }
    "wide";
    {
        "am";
        "صباحًا",
        "pm";
        "مساءً",
        "morning1";
        "فجرًا",
        "morning2";
        "صباحًا",
        "afternoon1";
        "ظهرًا",
        "afternoon2";
        "بعد الظهر",
        "evening1";
        "مساءً",
        "night1";
        "منتصف الليل",
        "night2";
        "ليلاً";
    }
}
"eras";
{
    "eraNames";
    {
        "0";
        "قبل الميلاد",
        "0-alt-variant";
        "BCE",
        "1";
        "ميلادي",
        "1-alt-variant";
        "بعد الميلاد";
    }
    "eraAbbr";
    {
        "0";
        "ق.م",
        "0-alt-variant";
        "BCE",
        "1";
        "م",
        "1-alt-variant";
    }
}

```

```

        "م.ب";
    }
    "eraNarrow";
    {
        "0";
        "م.ق",
        "0-alt-variant";
        "BCE",
        "1";
        "م",
        "1-alt-variant";
        "م.ب";
    }
}
"dateFormats";
{
    "full";
    "EEEE, d MMMM, y",
    "long";
    "d MMMM, y",
    "medium";
    "dd/MM/y",
    "short";
    "d/M/y";
}
"timeFormats";
{
    "full";
    "h:mm:ss a zzzz",
    "long";
    "h:mm:ss a z",
    "medium";
    "h:mm:ss a",
    "short";
    "h:mm a";
}
"dateTimeFormats";
{
    "full";
    "{1} {0}",
    "long";
    "{1} {0}",
    "medium";
    "{1} {0}",
    "short";
    "{1} {0}",
    "availableFormats";
    {
        "d";
        "d",
        "E";
        "ccc",
        "Ed";
        "E, d",
        "Ehm";
        "E h:mm a",
        "EHm";
    }
}

```

```

"E HH:mm",
  "Ehms";
"E h:mm:ss a",
  "EHms";
"E HH:mm:ss",
  "Gy";
"y G",
  "GyMMM";
"MMM y G",
  "GyMMMd";
"d MMM, y G",
  "GyMMMED";
"E, d MMM, y G",
  "h";
"h a",
  "H";
"HH",
  "hm";
"h:mm a",
  "Hm";
"HH:mm",
  "hms";
"h:mm:ss a",
  "Hms";
"HH:mm:ss",
  "hmsv";
"h:mm:ss a v",
  "Hmsv";
"HH:mm:ss v",
  "hmv";
"h:mm a v",
  "Hmv";
"HH:mm v",
  "M";
"L",
  "Md";
"d/M",
  "MEd";
"E, d/M",
  "MMdd";
"dd/MM",
  "MMM";
"LLL",
  "MMMd";
"d MMM",
  "MMMED";
"E, d MMM",
  "MMMMd";
"d MMMM",
  "MMMMEd";
"E, d MMMM",
  "MMMMW";
"MMM من الأسبوع",
  "MMMMW";
"MMM من الأسبوع",
  "MMMMW";
"MMM من الأسبوع",
  "MMMMW";

```



```

        "MMMMW";
        "من W الأسبوع MMM",
        "MMMMW";
        "من W الأسبوع MMM",
        "MMMMW";
        "من W الأسبوع MMM",
        "ms";
        "mm:ss",
        "y";
        "Y",
        "yM";
        "M/Y",
        "yMd";
        "d/M/Y",
        "yMEd";
        "E، d/M/Y",
        "yMM";
        "MM/Y",
        "yMMM";
        "MMM Y",
        "yMMMd";
        "d MMM، Y",
        "yMMMEd";
        "E، d MMM، Y",
        "yMMMM";
        "MMMM Y",
        "yQQQ";
        "QQQ Y",
        "yQQQQ";
        "QQQQ Y",
        "yw";
        "Y من سنة w الأسبوع",
        "yw";
        "Y من سنة w الأسبوع",
        "yw";
        "Y من سنة w الأسبوع",
        "yw";
        "Y من سنة w الأسبوع",
        "yw";
        "Y من سنة w الأسبوع",
        "yw";
        "Y من سنة w الأسبوع";
    }
    "appendItems";
    {
        "Day";
        "{0} ({2}: {1})",
        "Day-Of-Week";
        "{0} {1}",
        "Era";
        "{1} {0}",
        "Hour";
        "{0} ({2}: {1})",
        "Minute";
        "{0} ({2}: {1})",
        "Month";
        "{0} ({2}: {1})",

```

```

        "Quarter";
        "{0} ({2}: {1})",
        "Second";
        "{0} ({2}: {1})",
        "Timezone";
        "{0} {1}",
        "Week";
        "{0} ({2}: {1})",
        "Year";
        "{1} {0}";
    }
    "intervalFormats";
    {
        "intervalFormatFallback";
        "{0} - {1}",
        "d";
        {
            "d";
            "d-d";
        }
        "h";
        {
            "a";
            "h a - h a",
            "h";
            "h-h a";
        }
        "H";
        {
            "H";
            "HH-HH";
        }
        "hm";
        {
            "a";
            "h:mm a - h:mm a",
            "h";
            "h:mm-h:mm a",
            "m";
            "h:mm-h:mm a";
        }
        "Hm";
        {
            "H";
            "HH:mm-HH:mm",
            "m";
            "HH:mm-HH:mm";
        }
        "hmv";
        {
            "a";
            "h:mm a - h:mm a v",
            "h";
            "h:mm-h:mm a v",
            "m";
            "h:mm-h:mm a v";
        }
    }

```

```
"Hmv";
{
  "H";
  "HH:mm-HH:mm v",
  "m";
  "HH:mm-HH:mm v";
}
"hv";
{
  "a";
  "h a - h a v",
  "h";
  "h-h a v";
}
"Hv";
{
  "H";
  "HH-HH v";
}
"M";
{
  "M";
  "M-M";
}
"Md";
{
  "d";
  "M/d - M/d",
  "M";
  "M/d - M/d";
}
"MEd";
{
  "d";
  "E, d/M - E, d/M",
  "M";
  "E, d/M - E, d/M";
}
"MMM";
{
  "M";
  "MMM-MMM";
}
"MMMd";
{
  "d";
  "d-d MMM",
  "M";
  "d MMM - d MMM";
}
"MMMEd";
{
  "d";
  "E, d - E, d MMM",
  "M";
  "E, d MMM - E, d MMM";
}
```

```

"MMMM";
{
  "M";
  "LLLL-LLLL";
}
"Y";
{
  "Y";
  "Y-Y";
}
"yM";
{
  "M";
  "M/Y - M/Y",
  "Y";
  "M/Y - M/Y";
}
"yMd";
{
  "d";
  "d/M/Y - d/M/Y",
  "M";
  "d/M/Y - d/M/Y",
  "Y";
  "d/M/Y - d/M/Y";
}
"yMED";
{
  "d";
  "E, dd/MM/Y - E, dd/MM/Y",
  "M";
  "E, d/M/Y - E, d/M/Y",
  "Y";
  "E, d/M/Y - E, d/M/Y";
}
"yMMM";
{
  "M";
  "MMM - MMM, Y",
  "Y";
  "MMM, Y - MMM, Y";
}
"yMMMd";
{
  "d";
  "d-d MMM, Y",
  "M";
  "d MMM - d MMM, Y",
  "Y";
  "d MMM, Y - d MMM, Y";
}
"yMMMED";
{
  "d";
  "E, d - E, d MMM, Y",
  "M";
  "E, d MMM - E, d MMM, Y",

```

```

    "Y";
    "E. d MMM. y - E. d MMM. y";
}
"yMMMM";
{
    "M";
    "MMMM - MMMM. y",
    "y";
    "MMMM. y - MMMM. y";
}
}
}
}
}
}
}
}
}
}
```

## CURRENCIES.JSON

```
{
  "main": {
    "ar": {
      "identity": {
        "version": {
          "_number": "$Revision: 12879 $",
          "_cldrVersion": "30.0.3"
        },
        "language": "ar"
      },
      "numbers": {
        "currencies": {
          "ADP": {
            "displayName": "بيستا أندوري",
            "symbol": "ADP"
          },
          "AED": {
            "displayName": "درهم إماراتي",
            "displayName-count-zero": "درهم إماراتي",
            "displayName-count-one": "درهم إماراتي",
            "displayName-count-two": "درهم إماراتي",
            "displayName-count-few": "درهم إماراتي",
            "displayName-count-many": "درهم إماراتي",
            "displayName-count-other": "درهم إماراتي",
            "symbol": ".د.إ"
          },
          "AFA": {
            "displayName": "أفغاني - 2002-1927",
            "symbol": "AFA"
          },
          "AFN": {
            "displayName": "أفغاني",
            "displayName-count-zero": "أفغاني أفغانستاني",
            "displayName-count-one": "أفغاني أفغانستاني",
            "displayName-count-two": "أفغاني أفغانستاني",

```

```

    "displayName-count-few": "أفغاني أفغانستاني",
    "displayName-count-many": "أفغاني أفغانستاني",
    "displayName-count-other": "أفغاني أفغانستاني",
    "symbol": "AFN"
  },
  "ALL": {
    "displayName": "ليك ألباني",
    "displayName-count-zero": "ليك ألباني",
    "displayName-count-one": "ليك ألباني",
    "displayName-count-two": "ليك ألباني",
    "displayName-count-few": "ليك ألباني",
    "displayName-count-many": "ليك ألباني",
    "displayName-count-other": "ليك ألباني",
    "symbol": "ALL"
  },
  "AMD": {
    "displayName": "درام أرميني",
    "displayName-count-zero": "درام أرميني",
    "displayName-count-one": "درام أرميني",
    "displayName-count-two": "درام أرميني",
    "displayName-count-few": "درام أرميني",
    "displayName-count-many": "درام أرميني",
    "displayName-count-other": "درام أرميني",
    "symbol": "AMD"
  },
  "ANG": {
    "displayName": "غيلدر أنتيلي هولندي",
    "displayName-count-zero": "غيلدر أنتيلي هولندي",
    "displayName-count-one": "غيلدر أنتيلي هولندي",
    "displayName-count-two": "غيلدر أنتيلي هولندي",
    "displayName-count-few": "غيلدر أنتيلي هولندي",
    "displayName-count-many": "غيلدر أنتيلي هولندي",
    "displayName-count-other": "غيلدر أنتيلي هولندي",
    "symbol": "ANG"
  },
  "AOA": {
    "displayName": "كوانزا أنجولي",
    "displayName-count-zero": "كوانزا أنجولي",
    "displayName-count-one": "كوانزا أنجولي",
    "displayName-count-two": "كوانزا أنجولي",
    "displayName-count-few": "كوانزا أنجولي",
    "displayName-count-many": "كوانزا أنجولي",
    "displayName-count-other": "كوانزا أنجولي",
    "symbol": "AOA",
    "symbol-alt-narrow": "Kz"
  },
  "AOK": {
    "displayName": "كوانزا أنجولي - 1990-1977",
    "symbol": "AOK"
  },
  "AON": {
    "displayName": "كوانزا أنجولي جديدة - 2000-1990",
    "symbol": "AON"
  },
  "AOR": {
    "displayName": "كوانزا أنجولي معدلة - 1999 - 1995",
    "symbol": "AOR"
  }

```

```

    },
    "ARA": {
      "displayName": "استرال أرجنتيني",
      "symbol": "ARA"
    },
    "ARL": {
      "displayName": "ARL",
      "symbol": "ARL"
    },
    "ARM": {
      "displayName": "ARM",
      "symbol": "ARM"
    },
    "ARP": {
      "displayName": "1985-1983 - بيزو أرجنتيني",
      "symbol": "ARP"
    },
    "ARS": {
      "displayName": "بيزو أرجنتيني",
      "displayName-count-zero": "بيزو أرجنتيني",
      "displayName-count-one": "بيزو أرجنتيني",
      "displayName-count-two": "بيزو أرجنتيني",
      "displayName-count-few": "بيزو أرجنتيني",
      "displayName-count-many": "بيزو أرجنتيني",
      "displayName-count-other": "بيزو أرجنتيني",
      "symbol": "ARS",
      "symbol-alt-narrow": "AR$"
    },
    "ATS": {
      "displayName": "شلتن نمساوي",
      "symbol": "ATS"
    },
    "AUD": {
      "displayName": "دولار أسترالي",
      "displayName-count-zero": "دولار أسترالي",
      "displayName-count-one": "دولار أسترالي",
      "displayName-count-two": "دولار أسترالي",
      "displayName-count-few": "دولار أسترالي",
      "displayName-count-many": "دولار أسترالي",
      "displayName-count-other": "دولار أسترالي",
      "symbol": "AU$",
      "symbol-alt-narrow": "AU$"
    },
    "AWG": {
      "displayName": "فلورن أروبي",
      "displayName-count-zero": "فلورن أروبي",
      "displayName-count-one": "فلورن أروبي",
      "displayName-count-two": "فلورن أروبي",
      "displayName-count-few": "فلورن أروبي",
      "displayName-count-many": "فلورن أروبي",
      "displayName-count-other": "فلورن أروبي",
      "symbol": "AWG"
    },
    "AZM": {
      "displayName": "مانات أذربيجاني",
      "symbol": "AZM"
    },
  },

```

```

"AZN": {
  "displayName": "مانات أذربيجان",
  "displayName-count-zero": "مانت أذربيجاني",
  "displayName-count-one": "مانت أذربيجاني",
  "displayName-count-two": "مانت أذربيجاني",
  "displayName-count-few": "مانت أذربيجاني",
  "displayName-count-many": "مانت أذربيجاني",
  "displayName-count-other": "مانت أذربيجاني",
  "symbol": "AZN"
},
"BAD": {
  "displayName": "دينار البوسنة والهرسك",
  "symbol": "BAD"
},
"BAM": {
  "displayName": "مارك البوسنة والهرسك قابل للتحويل",
  "displayName-count-zero": "مارك البوسنة والهرسك قابل للتحويل",
  "displayName-count-one": "مارك البوسنة والهرسك قابل للتحويل",
  "displayName-count-two": "مارك البوسنة والهرسك قابل للتحويل",
  "displayName-count-few": "مارك البوسنة والهرسك قابل للتحويل",
  "displayName-count-many": "مارك البوسنة والهرسك قابل للتحويل",
  "displayName-count-other": "مارك البوسنة والهرسك قابل للتحويل",
  "symbol": "BAM",
  "symbol-alt-narrow": "KM"
},
"BAN": {
  "displayName": "BAN",
  "symbol": "BAN"
},
"BBD": {
  "displayName": "دولار بربادوسي",
  "displayName-count-zero": "دولار بربادوسي",
  "displayName-count-one": "دولار بربادوسي",
  "displayName-count-two": "دولار بربادوسي",
  "displayName-count-few": "دولار بربادوسي",
  "displayName-count-many": "دولار بربادوسي",
  "displayName-count-other": "دولار بربادوسي",
  "symbol": "BBD",
  "symbol-alt-narrow": "BB$"
},
"BDT": {
  "displayName": "تাকা بنجلاديشي",
  "displayName-count-zero": "تাকা بنجلاديشي",
  "displayName-count-one": "تাকা بنجلاديشي",
  "displayName-count-two": "تাকা بنجلاديشي",
  "displayName-count-few": "تাকা بنجلاديشي",
  "displayName-count-many": "تাকা بنجلاديشي",
  "displayName-count-other": "تাকা بنجلاديشي",
  "symbol": "BDT",
  "symbol-alt-narrow": "ট"
},
"BEC": {
  "displayName": "فرنك بلجيكي قابل للتحويل",
  "symbol": "BEC"
},
"BEF": {

```



```

    "displayName": "فرنك بلجيكي",
    "symbol": "BEF"
  },
  "BEL": {
    "displayName": "فرنك بلجيكي مالي",
    "symbol": "BEL"
  },
  "BGL": {
    "displayName": "BGL",
    "symbol": "BGL"
  },
  "BGM": {
    "displayName": "BGM",
    "symbol": "BGM"
  },
  "BGN": {
    "displayName": "ليف بلغاري",
    "displayName-count-zero": "ليف بلغاري",
    "displayName-count-one": "ليف بلغاري",
    "displayName-count-two": "ليف بلغاري",
    "displayName-count-few": "ليف بلغاري",
    "displayName-count-many": "ليف بلغاري",
    "displayName-count-other": "ليف بلغاري",
    "symbol": "BGN"
  },
  "BGO": {
    "displayName": "BGO",
    "symbol": "BGO"
  },
  "BHD": {
    "displayName": "دينار بحريني",
    "displayName-count-zero": "دينار بحريني",
    "displayName-count-one": "دينار بحريني",
    "displayName-count-two": "دينار بحريني",
    "displayName-count-few": "دينار بحريني",
    "displayName-count-many": "دينار بحريني",
    "displayName-count-other": "دينار بحريني",
    "symbol": "د.ب."
  },
  "BIF": {
    "displayName": "فرنك بروندي",
    "displayName-count-zero": "فرنك بروندي",
    "displayName-count-one": "فرنك بروندي",
    "displayName-count-two": "فرنك بروندي",
    "displayName-count-few": "فرنك بروندي",
    "displayName-count-many": "فرنك بروندي",
    "displayName-count-other": "فرنك بروندي",
    "symbol": "BIF"
  },
  "BMD": {
    "displayName": "دولار برمودي",
    "displayName-count-zero": "دولار برمودي",
    "displayName-count-one": "دولار برمودي",
    "displayName-count-two": "دولار برمودي",
    "displayName-count-few": "دولار برمودي",
    "displayName-count-many": "دولار برمودي",
    "displayName-count-other": "دولار برمودي",

```

```

    "symbol": "BMD",
    "symbol-alt-narrow": "BM$"
  },
  "BND": {
    "displayName": "دولار بروناي",
    "displayName-count-zero": "دولار بروناي",
    "displayName-count-one": "دولار بروناي",
    "displayName-count-two": "دولار بروناي",
    "displayName-count-few": "دولار بروناي",
    "displayName-count-many": "دولار بروناي",
    "displayName-count-other": "دولار بروناي",
    "symbol": "BND",
    "symbol-alt-narrow": "BN$"
  },
  "BOB": {
    "displayName": "بوليفيانو بوليفي",
    "displayName-count-zero": "بوليفيانو بوليفي",
    "displayName-count-one": "بوليفيانو بوليفي",
    "displayName-count-two": "بوليفيانو بوليفي",
    "displayName-count-few": "بوليفيانو بوليفي",
    "displayName-count-many": "بوليفيانو بوليفي",
    "displayName-count-other": "بوليفيانو بوليفي",
    "symbol": "BOB",
    "symbol-alt-narrow": "Bs"
  },
  "BOL": {
    "displayName": "BOL",
    "symbol": "BOL"
  },
  "BOP": {
    "displayName": "بيزو بوليفي",
    "symbol": "BOP"
  },
  "BOV": {
    "displayName": "مفدول بوليفي",
    "symbol": "BOV"
  },
  "BRB": {
    "displayName": "نوفو كروزايرو برازيل - 1967-1986",
    "symbol": "BRB"
  },
  "BRC": {
    "displayName": "كروزادو برازيل",
    "symbol": "BRC"
  },
  "BRE": {
    "displayName": "كروزايرو برازيل - 1990-1993",
    "symbol": "BRE"
  },
  "BRL": {
    "displayName": "ريال برازيل",
    "displayName-count-zero": "ريال برازيل",
    "displayName-count-one": "ريال برازيل",
    "displayName-count-two": "ريال برازيل",
    "displayName-count-few": "ريال برازيل",
    "displayName-count-many": "ريال برازيل",
    "displayName-count-other": "ريال برازيل",

```

```

    "symbol": "R$",
    "symbol-alt-narrow": "R$"
  },
  "BRN": {
    "displayName": "BRN",
    "symbol": "BRN"
  },
  "BRR": {
    "displayName": "BRR",
    "symbol": "BRR"
  },
  "BRZ": {
    "displayName": "BRZ",
    "symbol": "BRZ"
  },
  "BSD": {
    "displayName": "دولار باهامي",
    "displayName-count-zero": "دولار باهامي",
    "displayName-count-one": "دولار باهامي",
    "displayName-count-two": "دولار باهامي",
    "displayName-count-few": "دولار باهامي",
    "displayName-count-many": "دولار باهامي",
    "displayName-count-other": "دولار باهامي",
    "symbol": "BSD",
    "symbol-alt-narrow": "BS$"
  },
  "BTN": {
    "displayName": "نولتوم بوتاني",
    "displayName-count-zero": "نولتوم بوتاني",
    "displayName-count-one": "نولتوم بوتاني",
    "displayName-count-two": "نولتوم بوتاني",
    "displayName-count-few": "نولتوم بوتاني",
    "displayName-count-many": "نولتوم بوتاني",
    "displayName-count-other": "نولتوم بوتاني",
    "symbol": "BTN"
  },
  "BUK": {
    "displayName": "كيات بورمي",
    "symbol": "BUK"
  },
  "BWP": {
    "displayName": "بولا بتسواني",
    "displayName-count-zero": "بولا بتسواني",
    "displayName-count-one": "بولا بتسواني",
    "displayName-count-two": "بولا بتسواني",
    "displayName-count-few": "بولا بتسواني",
    "displayName-count-many": "بولا بتسواني",
    "displayName-count-other": "بولا بتسواني",
    "symbol": "BWP",
    "symbol-alt-narrow": "P"
  },
  "BYB": {
    "displayName": "روبل بيلاروسي جديد - 1999-1994",
    "symbol": "BYB"
  },
  "BYN": {
    "displayName": "روبل بيلاروسي",

```

```

    "displayName-count-zero": "روبل بيلاروسي",
    "displayName-count-one": "روبل بيلاروسي",
    "displayName-count-two": "روبل بيلاروسي",
    "displayName-count-few": "روبل بيلاروسي",
    "displayName-count-many": "روبل بيلاروسي",
    "displayName-count-other": "روبل بيلاروسي",
    "symbol": "BYN",
    "symbol-alt-narrow": "p."
  },
  "BYR": {
    "displayName": ")٢٠١٦-٢٠٠٠ (روبل بيلاروسي",
    "displayName-count-zero": ")٢٠١٦-٢٠٠٠ (روبل بيلاروسي",
    "displayName-count-one": ")٢٠١٦-٢٠٠٠ (روبل بيلاروسي",
    "displayName-count-two": ")٢٠١٦-٢٠٠٠ (روبل بيلاروسي",
    "displayName-count-few": ")٢٠١٦-٢٠٠٠ (روبل بيلاروسي",
    "displayName-count-many": ")٢٠١٦-٢٠٠٠ (روبل بيلاروسي",
    "displayName-count-other": ")٢٠١٦-٢٠٠٠ (روبل بيلاروسي",
    "symbol": "BYR"
  },
  "BZD": {
    "displayName": "دولار بليزي",
    "displayName-count-zero": "دولار بليزي",
    "displayName-count-one": "دولار بليزي",
    "displayName-count-two": "دولاران بليزيان",
    "displayName-count-few": "دولار بليزي",
    "displayName-count-many": "دولار بليزي",
    "displayName-count-other": "دولار بليزي",
    "symbol": "BZD",
    "symbol-alt-narrow": "BZ$"
  },
  "CAD": {
    "displayName": "دولار كندي",
    "displayName-count-zero": "دولار كندي",
    "displayName-count-one": "دولار كندي",
    "displayName-count-two": "دولار كندي",
    "displayName-count-few": "دولار كندي",
    "displayName-count-many": "دولار كندي",
    "displayName-count-other": "دولار كندي",
    "symbol": "CA$",
    "symbol-alt-narrow": "CA$"
  },
  "CDF": {
    "displayName": "فرنك كونغولي",
    "displayName-count-zero": "فرنك كونغولي",
    "displayName-count-one": "فرنك كونغولي",
    "displayName-count-two": "فرنك كونغولي",
    "displayName-count-few": "فرنك كونغولي",
    "displayName-count-many": "فرنك كونغولي",
    "displayName-count-other": "فرنك كونغولي",
    "symbol": "CDF"
  },
  "CHE": {
    "displayName": "CHE",
    "symbol": "CHE"
  },
  "CHF": {
    "displayName": "فرنك سويسري",

```

```

    "displayName-count-zero": "فرنك سويسري",
    "displayName-count-one": "فرنك سويسري",
    "displayName-count-two": "فرنك سويسري",
    "displayName-count-few": "فرنك سويسري",
    "displayName-count-many": "فرنك سويسري",
    "displayName-count-other": "فرنك سويسري",
    "symbol": "CHF"
  },
  "CHW": {
    "displayName": "CHW",
    "symbol": "CHW"
  },
  "CLE": {
    "displayName": "CLE",
    "symbol": "CLE"
  },
  "CLF": {
    "displayName": "CLF",
    "symbol": "CLF"
  },
  "CLP": {
    "displayName": "بيزو شيلي",
    "displayName-count-zero": "بيزو شيلي",
    "displayName-count-one": "بيزو شيلي",
    "displayName-count-two": "بيزو شيلي",
    "displayName-count-few": "بيزو شيلي",
    "displayName-count-many": "بيزو شيلي",
    "displayName-count-other": "بيزو شيلي",
    "symbol": "CLP",
    "symbol-alt-narrow": "CL$"
  },
  "CNY": {
    "displayName": "يوان صيني",
    "displayName-count-zero": "يوان صيني",
    "displayName-count-one": "يوان صيني",
    "displayName-count-two": "يوان صيني",
    "displayName-count-few": "يوان صيني",
    "displayName-count-many": "يوان صيني",
    "displayName-count-other": "يوان صيني",
    "symbol": "CN¥",
    "symbol-alt-narrow": "CN¥"
  },
  "COP": {
    "displayName": "بيزو كولومبي",
    "displayName-count-zero": "بيزو كولومبي",
    "displayName-count-one": "بيزو كولومبي",
    "displayName-count-two": "بيزو كولومبي",
    "displayName-count-few": "بيزو كولومبي",
    "displayName-count-many": "بيزو كولومبي",
    "displayName-count-other": "بيزو كولومبي",
    "symbol": "COP",
    "symbol-alt-narrow": "CO$"
  },
  "COU": {
    "displayName": "COU",
    "symbol": "COU"
  },

```

```

"CRC": {
  "displayName": "كولن كوستا ريكي",
  "displayName-count-zero": "كولن كوستا ريكي",
  "displayName-count-one": "كولن كوستا ريكي",
  "displayName-count-two": "كولن كوستا ريكي",
  "displayName-count-few": "كولن كوستا ريكي",
  "displayName-count-many": "كولن كوستا ريكي",
  "displayName-count-other": "كولن كوستا ريكي",
  "symbol": "CRC",
  "symbol-alt-narrow": "¢"
},
"CSD": {
  "displayName": "دينار صربي قديم",
  "symbol": "CSD"
},
"CSK": {
  "displayName": "كرونه تشيكوسلوفاكيا",
  "symbol": "CSK"
},
"CUC": {
  "displayName": "بيزو كوبي قابل للتحويل",
  "displayName-count-zero": "بيزو كوبي قابل للتحويل",
  "displayName-count-one": "بيزو كوبي قابل للتحويل",
  "displayName-count-two": "بيزو كوبي قابل للتحويل",
  "displayName-count-few": "بيزو كوبي قابل للتحويل",
  "displayName-count-many": "بيزو كوبي قابل للتحويل",
  "displayName-count-other": "بيزو كوبي قابل للتحويل",
  "symbol": "CUC",
  "symbol-alt-narrow": "$"
},
"CUP": {
  "displayName": "بيزو كوبي",
  "displayName-count-zero": "بيزو كوبي",
  "displayName-count-one": "بيزو كوبي",
  "displayName-count-two": "بيزو كوبي",
  "displayName-count-few": "بيزو كوبي",
  "displayName-count-many": "بيزو كوبي",
  "displayName-count-other": "بيزو كوبي",
  "symbol": "CUP",
  "symbol-alt-narrow": "CU$"
},
"CVE": {
  "displayName": "اسكودو الرأس الخضراء",
  "displayName-count-zero": "اسكودو الرأس الخضراء",
  "displayName-count-one": "اسكودو الرأس الخضراء",
  "displayName-count-two": "اسكودو الرأس الخضراء",
  "displayName-count-few": "اسكودو الرأس الخضراء",
  "displayName-count-many": "اسكودو الرأس الخضراء",
  "displayName-count-other": "اسكودو الرأس الخضراء",
  "symbol": "CVE"
},
"CYP": {
  "displayName": "جنيه قبرصي",
  "symbol": "CYP"
},
"CZK": {
  "displayName": "كرونه تشيكية",

```

```

    "displayName-count-zero": "كرونة تشيكية",
    "displayName-count-one": "كرونة تشيكية",
    "displayName-count-two": "كرونة تشيكية",
    "displayName-count-few": "كرونة تشيكية",
    "displayName-count-many": "كرونة تشيكية",
    "displayName-count-other": "كرونة تشيكية",
    "symbol": "CZK",
    "symbol-alt-narrow": "Kč"
  },
  "DDM": {
    "displayName": "أوستمارك ألماني شرقي",
    "symbol": "DDM"
  },
  "DEM": {
    "displayName": "مارك ألماني",
    "symbol": "DEM"
  },
  "DJF": {
    "displayName": "فرنك جيبوتي",
    "displayName-count-zero": "فرنك جيبوتي",
    "displayName-count-one": "فرنك جيبوتي",
    "displayName-count-two": "فرنك جيبوتي",
    "displayName-count-few": "فرنك جيبوتي",
    "displayName-count-many": "فرنك جيبوتي",
    "displayName-count-other": "فرنك جيبوتي",
    "symbol": "DJF"
  },
  "DKK": {
    "displayName": "كرونة دانماركي",
    "displayName-count-zero": "كرونة دانماركي",
    "displayName-count-one": "كرونة دانماركي",
    "displayName-count-two": "كرونة دانماركي",
    "displayName-count-few": "كرونة دانماركي",
    "displayName-count-many": "كرونة دانماركي",
    "displayName-count-other": "كرونة دانماركي",
    "symbol": "DKK",
    "symbol-alt-narrow": "kr"
  },
  "DOP": {
    "displayName": "بيزو الدومنيكان",
    "displayName-count-zero": "بيزو الدومنيكان",
    "displayName-count-one": "بيزو الدومنيكان",
    "displayName-count-two": "بيزو الدومنيكان",
    "displayName-count-few": "بيزو الدومنيكان",
    "displayName-count-many": "بيزو الدومنيكان",
    "displayName-count-other": "بيزو الدومنيكان",
    "symbol": "DOP",
    "symbol-alt-narrow": "DO$"
  },
  "DZD": {
    "displayName": "دينار جزائري",
    "displayName-count-zero": "دينار جزائري",
    "displayName-count-one": "دينار جزائري",
    "displayName-count-two": "ديناران جزائريان",
    "displayName-count-few": "دينارات جزائرية",
    "displayName-count-many": "دينارًا جزائريًا",
    "displayName-count-other": "دينار جزائري",

```

```

    "symbol": "د.ج."
  },
  "ECS": {
    "displayName": "ECS",
    "symbol": "ECS"
  },
  "ECV": {
    "displayName": "ECV",
    "symbol": "ECV"
  },
  "EEK": {
    "displayName": "كرونة استونية",
    "symbol": "EEK"
  },
  "EGP": {
    "displayName": "جنيه مصري",
    "displayName-count-zero": "جنيه مصري",
    "displayName-count-one": "جنيه مصري",
    "displayName-count-two": "جنيهان مصريان",
    "displayName-count-few": "جنيهات مصرية",
    "displayName-count-many": "جنيهاً مصرياً",
    "displayName-count-other": "جنيه مصري",
    "symbol": "ج.م.",
    "symbol-alt-narrow": "££"
  },
  "ERN": {
    "displayName": "ناكفا أريتري",
    "displayName-count-zero": "ناكفا أريتري",
    "displayName-count-one": "ناكفا أريتري",
    "displayName-count-two": "ناكفا أريتري",
    "displayName-count-few": "ناكفا أريتري",
    "displayName-count-many": "ناكفا أريتري",
    "displayName-count-other": "ناكفا أريتري",
    "symbol": "ERN"
  },
  "ESA": {
    "displayName": "ESA",
    "symbol": "ESA"
  },
  "ESB": {
    "displayName": "ESB",
    "symbol": "ESB"
  },
  "ESP": {
    "displayName": "بيزيتا إسباني",
    "symbol": "ESP",
    "symbol-alt-narrow": "₧"
  },
  "ETB": {
    "displayName": "بير أثيوبي",
    "displayName-count-zero": "بير أثيوبي",
    "displayName-count-one": "بير أثيوبي",
    "displayName-count-two": "بير أثيوبي",
    "displayName-count-few": "بير أثيوبي",
    "displayName-count-many": "بير أثيوبي",
    "displayName-count-other": "بير أثيوبي",
    "symbol": "ETB"
  }

```



```

    },
    "EUR": {
      "displayName": "يورو",
      "displayName-count-zero": "يورو",
      "displayName-count-one": "يورو",
      "displayName-count-two": "يورو",
      "displayName-count-few": "يورو",
      "displayName-count-many": "يورو",
      "displayName-count-other": "يورو",
      "symbol": "€",
      "symbol-alt-narrow": "€"
    },
    "FIM": {
      "displayName": "ماركا فنلندي",
      "symbol": "FIM"
    },
    "FJD": {
      "displayName": "دولار فيجي",
      "displayName-count-zero": "دولار فيجي",
      "displayName-count-one": "دولار فيجي",
      "displayName-count-two": "دولار فيجي",
      "displayName-count-few": "دولار فيجي",
      "displayName-count-many": "دولار فيجي",
      "displayName-count-other": "دولار فيجي",
      "symbol": "FJD",
      "symbol-alt-narrow": "FJ$"
    },
    "FKP": {
      "displayName": "جنيه جزر فوكلاند",
      "displayName-count-zero": "جنيه جزر فوكلاند",
      "displayName-count-one": "جنيه جزر فوكلاند",
      "displayName-count-two": "جنيه جزر فوكلاند",
      "displayName-count-few": "جنيه جزر فوكلاند",
      "displayName-count-many": "جنيه جزر فوكلاند",
      "displayName-count-other": "جنيه جزر فوكلاند",
      "symbol": "FKP",
      "symbol-alt-narrow": "£"
    },
    "FRF": {
      "displayName": "فرنك فرنسي",
      "symbol": "FRF"
    },
    "GBP": {
      "displayName": "جنيه إسترليني",
      "displayName-count-zero": "جنيه إسترليني",
      "displayName-count-one": "جنيه إسترليني",
      "displayName-count-two": "جنيه إسترليني",
      "displayName-count-few": "جنيه إسترليني",
      "displayName-count-many": "جنيه إسترليني",
      "displayName-count-other": "جنيه إسترليني",
      "symbol": "£",
      "symbol-alt-narrow": "UK£"
    },
    "GEK": {
      "displayName": "GEK",
      "symbol": "GEK"
    }
  },

```

```

"GEL": {
  "displayName": "لاری جورجی",
  "displayName-count-zero": "لاری جورجی",
  "displayName-count-one": "لاری جورجی",
  "displayName-count-two": "لاری جورجی",
  "displayName-count-few": "لاری جورجی",
  "displayName-count-many": "لاری جورجی",
  "displayName-count-other": "لاری جورجی",
  "symbol": "GEL",
  "symbol-alt-narrow": "₾",
  "symbol-alt-variant": "₾"
},
"GHC": {
  "displayName": "سیدی غانی",
  "symbol": "GHC"
},
"GHS": {
  "displayName": "سیدی غانا",
  "displayName-count-zero": "سیدی غانا",
  "displayName-count-one": "سیدی غانا",
  "displayName-count-two": "سیدی غانا",
  "displayName-count-few": "سیدی غانا",
  "displayName-count-many": "سیدی غانا",
  "displayName-count-other": "سیدی غانا",
  "symbol": "GHS"
},
"GIP": {
  "displayName": "جنیه جبل طارق",
  "displayName-count-zero": "جنیه جبل طارق",
  "displayName-count-one": "جنیه جبل طارق",
  "displayName-count-two": "جنیه جبل طارق",
  "displayName-count-few": "جنیه جبل طارق",
  "displayName-count-many": "جنیه جبل طارق",
  "displayName-count-other": "جنیه جبل طارق",
  "symbol": "GIP",
  "symbol-alt-narrow": "£"
},
"GMD": {
  "displayName": "دلasi جامبی",
  "displayName-count-zero": "دلasi جامبی",
  "displayName-count-one": "دلasi جامبی",
  "displayName-count-two": "دلasi جامبی",
  "displayName-count-few": "دلasi جامبی",
  "displayName-count-many": "دلasi جامبی",
  "displayName-count-other": "دلasi جامبی",
  "symbol": "GMD"
},
"GNF": {
  "displayName": "فرنك غينيا",
  "displayName-count-zero": "فرنك غينيا",
  "displayName-count-one": "فرنك غينيا",
  "displayName-count-two": "فرنك غينيا",
  "displayName-count-few": "فرنك غينيا",
  "displayName-count-many": "فرنك غينيا",
  "displayName-count-other": "فرنك غينيا",
  "symbol": "GNF",
  "symbol-alt-narrow": "FG"
}

```

```

    },
    "GNS": {
      "displayName": "سيلى غينيا",
      "symbol": "GNS"
    },
    },
    "GQE": {
      "displayName": "اكويل جونينا غينيا الاستوائية",
      "symbol": "GQE"
    },
    },
    "GRD": {
      "displayName": "دراخما يوناني",
      "symbol": "GRD"
    },
    },
    "GTQ": {
      "displayName": "كوتزال جواتيمالا",
      "displayName-count-zero": "كوتزال جواتيمالا",
      "displayName-count-one": "كوتزال جواتيمالا",
      "displayName-count-two": "كوتزال جواتيمالا",
      "displayName-count-few": "كوتزال جواتيمالا",
      "displayName-count-many": "كوتزال جواتيمالا",
      "displayName-count-other": "كوتزال جواتيمالا",
      "symbol": "GTQ",
      "symbol-alt-narrow": "Q"
    },
    },
    "GWE": {
      "displayName": "اسكود برتغالي غينيا",
      "symbol": "GWE"
    },
    },
    "GWP": {
      "displayName": "بيزو غينيا بيساو",
      "symbol": "GWP"
    },
    },
    "GYD": {
      "displayName": "دولار غيانا",
      "displayName-count-zero": "دولار غيانا",
      "displayName-count-one": "دولار غيانا",
      "displayName-count-two": "دولار غيانا",
      "displayName-count-few": "دولار غيانا",
      "displayName-count-many": "دولار غيانا",
      "displayName-count-other": "دولار غيانا",
      "symbol": "GYD",
      "symbol-alt-narrow": "GY$"
    },
    },
    "HKD": {
      "displayName": "دولار هونغ كونغ",
      "displayName-count-zero": "دولار هونغ كونغ",
      "displayName-count-one": "دولار هونغ كونغ",
      "displayName-count-two": "دولار هونغ كونغ",
      "displayName-count-few": "دولار هونغ كونغ",
      "displayName-count-many": "دولار هونغ كونغ",
      "displayName-count-other": "دولار هونغ كونغ",
      "symbol": "HK$",
      "symbol-alt-narrow": "HK$"
    },
    },
    "HNL": {
      "displayName": "ليمبيرا هنداروس",
      "displayName-count-zero": "ليمبيرا هندوراس",

```

```

    "displayName-count-one": "ليمبيرا هندوراس",
    "displayName-count-two": "ليمبيرا هندوراس",
    "displayName-count-few": "ليمبيرا هندوراس",
    "displayName-count-many": "ليمبيرا هندوراس",
    "displayName-count-other": "ليمبيرا هندوراس",
    "symbol": "HNL",
    "symbol-alt-narrow": "L"
  },
  "HRD": {
    "displayName": "دينار كرواتي",
    "symbol": "HRD"
  },
  "HRK": {
    "displayName": "كونا كرواتي",
    "displayName-count-zero": "كونا كرواتي",
    "displayName-count-one": "كونا كرواتي",
    "displayName-count-two": "كونا كرواتي",
    "displayName-count-few": "كونا كرواتي",
    "displayName-count-many": "كونا كرواتي",
    "displayName-count-other": "كونا كرواتي",
    "symbol": "HRK",
    "symbol-alt-narrow": "kn"
  },
  "HTG": {
    "displayName": "جوردي هايتي",
    "displayName-count-zero": "جوردي هايتي",
    "displayName-count-one": "جوردي هايتي",
    "displayName-count-two": "جوردي هايتي",
    "displayName-count-few": "جوردي هايتي",
    "displayName-count-many": "جوردي هايتي",
    "displayName-count-other": "جوردي هايتي",
    "symbol": "HTG"
  },
  "HUF": {
    "displayName": "فورينت مجري",
    "displayName-count-zero": "فورينت مجري",
    "displayName-count-one": "فورينت مجري",
    "displayName-count-two": "فورينت مجري",
    "displayName-count-few": "فورينت مجري",
    "displayName-count-many": "فورينت مجري",
    "displayName-count-other": "فورينت مجري",
    "symbol": "HUF",
    "symbol-alt-narrow": "Ft"
  },
  "IDR": {
    "displayName": "روبية إندونيسية",
    "displayName-count-zero": "روبية إندونيسية",
    "displayName-count-one": "روبية إندونيسية",
    "displayName-count-two": "روبية إندونيسية",
    "displayName-count-few": "روبية إندونيسية",
    "displayName-count-many": "روبية إندونيسية",
    "displayName-count-other": "روبية إندونيسية",
    "symbol": "IDR",
    "symbol-alt-narrow": "Rp"
  },
  "IEP": {
    "displayName": "جنيه إيرلندي",

```

```

    "symbol": "IEP"
  },
  "ILP": {
    "displayName": "جنیه اسرائیلی",
    "symbol": "ILP"
  },
  "ILS": {
    "displayName": "شیکل اسرائیلی جدید",
    "displayName-count-zero": "شیکل اسرائیلی جدید",
    "displayName-count-one": "شیکل اسرائیلی جدید",
    "displayName-count-two": "شیکل اسرائیلی جدید",
    "displayName-count-few": "شیکل اسرائیلی جدید",
    "displayName-count-many": "شیکل اسرائیلی جدید",
    "displayName-count-other": "شیکل اسرائیلی جدید",
    "symbol": "₪",
    "symbol-alt-narrow": "₪"
  },
  "INR": {
    "displayName": "روبیة هندی",
    "displayName-count-zero": "روبیة هندی",
    "displayName-count-one": "روبیة هندی",
    "displayName-count-two": "روبیة هندی",
    "displayName-count-few": "روبیة هندی",
    "displayName-count-many": "روبیة هندی",
    "displayName-count-other": "روبیة هندی",
    "symbol": "₹",
    "symbol-alt-narrow": "₹"
  },
  "IQD": {
    "displayName": "دينار عراقي",
    "displayName-count-zero": "دينار عراقي",
    "displayName-count-one": "دينار عراقي",
    "displayName-count-two": "دينار عراقي",
    "displayName-count-few": "دينار عراقي",
    "displayName-count-many": "دينار عراقي",
    "displayName-count-other": "دينار عراقي",
    "symbol": "د.ع."
  },
  "IRR": {
    "displayName": "ریال ایرانی",
    "displayName-count-zero": "ریال ایرانی",
    "displayName-count-one": "ریال ایرانی",
    "displayName-count-two": "ریال ایرانی",
    "displayName-count-few": "ریال ایرانی",
    "displayName-count-many": "ریال ایرانی",
    "displayName-count-other": "ریال ایرانی",
    "symbol": "ر.ا."
  },
  "ISK": {
    "displayName": "کرونه ایسلندی",
    "displayName-count-zero": "کرونه ایسلندی",
    "displayName-count-one": "کرونه ایسلندی",
    "displayName-count-two": "کرونه ایسلندی",
    "displayName-count-few": "کرونه ایسلندی",
    "displayName-count-many": "کرونه ایسلندی",
    "displayName-count-other": "کرونه ایسلندی",
    "symbol": "ISK",

```

```

    "symbol-alt-narrow": "kr"
  },
  "ITL": {
    "displayName": "ليرة إيطالية",
    "symbol": "ITL"
  },
  "JMD": {
    "displayName": "دولار جامايكي",
    "displayName-count-zero": "دولار جامايكي",
    "displayName-count-one": "دولار جامايكي",
    "displayName-count-two": "دولار جامايكي",
    "displayName-count-few": "دولار جامايكي",
    "displayName-count-many": "دولار جامايكي",
    "displayName-count-other": "دولار جامايكي",
    "symbol": "JMD",
    "symbol-alt-narrow": "JM$"
  },
  "JOD": {
    "displayName": "دينار أردني",
    "displayName-count-zero": "دينار أردني",
    "displayName-count-one": "دينار أردني",
    "displayName-count-two": "دينار أردني",
    "displayName-count-few": "دينار أردني",
    "displayName-count-many": "دينار أردني",
    "displayName-count-other": "دينار أردني",
    "symbol": "د.أ."
  },
  "JPY": {
    "displayName": "ين ياباني",
    "displayName-count-zero": "ين ياباني",
    "displayName-count-one": "ين ياباني",
    "displayName-count-two": "ين ياباني",
    "displayName-count-few": "ين ياباني",
    "displayName-count-many": "ين ياباني",
    "displayName-count-other": "ين ياباني",
    "symbol": "JP¥",
    "symbol-alt-narrow": "JP¥"
  },
  "KES": {
    "displayName": "شلن كيني",
    "displayName-count-zero": "شلن كيني",
    "displayName-count-one": "شلن كيني",
    "displayName-count-two": "شلن كيني",
    "displayName-count-few": "شلن كيني",
    "displayName-count-many": "شلن كيني",
    "displayName-count-other": "شلن كيني",
    "symbol": "KES"
  },
  "KGS": {
    "displayName": "سوم قيرغستاني",
    "displayName-count-zero": "سوم قيرغستاني",
    "displayName-count-one": "سوم قيرغستاني",
    "displayName-count-two": "سوم قيرغستاني",
    "displayName-count-few": "سوم قيرغستاني",
    "displayName-count-many": "سوم قيرغستاني",
    "displayName-count-other": "سوم قيرغستاني",
    "symbol": "KGS"
  }

```

```

    },
    "KHR": {
      "displayName": "ريال كمبودي",
      "displayName-count-zero": "ريال كمبودي",
      "displayName-count-one": "ريال كمبودي",
      "displayName-count-two": "ريال كمبودي",
      "displayName-count-few": "ريال كمبودي",
      "displayName-count-many": "ريال كمبودي",
      "displayName-count-other": "ريال كمبودي",
      "symbol": "KHR",
      "symbol-alt-narrow": "₭"
    },
    "KMF": {
      "displayName": "فرنك جزر القمر",
      "displayName-count-zero": "فرنك جزر القمر",
      "displayName-count-one": "فرنك جزر القمر",
      "displayName-count-two": "فرنك جزر القمر",
      "displayName-count-few": "فرنك جزر القمر",
      "displayName-count-many": "فرنك جزر القمر",
      "displayName-count-other": "فرنك جزر القمر",
      "symbol": "ف.ج.ق.",
      "symbol-alt-narrow": "CF"
    },
    "KPW": {
      "displayName": "وون كوريا الشمالية",
      "displayName-count-zero": "وون كوريا الشمالية",
      "displayName-count-one": "وون كوريا الشمالية",
      "displayName-count-two": "وون كوريا الشمالية",
      "displayName-count-few": "وون كوريا الشمالية",
      "displayName-count-many": "وون كوريا الشمالية",
      "displayName-count-other": "وون كوريا الشمالية",
      "symbol": "KPW",
      "symbol-alt-narrow": "₩"
    },
    "KRH": {
      "displayName": "KRH",
      "symbol": "KRH"
    },
    "KRO": {
      "displayName": "KRO",
      "symbol": "KRO"
    },
    "KRW": {
      "displayName": "وون كوريا الجنوبية",
      "displayName-count-zero": "وون كوريا الجنوبية",
      "displayName-count-one": "وون كوريا الجنوبية",
      "displayName-count-two": "وون كوريا الجنوبية",
      "displayName-count-few": "وون كوريا الجنوبية",
      "displayName-count-many": "وون كوريا الجنوبية",
      "displayName-count-other": "وون كوريا الجنوبية",
      "symbol": "₩",
      "symbol-alt-narrow": "₩"
    },
    "KWD": {
      "displayName": "دينار كويتي",
      "displayName-count-zero": "دينار كويتي",

```

```

    "displayName-count-one": "دينار كويتي",
    "displayName-count-two": "دينار كويتي",
    "displayName-count-few": "دينار كويتي",
    "displayName-count-many": "دينار كويتي",
    "displayName-count-other": "دينار كويتي",
    "symbol": "د.ك.",
  },
  "KYD": {
    "displayName": "دولار جزر كيمن",
    "displayName-count-zero": "دولار جزر كيمن",
    "displayName-count-one": "دولار جزر كيمن",
    "displayName-count-two": "دولار جزر كيمن",
    "displayName-count-few": "دولار جزر كيمن",
    "displayName-count-many": "دولار جزر كيمن",
    "displayName-count-other": "دولار جزر كيمن",
    "symbol": "KYD",
    "symbol-alt-narrow": "KY$"
  },
  "KZT": {
    "displayName": "تينغ كازاخستاني",
    "displayName-count-zero": "تينغ كازاخستاني",
    "displayName-count-one": "تينغ كازاخستاني",
    "displayName-count-two": "تينغ كازاخستاني",
    "displayName-count-few": "تينغ كازاخستاني",
    "displayName-count-many": "تينغ كازاخستاني",
    "displayName-count-other": "تينغ كازاخستاني",
    "symbol": "KZT",
    "symbol-alt-narrow": "Т"
  },
  "LAK": {
    "displayName": "كيب لاوسي",
    "displayName-count-zero": "كيب لاوسي",
    "displayName-count-one": "كيب لاوسي",
    "displayName-count-two": "كيب لاوسي",
    "displayName-count-few": "كيب لاوسي",
    "displayName-count-many": "كيب لاوسي",
    "displayName-count-other": "كيب لاوسي",
    "symbol": "LAK",
    "symbol-alt-narrow": "₭"
  },
  "LBP": {
    "displayName": "جنيه لبناني",
    "displayName-count-zero": "جنيه لبناني",
    "displayName-count-one": "جنيه لبناني",
    "displayName-count-two": "جنيه لبناني",
    "displayName-count-few": "جنيه لبناني",
    "displayName-count-many": "جنيه لبناني",
    "displayName-count-other": "جنيه لبناني",
    "symbol": "ل.ل.",
    "symbol-alt-narrow": "L£"
  },
  "LKR": {
    "displayName": "روبية سريلانكية",
    "displayName-count-zero": "روبية سريلانكية",
    "displayName-count-one": "روبية سريلانكية",
    "displayName-count-two": "روبية سريلانكية",
    "displayName-count-few": "روبية سريلانكية",

```



```

    "displayName-count-many": "روبية سريلانكية",
    "displayName-count-other": "روبية سريلانكية",
    "symbol": "LKR",
    "symbol-alt-narrow": "Rs"
  },
  "LRD": {
    "displayName": "دولار ليبيري",
    "displayName-count-zero": "دولار ليبيري",
    "displayName-count-one": "دولار ليبيري",
    "displayName-count-two": "دولاران ليبيريان",
    "displayName-count-few": "دولارات ليبيرية",
    "displayName-count-many": "دولارًا ليبيريًا",
    "displayName-count-other": "دولار ليبيري",
    "symbol": "LRD",
    "symbol-alt-narrow": "$"
  },
  "LSL": {
    "displayName": "لوتي ليسوتو",
    "symbol": "LSL"
  },
  "LTL": {
    "displayName": "ليتا ليتوانية",
    "displayName-count-zero": "ليتا ليتوانية",
    "displayName-count-one": "ليتا ليتوانية",
    "displayName-count-two": "ليتا ليتوانية",
    "displayName-count-few": "ليتا ليتوانية",
    "displayName-count-many": "ليتا ليتوانية",
    "displayName-count-other": "ليتا ليتوانية",
    "symbol": "LTL",
    "symbol-alt-narrow": "Lt"
  },
  "LTT": {
    "displayName": "تالوناس ليتواني",
    "symbol": "LTT"
  },
  "LUC": {
    "displayName": "فرنك لوكسمبرج قابل للتحويل",
    "symbol": "LUC"
  },
  "LUF": {
    "displayName": "فرنك لوكسمبرج",
    "symbol": "LUF"
  },
  "LUL": {
    "displayName": "فرنك لوكسمبرج المالي",
    "symbol": "LUL"
  },
  "LVL": {
    "displayName": "لاتس لاتفيا",
    "displayName-count-zero": "لاتس لاتفي",
    "displayName-count-one": "لاتس لاتفي",
    "displayName-count-two": "لاتس لاتفي",
    "displayName-count-few": "لاتس لاتفي",
    "displayName-count-many": "لاتس لاتفي",
    "displayName-count-other": "لاتس لاتفي",
    "symbol": "LVL",
    "symbol-alt-narrow": "Ls"
  }

```

```

    },
    "LVR": {
      "displayName": "روبل لاتفيا",
      "symbol": "LVR"
    },
    },
    "LYD": {
      "displayName": "دينار ليبي",
      "displayName-count-zero": "دينار ليبي",
      "displayName-count-one": "دينار ليبي",
      "displayName-count-two": "ديناران ليبيان",
      "displayName-count-few": "دينارات ليبية",
      "displayName-count-many": "دينارًا ليبيا",
      "displayName-count-other": "دينار ليبي",
      "symbol": "د.ل."
    },
    },
    "MAD": {
      "displayName": "درهم مغربي",
      "displayName-count-zero": "درهم مغربي",
      "displayName-count-one": "درهم مغربي",
      "displayName-count-two": "درهمان مغربيان",
      "displayName-count-few": "دراهم مغربية",
      "displayName-count-many": "درهمًا مغربيًا",
      "displayName-count-other": "درهم مغربي",
      "symbol": "د.م."
    },
    },
    "MAF": {
      "displayName": "فرنك مغربي",
      "symbol": "MAF"
    },
    },
    "MCF": {
      "displayName": "MCF",
      "symbol": "MCF"
    },
    },
    "MDC": {
      "displayName": "MDC",
      "symbol": "MDC"
    },
    },
    "MDL": {
      "displayName": "ليو مولدوفي",
      "displayName-count-zero": "ليو مولدوفي",
      "displayName-count-one": "ليو مولدوفي",
      "displayName-count-two": "ليو مولدوفي",
      "displayName-count-few": "ليو مولدوفي",
      "displayName-count-many": "ليو مولدوفي",
      "displayName-count-other": "ليو مولدوفي",
      "symbol": "MDL"
    },
    },
    "MGA": {
      "displayName": "أرياري مدغشقر",
      "displayName-count-zero": "أرياري مدغشقر",
      "displayName-count-one": "أرياري مدغشقر",
      "displayName-count-two": "أرياري مدغشقر",
      "displayName-count-few": "أرياري مدغشقر",
      "displayName-count-many": "أرياري مدغشقر",
      "displayName-count-other": "أرياري مدغشقر",
      "symbol": "MGA",
      "symbol-alt-narrow": "Ar"
    }
  }

```

```

    },
    "MGF": {
      "displayName": "فرنك مدغشقر",
      "symbol": "MGF"
    },
    "MKD": {
      "displayName": "دينار مقدوني",
      "displayName-count-zero": "دينار مقدوني",
      "displayName-count-one": "دينار مقدوني",
      "displayName-count-two": "ديناران مقدونيان",
      "displayName-count-few": "دينارات مقدونية",
      "displayName-count-many": "دينارًا مقدونيًا",
      "displayName-count-other": "دينار مقدوني",
      "symbol": "MKD"
    },
    "MKN": {
      "displayName": "MKN",
      "symbol": "MKN"
    },
    "MLF": {
      "displayName": "فرنك مالي",
      "symbol": "MLF"
    },
    "MMK": {
      "displayName": "كيات ميانمار",
      "displayName-count-zero": "كيات ميانمار",
      "displayName-count-one": "كيات ميانمار",
      "displayName-count-two": "كيات ميانمار",
      "displayName-count-few": "كيات ميانمار",
      "displayName-count-many": "كيات ميانمار",
      "displayName-count-other": "كيات ميانمار",
      "symbol": "MMK",
      "symbol-alt-narrow": "K"
    },
    "MNT": {
      "displayName": "توغروغ منغولي",
      "displayName-count-zero": "توغروغ منغولي",
      "displayName-count-one": "توغروغ منغولي",
      "displayName-count-two": "توغروغ منغولي",
      "displayName-count-few": "توغروغ منغولي",
      "displayName-count-many": "توغروغ منغولي",
      "displayName-count-other": "توغروغ منغولي",
      "symbol": "MNT",
      "symbol-alt-narrow": "₮"
    },
    "MOP": {
      "displayName": "باتاكا مكاوي",
      "displayName-count-zero": "باتاكا مكاوي",
      "displayName-count-one": "باتاكا مكاوي",
      "displayName-count-two": "باتاكا مكاوي",
      "displayName-count-few": "باتاكا مكاوي",
      "displayName-count-many": "باتاكا مكاوي",
      "displayName-count-other": "باتاكا مكاوي",
      "symbol": "MOP"
    },
    "MRO": {
      "displayName": "أوقية موريتانية",

```

```

    "displayName-count-zero": "أوقية موريتانية",
    "displayName-count-one": "أوقية موريتانية",
    "displayName-count-two": "أوقية موريتانية",
    "displayName-count-few": "أوقية موريتانية",
    "displayName-count-many": "أوقية موريتانية",
    "displayName-count-other": "أوقية موريتانية",
    "symbol": "أ.م.",
  },
  "MTL": {
    "displayName": "ليرة مالطية",
    "symbol": "MTL"
  },
  "MTP": {
    "displayName": "جنيه مالطي",
    "symbol": "MTP"
  },
  "MUR": {
    "displayName": "روبية موريشيوسية",
    "displayName-count-zero": "روبية موريشيوسية",
    "displayName-count-one": "روبية موريشيوسية",
    "displayName-count-two": "روبية موريشيوسية",
    "displayName-count-few": "روبية موريشيوسية",
    "displayName-count-many": "روبية موريشيوسية",
    "displayName-count-other": "روبية موريشيوسية",
    "symbol": "MUR",
    "symbol-alt-narrow": "Rs"
  },
  "MVR": {
    "displayName": "روفيه جزر المالديف",
    "displayName-count-zero": "روفيه جزر المالديف",
    "displayName-count-one": "روفيه جزر المالديف",
    "displayName-count-two": "روفيه جزر المالديف",
    "displayName-count-few": "روفيه جزر المالديف",
    "displayName-count-many": "روفيه جزر المالديف",
    "displayName-count-other": "روفيه جزر المالديف",
    "symbol": "MVR"
  },
  "MWK": {
    "displayName": "كواشا مالاوي",
    "displayName-count-zero": "كواشا مالاوي",
    "displayName-count-one": "كواشا مالاوي",
    "displayName-count-two": "كواشا مالاوي",
    "displayName-count-few": "كواشا مالاوي",
    "displayName-count-many": "كواشا مالاوي",
    "displayName-count-other": "كواشا مالاوي",
    "symbol": "MWK"
  },
  "MXN": {
    "displayName": "بيزو مكسيكي",
    "displayName-count-zero": "بيزو مكسيكي",
    "displayName-count-one": "بيزو مكسيكي",
    "displayName-count-two": "بيزو مكسيكي",
    "displayName-count-few": "بيزو مكسيكي",
    "displayName-count-many": "بيزو مكسيكي",
    "displayName-count-other": "بيزو مكسيكي",
    "symbol": "MX$",
    "symbol-alt-narrow": "MX$"
  }

```

```

},
"MXP": {
  "displayName": "1992-1861 - بيزو فضي مكسيكي",
  "symbol": "MXP"
},
"MXV": {
  "displayName": "MXV",
  "symbol": "MXV"
},
"MYR": {
  "displayName": "رينغيت ماليزي",
  "displayName-count-zero": "رينغيت ماليزي",
  "displayName-count-one": "رينغيت ماليزي",
  "displayName-count-two": "رينغيت ماليزي",
  "displayName-count-few": "رينغيت ماليزي",
  "displayName-count-many": "رينغيت ماليزي",
  "displayName-count-other": "رينغيت ماليزي",
  "symbol": "MYR",
  "symbol-alt-narrow": "RM"
},
"MZE": {
  "displayName": "اسكود موزمبيقي",
  "symbol": "MZE"
},
"MZM": {
  "displayName": "MZM",
  "symbol": "MZM"
},
"MZN": {
  "displayName": "متكال موزمبيقي",
  "displayName-count-zero": "متكال موزمبيقي",
  "displayName-count-one": "متكال موزمبيقي",
  "displayName-count-two": "متكال موزمبيقي",
  "displayName-count-few": "متكال موزمبيقي",
  "displayName-count-many": "متكال موزمبيقي",
  "displayName-count-other": "متكال موزمبيقي",
  "symbol": "MZN"
},
"NAD": {
  "displayName": "دولار ناميبي",
  "displayName-count-zero": "دولار ناميبي",
  "displayName-count-one": "دولار ناميبي",
  "displayName-count-two": "دولار ناميبي",
  "displayName-count-few": "دولار ناميبي",
  "displayName-count-many": "دولار ناميبي",
  "displayName-count-other": "دولار ناميبي",
  "symbol": "NAD",
  "symbol-alt-narrow": "$"
},
"NGN": {
  "displayName": "نايرا نيجيري",
  "displayName-count-zero": "نايرا نيجيري",
  "displayName-count-one": "نايرا نيجيري",
  "displayName-count-two": "نايرا نيجيري",
  "displayName-count-few": "نايرا نيجيري",
  "displayName-count-many": "نايرا نيجيري",
  "displayName-count-other": "نايرا نيجيري",

```

```

    "symbol": "NGN",
    "symbol-alt-narrow": "₦"
  },
  "NIC": {
    "displayName": "كوردوبه نيكاراغوا",
    "symbol": "NIC"
  },
  "NIO": {
    "displayName": "قرطبة نيكاراغوا",
    "displayName-count-zero": "قرطبة نيكاراغوا",
    "displayName-count-one": "قرطبة نيكاراغوا",
    "displayName-count-two": "قرطبة نيكاراغوا",
    "displayName-count-few": "قرطبة نيكاراغوا",
    "displayName-count-many": "قرطبة نيكاراغوا",
    "displayName-count-other": "قرطبة نيكاراغوا",
    "symbol": "NIO",
    "symbol-alt-narrow": "C$"
  },
  "NLG": {
    "displayName": "جلدر هولندي",
    "symbol": "NLG"
  },
  "NOK": {
    "displayName": "كرونه نرويجية",
    "displayName-count-zero": "كرونه نرويجية",
    "displayName-count-one": "كرونه نرويجية",
    "displayName-count-two": "كرونه نرويجية",
    "displayName-count-few": "كرونه نرويجية",
    "displayName-count-many": "كرونه نرويجية",
    "displayName-count-other": "كرونه نرويجية",
    "symbol": "NOK",
    "symbol-alt-narrow": "kr"
  },
  "NPR": {
    "displayName": "روبية نيبالي",
    "displayName-count-zero": "روبية نيبالي",
    "displayName-count-one": "روبية نيبالي",
    "displayName-count-two": "روبية نيبالي",
    "displayName-count-few": "روبية نيبالي",
    "displayName-count-many": "روبية نيبالي",
    "displayName-count-other": "روبية نيبالي",
    "symbol": "NPR",
    "symbol-alt-narrow": "Rs"
  },
  "NZD": {
    "displayName": "دولار نيوزيلندي",
    "displayName-count-zero": "دولار نيوزيلندي",
    "displayName-count-one": "دولار نيوزيلندي",
    "displayName-count-two": "دولار نيوزيلندي",
    "displayName-count-few": "دولار نيوزيلندي",
    "displayName-count-many": "دولار نيوزيلندي",
    "displayName-count-other": "دولار نيوزيلندي",
    "symbol": "NZ$",
    "symbol-alt-narrow": "NZ$"
  },
  "OMR": {
    "displayName": "ريال عماني",

```

```

    "displayName-count-zero": "٥٥ عماري",
    "displayName-count-one": "٥٥ عماري",
    "displayName-count-two": "٥٥ عماري",
    "displayName-count-few": "٥٥ عماري",
    "displayName-count-many": "٥٥ عماري",
    "displayName-count-other": "٥٥ عماري",
    "symbol": "ع.ع."
  },
  "PAB": {
    "displayName": "بالبوا بنمي",
    "displayName-count-zero": "بالبوا بنمي",
    "displayName-count-one": "بالبوا بنمي",
    "displayName-count-two": "بالبوا بنمي",
    "displayName-count-few": "بالبوا بنمي",
    "displayName-count-many": "بالبوا بنمي",
    "displayName-count-other": "بالبوا بنمي",
    "symbol": "PAB"
  },
  "PEI": {
    "displayName": "PEI",
    "symbol": "PEI"
  },
  "PEN": {
    "displayName": "سول جديد البيرو",
    "displayName-count-zero": "سول جديد البيرو",
    "displayName-count-one": "سول جديد البيرو",
    "displayName-count-two": "سول جديد البيرو",
    "displayName-count-few": "سول جديد البيرو",
    "displayName-count-many": "سول جديد البيرو",
    "displayName-count-other": "سول جديد البيرو",
    "symbol": "PEN"
  },
  "PES": {
    "displayName": "PES",
    "symbol": "PES"
  },
  "PGK": {
    "displayName": "كيننا بابوا غينيا الجديدة",
    "displayName-count-zero": "كيننا بابوا غينيا الجديدة",
    "displayName-count-one": "كيننا بابوا غينيا الجديدة",
    "displayName-count-two": "كيننا بابوا غينيا الجديدة",
    "displayName-count-few": "كيننا بابوا غينيا الجديدة",
    "displayName-count-many": "كيننا بابوا غينيا الجديدة",
    "displayName-count-other": "كيننا بابوا غينيا الجديدة",
    "symbol": "PGK"
  },
  "PHP": {
    "displayName": "بيزو فلبيني",
    "displayName-count-zero": "بيزو فلبيني",
    "displayName-count-one": "بيزو فلبيني",
    "displayName-count-two": "بيزو فلبيني",
    "displayName-count-few": "بيزو فلبيني",
    "displayName-count-many": "بيزو فلبيني",
    "displayName-count-other": "بيزو فلبيني",
    "symbol": "PHP",
    "symbol-alt-narrow": "₱"
  },
},

```

```

"PKR": {
  "displayName": "روبية باستاني",
  "displayName-count-zero": "روبية باستاني",
  "displayName-count-one": "روبية باستاني",
  "displayName-count-two": "روبية باستاني",
  "displayName-count-few": "روبية باستاني",
  "displayName-count-many": "روبية باستاني",
  "displayName-count-other": "روبية باستاني",
  "symbol": "PKR",
  "symbol-alt-narrow": "Rs"
},
"PLN": {
  "displayName": "زلوتي بولندي",
  "displayName-count-zero": "زلوتي بولندي",
  "displayName-count-one": "زلوتي بولندي",
  "displayName-count-two": "زلوتي بولندي",
  "displayName-count-few": "زلوتي بولندي",
  "displayName-count-many": "زلوتي بولندي",
  "displayName-count-other": "زلوتي بولندي",
  "symbol": "PLN",
  "symbol-alt-narrow": "zł"
},
"PLZ": {
  "displayName": "زلوتي بولندي - 1950-1995",
  "symbol": "PLZ"
},
"PTE": {
  "displayName": "اسكود برتغالي",
  "symbol": "PTE"
},
"PYG": {
  "displayName": "جواراني باراجواي",
  "displayName-count-zero": "جواراني باراجواي",
  "displayName-count-one": "جواراني باراجواي",
  "displayName-count-two": "جواراني باراجواي",
  "displayName-count-few": "جواراني باراجواي",
  "displayName-count-many": "جواراني باراجواي",
  "displayName-count-other": "جواراني باراجواي",
  "symbol": "PYG",
  "symbol-alt-narrow": "₲"
},
"QAR": {
  "displayName": "وقطري",
  "displayName-count-zero": "وقطري",
  "displayName-count-one": "وقطري",
  "displayName-count-two": "وقطري",
  "displayName-count-few": "وقطري",
  "displayName-count-many": "وقطري",
  "displayName-count-other": "وقطري",
  "symbol": "ر.ق."
},
"RHD": {
  "displayName": "دولار روديسي",
  "symbol": "RHD"
},
"ROL": {
  "displayName": "ليو روماني قديم",

```



```

    "symbol": "ROL"
  },
  "RON": {
    "displayName": "ليو روماني",
    "displayName-count-zero": "ليو روماني",
    "displayName-count-one": "ليو روماني",
    "displayName-count-two": "ليو روماني",
    "displayName-count-few": "ليو روماني",
    "displayName-count-many": "ليو روماني",
    "displayName-count-other": "ليو روماني",
    "symbol": "RON",
    "symbol-alt-narrow": "lei"
  },
  "RSD": {
    "displayName": "دينار صربي",
    "displayName-count-zero": "دينار صربي",
    "displayName-count-one": "دينار صربي",
    "displayName-count-two": "ديناران صربيان",
    "displayName-count-few": "دينارات صربية",
    "displayName-count-many": "دينارًا صربيًا",
    "displayName-count-other": "دينار صربي",
    "symbol": "RSD"
  },
  "RUB": {
    "displayName": "روبل روسي",
    "displayName-count-zero": "روبل روسي",
    "displayName-count-one": "روبل روسي",
    "displayName-count-two": "روبل روسي",
    "displayName-count-few": "روبل روسي",
    "displayName-count-many": "روبل روسي",
    "displayName-count-other": "روبل روسي",
    "symbol": "RUB",
    "symbol-alt-narrow": "₽"
  },
  "RUR": {
    "displayName": "روبل روسي - 1998-1991",
    "symbol": "RUR",
    "symbol-alt-narrow": "p."
  },
  "RWF": {
    "displayName": "فرنك رواندي",
    "displayName-count-zero": "فرنك رواندي",
    "displayName-count-one": "فرنك رواندي",
    "displayName-count-two": "فرنك رواندي",
    "displayName-count-few": "فرنك رواندي",
    "displayName-count-many": "فرنك رواندي",
    "displayName-count-other": "فرنك رواندي",
    "symbol": "RWF",
    "symbol-alt-narrow": "RF"
  },
  "SAR": {
    "displayName": "ول سعودي",
    "displayName-count-zero": "ول سعودي",
    "displayName-count-one": "ول سعودي",
    "displayName-count-two": "ول سعودي",
    "displayName-count-few": "ول سعودي",
    "displayName-count-many": "ول سعودي",

```

```

    "displayName-count-other": "ريال سعودي",
    "symbol": "ر.س."
  },
  "SBD": {
    "displayName": "دولار جزر سليمان",
    "displayName-count-zero": "دولار جزر سليمان",
    "displayName-count-one": "دولار جزر سليمان",
    "displayName-count-two": "دولار جزر سليمان",
    "displayName-count-few": "دولار جزر سليمان",
    "displayName-count-many": "دولار جزر سليمان",
    "displayName-count-other": "دولار جزر سليمان",
    "symbol": "SBD",
    "symbol-alt-narrow": "SB$"
  },
  "SCR": {
    "displayName": "روبية سيشيلية",
    "displayName-count-zero": "روبية سيشيلية",
    "displayName-count-one": "روبية سيشيلية",
    "displayName-count-two": "روبية سيشيلية",
    "displayName-count-few": "روبية سيشيلية",
    "displayName-count-many": "روبية سيشيلية",
    "displayName-count-other": "روبية سيشيلية",
    "symbol": "SCR"
  },
  "SDD": {
    "displayName": "دينار سوداني",
    "symbol": "د.س."
  },
  "SDG": {
    "displayName": "جنيه سوداني",
    "displayName-count-zero": "جنيه سوداني",
    "displayName-count-one": "جنيه سوداني",
    "displayName-count-two": "جنيه سوداني",
    "displayName-count-few": "جنيهات سودانية",
    "displayName-count-many": "جنيهًا سودانيًا",
    "displayName-count-other": "جنيه سوداني",
    "symbol": "ج.س."
  },
  "SDP": {
    "displayName": "جنيه سوداني قديم",
    "symbol": "SDP"
  },
  "SEK": {
    "displayName": "كرونة سويدية",
    "displayName-count-zero": "كرونة سويدية",
    "displayName-count-one": "كرونة سويدية",
    "displayName-count-two": "كرونة سويدية",
    "displayName-count-few": "كرونة سويدية",
    "displayName-count-many": "كرونة سويدية",
    "displayName-count-other": "كرونة سويدية",
    "symbol": "SEK",
    "symbol-alt-narrow": "kr"
  },
  "SGD": {
    "displayName": "دولار سنغافوري",
    "displayName-count-zero": "دولار سنغافوري",
    "displayName-count-one": "دولار سنغافوري",

```

```

    "displayName-count-two": "دولار سنغافوري",
    "displayName-count-few": "دولار سنغافوري",
    "displayName-count-many": "دولار سنغافوري",
    "displayName-count-other": "دولار سنغافوري",
    "symbol": "SGD",
    "symbol-alt-narrow": "$",
  },
  "SHP": {
    "displayName": "جنيه سانت هيلين",
    "displayName-count-zero": "جنيه سانت هيلين",
    "displayName-count-one": "جنيه سانت هيلين",
    "displayName-count-two": "جنيه سانت هيلين",
    "displayName-count-few": "جنيه سانت هيلين",
    "displayName-count-many": "جنيه سانت هيلين",
    "displayName-count-other": "جنيه سانت هيلين",
    "symbol": "SHP",
    "symbol-alt-narrow": "£",
  },
  "SIT": {
    "displayName": "تولار سلوفيني",
    "symbol": "SIT",
  },
  "SKK": {
    "displayName": "كرونه سلوفاكية",
    "symbol": "SKK",
  },
  "SLL": {
    "displayName": "ليون سيراليوني",
    "displayName-count-zero": "ليون سيراليوني",
    "displayName-count-one": "ليون سيراليوني",
    "displayName-count-two": "ليون سيراليوني",
    "displayName-count-few": "ليون سيراليوني",
    "displayName-count-many": "ليون سيراليوني",
    "displayName-count-other": "ليون سيراليوني",
    "symbol": "SLL",
  },
  "SOS": {
    "displayName": "شلن صومالي",
    "displayName-count-zero": "شلن صومالي",
    "displayName-count-one": "شلن صومالي",
    "displayName-count-two": "شلن صومالي",
    "displayName-count-few": "شلن صومالي",
    "displayName-count-many": "شلن صومالي",
    "displayName-count-other": "شلن صومالي",
    "symbol": "SOS",
  },
  "SRD": {
    "displayName": "دولار سورينامي",
    "displayName-count-zero": "دولار سورينامي",
    "displayName-count-one": "دولار سورينامي",
    "displayName-count-two": "دولار سورينامي",
    "displayName-count-few": "دولار سورينامي",
    "displayName-count-many": "دولار سورينامي",
    "displayName-count-other": "دولار سورينامي",
    "symbol": "SRD",
    "symbol-alt-narrow": "SR$",
  },
},

```

```

"SRG": {
  "displayName": "جلدر سورينامي",
  "symbol": "SRG"
},
"SSP": {
  "displayName": "جنيه جنوب السودان",
  "displayName-count-zero": "جنيه جنوب السودان",
  "displayName-count-one": "جنيه جنوب السودان",
  "displayName-count-two": "جنيهان جنوب السودان",
  "displayName-count-few": "جنيهات جنوب السودان",
  "displayName-count-many": "جنيهًا جنوب السودان",
  "displayName-count-other": "جنيه جنوب السودان",
  "symbol": "SSP",
  "symbol-alt-narrow": "£"
},
"STD": {
  "displayName": "دوبرا ساو تومي وبرينسيبي",
  "displayName-count-zero": "دوبرا ساو تومي وبرينسيبي",
  "displayName-count-one": "دوبرا ساو تومي وبرينسيبي",
  "displayName-count-two": "دوبرا ساو تومي وبرينسيبي",
  "displayName-count-few": "دوبرا ساو تومي وبرينسيبي",
  "displayName-count-many": "دوبرا ساو تومي وبرينسيبي",
  "displayName-count-other": "دوبرا ساو تومي وبرينسيبي",
  "symbol": "STD",
  "symbol-alt-narrow": "Db"
},
"SUR": {
  "displayName": "روبل سوفيتي",
  "symbol": "SUR"
},
"SVC": {
  "displayName": "كولون سلفادوري",
  "symbol": "SVC"
},
"SYP": {
  "displayName": "ليرة سورية",
  "displayName-count-zero": "ليرة سورية",
  "displayName-count-one": "ليرة سورية",
  "displayName-count-two": "ليرة سورية",
  "displayName-count-few": "ليرة سورية",
  "displayName-count-many": "ليرة سورية",
  "displayName-count-other": "ليرة سورية",
  "symbol": "ل.س.",
  "symbol-alt-narrow": "£"
},
"SZL": {
  "displayName": "ليلانجيني سوازيلندي",
  "displayName-count-zero": "ليلانجيني سوازيلندي",
  "displayName-count-one": "ليلانجيني سوازيلندي",
  "displayName-count-two": "ليلانجيني سوازيلندي",
  "displayName-count-few": "ليلانجيني سوازيلندي",
  "displayName-count-many": "ليلانجيني سوازيلندي",
  "displayName-count-other": "ليلانجيني سوازيلندي",
  "symbol": "SZL"
},
"THB": {
  "displayName": "باخت تايلاندي",

```

```

    "displayName-count-zero": "باخت تایلاندي",
    "displayName-count-one": "باخت تایلاندي",
    "displayName-count-two": "باخت تایلاندي",
    "displayName-count-few": "باخت تایلاندي",
    "displayName-count-many": "باخت تایلاندي",
    "displayName-count-other": "باخت تایلاندي",
    "symbol": "฿",
    "symbol-alt-narrow": "฿"
  },
  "TJR": {
    "displayName": "روبل طاجیکستانی",
    "symbol": "TJR"
  },
  "TJS": {
    "displayName": "سومونی طاجیکستانی",
    "displayName-count-zero": "سومونی طاجیکستانی",
    "displayName-count-one": "سومونی طاجیکستانی",
    "displayName-count-two": "سومونی طاجیکستانی",
    "displayName-count-few": "سومونی طاجیکستانی",
    "displayName-count-many": "سومونی طاجیکستانی",
    "displayName-count-other": "سومونی طاجیکستانی",
    "symbol": "TJS"
  },
  "TMM": {
    "displayName": "مانات ترکمنستانی",
    "symbol": "TMM"
  },
  "TMT": {
    "displayName": "مانات ترکمانستان",
    "displayName-count-zero": "مانات ترکمانستان",
    "displayName-count-one": "مانات ترکمانستان",
    "displayName-count-two": "مانات ترکمانستان",
    "displayName-count-few": "مانات ترکمانستان",
    "displayName-count-many": "مانات ترکمانستان",
    "displayName-count-other": "مانات ترکمانستان",
    "symbol": "TMT"
  },
  "TND": {
    "displayName": "دینار تونس",
    "displayName-count-zero": "دینار تونس",
    "displayName-count-one": "دینار تونس",
    "displayName-count-two": "دیناران تونس",
    "displayName-count-few": "دینارات تونس",
    "displayName-count-many": "دینارًا تونس",
    "displayName-count-other": "دینار تونس",
    "symbol": "د.ت."
  },
  "TOP": {
    "displayName": "بانغا تونگا",
    "displayName-count-zero": "بانغا تونگا",
    "displayName-count-one": "بانغا تونگا",
    "displayName-count-two": "بانغا تونگا",
    "displayName-count-few": "بانغا تونگا",
    "displayName-count-many": "بانغا تونگا",
    "displayName-count-other": "بانغا تونگا",
    "symbol": "TOP",
    "symbol-alt-narrow": "T$"
  }

```

```

    },
    "TPE": {
      "displayName": "اسكود تيموري",
      "symbol": "TPE"
    },
    "TRL": {
      "displayName": "ليرة تركي",
      "symbol": "TRL"
    },
    "TRY": {
      "displayName": "ليرة تركية",
      "displayName-count-zero": "ليرة تركية",
      "displayName-count-one": "ليرة تركية",
      "displayName-count-two": "ليرة تركية",
      "displayName-count-few": "ليرة تركية",
      "displayName-count-many": "ليرة تركية",
      "displayName-count-other": "ليرة تركية",
      "symbol": "TRY",
      "symbol-alt-narrow": "₺",
      "symbol-alt-variant": "TL"
    },
    "TTD": {
      "displayName": "دولار ترينداد وتوباغو",
      "displayName-count-zero": "دولار ترينداد وتوباغو",
      "displayName-count-one": "دولار ترينداد وتوباغو",
      "displayName-count-two": "دولار ترينداد وتوباغو",
      "displayName-count-few": "دولار ترينداد وتوباغو",
      "displayName-count-many": "دولار ترينداد وتوباغو",
      "displayName-count-other": "دولار ترينداد وتوباغو",
      "symbol": "TTD",
      "symbol-alt-narrow": "TT$"
    },
    "TWD": {
      "displayName": "دولار تايواني",
      "displayName-count-zero": "دولار تايواني",
      "displayName-count-one": "دولار تايواني",
      "displayName-count-two": "دولار تايواني",
      "displayName-count-few": "دولار تايواني",
      "displayName-count-many": "دولار تايواني",
      "displayName-count-other": "دولار تايواني",
      "symbol": "NT$",
      "symbol-alt-narrow": "NT$"
    },
    "TZS": {
      "displayName": "شلن تنزاني",
      "displayName-count-zero": "شلن تنزاني",
      "displayName-count-one": "شلن تنزاني",
      "displayName-count-two": "شلن تنزاني",
      "displayName-count-few": "شلن تنزاني",
      "displayName-count-many": "شلن تنزاني",
      "displayName-count-other": "شلن تنزاني",
      "symbol": "TZS"
    },
    "UAH": {
      "displayName": "هريفنيا أوكراني",
      "displayName-count-zero": "هريفنيا أوكراني",
      "displayName-count-one": "هريفنيا أوكراني",

```

```

    "displayName-count-two": "هريفنيا أوكراني",
    "displayName-count-few": "هريفنيا أوكراني",
    "displayName-count-many": "هريفنيا أوكراني",
    "displayName-count-other": "هريفنيا أوكراني",
    "symbol": "UAH",
    "symbol-alt-narrow": "₴"
  },
  "UAK": {
    "displayName": "UAK",
    "symbol": "UAK"
  },
  "UGS": {
    "displayName": "1987-1966 - شلن أوغندي",
    "symbol": "UGS"
  },
  "UGX": {
    "displayName": "شلن أوغندي",
    "displayName-count-zero": "شلن أوغندي",
    "displayName-count-one": "شلن أوغندي",
    "displayName-count-two": "شلن أوغندي",
    "displayName-count-few": "شلن أوغندي",
    "displayName-count-many": "شلن أوغندي",
    "displayName-count-other": "شلن أوغندي",
    "symbol": "UGX"
  },
  "USD": {
    "displayName": "دولار أمريكي",
    "displayName-count-zero": "دولار أمريكي",
    "displayName-count-one": "دولار أمريكي",
    "displayName-count-two": "دولار أمريكي",
    "displayName-count-few": "دولار أمريكي",
    "displayName-count-many": "دولار أمريكي",
    "displayName-count-other": "دولار أمريكي",
    "symbol": "US$",
    "symbol-alt-narrow": "US$"
  },
  "USN": {
    "displayName": "دولار أمريكي (اليوم التالي)",
    "symbol": "USN"
  },
  "USS": {
    "displayName": "دولار أمريكي (نفس اليوم)",
    "symbol": "USS"
  },
  "UYI": {
    "displayName": "UYI",
    "symbol": "UYI"
  },
  "UYU": {
    "displayName": "بيزو أوروجواي",
    "symbol": "UYU"
  },
  "UYU": {
    "displayName": "بيزو أوروجواي",
    "displayName-count-zero": "بيزو أوروجواي",
    "displayName-count-one": "بيزو أوروجواي",
    "displayName-count-two": "بيزو أوروجواي",
  }

```

```

    "displayName-count-few": "بىزو اوروغوڭاي",
    "displayName-count-many": "بىزو اوروغوڭاي",
    "displayName-count-other": "بىزو اوروغوڭاي",
    "symbol": "UYU",
    "symbol-alt-narrow": "UY$"
  },
  "UZS": {
    "displayName": "سوم اۆزبىكستاني",
    "displayName-count-zero": "سوم اۆزبىكستاني",
    "displayName-count-one": "سوم اۆزبىكستاني",
    "displayName-count-two": "سوم اۆزبىكستاني",
    "displayName-count-few": "سوم اۆزبىكستاني",
    "displayName-count-many": "سوم اۆزبىكستاني",
    "displayName-count-other": "سوم اۆزبىكستاني",
    "symbol": "UZS"
  },
  "VEB": {
    "displayName": "بوليفار فنزوئيلي - 2008-1871",
    "symbol": "VEB"
  },
  "VEF": {
    "displayName": "بوليفار فنزوئيلي",
    "displayName-count-zero": "بوليفار فنزوئيلي",
    "displayName-count-one": "بوليفار فنزوئيلي",
    "displayName-count-two": "بوليفار فنزوئيلي",
    "displayName-count-few": "بوليفار فنزوئيلي",
    "displayName-count-many": "بوليفار فنزوئيلي",
    "displayName-count-other": "بوليفار فنزوئيلي",
    "symbol": "VEF",
    "symbol-alt-narrow": "Bs"
  },
  "VND": {
    "displayName": "دونج فيتنامي",
    "displayName-count-zero": "دونج فيتنامي",
    "displayName-count-one": "دونج فيتنامي",
    "displayName-count-two": "دونج فيتنامي",
    "displayName-count-few": "دونج فيتنامي",
    "displayName-count-many": "دونج فيتنامي",
    "displayName-count-other": "دونج فيتنامي",
    "symbol": "₫",
    "symbol-alt-narrow": "₫"
  },
  "VNN": {
    "displayName": "VNN",
    "symbol": "VNN"
  },
  "VUV": {
    "displayName": "فاتو فانواتو",
    "displayName-count-zero": "فاتو فانواتو",
    "displayName-count-one": "فاتو فانواتو",
    "displayName-count-two": "فاتو فانواتو",
    "displayName-count-few": "فاتو فانواتو",
    "displayName-count-many": "فاتو فانواتو",
    "displayName-count-other": "فاتو فانواتو",
    "symbol": "VUV"
  },
  "WST": {

```



```

    "displayName": "تالا ساموا",
    "displayName-count-zero": "تالا ساموا",
    "displayName-count-one": "تالا ساموا",
    "displayName-count-two": "تالا ساموا",
    "displayName-count-few": "تالا ساموا",
    "displayName-count-many": "تالا ساموا",
    "displayName-count-other": "تالا ساموا",
    "symbol": "WST"
  },
  "XAF": {
    "displayName": "فرنك وسط أفريقي",
    "displayName-count-zero": "فرنك وسط أفريقي",
    "displayName-count-one": "فرنك وسط أفريقي",
    "displayName-count-two": "فرنك وسط أفريقي",
    "displayName-count-few": "فرنك وسط أفريقي",
    "displayName-count-many": "فرنك وسط أفريقي",
    "displayName-count-other": "فرنك وسط أفريقي",
    "symbol": "FCFA"
  },
  "XAG": {
    "displayName": "فضة",
    "symbol": "XAG"
  },
  "XAU": {
    "displayName": "ذهب",
    "symbol": "XAU"
  },
  "XBA": {
    "displayName": "الوحدة الأوروبية المركبة",
    "symbol": "XBA"
  },
  "XBB": {
    "displayName": "الوحدة المالية الأوروبية",
    "symbol": "XBB"
  },
  "XBC": {
    "displayName": "الوحدة الحسابية الأوروبية",
    "symbol": "XBC"
  },
  "XBD": {
    "displayName": "وحدة الحساب الأوروبية (XBD)",
    "symbol": "XBD"
  },
  "XCD": {
    "displayName": "دولار شرق الكاريبي",
    "displayName-count-zero": "دولار شرق الكاريبي",
    "displayName-count-one": "دولار شرق الكاريبي",
    "displayName-count-two": "دولار شرق الكاريبي",
    "displayName-count-few": "دولار شرق الكاريبي",
    "displayName-count-many": "دولار شرق الكاريبي",
    "displayName-count-other": "دولار شرق الكاريبي",
    "symbol": "EC$",
    "symbol-alt-narrow": "$"
  },
  "XDR": {
    "displayName": "حقوق السحب الخاصة",
    "symbol": "XDR"
  }

```

```

    },
    "XEU": {
      "displayName": "وحدة النقد الأوروبية",
      "symbol": "XEU"
    },
    },
    "XFO": {
      "displayName": "فرنك فرنسي ذهبي",
      "symbol": "XFO"
    },
    },
    "XFU": {
      "displayName": "فرنك فرنسي (UIC)",
      "symbol": "XFU"
    },
    },
    "XOF": {
      "displayName": "فرنك غرب أفريقي",
      "displayName-count-zero": "فرنك غرب أفريقي",
      "displayName-count-one": "فرنك غرب أفريقي",
      "displayName-count-two": "فرنك غرب أفريقي",
      "displayName-count-few": "فرنك غرب أفريقي",
      "displayName-count-many": "فرنك غرب أفريقي",
      "displayName-count-other": "فرنك غرب أفريقي",
      "symbol": "CFA"
    },
    },
    "XPD": {
      "displayName": "بالاديوم",
      "symbol": "XPD"
    },
    },
    "XPF": {
      "displayName": "فرنك سي إف بي",
      "displayName-count-zero": "فرنك سي إف بي",
      "displayName-count-one": "فرنك سي إف بي",
      "displayName-count-two": "فرنك سي إف بي",
      "displayName-count-few": "فرنك سي إف بي",
      "displayName-count-many": "فرنك سي إف بي",
      "displayName-count-other": "فرنك سي إف بي",
      "symbol": "CFPF"
    },
    },
    "XPT": {
      "displayName": "البلاتين",
      "symbol": "XPT"
    },
    },
    "XRE": {
      "displayName": "XRE",
      "symbol": "XRE"
    },
    },
    "XSU": {
      "displayName": "XSU",
      "symbol": "XSU"
    },
    },
    "XTS": {
      "displayName": "كود اختبار العملة",
      "symbol": "XTS"
    },
    },
    "XUA": {
      "displayName": "XUA",
      "symbol": "XUA"
    },
    },
  },

```

```

"XXX": {
  "displayName": "عملة غير معروفة",
  "displayName-count-zero": "(عملة غير معروفة)",
  "displayName-count-one": "(عملة غير معروفة)",
  "displayName-count-two": "(عملة غير معروفة)",
  "displayName-count-few": "(عملة غير معروفة)",
  "displayName-count-many": "(عملة غير معروفة)",
  "displayName-count-other": "(عملة غير معروفة)",
  "symbol": "***"
},
"YDD": {
  "displayName": "دينار يمني",
  "symbol": "YDD"
},
"YER": {
  "displayName": "ول يمني",
  "displayName-count-zero": "ول يمني",
  "displayName-count-one": "ول يمني",
  "displayName-count-two": "ول يمني",
  "displayName-count-few": "ول يمني",
  "displayName-count-many": "ول يمني",
  "displayName-count-other": "ول يمني",
  "symbol": "ر.ي."
},
"YUD": {
  "displayName": "دينار يوغسلافي",
  "symbol": "YUD"
},
"YUM": {
  "displayName": "YUM",
  "symbol": "YUM"
},
"YUN": {
  "displayName": "دينار يوغسلافي قابل للتحويل",
  "symbol": "YUN"
},
"YUR": {
  "displayName": "YUR",
  "symbol": "YUR"
},
"ZAL": {
  "displayName": "راند جنوب أفريقيا -مالي",
  "symbol": "ZAL"
},
"ZAR": {
  "displayName": "راند جنوب أفريقيا",
  "displayName-count-zero": "راند جنوب أفريقيا",
  "displayName-count-one": "راند جنوب أفريقيا",
  "displayName-count-two": "راند جنوب أفريقيا",
  "displayName-count-few": "راند جنوب أفريقيا",
  "displayName-count-many": "راند جنوب أفريقيا",
  "displayName-count-other": "راند جنوب أفريقيا",
  "symbol": "ZAR",
  "symbol-alt-narrow": "R"
},
"ZMK": {
  "displayName": "كواشا زامبي - 1968-2012",

```

```

    "displayName-count-zero": "2012-1968 - کواشا زامبی",
    "displayName-count-one": "2012-1968 - کواشا زامبی",
    "displayName-count-two": "2012-1968 - کواشا زامبی",
    "displayName-count-few": "2012-1968 - کواشا زامبی",
    "displayName-count-many": "2012-1968 - کواشا زامبی",
    "displayName-count-other": "2012-1968 - کواشا زامبی",
    "symbol": "ZMK"
  },
  "ZMW": {
    "displayName": "کواشا زامبی",
    "displayName-count-zero": "کواشا زامبی",
    "displayName-count-one": "کواشا زامبی",
    "displayName-count-two": "کواشا زامبی",
    "displayName-count-few": "کواشا زامبی",
    "displayName-count-many": "کواشا زامبی",
    "displayName-count-other": "کواشا زامبی",
    "symbol": "ZMW",
    "symbol-alt-narrow": "ZK"
  },
  "ZRN": {
    "displayName": "زائیر زائیری جدید",
    "symbol": "ZRN"
  },
  "ZRZ": {
    "displayName": "زائیر زائیری",
    "symbol": "ZRZ"
  },
  "ZWD": {
    "displayName": "دولار زمبابوی",
    "symbol": "ZWD"
  },
  "ZWL": {
    "displayName": "دولار زمبابوی 2009",
    "symbol": "ZWL"
  },
  "ZWR": {
    "displayName": "ZWR",
    "symbol": "ZWR"
  }
}
}
}
}

```

## CURRENCIES.JSX

```
{
  "main":
  {
    "ar":
    {
      "identity":
      {
        "version":
        {
```

```

        "_number";
        "$Revision: 12879 $",
        "_cldrVersion";
        "30.0.3";
    }
    "language";
    "ar";
}
"numbers";
{
    "currencies";
    {
        "ADP";
        {
            "displayName";
            "بيستا أندوري",
            "symbol";
            "ADP";
        }
        "AED";
        {
            "displayName";
            "درهم إماراتي",
            "displayName-count-zero";
            "درهم إماراتي",
            "displayName-count-one";
            "درهم إماراتي",
            "displayName-count-two";
            "درهم إماراتي",
            "displayName-count-few";
            "درهم إماراتي",
            "displayName-count-many";
            "درهم إماراتي",
            "displayName-count-other";
            "درهم إماراتي",
            "symbol";
            "د.إ.";
        }
        "AFA";
        {
            "displayName";
            "أفغاني - 2002-1927",
            "symbol";
            "AFA";
        }
        "AFN";
        {
            "displayName";
            "أفغاني",
            "displayName-count-zero";
            "أفغاني أفغانستان",
            "displayName-count-one";
            "أفغاني أفغانستان",
            "displayName-count-two";
            "أفغاني أفغانستان",
            "displayName-count-few";
            "أفغاني أفغانستان",
        }
    }
}

```

```

        "displayName-count-many";
        "أفغاني أفغانستاني",
        "displayName-count-other";
        "أفغاني أفغانستاني",
        "symbol";
        "AFN";
    }
    "ALL";
    {
        "displayName";
        "ليك ألباني",
        "displayName-count-zero";
        "ليك ألباني",
        "displayName-count-one";
        "ليك ألباني",
        "displayName-count-two";
        "ليك ألباني",
        "displayName-count-few";
        "ليك ألباني",
        "displayName-count-many";
        "ليك ألباني",
        "displayName-count-other";
        "ليك ألباني",
        "symbol";
        "ALL";
    }
    "AMD";
    {
        "displayName";
        "درام أرميني",
        "displayName-count-zero";
        "درام أرميني",
        "displayName-count-one";
        "درام أرميني",
        "displayName-count-two";
        "درام أرميني",
        "displayName-count-few";
        "درام أرميني",
        "displayName-count-many";
        "درام أرميني",
        "displayName-count-other";
        "درام أرميني",
        "symbol";
        "AMD";
    }
    "ANG";
    {
        "displayName";
        "غيلدر أنتيلي هولندي",
        "displayName-count-zero";
        "غيلدر أنتيلي هولندي",
        "displayName-count-one";
        "غيلدر أنتيلي هولندي",
        "displayName-count-two";
        "غيلدر أنتيلي هولندي",
        "displayName-count-few";
        "غيلدر أنتيلي هولندي",
    }

```

```

        "displayName-count-many";
        "غيلدر أنتيلي هولندي",
        "displayName-count-other";
        "غيلدر أنتيلي هولندي",
        "symbol";
        "ANG";
    }
    "AOA";
    {
        "displayName";
        "كوانزا أنجولي",
        "displayName-count-zero";
        "كوانزا أنجولي",
        "displayName-count-one";
        "كوانزا أنجولي",
        "displayName-count-two";
        "كوانزا أنجولي",
        "displayName-count-few";
        "كوانزا أنجولي",
        "displayName-count-many";
        "كوانزا أنجولي",
        "displayName-count-other";
        "كوانزا أنجولي",
        "symbol";
        "AOA",
        "symbol-alt-narrow";
        "Kz";
    }
    "AOK";
    {
        "displayName";
        "كوانزا أنجولي - 1990-1977",
        "symbol";
        "AOK";
    }
    "AON";
    {
        "displayName";
        "كوانزا أنجولي جديدة - 2000-1990",
        "symbol";
        "AON";
    }
    "AOR";
    {
        "displayName";
        "كوانزا أنجولي معدلة - 1999 - 1995",
        "symbol";
        "AOR";
    }
    "ARA";
    {
        "displayName";
        "استرال أرجنتيني",
        "symbol";
        "ARA";
    }
    "ARL";

```

```

{
  "displayName";
  "ARL",
  "symbol";
  "ARL";
}
"ARM";
{
  "displayName";
  "ARM",
  "symbol";
  "ARM";
}
"ARP";
{
  "displayName";
  "1985-1983 - أرنجنتيني",
  "symbol";
  "ARP";
}
"ARS";
{
  "displayName";
  "بيزو أرنجنتيني",
  "displayName-count-zero";
  "بيزو أرنجنتيني",
  "displayName-count-one";
  "بيزو أرنجنتيني",
  "displayName-count-two";
  "بيزو أرنجنتيني",
  "displayName-count-few";
  "بيزو أرنجنتيني",
  "displayName-count-many";
  "بيزو أرنجنتيني",
  "displayName-count-other";
  "بيزو أرنجنتيني",
  "symbol";
  "ARS",
  "symbol-alt-narrow";
  "AR$";
}
"ATS";
{
  "displayName";
  "شلن نمساوي",
  "symbol";
  "ATS";
}
"AUD";
{
  "displayName";
  "دولار أسترالي",
  "displayName-count-zero";
  "دولار أسترالي",
  "displayName-count-one";
  "دولار أسترالي",
  "displayName-count-two";
}

```



```

        "دولار أسترالي",
        "displayName-count-few";
        "دولار أسترالي",
        "displayName-count-many";
        "دولار أسترالي",
        "displayName-count-other";
        "دولار أسترالي",
        "symbol";
        "AU$";
        "symbol-alt-narrow";
        "AU$";
    }
    "AWG";
    {
        "displayName";
        "فلورن أروبي",
        "displayName-count-zero";
        "فلورن أروبي",
        "displayName-count-one";
        "فلورن أروبي",
        "displayName-count-two";
        "فلورن أروبي",
        "displayName-count-few";
        "فلورن أروبي",
        "displayName-count-many";
        "فلورن أروبي",
        "displayName-count-other";
        "فلورن أروبي",
        "symbol";
        "AWG";
    }
    "AZM";
    {
        "displayName";
        "مانات أذربيجاني",
        "symbol";
        "AZM";
    }
    "AZN";
    {
        "displayName";
        "مانات أذربيجان",
        "displayName-count-zero";
        "مانت أذربيجاني",
        "displayName-count-one";
        "مانت أذربيجاني",
        "displayName-count-two";
        "مانت أذربيجاني",
        "displayName-count-few";
        "مانت أذربيجاني",
        "displayName-count-many";
        "مانت أذربيجاني",
        "displayName-count-other";
        "مانت أذربيجاني",
        "symbol";
        "AZN";
    }
}

```

```

"BAD";
{
  "displayName";
  "دينار البوسنة والهرسك",
  "symbol";
  "BAD";
}
"BAM";
{
  "displayName";
  "مارك البوسنة والهرسك قابل للتحويل",
  "displayName-count-zero";
  "مارك البوسنة والهرسك قابل للتحويل",
  "displayName-count-one";
  "مارك البوسنة والهرسك قابل للتحويل",
  "displayName-count-two";
  "مارك البوسنة والهرسك قابل للتحويل",
  "displayName-count-few";
  "مارك البوسنة والهرسك قابل للتحويل",
  "displayName-count-many";
  "مارك البوسنة والهرسك قابل للتحويل",
  "displayName-count-other";
  "مارك البوسنة والهرسك قابل للتحويل",
  "symbol";
  "BAM",
  "symbol-alt-narrow";
  "KM";
}
"BAN";
{
  "displayName";
  "BAN",
  "symbol";
  "BAN";
}
"BBD";
{
  "displayName";
  "دولار بربادوسي",
  "displayName-count-zero";
  "دولار بربادوسي",
  "displayName-count-one";
  "دولار بربادوسي",
  "displayName-count-two";
  "دولار بربادوسي",
  "displayName-count-few";
  "دولار بربادوسي",
  "displayName-count-many";
  "دولار بربادوسي",
  "displayName-count-other";
  "دولار بربادوسي",
  "symbol";
  "BBD",
  "symbol-alt-narrow";
  "BB$";
}
"BDT";

```

```

    {
      "displayName";
      "তাকা বঙ্গলাদেশি",
      "displayName-count-zero";
      "তাকা বঙ্গলাদেশি",
      "displayName-count-one";
      "তাকা বঙ্গলাদেশি",
      "displayName-count-two";
      "তাকা বঙ্গলাদেশি",
      "displayName-count-few";
      "তাকা বঙ্গলাদেশি",
      "displayName-count-many";
      "তাকা বঙ্গলাদেশি",
      "displayName-count-other";
      "তাকা বঙ্গলাদেশি",
      "symbol";
      "BDT",
      "symbol-alt-narrow";
      "৳";
    }
    "BEC";
    {
      "displayName";
      "ফ্রনক বেলজিকি قابل للتحويل",
      "symbol";
      "BEC";
    }
    "BEF";
    {
      "displayName";
      "ফ্রনক বেলজিকি",
      "symbol";
      "BEF";
    }
    "BEL";
    {
      "displayName";
      "ফ্রনক বেলজিকি مالي",
      "symbol";
      "BEL";
    }
    "BGL";
    {
      "displayName";
      "BGL",
      "symbol";
      "BGL";
    }
    "BGM";
    {
      "displayName";
      "BGM",
      "symbol";
      "BGM";
    }
    "BGN";

```

```
{
  "displayName";
  "ليف بلغاري",
  "displayName-count-zero";
  "ليف بلغاري",
  "displayName-count-one";
  "ليف بلغاري",
  "displayName-count-two";
  "ليف بلغاري",
  "displayName-count-few";
  "ليف بلغاري",
  "displayName-count-many";
  "ليف بلغاري",
  "displayName-count-other";
  "ليف بلغاري",
  "symbol";
  "BGN";
}
"BGO";
{
  "displayName";
  "BGO",
  "symbol";
  "BGO";
}
"BHD";
{
  "displayName";
  "دينار بحريني",
  "displayName-count-zero";
  "دينار بحريني",
  "displayName-count-one";
  "دينار بحريني",
  "displayName-count-two";
  "دينار بحريني",
  "displayName-count-few";
  "دينار بحريني",
  "displayName-count-many";
  "دينار بحريني",
  "displayName-count-other";
  "دينار بحريني",
  "symbol";
  "د.ب.";
}
"BIF";
{
  "displayName";
  "فرنك بروندي",
  "displayName-count-zero";
  "فرنك بروندي",
  "displayName-count-one";
  "فرنك بروندي",
  "displayName-count-two";
  "فرنك بروندي",
  "displayName-count-few";
  "فرنك بروندي",
  "displayName-count-many";
}
```

```

        "فرنك بروندي",
        "displayName-count-other";
        "فرنك بروندي",
        "symbol";
        "BIF";
    }
    "BMD";
    {
        "displayName";
        "دولار برمودي",
        "displayName-count-zero";
        "دولار برمودي",
        "displayName-count-one";
        "دولار برمودي",
        "displayName-count-two";
        "دولار برمودي",
        "displayName-count-few";
        "دولار برمودي",
        "displayName-count-many";
        "دولار برمودي",
        "displayName-count-other";
        "دولار برمودي",
        "symbol";
        "BMD",
        "symbol-alt-narrow";
        "BM$";
    }
    "BND";
    {
        "displayName";
        "دولار برونائي",
        "displayName-count-zero";
        "دولار برونائي",
        "displayName-count-one";
        "دولار برونائي",
        "displayName-count-two";
        "دولار برونائي",
        "displayName-count-few";
        "دولار برونائي",
        "displayName-count-many";
        "دولار برونائي",
        "displayName-count-other";
        "دولار برونائي",
        "symbol";
        "BND",
        "symbol-alt-narrow";
        "BN$";
    }
    "BOB";
    {
        "displayName";
        "بوليفيانو بوليفي",
        "displayName-count-zero";
        "بوليفيانو بوليفي",
        "displayName-count-one";
        "بوليفيانو بوليفي",
        "displayName-count-two";
    }

```

```

        "بوليفيانو بوليفي",
        "displayName-count-few";
        "بوليفيانو بوليفي",
        "displayName-count-many";
        "بوليفيانو بوليفي",
        "displayName-count-other";
        "بوليفيانو بوليفي",
        "symbol";
        "BOB",
        "symbol-alt-narrow";
        "Bs";
    }
    "BOL";
    {
        "displayName";
        "BOL",
        "symbol";
        "BOL";
    }
    "BOP";
    {
        "displayName";
        "بيزو بوليفي",
        "symbol";
        "BOP";
    }
    "BOV";
    {
        "displayName";
        "مفدول بوليفي",
        "symbol";
        "BOV";
    }
    "BRB";
    {
        "displayName";
        "نوفو كروزايرو برازيلى - 1986-1967",
        "symbol";
        "BRB";
    }
    "BRC";
    {
        "displayName";
        "كروزادو برازيلى",
        "symbol";
        "BRC";
    }
    "BRE";
    {
        "displayName";
        "كروزايرو برازيلى - 1993-1990",
        "symbol";
        "BRE";
    }
    "BRL";
    {
        "displayName";

```

```

        "دولار برازيلی",
        "displayName-count-zero";
        "دولار برازيلی",
        "displayName-count-one";
        "دولار برازيلی",
        "displayName-count-two";
        "دولار برازيلی",
        "displayName-count-few";
        "دولار برازيلی",
        "displayName-count-many";
        "دولار برازيلی",
        "displayName-count-other";
        "دولار برازيلی",
        "symbol";
        "R$";
        "symbol-alt-narrow";
        "R$";
    }
    "BRN";
    {
        "displayName";
        "BRN",
        "symbol";
        "BRN";
    }
    "BRR";
    {
        "displayName";
        "BRR",
        "symbol";
        "BRR";
    }
    "BRZ";
    {
        "displayName";
        "BRZ",
        "symbol";
        "BRZ";
    }
    "BSD";
    {
        "displayName";
        "دولار باهامی",
        "displayName-count-zero";
        "دولار باهامی",
        "displayName-count-one";
        "دولار باهامی",
        "displayName-count-two";
        "دولار باهامی",
        "displayName-count-few";
        "دولار باهامی",
        "displayName-count-many";
        "دولار باهامی",
        "displayName-count-other";
        "دولار باهامی",
        "symbol";
        "BSD",

```

```

        "symbol-alt-narrow";
        "BS$";
    }
    "BTN";
    {
        "displayName";
        "نولتوم بوتاني",
        "displayName-count-zero";
        "نولتوم بوتاني",
        "displayName-count-one";
        "نولتوم بوتاني",
        "displayName-count-two";
        "نولتوم بوتاني",
        "displayName-count-few";
        "نولتوم بوتاني",
        "displayName-count-many";
        "نولتوم بوتاني",
        "displayName-count-other";
        "نولتوم بوتاني",
        "symbol";
        "BTN";
    }
    "BUK";
    {
        "displayName";
        "کیات بورمی",
        "symbol";
        "BUK";
    }
    "BWP";
    {
        "displayName";
        "بولا بتسواني",
        "displayName-count-zero";
        "بولا بتسواني",
        "displayName-count-one";
        "بولا بتسواني",
        "displayName-count-two";
        "بولا بتسواني",
        "displayName-count-few";
        "بولا بتسواني",
        "displayName-count-many";
        "بولا بتسواني",
        "displayName-count-other";
        "بولا بتسواني",
        "symbol";
        "BWP",
        "symbol-alt-narrow";
        "P";
    }
    "BYB";
    {
        "displayName";
        "روبل بیلاروسی جدید - 1999-1994",
        "symbol";
        "BYB";
    }
}

```



```
"BYN";
{
  "displayName";
  "روبل بيلاروسي",
  "displayName-count-zero";
  "روبل بيلاروسي",
  "displayName-count-one";
  "روبل بيلاروسي",
  "displayName-count-two";
  "روبل بيلاروسي",
  "displayName-count-few";
  "روبل بيلاروسي",
  "displayName-count-many";
  "روبل بيلاروسي",
  "displayName-count-other";
  "روبل بيلاروسي",
  "symbol";
  "BYN",
  "symbol-alt-narrow";
  "p.";
}
"BYR";
{
  "displayName";
  " ) ۲۰۱۶-۲۰۰۰ ( روبل بيلاروسي",
  "displayName-count-zero";
  " ) ۲۰۱۶-۲۰۰۰ ( روبل بيلاروسي",
  "displayName-count-one";
  " ) ۲۰۱۶-۲۰۰۰ ( روبل بيلاروسي",
  "displayName-count-two";
  " ) ۲۰۱۶-۲۰۰۰ ( روبل بيلاروسي",
  "displayName-count-few";
  " ) ۲۰۱۶-۲۰۰۰ ( روبل بيلاروسي",
  "displayName-count-many";
  " ) ۲۰۱۶-۲۰۰۰ ( روبل بيلاروسي",
  "displayName-count-other";
  " ) ۲۰۱۶-۲۰۰۰ ( روبل بيلاروسي",
  "symbol";
  "BYR";
}
"BZD";
{
  "displayName";
  "دولار بليزي",
  "displayName-count-zero";
  "دولار بليزي",
  "displayName-count-one";
  "دولار بليزي",
  "displayName-count-two";
  "دولاران بليزيان",
  "displayName-count-few";
  "دولار بليزي",
  "displayName-count-many";
  "دولار بليزي",
  "displayName-count-other";
  "دولار بليزي",
  "symbol";
}
```

```

        "BZD",
        "symbol-alt-narrow";
        "BZ$";
    }
    "CAD";
    {
        "displayName";
        "دولار كندي",
        "displayName-count-zero";
        "دولار كندي",
        "displayName-count-one";
        "دولار كندي",
        "displayName-count-two";
        "دولار كندي",
        "displayName-count-few";
        "دولار كندي",
        "displayName-count-many";
        "دولار كندي",
        "displayName-count-other";
        "دولار كندي",
        "symbol";
        "CA$",
        "symbol-alt-narrow";
        "CA$";
    }
    "CDF";
    {
        "displayName";
        "فرنك كونغولي",
        "displayName-count-zero";
        "فرنك كونغولي",
        "displayName-count-one";
        "فرنك كونغولي",
        "displayName-count-two";
        "فرنك كونغولي",
        "displayName-count-few";
        "فرنك كونغولي",
        "displayName-count-many";
        "فرنك كونغولي",
        "displayName-count-other";
        "فرنك كونغولي",
        "symbol";
        "CDF";
    }
    "CHE";
    {
        "displayName";
        "CHE",
        "symbol";
        "CHE";
    }
    "CHF";
    {
        "displayName";
        "فرنك سويسري",
        "displayName-count-zero";
        "فرنك سويسري",

```

```

        "displayName-count-one";
        "فرنك سويسري",
        "displayName-count-two";
        "فرنك سويسري",
        "displayName-count-few";
        "فرنك سويسري",
        "displayName-count-many";
        "فرنك سويسري",
        "displayName-count-other";
        "فرنك سويسري",
        "symbol";
        "CHF";
    }
    "CHW";
    {
        "displayName";
        "CHW",
        "symbol";
        "CHW";
    }
    "CLE";
    {
        "displayName";
        "CLE",
        "symbol";
        "CLE";
    }
    "CLF";
    {
        "displayName";
        "CLF",
        "symbol";
        "CLF";
    }
    "CLP";
    {
        "displayName";
        "بيزو شيلي",
        "displayName-count-zero";
        "بيزو شيلي",
        "displayName-count-one";
        "بيزو شيلي",
        "displayName-count-two";
        "بيزو شيلي",
        "displayName-count-few";
        "بيزو شيلي",
        "displayName-count-many";
        "بيزو شيلي",
        "displayName-count-other";
        "بيزو شيلي",
        "symbol";
        "CLP",
        "symbol-alt-narrow";
        "CL$";
    }
    "CNY";
    {

```

```

        "displayName";
        "يوان صيني",
        "displayName-count-zero";
        "يوان صيني",
        "displayName-count-one";
        "يوان صيني",
        "displayName-count-two";
        "يوان صيني",
        "displayName-count-few";
        "يوان صيني",
        "displayName-count-many";
        "يوان صيني",
        "displayName-count-other";
        "يوان صيني",
        "symbol";
        "CN¥",
        "symbol-alt-narrow";
        "CN¥";
    }
    "COP";
    {
        "displayName";
        "بيزو كولومبي",
        "displayName-count-zero";
        "بيزو كولومبي",
        "displayName-count-one";
        "بيزو كولومبي",
        "displayName-count-two";
        "بيزو كولومبي",
        "displayName-count-few";
        "بيزو كولومبي",
        "displayName-count-many";
        "بيزو كولومبي",
        "displayName-count-other";
        "بيزو كولومبي",
        "symbol";
        "COP",
        "symbol-alt-narrow";
        "CO$";
    }
    "COU";
    {
        "displayName";
        "COU",
        "symbol";
        "COU";
    }
    "CRC";
    {
        "displayName";
        "كولن كوستا ريكي",
        "displayName-count-zero";
        "كولن كوستا ريكي",
        "displayName-count-one";
        "كولن كوستا ريكي",
        "displayName-count-two";
        "كولن كوستا ريكي",
    }

```

```

        "displayName-count-few";
        "کولن کوستا ريکي",
        "displayName-count-many";
        "کولن کوستا ريکي",
        "displayName-count-other";
        "کولن کوستا ريکي",
        "symbol";
        "CRC",
        "symbol-alt-narrow";
        "¢";
    }
    "CSD";
    {
        "displayName";
        "دينار صربي قديم",
        "symbol";
        "CSD";
    }
    "CSK";
    {
        "displayName";
        "کرونة تشيکوسلواکيا",
        "symbol";
        "CSK";
    }
    "CUC";
    {
        "displayName";
        "بيزو کوبي قابل للتحويل",
        "displayName-count-zero";
        "بيزو کوبي قابل للتحويل",
        "displayName-count-one";
        "بيزو کوبي قابل للتحويل",
        "displayName-count-two";
        "بيزو کوبي قابل للتحويل",
        "displayName-count-few";
        "بيزو کوبي قابل للتحويل",
        "displayName-count-many";
        "بيزو کوبي قابل للتحويل",
        "displayName-count-other";
        "بيزو کوبي قابل للتحويل",
        "symbol";
        "CUC",
        "symbol-alt-narrow";
        "$";
    }
    "CUP";
    {
        "displayName";
        "بيزو کوبي",
        "displayName-count-zero";
        "بيزو کوبي",
        "displayName-count-one";
        "بيزو کوبي",
        "displayName-count-two";
        "بيزو کوبي",
        "displayName-count-few";
    }

```

```

        "بيزو كوبي",
        "displayName-count-many";
        "بيزو كوبي",
        "displayName-count-other";
        "بيزو كوبي",
        "symbol";
        "CUP",
        "symbol-alt-narrow";
        "CU$";
    }
    "CVE";
    {
        "displayName";
        "اسكودو الرأس الخضراء",
        "displayName-count-zero";
        "اسكودو الرأس الخضراء",
        "displayName-count-one";
        "اسكودو الرأس الخضراء",
        "displayName-count-two";
        "اسكودو الرأس الخضراء",
        "displayName-count-few";
        "اسكودو الرأس الخضراء",
        "displayName-count-many";
        "اسكودو الرأس الخضراء",
        "displayName-count-other";
        "اسكودو الرأس الخضراء",
        "symbol";
        "CVE";
    }
    "CYP";
    {
        "displayName";
        "جنيه قبرصي",
        "symbol";
        "CYP";
    }
    "CZK";
    {
        "displayName";
        "كرونة تشيكية",
        "displayName-count-zero";
        "كرونة تشيكية",
        "displayName-count-one";
        "كرونة تشيكية",
        "displayName-count-two";
        "كرونة تشيكية",
        "displayName-count-few";
        "كرونة تشيكية",
        "displayName-count-many";
        "كرونة تشيكية",
        "displayName-count-other";
        "كرونة تشيكية",
        "symbol";
        "CZK",
        "symbol-alt-narrow";
        "Kč";
    }
}

```

```

"DDM";
{
  "displayName";
  "أوستمارك ألماني شرقي",
  "symbol";
  "DDM";
}
"DEM";
{
  "displayName";
  "مارك ألماني",
  "symbol";
  "DEM";
}
"DJF";
{
  "displayName";
  "فرنك جيبوتي",
  "displayName-count-zero";
  "فرنك جيبوتي",
  "displayName-count-one";
  "فرنك جيبوتي",
  "displayName-count-two";
  "فرنك جيبوتي",
  "displayName-count-few";
  "فرنك جيبوتي",
  "displayName-count-many";
  "فرنك جيبوتي",
  "displayName-count-other";
  "فرنك جيبوتي",
  "symbol";
  "DJF";
}
"DKK";
{
  "displayName";
  "كرونة دانماركي",
  "displayName-count-zero";
  "كرونة دانماركي",
  "displayName-count-one";
  "كرونة دانماركي",
  "displayName-count-two";
  "كرونة دانماركي",
  "displayName-count-few";
  "كرونة دانماركي",
  "displayName-count-many";
  "كرونة دانماركي",
  "displayName-count-other";
  "كرونة دانماركي",
  "symbol";
  "DKK",
  "symbol-alt-narrow";
  "kr";
}
"DOP";
{
  "displayName";

```

```

        "بيزو الدومنيكان",
        "displayName-count-zero";
        "بيزو الدومنيكان",
        "displayName-count-one";
        "بيزو الدومنيكان",
        "displayName-count-two";
        "بيزو الدومنيكان",
        "displayName-count-few";
        "بيزو الدومنيكان",
        "displayName-count-many";
        "بيزو الدومنيكان",
        "displayName-count-other";
        "بيزو الدومنيكان",
        "symbol";
        "DOP",
        "symbol-alt-narrow";
        "DO$";
    }
    "DZD";
    {
        "displayName";
        "دينار جزائري",
        "displayName-count-zero";
        "دينار جزائري",
        "displayName-count-one";
        "دينار جزائري",
        "displayName-count-two";
        "ديناران جزائريان",
        "displayName-count-few";
        "دينارات جزائرية",
        "displayName-count-many";
        "دينارًا جزائريًا",
        "displayName-count-other";
        "دينار جزائري",
        "symbol";
        "د.ج.";
    }
    "ECS";
    {
        "displayName";
        "ECS",
        "symbol";
        "ECS";
    }
    "ECV";
    {
        "displayName";
        "ECV",
        "symbol";
        "ECV";
    }
    "EEK";
    {
        "displayName";
        "كرونة استونية",
        "symbol";
        "EEK";
    }

```



```

    }
    "EGP";
    {
        "displayName";
        "جنيه مصري",
        "displayName-count-zero";
        "جنيه مصري",
        "displayName-count-one";
        "جنيه مصري",
        "displayName-count-two";
        "جنيهان مصريان",
        "displayName-count-few";
        "جنيهات مصرية",
        "displayName-count-many";
        "جنيهاً مصرياً",
        "displayName-count-other";
        "جنيه مصري",
        "symbol";
        "ج.م.ع",
        "symbol-alt-narrow";
        "E£";
    }
    "ERN";
    {
        "displayName";
        "ناكفا أريتري",
        "displayName-count-zero";
        "ناكفا أريتري",
        "displayName-count-one";
        "ناكفا أريتري",
        "displayName-count-two";
        "ناكفا أريتري",
        "displayName-count-few";
        "ناكفا أريتري",
        "displayName-count-many";
        "ناكفا أريتري",
        "displayName-count-other";
        "ناكفا أريتري",
        "symbol";
        "ERN";
    }
    "ESA";
    {
        "displayName";
        "ESA",
        "symbol";
        "ESA";
    }
    "ESB";
    {
        "displayName";
        "ESB",
        "symbol";
        "ESB";
    }
    "ESP";
    {

```

```

        "displayName";
        "بیزیتا إسبانی",
        "symbol";
        "ESP",
        "symbol-alt-narrow";
        "₧";
    }
    "ETB";
    {
        "displayName";
        "بیر أثیوبی",
        "displayName-count-zero";
        "بیر أثیوبی",
        "displayName-count-one";
        "بیر أثیوبی",
        "displayName-count-two";
        "بیر أثیوبی",
        "displayName-count-few";
        "بیر أثیوبی",
        "displayName-count-many";
        "بیر أثیوبی",
        "displayName-count-other";
        "بیر أثیوبی",
        "symbol";
        "ETB";
    }
    "EUR";
    {
        "displayName";
        "یورو",
        "displayName-count-zero";
        "یورو",
        "displayName-count-one";
        "یورو",
        "displayName-count-two";
        "یورو",
        "displayName-count-few";
        "یورو",
        "displayName-count-many";
        "یورو",
        "displayName-count-other";
        "یورو",
        "symbol";
        "€",
        "symbol-alt-narrow";
        "€";
    }
    "FIM";
    {
        "displayName";
        "مارکا فنلندی",
        "symbol";
        "FIM";
    }
    "FJD";
    {
        "displayName";

```

```

        "دولار فيجي",
        "displayName-count-zero";
        "دولار فيجي",
        "displayName-count-one";
        "دولار فيجي",
        "displayName-count-two";
        "دولار فيجي",
        "displayName-count-few";
        "دولار فيجي",
        "displayName-count-many";
        "دولار فيجي",
        "displayName-count-other";
        "دولار فيجي",
        "symbol";
        "FJD",
        "symbol-alt-narrow";
        "FJ$";
    }
    "FKP";
    {
        "displayName";
        "جنيه جزر فوكلاند",
        "displayName-count-zero";
        "جنيه جزر فوكلاند",
        "displayName-count-one";
        "جنيه جزر فوكلاند",
        "displayName-count-two";
        "جنيه جزر فوكلاند",
        "displayName-count-few";
        "جنيه جزر فوكلاند",
        "displayName-count-many";
        "جنيه جزر فوكلاند",
        "displayName-count-other";
        "جنيه جزر فوكلاند",
        "symbol";
        "FKP",
        "symbol-alt-narrow";
        "£";
    }
    "FRF";
    {
        "displayName";
        "فرنك فرنسي",
        "symbol";
        "FRF";
    }
    "GBP";
    {
        "displayName";
        "جنيه إسترليني",
        "displayName-count-zero";
        "جنيه إسترليني",
        "displayName-count-one";
        "جنيه إسترليني",
        "displayName-count-two";
        "جنيه إسترليني",
        "displayName-count-few";
    }

```

```

        "جنيه إسترليني",
        "displayName-count-many";
        "جنيه إسترليني",
        "displayName-count-other";
        "جنيه إسترليني",
        "symbol";
        "£",
        "symbol-alt-narrow";
        "UK£";
    }
    "GEK";
    {
        "displayName";
        "GEK",
        "symbol";
        "GEK";
    }
    "GEL";
    {
        "displayName";
        "ლარი جورجი",
        "displayName-count-zero";
        "ლარი جورجი",
        "displayName-count-one";
        "ლარი جورجი",
        "displayName-count-two";
        "ლარი جورجი",
        "displayName-count-few";
        "ლარი جورجი",
        "displayName-count-many";
        "ლარი جورجი",
        "displayName-count-other";
        "ლარი جورجი",
        "symbol";
        "GEL",
        "symbol-alt-narrow";
        "ლ",
        "symbol-alt-variant";
        "ლ";
    }
    "GHC";
    {
        "displayName";
        "სიდი განი",
        "symbol";
        "GHC";
    }
    "GHS";
    {
        "displayName";
        "სიდი განა",
        "displayName-count-zero";
        "სიდი განა",
        "displayName-count-one";
        "სიდი განა",
        "displayName-count-two";
        "სიდი განა",
    }

```

```

        "displayName-count-few";
        "سيدي غانا",
        "displayName-count-many";
        "سيدي غانا",
        "displayName-count-other";
        "سيدي غانا",
        "symbol";
        "GHS";
    }
    "GIP";
    {
        "displayName";
        "جنيه جبل طارق",
        "displayName-count-zero";
        "جنيه جبل طارق",
        "displayName-count-one";
        "جنيه جبل طارق",
        "displayName-count-two";
        "جنيه جبل طارق",
        "displayName-count-few";
        "جنيه جبل طارق",
        "displayName-count-many";
        "جنيه جبل طارق",
        "displayName-count-other";
        "جنيه جبل طارق",
        "symbol";
        "GIP",
        "symbol-alt-narrow";
        "£";
    }
    "GMD";
    {
        "displayName";
        "دلاسي جامبي",
        "displayName-count-zero";
        "دلاسي جامبي",
        "displayName-count-one";
        "دلاسي جامبي",
        "displayName-count-two";
        "دلاسي جامبي",
        "displayName-count-few";
        "دلاسي جامبي",
        "displayName-count-many";
        "دلاسي جامبي",
        "displayName-count-other";
        "دلاسي جامبي",
        "symbol";
        "GMD";
    }
    "GNF";
    {
        "displayName";
        "فرنك غينيا",
        "displayName-count-zero";
        "فرنك غينيا",
        "displayName-count-one";
        "فرنك غينيا",
    }

```

```

        "displayName-count-two";
        "فرنك غينيا",
        "displayName-count-few";
        "فرنك غينيا",
        "displayName-count-many";
        "فرنك غينيا",
        "displayName-count-other";
        "فرنك غينيا",
        "symbol";
        "GNF",
        "symbol-alt-narrow";
        "FG";
    }
    "GNS";
    {
        "displayName";
        "سيلي غينيا",
        "symbol";
        "GNS";
    }
    "GQE";
    {
        "displayName";
        "اكويل جونيئا غينيا الاستوائية",
        "symbol";
        "GQE";
    }
    "GRD";
    {
        "displayName";
        "دراخما يوناني",
        "symbol";
        "GRD";
    }
    "GTQ";
    {
        "displayName";
        "كوتزال جواتيمالا",
        "displayName-count-zero";
        "كوتزال جواتيمالا",
        "displayName-count-one";
        "كوتزال جواتيمالا",
        "displayName-count-two";
        "كوتزال جواتيمالا",
        "displayName-count-few";
        "كوتزال جواتيمالا",
        "displayName-count-many";
        "كوتزال جواتيمالا",
        "displayName-count-other";
        "كوتزال جواتيمالا",
        "symbol";
        "GTQ",
        "symbol-alt-narrow";
        "Q";
    }
    "GWE";
    {

```

```

        "displayName";
        "اسکود برتغالي غينيا",
        "symbol";
        "GWE";
    }
    "GWP";
    {
        "displayName";
        "بیزو غینیا بیساو",
        "symbol";
        "GWP";
    }
    "GYD";
    {
        "displayName";
        "دولار غيانا",
        "displayName-count-zero";
        "دولار غيانا",
        "displayName-count-one";
        "دولار غيانا",
        "displayName-count-two";
        "دولار غيانا",
        "displayName-count-few";
        "دولار غيانا",
        "displayName-count-many";
        "دولار غيانا",
        "displayName-count-other";
        "دولار غيانا",
        "symbol";
        "GYD",
        "symbol-alt-narrow";
        "GY$";
    }
    "HKD";
    {
        "displayName";
        "دولار هونغ كونغ",
        "displayName-count-zero";
        "دولار هونغ كونغ",
        "displayName-count-one";
        "دولار هونغ كونغ",
        "displayName-count-two";
        "دولار هونغ كونغ",
        "displayName-count-few";
        "دولار هونغ كونغ",
        "displayName-count-many";
        "دولار هونغ كونغ",
        "displayName-count-other";
        "دولار هونغ كونغ",
        "symbol";
        "HK$",
        "symbol-alt-narrow";
        "HK$";
    }
    "HNL";
    {
        "displayName";

```

```

        "ليمبيرا هندوراس",
        "displayName-count-zero";
        "ليمبيرا هندوراس",
        "displayName-count-one";
        "ليمبيرا هندوراس",
        "displayName-count-two";
        "ليمبيرا هندوراس",
        "displayName-count-few";
        "ليمبيرا هندوراس",
        "displayName-count-many";
        "ليمبيرا هندوراس",
        "displayName-count-other";
        "ليمبيرا هندوراس",
        "symbol";
        "HNL",
        "symbol-alt-narrow";
        "L";
    }
    "HRD";
    {
        "displayName";
        "دينار كرواتي",
        "symbol";
        "HRD";
    }
    "HRK";
    {
        "displayName";
        "كونا كرواتي",
        "displayName-count-zero";
        "كونا كرواتي",
        "displayName-count-one";
        "كونا كرواتي",
        "displayName-count-two";
        "كونا كرواتي",
        "displayName-count-few";
        "كونا كرواتي",
        "displayName-count-many";
        "كونا كرواتي",
        "displayName-count-other";
        "كونا كرواتي",
        "symbol";
        "HRK",
        "symbol-alt-narrow";
        "kn";
    }
    "HTG";
    {
        "displayName";
        "جوردی هاييتي",
        "displayName-count-zero";
        "جوردی هاييتي",
        "displayName-count-one";
        "جوردی هاييتي",
        "displayName-count-two";
        "جوردی هاييتي",
        "displayName-count-few";
    }

```



```

        "جوردی هایتي",
        "displayName-count-many";
        "جوردی هایتي",
        "displayName-count-other";
        "جوردی هایتي",
        "symbol";
        "HTG";
    }
    "HUF";
    {
        "displayName";
        "فورينت مجري",
        "displayName-count-zero";
        "فورينت مجري",
        "displayName-count-one";
        "فورينت مجري",
        "displayName-count-two";
        "فورينت مجري",
        "displayName-count-few";
        "فورينت مجري",
        "displayName-count-many";
        "فورينت مجري",
        "displayName-count-other";
        "فورينت مجري",
        "symbol";
        "HUF",
        "symbol-alt-narrow";
        "Ft";
    }
    "IDR";
    {
        "displayName";
        "روبية إندونيسية",
        "displayName-count-zero";
        "روبية إندونيسية",
        "displayName-count-one";
        "روبية إندونيسية",
        "displayName-count-two";
        "روبية إندونيسية",
        "displayName-count-few";
        "روبية إندونيسية",
        "displayName-count-many";
        "روبية إندونيسية",
        "displayName-count-other";
        "روبية إندونيسية",
        "symbol";
        "IDR",
        "symbol-alt-narrow";
        "Rp";
    }
    "IEP";
    {
        "displayName";
        "جنيه إيرلندي",
        "symbol";
        "IEP";
    }
}

```

```

"ILP";
{
  "displayName";
  "جنيه إسرائيلي",
  "symbol";
  "ILP";
}
"ILS";
{
  "displayName";
  "שילקל ישראליל جليل",
  "displayName-count-zero";
  "שילקל ישראליל جليل",
  "displayName-count-one";
  "שילקל ישראליל جليل",
  "displayName-count-two";
  "שילקל ישראליל جليل",
  "displayName-count-few";
  "שילקל ישראליל جليل",
  "displayName-count-many";
  "שילקל ישראליל جليل",
  "displayName-count-other";
  "שילקל ישראליל جليل",
  "symbol";
  "₪",
  "symbol-alt-narrow";
  "₪";
}
"INR";
{
  "displayName";
  "روبية هنلي",
  "displayName-count-zero";
  "روبية هنلي",
  "displayName-count-one";
  "روبية هنلي",
  "displayName-count-two";
  "روبية هنلي",
  "displayName-count-few";
  "روبية هنلي",
  "displayName-count-many";
  "روبية هنلي",
  "displayName-count-other";
  "روبية هنلي",
  "symbol";
  "₹",
  "symbol-alt-narrow";
  "₹";
}
"IQD";
{
  "displayName";
  "للىنار عراقلي",
  "displayName-count-zero";
  "للىنار عراقلي",
  "displayName-count-one";
  "للىنار عراقلي",

```

```

        "displayName-count-two";
        "دينار عراقي",
        "displayName-count-few";
        "دينار عراقي",
        "displayName-count-many";
        "دينار عراقي",
        "displayName-count-other";
        "دينار عراقي",
        "symbol";
        "د.ع.";
    }
    "IRR";
    {
        "displayName";
        "هلد إيراني",
        "displayName-count-zero";
        "هلد إيراني",
        "displayName-count-one";
        "هلد إيراني",
        "displayName-count-two";
        "هلد إيراني",
        "displayName-count-few";
        "هلد إيراني",
        "displayName-count-many";
        "هلد إيراني",
        "displayName-count-other";
        "هلد إيراني",
        "symbol";
        "ر.إ.";
    }
    "ISK";
    {
        "displayName";
        "كرونه أيسلندي",
        "displayName-count-zero";
        "كرونه أيسلندي",
        "displayName-count-one";
        "كرونه أيسلندي",
        "displayName-count-two";
        "كرونه أيسلندي",
        "displayName-count-few";
        "كرونه أيسلندي",
        "displayName-count-many";
        "كرونه أيسلندي",
        "displayName-count-other";
        "كرونه أيسلندي",
        "symbol";
        "ISK",
        "symbol-alt-narrow";
        "kr";
    }
    "ITL";
    {
        "displayName";
        "ليرة إيطالية",
        "symbol";
        "ITL";
    }

```

```

    }
    "JMD";
    {
        "displayName";
        "دولار جامايكي",
        "displayName-count-zero";
        "دولار جامايكي",
        "displayName-count-one";
        "دولار جامايكي",
        "displayName-count-two";
        "دولار جامايكي",
        "displayName-count-few";
        "دولار جامايكي",
        "displayName-count-many";
        "دولار جامايكي",
        "displayName-count-other";
        "دولار جامايكي",
        "symbol";
        "JMD",
        "symbol-alt-narrow";
        "JM$";
    }
    "JOD";
    {
        "displayName";
        "دينار أردني",
        "displayName-count-zero";
        "دينار أردني",
        "displayName-count-one";
        "دينار أردني",
        "displayName-count-two";
        "دينار أردني",
        "displayName-count-few";
        "دينار أردني",
        "displayName-count-many";
        "دينار أردني",
        "displayName-count-other";
        "دينار أردني",
        "symbol";
        "د.أ.";
    }
    "JPY";
    {
        "displayName";
        "ين ياباني",
        "displayName-count-zero";
        "ين ياباني",
        "displayName-count-one";
        "ين ياباني",
        "displayName-count-two";
        "ين ياباني",
        "displayName-count-few";
        "ين ياباني",
        "displayName-count-many";
        "ين ياباني",
        "displayName-count-other";
        "ين ياباني",
    }

```

```
        "symbol";
        "JP¥",
        "symbol-alt-narrow";
        "JP¥";
    }
    "KES";
    {
        "displayName";
        "شلن كيني",
        "displayName-count-zero";
        "شلن كيني",
        "displayName-count-one";
        "شلن كيني",
        "displayName-count-two";
        "شلن كيني",
        "displayName-count-few";
        "شلن كيني",
        "displayName-count-many";
        "شلن كيني",
        "displayName-count-other";
        "شلن كيني",
        "symbol";
        "KES";
    }
    "KGS";
    {
        "displayName";
        "سوم قيرغستاني",
        "displayName-count-zero";
        "سوم قيرغستاني",
        "displayName-count-one";
        "سوم قيرغستاني",
        "displayName-count-two";
        "سوم قيرغستاني",
        "displayName-count-few";
        "سوم قيرغستاني",
        "displayName-count-many";
        "سوم قيرغستاني",
        "displayName-count-other";
        "سوم قيرغستاني",
        "symbol";
        "KGS";
    }
    "KHR";
    {
        "displayName";
        "ريال كمبودي",
        "displayName-count-zero";
        "ريال كمبودي",
        "displayName-count-one";
        "ريال كمبودي",
        "displayName-count-two";
        "ريال كمبودي",
        "displayName-count-few";
        "ريال كمبودي",
        "displayName-count-many";
        "ريال كمبودي",
    }
```

```

        "displayName-count-other";
        "ريال كمبودي",
        "symbol";
        "KHR",
        "symbol-alt-narrow";
        "៛";
    }
    "KMF";
    {
        "displayName";
        "فرنك جزر القمر",
        "displayName-count-zero";
        "فرنك جزر القمر",
        "displayName-count-one";
        "فرنك جزر القمر",
        "displayName-count-two";
        "فرنك جزر القمر",
        "displayName-count-few";
        "فرنك جزر القمر",
        "displayName-count-many";
        "فرنك جزر القمر",
        "displayName-count-other";
        "فرنك جزر القمر",
        "symbol";
        "ف.ج.ق.",
        "symbol-alt-narrow";
        "CF";
    }
    "KPW";
    {
        "displayName";
        "وون كوريا الشمالية",
        "displayName-count-zero";
        "وون كوريا الشمالية",
        "displayName-count-one";
        "وون كوريا الشمالية",
        "displayName-count-two";
        "وون كوريا الشمالية",
        "displayName-count-few";
        "وون كوريا الشمالية",
        "displayName-count-many";
        "وون كوريا الشمالية",
        "displayName-count-other";
        "وون كوريا الشمالية",
        "symbol";
        "KPW",
        "symbol-alt-narrow";
        "₩";
    }
    "KRH";
    {
        "displayName";
        "KRH",
        "symbol";
        "KRH";
    }
}

```

```

    "KRO";
    {
      "displayName";
      "KRO",
      "symbol";
      "KRO";
    }
    "KRW";
    {
      "displayName";
      "وون كوريا الجنوبية",
      "displayName-count-zero";
      "وون كوريا الجنوبية",
      "displayName-count-one";
      "وون كوريا الجنوبية",
      "displayName-count-two";
      "وون كوريا الجنوبية",
      "displayName-count-few";
      "وون كوريا الجنوبية",
      "displayName-count-many";
      "وون كوريا الجنوبية",
      "displayName-count-other";
      "وون كوريا الجنوبية",
      "symbol";
      "₩",
      "symbol-alt-narrow";
      "₩";
    }
    "KWD";
    {
      "displayName";
      "دينار كويتي",
      "displayName-count-zero";
      "دينار كويتي",
      "displayName-count-one";
      "دينار كويتي",
      "displayName-count-two";
      "دينار كويتي",
      "displayName-count-few";
      "دينار كويتي",
      "displayName-count-many";
      "دينار كويتي",
      "displayName-count-other";
      "دينار كويتي",
      "symbol";
      "د.ك.";
    }
    "KYD";
    {
      "displayName";
      "دولار جزر كيمن",
      "displayName-count-zero";
      "دولار جزر كيمن",
      "displayName-count-one";
      "دولار جزر كيمن",
      "displayName-count-two";
      "دولار جزر كيمن",
    }

```

```

        "displayName-count-few";
        "دولار جزر كيمن",
        "displayName-count-many";
        "دولار جزر كيمن",
        "displayName-count-other";
        "دولار جزر كيمن",
        "symbol";
        "KYD",
        "symbol-alt-narrow";
        "KY$";
    }
    "KZT";
    {
        "displayName";
        "تينغ كازاخستاني",
        "displayName-count-zero";
        "تينغ كازاخستاني",
        "displayName-count-one";
        "تينغ كازاخستاني",
        "displayName-count-two";
        "تينغ كازاخستاني",
        "displayName-count-few";
        "تينغ كازاخستاني",
        "displayName-count-many";
        "تينغ كازاخستاني",
        "displayName-count-other";
        "تينغ كازاخستاني",
        "symbol";
        "KZT",
        "symbol-alt-narrow";
        "Т";
    }
    "LAK";
    {
        "displayName";
        "كيب لاوسي",
        "displayName-count-zero";
        "كيب لاوسي",
        "displayName-count-one";
        "كيب لاوسي",
        "displayName-count-two";
        "كيب لاوسي",
        "displayName-count-few";
        "كيب لاوسي",
        "displayName-count-many";
        "كيب لاوسي",
        "displayName-count-other";
        "كيب لاوسي",
        "symbol";
        "LAK",
        "symbol-alt-narrow";
        "₭";
    }
    "LBP";
    {
        "displayName";
        "جنيه لبناني",

```



```

        "displayName-count-zero";
        "جنيه لبناني",
        "displayName-count-one";
        "جنيه لبناني",
        "displayName-count-two";
        "جنيه لبناني",
        "displayName-count-few";
        "جنيه لبناني",
        "displayName-count-many";
        "جنيه لبناني",
        "displayName-count-other";
        "جنيه لبناني",
        "symbol";
        "ل.ل.",
        "symbol-alt-narrow";
        "L£";
    }
    "LKR";
    {
        "displayName";
        "روبية سريلانكية",
        "displayName-count-zero";
        "روبية سريلانكية",
        "displayName-count-one";
        "روبية سريلانكية",
        "displayName-count-two";
        "روبية سريلانكية",
        "displayName-count-few";
        "روبية سريلانكية",
        "displayName-count-many";
        "روبية سريلانكية",
        "displayName-count-other";
        "روبية سريلانكية",
        "symbol";
        "LKR",
        "symbol-alt-narrow";
        "Rs";
    }
    "LRD";
    {
        "displayName";
        "دولار ليبيري",
        "displayName-count-zero";
        "دولار ليبيري",
        "displayName-count-one";
        "دولار ليبيري",
        "displayName-count-two";
        "دولار ان ليبيريان",
        "displayName-count-few";
        "دولارات ليبيرية",
        "displayName-count-many";
        "دولارًا ليبيريًا",
        "displayName-count-other";
        "دولار ليبيري",
        "symbol";
        "LRD",
        "symbol-alt-narrow";
    }

```

```

        "$";
    }
    "LSL";
    {
        "displayName";
        "لوتي ليسوتو",
        "symbol";
        "LSL";
    }
    "LTL";
    {
        "displayName";
        "ليتا ليتوانية",
        "displayName-count-zero";
        "ليتا ليتوانية",
        "displayName-count-one";
        "ليتا ليتوانية",
        "displayName-count-two";
        "ليتا ليتوانية",
        "displayName-count-few";
        "ليتا ليتوانية",
        "displayName-count-many";
        "ليتا ليتوانية",
        "displayName-count-other";
        "ليتا ليتوانية",
        "symbol";
        "LTL",
        "symbol-alt-narrow";
        "Lt";
    }
    "LTT";
    {
        "displayName";
        "تالوناس ليتواني",
        "symbol";
        "LTT";
    }
    "LUC";
    {
        "displayName";
        "فرنك لوكسمبرج قابل للتحويل",
        "symbol";
        "LUC";
    }
    "LUF";
    {
        "displayName";
        "فرنك لوكسمبرج",
        "symbol";
        "LUF";
    }
    "LUL";
    {
        "displayName";
        "فرنك لوكسمبرج المالي",
        "symbol";
        "LUL";
    }

```

```
}
"LVL";
{
  "displayName";
  "لاتس لاتفيا",
  "displayName-count-zero";
  "لاتس لاتفي",
  "displayName-count-one";
  "لاتس لاتفي",
  "displayName-count-two";
  "لاتس لاتفي",
  "displayName-count-few";
  "لاتس لاتفي",
  "displayName-count-many";
  "لاتس لاتفي",
  "displayName-count-other";
  "لاتس لاتفي",
  "symbol";
  "LVL",
  "symbol-alt-narrow";
  "Ls";
}
"LVR";
{
  "displayName";
  "روبل لاتفيا",
  "symbol";
  "LVR";
}
"LYD";
{
  "displayName";
  "دينار ليبي",
  "displayName-count-zero";
  "دينار ليبي",
  "displayName-count-one";
  "دينار ليبي",
  "displayName-count-two";
  "ديناران ليبيان",
  "displayName-count-few";
  "دينارات ليبية",
  "displayName-count-many";
  "دينارًا ليبيًا",
  "displayName-count-other";
  "دينار ليبي",
  "symbol";
  "د.ل.";
}
"MAD";
{
  "displayName";
  "درهم مغربي",
  "displayName-count-zero";
  "درهم مغربي",
  "displayName-count-one";
  "درهم مغربي",
  "displayName-count-two";
}
```

```

        "درهمان مغربيان",
        "displayName-count-few";
        "دراهم مغربية",
        "displayName-count-many";
        "درهما مغربيا",
        "displayName-count-other";
        "درهم مغربي",
        "symbol";
        ". د.م.";
    }
    "MAF";
    {
        "displayName";
        "فرنك مغربي",
        "symbol";
        "MAF";
    }
    "MCF";
    {
        "displayName";
        "MCF",
        "symbol";
        "MCF";
    }
    "MDC";
    {
        "displayName";
        "MDC",
        "symbol";
        "MDC";
    }
    "MDL";
    {
        "displayName";
        "ليو مولدوفي",
        "displayName-count-zero";
        "ليو مولدوفي",
        "displayName-count-one";
        "ليو مولدوفي",
        "displayName-count-two";
        "ليو مولدوفي",
        "displayName-count-few";
        "ليو مولدوفي",
        "displayName-count-many";
        "ليو مولدوفي",
        "displayName-count-other";
        "ليو مولدوفي",
        "symbol";
        "MDL";
    }
    "MGA";
    {
        "displayName";
        "أرياري مدغشقر",
        "displayName-count-zero";
        "أرياري مدغشقر",
        "displayName-count-one";
    }

```

```

        "أرياري مدغشقر",
        "displayName-count-two";
        "أرياري مدغشقر",
        "displayName-count-few";
        "أرياري مدغشقر",
        "displayName-count-many";
        "أرياري مدغشقر",
        "displayName-count-other";
        "أرياري مدغشقر",
        "symbol";
        "MGA",
        "symbol-alt-narrow";
        "Ar";
    }
    "MGF";
    {
        "displayName";
        "فرنك مدغشقر",
        "symbol";
        "MGF";
    }
    "MKD";
    {
        "displayName";
        "دينار مقدوني",
        "displayName-count-zero";
        "دينار مقدوني",
        "displayName-count-one";
        "دينار مقدوني",
        "displayName-count-two";
        "ديناران مقدونيان",
        "displayName-count-few";
        "دينارات مقدونية",
        "displayName-count-many";
        "دينارًا مقدونيا",
        "displayName-count-other";
        "دينار مقدوني",
        "symbol";
        "MKD";
    }
    "MKN";
    {
        "displayName";
        "MKN",
        "symbol";
        "MKN";
    }
    "MLF";
    {
        "displayName";
        "فرنك مالي",
        "symbol";
        "MLF";
    }
    "MMK";
    {
        "displayName";

```

```

        "кият میانمار",
        "displayName-count-zero";
        "кият میانمار",
        "displayName-count-one";
        "кият میانمار",
        "displayName-count-two";
        "кият میانمار",
        "displayName-count-few";
        "кият میانمار",
        "displayName-count-many";
        "кият میانمار",
        "displayName-count-other";
        "кият میانمار",
        "symbol";
        "MMK",
        "symbol-alt-narrow";
        "K";
    }
    "MNT";
    {
        "displayName";
        "туугрог манголи",
        "displayName-count-zero";
        "туугрог манголи",
        "displayName-count-one";
        "туугрог манголи",
        "displayName-count-two";
        "туугрог манголи",
        "displayName-count-few";
        "туугрог манголи",
        "displayName-count-many";
        "туугрог манголи",
        "displayName-count-other";
        "туугрог манголи",
        "symbol";
        "MNT",
        "symbol-alt-narrow";
        "₮";
    }
    "MOP";
    {
        "displayName";
        "батата макауи",
        "displayName-count-zero";
        "батата макауи",
        "displayName-count-one";
        "батата макауи",
        "displayName-count-two";
        "батата макауи",
        "displayName-count-few";
        "батата макауи",
        "displayName-count-many";
        "батата макауи",
        "displayName-count-other";
        "батата макауи",
        "symbol";
        "MOP";
    }

```

```

    }
    "MRO";
    {
        "displayName";
        "أوقية موريتانية",
        "displayName-count-zero";
        "أوقية موريتانية",
        "displayName-count-one";
        "أوقية موريتانية",
        "displayName-count-two";
        "أوقية موريتانية",
        "displayName-count-few";
        "أوقية موريتانية",
        "displayName-count-many";
        "أوقية موريتانية",
        "displayName-count-other";
        "أوقية موريتانية",
        "symbol";
        "أ.م.";
    }
    "MTL";
    {
        "displayName";
        "ليرة مالطية",
        "symbol";
        "MTL";
    }
    "MTP";
    {
        "displayName";
        "جنيه مالطي",
        "symbol";
        "MTP";
    }
    "MUR";
    {
        "displayName";
        "روبية موريشيوسية",
        "displayName-count-zero";
        "روبية موريشيوسية",
        "displayName-count-one";
        "روبية موريشيوسية",
        "displayName-count-two";
        "روبية موريشيوسية",
        "displayName-count-few";
        "روبية موريشيوسية",
        "displayName-count-many";
        "روبية موريشيوسية",
        "displayName-count-other";
        "روبية موريشيوسية",
        "symbol";
        "MUR",
        "symbol-alt-narrow";
        "Rs";
    }
    "MVR";
    {

```

```

        "displayName";
        "روفيه جزر المالديف",
        "displayName-count-zero";
        "روفيه جزر المالديف",
        "displayName-count-one";
        "روفيه جزر المالديف",
        "displayName-count-two";
        "روفيه جزر المالديف",
        "displayName-count-few";
        "روفيه جزر المالديف",
        "displayName-count-many";
        "روفيه جزر المالديف",
        "displayName-count-other";
        "روفيه جزر المالديف",
        "symbol";
        "MVR";
    }
    "MWK";
    {
        "displayName";
        "كواشا مالاوي",
        "displayName-count-zero";
        "كواشا مالاوي",
        "displayName-count-one";
        "كواشا مالاوي",
        "displayName-count-two";
        "كواشا مالاوي",
        "displayName-count-few";
        "كواشا مالاوي",
        "displayName-count-many";
        "كواشا مالاوي",
        "displayName-count-other";
        "كواشا مالاوي",
        "symbol";
        "MWK";
    }
    "MXN";
    {
        "displayName";
        "بيزو مكسيكي",
        "displayName-count-zero";
        "بيزو مكسيكي",
        "displayName-count-one";
        "بيزو مكسيكي",
        "displayName-count-two";
        "بيزو مكسيكي",
        "displayName-count-few";
        "بيزو مكسيكي",
        "displayName-count-many";
        "بيزو مكسيكي",
        "displayName-count-other";
        "بيزو مكسيكي",
        "symbol";
        "MX$",
        "symbol-alt-narrow";
        "MX$";
    }
}

```



```

"MXP";
{
  "displayName";
  "1992-1861 - بيزو فضي مكسيكي",
  "symbol";
  "MXP";
}
"MXV";
{
  "displayName";
  "MXV",
  "symbol";
  "MXV";
}
"MYR";
{
  "displayName";
  "رينغيت ماليزي",
  "displayName-count-zero";
  "رينغيت ماليزي",
  "displayName-count-one";
  "رينغيت ماليزي",
  "displayName-count-two";
  "رينغيت ماليزي",
  "displayName-count-few";
  "رينغيت ماليزي",
  "displayName-count-many";
  "رينغيت ماليزي",
  "displayName-count-other";
  "رينغيت ماليزي",
  "symbol";
  "MYR",
  "symbol-alt-narrow";
  "RM";
}
"MZE";
{
  "displayName";
  "اسكود موزمبيقي",
  "symbol";
  "MZE";
}
"MZM";
{
  "displayName";
  "MZM",
  "symbol";
  "MZM";
}
"MZN";
{
  "displayName";
  "متكال موزمبيقي",
  "displayName-count-zero";
  "متكال موزمبيقي",
  "displayName-count-one";
  "متكال موزمبيقي",

```

```

        "displayName-count-two";
        "متكال موزمبيقى",
        "displayName-count-few";
        "متكال موزمبيقى",
        "displayName-count-many";
        "متكال موزمبيقى",
        "displayName-count-other";
        "متكال موزمبيقى",
        "symbol";
        "MZN";
    }
    "NAD";
    {
        "displayName";
        "دولار نامىبى",
        "displayName-count-zero";
        "دولار نامىبى",
        "displayName-count-one";
        "دولار نامىبى",
        "displayName-count-two";
        "دولار نامىبى",
        "displayName-count-few";
        "دولار نامىبى",
        "displayName-count-many";
        "دولار نامىبى",
        "displayName-count-other";
        "دولار نامىبى",
        "symbol";
        "NAD",
        "symbol-alt-narrow";
        "$";
    }
    "NGN";
    {
        "displayName";
        "نايرا نيجيرى",
        "displayName-count-zero";
        "نايرا نيجيرى",
        "displayName-count-one";
        "نايرا نيجيرى",
        "displayName-count-two";
        "نايرا نيجيرى",
        "displayName-count-few";
        "نايرا نيجيرى",
        "displayName-count-many";
        "نايرا نيجيرى",
        "displayName-count-other";
        "نايرا نيجيرى",
        "symbol";
        "NGN",
        "symbol-alt-narrow";
        "₦";
    }
    "NIC";
    {
        "displayName";
        "كوردوبه نيكاراچوا",

```

```

        "symbol";
        "NIC";
    }
    "NIO";
    {
        "displayName";
        "قرطبة نيكاراغوا",
        "displayName-count-zero";
        "قرطبة نيكاراغوا",
        "displayName-count-one";
        "قرطبة نيكاراغوا",
        "displayName-count-two";
        "قرطبة نيكاراغوا",
        "displayName-count-few";
        "قرطبة نيكاراغوا",
        "displayName-count-many";
        "قرطبة نيكاراغوا",
        "displayName-count-other";
        "قرطبة نيكاراغوا",
        "symbol";
        "NIO",
        "symbol-alt-narrow";
        "C$";
    }
    "NLG";
    {
        "displayName";
        "جلدر هولندي",
        "symbol";
        "NLG";
    }
    "NOK";
    {
        "displayName";
        "كرونة نرويجية",
        "displayName-count-zero";
        "كرونة نرويجية",
        "displayName-count-one";
        "كرونة نرويجية",
        "displayName-count-two";
        "كرونة نرويجية",
        "displayName-count-few";
        "كرونة نرويجية",
        "displayName-count-many";
        "كرونة نرويجية",
        "displayName-count-other";
        "كرونة نرويجية",
        "symbol";
        "NOK",
        "symbol-alt-narrow";
        "kr";
    }
    "NPR";
    {
        "displayName";
        "روبية نيبالي",
        "displayName-count-zero";

```

```

        "روبية نيبالي",
        "displayName-count-one";
        "روبية نيبالي",
        "displayName-count-two";
        "روبية نيبالي",
        "displayName-count-few";
        "روبية نيبالي",
        "displayName-count-many";
        "روبية نيبالي",
        "displayName-count-other";
        "روبية نيبالي",
        "symbol";
        "NPR",
        "symbol-alt-narrow";
        "Rs";
    }
    "NZD";
    {
        "displayName";
        "دولار نيوزيلندي",
        "displayName-count-zero";
        "دولار نيوزيلندي",
        "displayName-count-one";
        "دولار نيوزيلندي",
        "displayName-count-two";
        "دولار نيوزيلندي",
        "displayName-count-few";
        "دولار نيوزيلندي",
        "displayName-count-many";
        "دولار نيوزيلندي",
        "displayName-count-other";
        "دولار نيوزيلندي",
        "symbol";
        "NZ$",
        "symbol-alt-narrow";
        "NZ$";
    }
    "OMR";
    {
        "displayName";
        "رول عماني",
        "displayName-count-zero";
        "رول عماني",
        "displayName-count-one";
        "رول عماني",
        "displayName-count-two";
        "رول عماني",
        "displayName-count-few";
        "رول عماني",
        "displayName-count-many";
        "رول عماني",
        "displayName-count-other";
        "رول عماني",
        "symbol";
        "ر.ع.";
    }
    "PAB";

```

```

{
  "displayName";
  "بالبوا بنمي",
  "displayName-count-zero";
  "بالبوا بنمي",
  "displayName-count-one";
  "بالبوا بنمي",
  "displayName-count-two";
  "بالبوا بنمي",
  "displayName-count-few";
  "بالبوا بنمي",
  "displayName-count-many";
  "بالبوا بنمي",
  "displayName-count-other";
  "بالبوا بنمي",
  "symbol";
  "PAB";
}
"PEI";
{
  "displayName";
  "PEI",
  "symbol";
  "PEI";
}
"PEN";
{
  "displayName";
  "سول جديد البيرو",
  "displayName-count-zero";
  "سول جديد البيرو",
  "displayName-count-one";
  "سول جديد البيرو",
  "displayName-count-two";
  "سول جديد البيرو",
  "displayName-count-few";
  "سول جديد البيرو",
  "displayName-count-many";
  "سول جديد البيرو",
  "displayName-count-other";
  "سول جديد البيرو",
  "symbol";
  "PEN";
}
"PES";
{
  "displayName";
  "PES",
  "symbol";
  "PES";
}
"PGK";
{
  "displayName";
  "كيننا بابوا غينيا الجديدة",
  "displayName-count-zero";
  "كيننا بابوا غينيا الجديدة",

```

```

        "displayName-count-one";
        "كينا بابوا غينيا الجديدة",
        "displayName-count-two";
        "كينا بابوا غينيا الجديدة",
        "displayName-count-few";
        "كينا بابوا غينيا الجديدة",
        "displayName-count-many";
        "كينا بابوا غينيا الجديدة",
        "displayName-count-other";
        "كينا بابوا غينيا الجديدة",
        "symbol";
        "PGK";
    }
    "PHP";
    {
        "displayName";
        "بيزو فلبيني",
        "displayName-count-zero";
        "بيزو فلبيني",
        "displayName-count-one";
        "بيزو فلبيني",
        "displayName-count-two";
        "بيزو فلبيني",
        "displayName-count-few";
        "بيزو فلبيني",
        "displayName-count-many";
        "بيزو فلبيني",
        "displayName-count-other";
        "بيزو فلبيني",
        "symbol";
        "PHP",
        "symbol-alt-narrow";
        "₱";
    }
    "PKR";
    {
        "displayName";
        "روبية باكستاني",
        "displayName-count-zero";
        "روبية باكستاني",
        "displayName-count-one";
        "روبية باكستاني",
        "displayName-count-two";
        "روبية باكستاني",
        "displayName-count-few";
        "روبية باكستاني",
        "displayName-count-many";
        "روبية باكستاني",
        "displayName-count-other";
        "روبية باكستاني",
        "symbol";
        "PKR",
        "symbol-alt-narrow";
        "Rs";
    }
    "PLN";
    {

```

```

        "displayName";
        "زلوتي بولندي",
        "displayName-count-zero";
        "زلوتي بولندي",
        "displayName-count-one";
        "زلوتي بولندي",
        "displayName-count-two";
        "زلوتي بولندي",
        "displayName-count-few";
        "زلوتي بولندي",
        "displayName-count-many";
        "زلوتي بولندي",
        "displayName-count-other";
        "زلوتي بولندي",
        "symbol";
        "PLN",
        "symbol-alt-narrow";
        "zł";
    }
    "PLZ";
    {
        "displayName";
        "زلوتي بولندي - 1950-1995",
        "symbol";
        "PLZ";
    }
    "PTE";
    {
        "displayName";
        "اسكود برتغالي",
        "symbol";
        "PTE";
    }
    "PYG";
    {
        "displayName";
        "جواراني باراجواي",
        "displayName-count-zero";
        "جواراني باراجواي",
        "displayName-count-one";
        "جواراني باراجواي",
        "displayName-count-two";
        "جواراني باراجواي",
        "displayName-count-few";
        "جواراني باراجواي",
        "displayName-count-many";
        "جواراني باراجواي",
        "displayName-count-other";
        "جواراني باراجواي",
        "symbol";
        "PYG",
        "symbol-alt-narrow";
        "₲";
    }
    "QAR";
    {
        "displayName";

```

```

        "واڤ قطري",
        "displayName-count-zero";
        "واڤ قطري",
        "displayName-count-one";
        "واڤ قطري",
        "displayName-count-two";
        "واڤ قطري",
        "displayName-count-few";
        "واڤ قطري",
        "displayName-count-many";
        "واڤ قطري",
        "displayName-count-other";
        "واڤ قطري",
        "symbol";
        "ر.ق.";
    }
    "RHD";
    {
        "displayName";
        "دولار روڊيسي",
        "symbol";
        "RHD";
    }
    "ROL";
    {
        "displayName";
        "ليو روماني قديم",
        "symbol";
        "ROL";
    }
    "RON";
    {
        "displayName";
        "ليو روماني",
        "displayName-count-zero";
        "ليو روماني",
        "displayName-count-one";
        "ليو روماني",
        "displayName-count-two";
        "ليو روماني",
        "displayName-count-few";
        "ليو روماني",
        "displayName-count-many";
        "ليو روماني",
        "displayName-count-other";
        "ليو روماني",
        "symbol";
        "RON",
        "symbol-alt-narrow";
        "lei";
    }
    "RSD";
    {
        "displayName";
        "دينار صربي",
        "displayName-count-zero";
        "دينار صربي",

```



```

        "displayName-count-one";
        "دينار صربي",
        "displayName-count-two";
        "ديناران صربيان",
        "displayName-count-few";
        "دينارات صربية",
        "displayName-count-many";
        "دينارًا صربيًا",
        "displayName-count-other";
        "دينار صربي",
        "symbol";
        "RSD";
    }
    "RUB";
    {
        "displayName";
        "روبل روسي",
        "displayName-count-zero";
        "روبل روسي",
        "displayName-count-one";
        "روبل روسي",
        "displayName-count-two";
        "روبل روسي",
        "displayName-count-few";
        "روبل روسي",
        "displayName-count-many";
        "روبل روسي",
        "displayName-count-other";
        "روبل روسي",
        "symbol";
        "RUB",
        "symbol-alt-narrow";
        "₽";
    }
    "RUR";
    {
        "displayName";
        "1998-1991 - روبل روسي",
        "symbol";
        "RUR",
        "symbol-alt-narrow";
        "p.";
    }
    "RWF";
    {
        "displayName";
        "فرنك رواندي",
        "displayName-count-zero";
        "فرنك رواندي",
        "displayName-count-one";
        "فرنك رواندي",
        "displayName-count-two";
        "فرنك رواندي",
        "displayName-count-few";
        "فرنك رواندي",
        "displayName-count-many";
        "فرنك رواندي",
    }

```

```

        "displayName-count-other";
        "فرنك رواندي",
        "symbol";
        "RWF",
        "symbol-alt-narrow";
        "RF";
    }
    "SAR";
    {
        "displayName";
        "رول سعودي",
        "displayName-count-zero";
        "رول سعودي",
        "displayName-count-one";
        "رول سعودي",
        "displayName-count-two";
        "رول سعودي",
        "displayName-count-few";
        "رول سعودي",
        "displayName-count-many";
        "رول سعودي",
        "displayName-count-other";
        "رول سعودي",
        "symbol";
        "ر.س.";
    }
    "SBD";
    {
        "displayName";
        "دولار جزر سليمان",
        "displayName-count-zero";
        "دولار جزر سليمان",
        "displayName-count-one";
        "دولار جزر سليمان",
        "displayName-count-two";
        "دولار جزر سليمان",
        "displayName-count-few";
        "دولار جزر سليمان",
        "displayName-count-many";
        "دولار جزر سليمان",
        "displayName-count-other";
        "دولار جزر سليمان",
        "symbol";
        "SBD",
        "symbol-alt-narrow";
        "SB$";
    }
    "SCR";
    {
        "displayName";
        "روبية سيشيلية",
        "displayName-count-zero";
        "روبية سيشيلية",
        "displayName-count-one";
        "روبية سيشيلية",
        "displayName-count-two";
        "روبية سيشيلية",
    }

```

```

        "displayName-count-few";
        "روبية سيشيلية",
        "displayName-count-many";
        "روبية سيشيلية",
        "displayName-count-other";
        "روبية سيشيلية",
        "symbol";
        "SCR";
    }
    "SDD";
    {
        "displayName";
        "دينار سوداني",
        "symbol";
        "د.س.";
    }
    "SDG";
    {
        "displayName";
        "جنيه سوداني",
        "displayName-count-zero";
        "جنيه سوداني",
        "displayName-count-one";
        "جنيه سوداني",
        "displayName-count-two";
        "جنيه سوداني",
        "displayName-count-few";
        "جنيحات سودانية",
        "displayName-count-many";
        "جنيهاً سودانياً",
        "displayName-count-other";
        "جنيه سوداني",
        "symbol";
        "ج.س.";
    }
    "SDP";
    {
        "displayName";
        "جنيه سوداني قديم",
        "symbol";
        "SDP";
    }
    "SEK";
    {
        "displayName";
        "كرونة سويدية",
        "displayName-count-zero";
        "كرونة سويدية",
        "displayName-count-one";
        "كرونة سويدية",
        "displayName-count-two";
        "كرونة سويدية",
        "displayName-count-few";
        "كرونة سويدية",
        "displayName-count-many";
        "كرونة سويدية",
        "displayName-count-other";
    }

```

```

        "كرونة سويدية",
        "symbol";
    "SEK",
        "symbol-alt-narrow";
    "kr";
}
"SGD";
{
    "displayName";
    "دولار سنغافوري",
        "displayName-count-zero";
    "دولار سنغافوري",
        "displayName-count-one";
    "دولار سنغافوري",
        "displayName-count-two";
    "دولار سنغافوري",
        "displayName-count-few";
    "دولار سنغافوري",
        "displayName-count-many";
    "دولار سنغافوري",
        "displayName-count-other";
    "دولار سنغافوري",
        "symbol";
    "SGD",
        "symbol-alt-narrow";
    "$";
}
"SHP";
{
    "displayName";
    "جنيه سانت هيلين",
        "displayName-count-zero";
    "جنيه سانت هيلين",
        "displayName-count-one";
    "جنيه سانت هيلين",
        "displayName-count-two";
    "جنيه سانت هيلين",
        "displayName-count-few";
    "جنيه سانت هيلين",
        "displayName-count-many";
    "جنيه سانت هيلين",
        "displayName-count-other";
    "جنيه سانت هيلين",
        "symbol";
    "SHP",
        "symbol-alt-narrow";
    "£";
}
"SIT";
{
    "displayName";
    "تولار سلوفيني",
        "symbol";
    "SIT";
}
"SKK";
{

```

```

        "displayName";
        "كرونّة سلوفاكية",
        "symbol";
        "SKK";
    }
    "SLL";
    {
        "displayName";
        "ليون سيراليوني",
        "displayName-count-zero";
        "ليون سيراليوني",
        "displayName-count-one";
        "ليون سيراليوني",
        "displayName-count-two";
        "ليون سيراليوني",
        "displayName-count-few";
        "ليون سيراليوني",
        "displayName-count-many";
        "ليون سيراليوني",
        "displayName-count-other";
        "ليون سيراليوني",
        "symbol";
        "SLL";
    }
    "SOS";
    {
        "displayName";
        "شّلن صومالي",
        "displayName-count-zero";
        "شّلن صومالي",
        "displayName-count-one";
        "شّلن صومالي",
        "displayName-count-two";
        "شّلن صومالي",
        "displayName-count-few";
        "شّلن صومالي",
        "displayName-count-many";
        "شّلن صومالي",
        "displayName-count-other";
        "شّلن صومالي",
        "symbol";
        "SOS";
    }
    "SRD";
    {
        "displayName";
        "دولار سورينامي",
        "displayName-count-zero";
        "دولار سورينامي",
        "displayName-count-one";
        "دولار سورينامي",
        "displayName-count-two";
        "دولار سورينامي",
        "displayName-count-few";
        "دولار سورينامي",
        "displayName-count-many";
        "دولار سورينامي",
    }

```

```

        "displayName-count-other";
        "دولار سورينامي",
        "symbol";
        "SRD",
        "symbol-alt-narrow";
        "SR$";
    }
    "SRG";
    {
        "displayName";
        "جلدر سورينامي",
        "symbol";
        "SRG";
    }
    "SSP";
    {
        "displayName";
        "جنيه جنوب السودان",
        "displayName-count-zero";
        "جنيه جنوب السودان",
        "displayName-count-one";
        "جنيه جنوب السودان",
        "displayName-count-two";
        "جنيهان جنوب السودان",
        "displayName-count-few";
        "جنيهات جنوب السودان",
        "displayName-count-many";
        "جنيهًا جنوب السودان",
        "displayName-count-other";
        "جنيه جنوب السودان",
        "symbol";
        "SSP",
        "symbol-alt-narrow";
        "£";
    }
    "STD";
    {
        "displayName";
        "دوبرا ساو تومي وبرينسيبي",
        "displayName-count-zero";
        "دوبرا ساو تومي وبرينسيبي",
        "displayName-count-one";
        "دوبرا ساو تومي وبرينسيبي",
        "displayName-count-two";
        "دوبرا ساو تومي وبرينسيبي",
        "displayName-count-few";
        "دوبرا ساو تومي وبرينسيبي",
        "displayName-count-many";
        "دوبرا ساو تومي وبرينسيبي",
        "displayName-count-other";
        "دوبرا ساو تومي وبرينسيبي",
        "symbol";
        "STD",
        "symbol-alt-narrow";
        "Db";
    }
    "SUR";

```

```

    {
      "displayName";
      "روبل سوفيتي",
      "symbol";
      "SUR";
    }
    "SVC";
    {
      "displayName";
      "كولون سلفادوري",
      "symbol";
      "SVC";
    }
    "SYP";
    {
      "displayName";
      "ليرة سورية",
      "displayName-count-zero";
      "ليرة سورية",
      "displayName-count-one";
      "ليرة سورية",
      "displayName-count-two";
      "ليرة سورية",
      "displayName-count-few";
      "ليرة سورية",
      "displayName-count-many";
      "ليرة سورية",
      "displayName-count-other";
      "ليرة سورية",
      "symbol";
      "ل.س.",
      "symbol-alt-narrow";
      "£";
    }
    "SZL";
    {
      "displayName";
      "ليلانجيني سوازيلندي",
      "displayName-count-zero";
      "ليلانجيني سوازيلندي",
      "displayName-count-one";
      "ليلانجيني سوازيلندي",
      "displayName-count-two";
      "ليلانجيني سوازيلندي",
      "displayName-count-few";
      "ليلانجيني سوازيلندي",
      "displayName-count-many";
      "ليلانجيني سوازيلندي",
      "displayName-count-other";
      "ليلانجيني سوازيلندي",
      "symbol";
      "SZL";
    }
    "THB";
    {
      "displayName";
      "باخت تايلاندي",

```

```

        "displayName-count-zero";
        "باخت تایلاندي",
        "displayName-count-one";
        "باخت تایلاندي",
        "displayName-count-two";
        "باخت تایلاندي",
        "displayName-count-few";
        "باخت تایلاندي",
        "displayName-count-many";
        "باخت تایلاندي",
        "displayName-count-other";
        "باخت تایلاندي",
        "symbol";
        "฿",
        "symbol-alt-narrow";
        "฿";
    }
    "TJR";
    {
        "displayName";
        "روبل طاجیکستانی",
        "symbol";
        "TJR";
    }
    "TJS";
    {
        "displayName";
        "سومونی طاجیکستانی",
        "displayName-count-zero";
        "سومونی طاجیکستانی",
        "displayName-count-one";
        "سومونی طاجیکستانی",
        "displayName-count-two";
        "سومونی طاجیکستانی",
        "displayName-count-few";
        "سومونی طاجیکستانی",
        "displayName-count-many";
        "سومونی طاجیکستانی",
        "displayName-count-other";
        "سومونی طاجیکستانی",
        "symbol";
        "TJS";
    }
    "TMM";
    {
        "displayName";
        "مانات ترکمنستانی",
        "symbol";
        "TMM";
    }
    "TMT";
    {
        "displayName";
        "مانات ترکمانستان",
        "displayName-count-zero";
        "مانات ترکمانستان",
        "displayName-count-one";
    }

```



```

        "مانات ترکمانستان",
        "displayName-count-two";
        "مانات ترکمانستان",
        "displayName-count-few";
        "مانات ترکمانستان",
        "displayName-count-many";
        "مانات ترکمانستان",
        "displayName-count-other";
        "مانات ترکمانستان",
        "symbol";
        "TMT";
    }
    "TND";
    {
        "displayName";
        "دينار تونسي",
        "displayName-count-zero";
        "دينار تونسي",
        "displayName-count-one";
        "دينار تونسي",
        "displayName-count-two";
        "ديناران تونسيان",
        "displayName-count-few";
        "دينارات تونسية",
        "displayName-count-many";
        "دينارًا تونسيًا",
        "displayName-count-other";
        "دينار تونسي",
        "symbol";
        "د.ت.";
    }
    "TOP";
    {
        "displayName";
        "بانغا تونغا",
        "displayName-count-zero";
        "بانغا تونغا",
        "displayName-count-one";
        "بانغا تونغا",
        "displayName-count-two";
        "بانغا تونغا",
        "displayName-count-few";
        "بانغا تونغا",
        "displayName-count-many";
        "بانغا تونغا",
        "displayName-count-other";
        "بانغا تونغا",
        "symbol";
        "TOP",
        "symbol-alt-narrow";
        "T$";
    }
    "TPE";
    {
        "displayName";
        "اسكود تيموري",
        "symbol";
    }

```

```
        "TPE";
    }
    "TRL";
    {
        "displayName";
        "ليرة تركي",
        "symbol";
        "TRL";
    }
    "TRY";
    {
        "displayName";
        "ليرة تركية",
        "displayName-count-zero";
        "ليرة تركية",
        "displayName-count-one";
        "ليرة تركية",
        "displayName-count-two";
        "ليرة تركية",
        "displayName-count-few";
        "ليرة تركية",
        "displayName-count-many";
        "ليرة تركية",
        "displayName-count-other";
        "ليرة تركية",
        "symbol";
        "TRY",
        "symbol-alt-narrow";
        "₺",
        "symbol-alt-variant";
        "TL";
    }
    "TTD";
    {
        "displayName";
        "دولار ترينداد وتوباغو",
        "displayName-count-zero";
        "دولار ترينداد وتوباغو",
        "displayName-count-one";
        "دولار ترينداد وتوباغو",
        "displayName-count-two";
        "دولار ترينداد وتوباغو",
        "displayName-count-few";
        "دولار ترينداد وتوباغو",
        "displayName-count-many";
        "دولار ترينداد وتوباغو",
        "displayName-count-other";
        "دولار ترينداد وتوباغو",
        "symbol";
        "TTD",
        "symbol-alt-narrow";
        "TT$";
    }
    "TWD";
    {
        "displayName";
        "دولار تايواني",
```

```

        "displayName-count-zero";
        "دولار تایوانی",
        "displayName-count-one";
        "دولار تایوانی",
        "displayName-count-two";
        "دولار تایوانی",
        "displayName-count-few";
        "دولار تایوانی",
        "displayName-count-many";
        "دولار تایوانی",
        "displayName-count-other";
        "دولار تایوانی",
        "symbol";
        "NT$";
        "symbol-alt-narrow";
        "NT$";
    }
    "TZS";
    {
        "displayName";
        "شلن تنزانی",
        "displayName-count-zero";
        "شلن تنزانی",
        "displayName-count-one";
        "شلن تنزانی",
        "displayName-count-two";
        "شلن تنزانی",
        "displayName-count-few";
        "شلن تنزانی",
        "displayName-count-many";
        "شلن تنزانی",
        "displayName-count-other";
        "شلن تنزانی",
        "symbol";
        "TZS";
    }
    "UAH";
    {
        "displayName";
        "هریفنیا اوکرانی",
        "displayName-count-zero";
        "هریفنیا اوکرانی",
        "displayName-count-one";
        "هریفنیا اوکرانی",
        "displayName-count-two";
        "هریفنیا اوکرانی",
        "displayName-count-few";
        "هریفنیا اوکرانی",
        "displayName-count-many";
        "هریفنیا اوکرانی",
        "displayName-count-other";
        "هریفنیا اوکرانی",
        "symbol";
        "UAH";
        "symbol-alt-narrow";
        "₴";
    }
}

```

```
"UAK";
{
  "displayName";
  "UAK",
  "symbol";
  "UAK";
}
"UGS";
{
  "displayName";
  "1987-1966 - شلن أوغندي",
  "symbol";
  "UGS";
}
"UGX";
{
  "displayName";
  "شلن أوغندي",
  "displayName-count-zero";
  "شلن أوغندي",
  "displayName-count-one";
  "شلن أوغندي",
  "displayName-count-two";
  "شلن أوغندي",
  "displayName-count-few";
  "شلن أوغندي",
  "displayName-count-many";
  "شلن أوغندي",
  "displayName-count-other";
  "شلن أوغندي",
  "symbol";
  "UGX";
}
"USD";
{
  "displayName";
  "دولار أمريكي",
  "displayName-count-zero";
  "دولار أمريكي",
  "displayName-count-one";
  "دولار أمريكي",
  "displayName-count-two";
  "دولار أمريكي",
  "displayName-count-few";
  "دولار أمريكي",
  "displayName-count-many";
  "دولار أمريكي",
  "displayName-count-other";
  "دولار أمريكي",
  "symbol";
  "US$",
  "symbol-alt-narrow";
  "US$";
}
"USN";
{
  "displayName";
```

```

        "دولار أمريكي (اليوم التالي)",
        "symbol";
        "USN";
    }
    "USS";
    {
        "displayName";
        "دولار أمريكي (نفس اليوم)",
        "symbol";
        "USS";
    }
    "UYI";
    {
        "displayName";
        "UYI",
        "symbol";
        "UYI";
    }
    "UYP";
    {
        "displayName";
        "بيزو أوروغواي - 1993-1975",
        "symbol";
        "UYP";
    }
    "UYU";
    {
        "displayName";
        "بيزو أوروغواي",
        "displayName-count-zero";
        "بيزو أوروغواي",
        "displayName-count-one";
        "بيزو أوروغواي",
        "displayName-count-two";
        "بيزو أوروغواي",
        "displayName-count-few";
        "بيزو أوروغواي",
        "displayName-count-many";
        "بيزو أوروغواي",
        "displayName-count-other";
        "بيزو أوروغواي",
        "symbol";
        "UYU",
        "symbol-alt-narrow";
        "UY$";
    }
    "UZS";
    {
        "displayName";
        "سوم أوزبكستاني",
        "displayName-count-zero";
        "سوم أوزبكستاني",
        "displayName-count-one";
        "سوم أوزبكستاني",
        "displayName-count-two";
        "سوم أوزبكستاني",
        "displayName-count-few";
    }

```

```

        "سوم أوزبكستاني",
        "displayName-count-many";
        "سوم أوزبكستاني",
        "displayName-count-other";
        "سوم أوزبكستاني",
        "symbol";
        "UZS";
    }
    "VEB";
    {
        "displayName";
        "بوليفار فنزويلي - 2008-1871",
        "symbol";
        "VEB";
    }
    "VEF";
    {
        "displayName";
        "بوليفار فنزويلي",
        "displayName-count-zero";
        "بوليفار فنزويلي",
        "displayName-count-one";
        "بوليفار فنزويلي",
        "displayName-count-two";
        "بوليفار فنزويلي",
        "displayName-count-few";
        "بوليفار فنزويلي",
        "displayName-count-many";
        "بوليفار فنزويلي",
        "displayName-count-other";
        "بوليفار فنزويلي",
        "symbol";
        "VEF",
        "symbol-alt-narrow";
        "Bs";
    }
    "VND";
    {
        "displayName";
        "دونج فيتنامي",
        "displayName-count-zero";
        "دونج فيتنامي",
        "displayName-count-one";
        "دونج فيتنامي",
        "displayName-count-two";
        "دونج فيتنامي",
        "displayName-count-few";
        "دونج فيتنامي",
        "displayName-count-many";
        "دونج فيتنامي",
        "displayName-count-other";
        "دونج فيتنامي",
        "symbol";
        "₫",
        "symbol-alt-narrow";
        "₫";
    }
}

```

```

"VNN";
{
  "displayName";
  "VNN",
  "symbol";
  "VNN";
}
"VUV";
{
  "displayName";
  "فاتو فانواتو",
  "displayName-count-zero";
  "فاتو فانواتو",
  "displayName-count-one";
  "فاتو فانواتو",
  "displayName-count-two";
  "فاتو فانواتو",
  "displayName-count-few";
  "فاتو فانواتو",
  "displayName-count-many";
  "فاتو فانواتو",
  "displayName-count-other";
  "فاتو فانواتو",
  "symbol";
  "VUV";
}
"WST";
{
  "displayName";
  "تالا ساموا",
  "displayName-count-zero";
  "تالا ساموا",
  "displayName-count-one";
  "تالا ساموا",
  "displayName-count-two";
  "تالا ساموا",
  "displayName-count-few";
  "تالا ساموا",
  "displayName-count-many";
  "تالا ساموا",
  "displayName-count-other";
  "تالا ساموا",
  "symbol";
  "WST";
}
"XAF";
{
  "displayName";
  "فرنك وسط أفريقي",
  "displayName-count-zero";
  "فرنك وسط أفريقي",
  "displayName-count-one";
  "فرنك وسط أفريقي",
  "displayName-count-two";
  "فرنك وسط أفريقي",
  "displayName-count-few";
  "فرنك وسط أفريقي",

```

```

        "displayName-count-many";
        "فرنك وسط أفريقي",
        "displayName-count-other";
        "فرنك وسط أفريقي",
        "symbol";
        "FCFA";
    }
    "XAG";
    {
        "displayName";
        "فضة",
        "symbol";
        "XAG";
    }
    "XAU";
    {
        "displayName";
        "ذهب",
        "symbol";
        "XAU";
    }
    "XBA";
    {
        "displayName";
        "الوحدة الأوروبية المركبة",
        "symbol";
        "XBA";
    }
    "XBB";
    {
        "displayName";
        "الوحدة المالية الأوروبية",
        "symbol";
        "XBB";
    }
    "XBC";
    {
        "displayName";
        "الوحدة الحسابية الأوروبية",
        "symbol";
        "XBC";
    }
    "XBD";
    {
        "displayName";
        "وحدة الحساب الأوروبية (XBD)",
        "symbol";
        "XBD";
    }
    "XCD";
    {
        "displayName";
        "دولار شرق الكاريبي",
        "displayName-count-zero";
        "دولار شرق الكاريبي",
        "displayName-count-one";
        "دولار شرق الكاريبي",

```



```

        "displayName-count-two";
        "دولار شرق الكاريبي",
        "displayName-count-few";
        "دولار شرق الكاريبي",
        "displayName-count-many";
        "دولار شرق الكاريبي",
        "displayName-count-other";
        "دولار شرق الكاريبي",
        "symbol";
        "EC$";
        "symbol-alt-narrow";
        "$";
    }
    "XDR";
    {
        "displayName";
        "حقوق السحب الخاصة",
        "symbol";
        "XDR";
    }
    "XEU";
    {
        "displayName";
        "وحدة النقد الأوروبية",
        "symbol";
        "XEU";
    }
    "XFO";
    {
        "displayName";
        "فرنك فرنسي ذهبي",
        "symbol";
        "XFO";
    }
    "XFU";
    {
        "displayName";
        "فرنك فرنسي (UIC)",
        "symbol";
        "XFU";
    }
    "XOF";
    {
        "displayName";
        "فرنك غرب أفريقي",
        "displayName-count-zero";
        "فرنك غرب أفريقي",
        "displayName-count-one";
        "فرنك غرب أفريقي",
        "displayName-count-two";
        "فرنك غرب أفريقي",
        "displayName-count-few";
        "فرنك غرب أفريقي",
        "displayName-count-many";
        "فرنك غرب أفريقي",
        "displayName-count-other";
        "فرنك غرب أفريقي",
    }

```

```
        "symbol";
        "CFA";
    }
    "XPD";
    {
        "displayName";
        "بالاديوم",
        "symbol";
        "XPD";
    }
    "XPF";
    {
        "displayName";
        "فرنك سي إف بي",
        "displayName-count-zero";
        "فرنك سي إف بي",
        "displayName-count-one";
        "فرنك سي إف بي",
        "displayName-count-two";
        "فرنك سي إف بي",
        "displayName-count-few";
        "فرنك سي إف بي",
        "displayName-count-many";
        "فرنك سي إف بي",
        "displayName-count-other";
        "فرنك سي إف بي",
        "symbol";
        "CFPF";
    }
    "XPT";
    {
        "displayName";
        "البلاطين",
        "symbol";
        "XPT";
    }
    "XRE";
    {
        "displayName";
        "XRE",
        "symbol";
        "XRE";
    }
    "XSU";
    {
        "displayName";
        "XSU",
        "symbol";
        "XSU";
    }
    "XTS";
    {
        "displayName";
        "كود اختبار العملة",
        "symbol";
        "XTS";
    }
}
```

```

    "XUA";
    {
      "displayName";
      "XUA",
      "symbol";
      "XUA";
    }
    "XXX";
    {
      "displayName";
      "عملة غير معروفة",
      "displayName-count-zero";
      "(عملة غير معروفة)",
      "displayName-count-one";
      "(عملة غير معروفة)",
      "displayName-count-two";
      "(عملة غير معروفة)",
      "displayName-count-few";
      "(عملة غير معروفة)",
      "displayName-count-many";
      "(عملة غير معروفة)",
      "displayName-count-other";
      "(عملة غير معروفة)",
      "symbol";
      "***";
    }
    "YDD";
    {
      "displayName";
      "دينار يمني",
      "symbol";
      "YDD";
    }
    "YER";
    {
      "displayName";
      "ول يمني",
      "displayName-count-zero";
      "ول يمني",
      "displayName-count-one";
      "ول يمني",
      "displayName-count-two";
      "ول يمني",
      "displayName-count-few";
      "ول يمني",
      "displayName-count-many";
      "ول يمني",
      "displayName-count-other";
      "ول يمني",
      "symbol";
      "ر.ي.";
    }
    "YUD";
    {
      "displayName";
      "دينار يوغسلافي",
      "symbol";

```

```

        "YUD";
    }
    "YUM";
    {
        "displayName";
        "YUM",
        "symbol";
        "YUM";
    }
    "YUN";
    {
        "displayName";
        "دينار يوغسلافي قابل للتحويل",
        "symbol";
        "YUN";
    }
    "YUR";
    {
        "displayName";
        "YUR",
        "symbol";
        "YUR";
    }
    "ZAL";
    {
        "displayName";
        "راند جنوب أفريقيا -مالي",
        "symbol";
        "ZAL";
    }
    "ZAR";
    {
        "displayName";
        "راند جنوب أفريقيا",
        "displayName-count-zero";
        "راند جنوب أفريقيا",
        "displayName-count-one";
        "راند جنوب أفريقيا",
        "displayName-count-two";
        "راند جنوب أفريقيا",
        "displayName-count-few";
        "راند جنوب أفريقيا",
        "displayName-count-many";
        "راند جنوب أفريقيا",
        "displayName-count-other";
        "راند جنوب أفريقيا",
        "symbol";
        "ZAR",
        "symbol-alt-narrow";
        "R";
    }
    "ZMK";
    {
        "displayName";
        "كواشا زامبي - 2012-1968",
        "displayName-count-zero";
        "كواشا زامبي - 2012-1968",

```

```

        "displayName-count-one";
        "2012-1968 - كواشا زامبي",
        "displayName-count-two";
        "2012-1968 - كواشا زامبي",
        "displayName-count-few";
        "2012-1968 - كواشا زامبي",
        "displayName-count-many";
        "2012-1968 - كواشا زامبي",
        "displayName-count-other";
        "2012-1968 - كواشا زامبي",
        "symbol";
        "ZMK";
    }
    "ZMW";
    {
        "displayName";
        "كواشا زامبي",
        "displayName-count-zero";
        "كواشا زامبي",
        "displayName-count-one";
        "كواشا زامبي",
        "displayName-count-two";
        "كواشا زامبي",
        "displayName-count-few";
        "كواشا زامبي",
        "displayName-count-many";
        "كواشا زامبي",
        "displayName-count-other";
        "كواشا زامبي",
        "symbol";
        "ZMW",
        "symbol-alt-narrow";
        "ZK";
    }
    "ZRN";
    {
        "displayName";
        "زائير زائيري جديد",
        "symbol";
        "ZRN";
    }
    "ZRZ";
    {
        "displayName";
        "زائير زائيري",
        "symbol";
        "ZRZ";
    }
    "ZWD";
    {
        "displayName";
        "دولار زمبابوي",
        "symbol";
        "ZWD";
    }
    "ZWL";
    {

```

```
"displayName";  
    "دولار زمبابوي 2009",  
        "symbol";  
    "ZWL";  
}  
"ZWR";  
{  
    "displayName";  
    "ZWR",  
        "symbol";  
    "ZWR";  
}  
}  
}  
}
```

## INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
//Load the L10n, loadCldr from ej2-base
import { loadCldr, L10n } from '@syncfusion/ej2-base';
//import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
//import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
// enable ripple effect
enableRipple(true);
//load the CLDR data files.
// Here we have referred local json files for preview purpose
import * as numberingSystems from './numberingSystems.json';
import * as gregorian from './ca-gregorian.json';
import * as numbers from './numbers.json';
loadCldr(numberingSystems, gregorian, numbers);
L10n.load({
    'ar': {
        'timepicker': { placeholder: 'حدد الوقت' }
    }
});
//import the timepickercomponent
class App extends React.Component {
    rtl = true;
    render() {
        return <TimePickerComponent id="time" locale='ar'
enableRtl={this.rtl}/>;
    }
}
;
ReactDOM.render(<App />, document.getElementById('timer'));
```

**INDEX.TSX**

```
import * as React from "react";
import * as ReactDOM from "react-dom";
```

```

//Load the L10n, loadCldr from ej2-base
import { loadCldr, L10n } from '@syncfusion/ej2-base';
//import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
//import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
// enable ripple effect
enableRipple(true);
//load the CLDR data files.
// Here we have referred local json files for preview purpose
import * as numberingSystems from './numberingSystems.json';
import * as gregorian from './ca-gregorian.json';
import * as numbers from './numbers.json';
loadCldr(numberingSystems, gregorian, numbers);
L10n.load({
    'ar': {
        'timepicker': { placeholder: 'حدد الوقت' }
    }
});
//import the timepickercomponent
class App extends React.Component<{}, {}> {
    private rtl:boolean=true;
    render() {
        return <TimePickerComponent id="time" locale='ar'
enableRtl={this.rtl} />
    }
};
ReactDOM.render(<App />, document.getElementById('timer'));

```

### MAIN.JSX

```

import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
import { enableProdMode } from '@angular/core';
import { AppModule } from './app.module';
enableProdMode();
platformBrowserDynamic().bootstrapModule(AppModule);

```

### MAIN.TS

```

import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
import { enableProdMode } from '@angular/core';
import { AppModule } from './app.module';
enableProdMode();
platformBrowserDynamic().bootstrapModule(AppModule);

```

### NUMBERINGSYSTEMS.JSON

```

{
  "supplemental": {
    "version": {
      "_number": "$Revision: 12732 $",
      "_unicodeVersion": "9.0.0",
      "_cldrVersion": "31"
    },
    "numberingSystems": {

```

```

"adlm": {
  "_digits": "ᲠᲡᲢᲣᲤᲦᲧᲨᲩ",
  "_type": "numeric"
},
"ahom": {
  "_digits": "᱀᱁᱂᱃᱄᱅᱆᱇᱈᱉",
  "_type": "numeric"
},
"arab": {
  "_digits": "٠١٢٣٤٥٦٧٨٩",
  "_type": "numeric"
},
"arabext": {
  "_digits": "٠١٢٣٤٥٦٧٨٩",
  "_type": "numeric"
},
"armn": {
  "_rules": "armenian-upper",
  "_type": "algorithmic"
},
"armnlow": {
  "_rules": "armenian-lower",
  "_type": "algorithmic"
},
"bali": {
  "_digits": "ᬀᬁᬂᬃᬄᬅᬆᬇᬈᬉ",
  "_type": "numeric"
},
"beng": {
  "_digits": "০১২৩৪৫৬৭৮৯",
  "_type": "numeric"
},
"bhks": {
  "_digits": "᱀᱁᱂᱃᱄᱅᱆᱇᱈᱉",
  "_type": "numeric"
},
"brah": {
  "_digits": "᱀᱁᱂᱃᱄᱅᱆᱇᱈᱉",
  "_type": "numeric"
},
"cakm": {
  "_digits": "᱀᱁᱂᱃᱄᱅᱆᱇᱈᱉",
  "_type": "numeric"
},
"cham": {
  "_digits": "᱀᱁᱂᱃᱄᱅᱆᱇᱈᱉",
  "_type": "numeric"
},
"cyrl": {
  "_rules": "cyrillic-lower",
  "_type": "algorithmic"
},
"deva": {
  "_digits": "०१२३४५६७८९",
  "_type": "numeric"
},

```



```
"ethi": {
  "_rules": "ethiopic",
  "_type": "algorithmic"
},
"fullwide": {
  "_digits": "0 1 2 3 4 5 6 7 8 9",
  "_type": "numeric"
},
"geor": {
  "_rules": "georgian",
  "_type": "algorithmic"
},
"grek": {
  "_rules": "greek-upper",
  "_type": "algorithmic"
},
"greklow": {
  "_rules": "greek-lower",
  "_type": "algorithmic"
},
"gujr": {
  "_digits": "૦ ૧ ૨ ૩ ૪ ૫ ૬ ૭ ૮ ૯",
  "_type": "numeric"
},
"guru": {
  "_digits": "੦ ੧ ੨ ੩ ੪ ੫ ੬ ੭ ੮ ੯",
  "_type": "numeric"
},
"hanidays": {
  "_rules": "zh/SpelloutRules/spellout-numbering-days",
  "_type": "algorithmic"
},
"hanidec": {
  "_digits": "〇 一 二 三 四 五 六 七 八 九",
  "_type": "numeric"
},
"hans": {
  "_rules": "zh/SpelloutRules/spellout-cardinal",
  "_type": "algorithmic"
},
"hansfin": {
  "_rules": "zh/SpelloutRules/spellout-cardinal-financial",
  "_type": "algorithmic"
},
"hant": {
  "_rules": "zh_Hant/SpelloutRules/spellout-cardinal",
  "_type": "algorithmic"
},
"hantfin": {
  "_rules": "zh_Hant/SpelloutRules/spellout-cardinal-financial",
  "_type": "algorithmic"
},
"hebr": {
  "_rules": "hebrew",
  "_type": "algorithmic"
},
}
```

```

"hmng": {
  "_digits": "□□□□□□□□□□",
  "_type": "numeric"
},
"java": {
  "_digits": "၀၈၈၅၅၅၆၆၆၆၆၆၆၆၆",
  "_type": "numeric"
},
"jpan": {
  "_rules": "ja/SpelloutRules/spellout-cardinal",
  "_type": "algorithmic"
},
"jpanfin": {
  "_rules": "ja/SpelloutRules/spellout-cardinal-financial",
  "_type": "algorithmic"
},
"kali": {
  "_digits": "□□□□□□□□□□",
  "_type": "numeric"
},
"khmr": {
  "_digits": "០១២៣៤៥៦៧៨៩",
  "_type": "numeric"
},
"knda": {
  "_digits": "೦೧೨೩೪೫೬೭೮೯",
  "_type": "numeric"
},
"lana": {
  "_digits": "□□□□□□□□□□",
  "_type": "numeric"
},
"lanatham": {
  "_digits": "□□□□□□□□□□",
  "_type": "numeric"
},
"laoo": {
  "_digits": "໐໑໒໓໔໕໖໗໘໑",
  "_type": "numeric"
},
"latn": {
  "_digits": "0123456789",
  "_type": "numeric"
},
"lepc": {
  "_digits": "□□□□□□□□□□",
  "_type": "numeric"
},
"limb": {
  "_digits": "□□□□□□□□□□",
  "_type": "numeric"
},
"mathbold": {
  "_digits": "0123456789",
  "_type": "numeric"
}

```

```
},
"mathtdbl": {
  "_digits": "0123456789",
  "_type": "numeric"
},
"mathmono": {
  "_digits": "0123456789",
  "_type": "numeric"
},
"mathsanb": {
  "_digits": "0123456789",
  "_type": "numeric"
},
"mathsans": {
  "_digits": "0123456789",
  "_type": "numeric"
},
"mlym": {
  "_digits": "൦൧൨൩൪൫൬൭൮൯",
  "_type": "numeric"
},
"modi": {
  "_digits": "□□□□□□□□□□",
  "_type": "numeric"
},
"mong": {
  "_digits": "᠐᠑᠒᠓᠐ᠠᠨᠳᠤᠯᠤᠰ",
  "_type": "numeric"
},
"mroo": {
  "_digits": "□□□□□□□□□□",
  "_type": "numeric"
},
"mtei": {
  "_digits": "ᐐᐑᐒᐓᐔᐕᐖᐗᐘᐙᐚ",
  "_type": "numeric"
},
"mymr": {
  "_digits": "၀၁၂၃၄၅၆၇၈၉",
  "_type": "numeric"
},
"mymrshan": {
  "_digits": "0123456789",
  "_type": "numeric"
},
"mymrtlng": {
  "_digits": "ᐐᐑᐒᐓᐔᐕᐖᐗᐘᐙᐚ",
  "_type": "numeric"
},
"newa": {
  "_digits": "□□□□□□□□□□",
  "_type": "numeric"
},
"nkoo": {
```

```

    "_digits": "᠑ᠡᠨᠢᠯᠠᠭᠤᠨᠠᠨᠠᠭᠤᠨ",
    "_type": "numeric"
  },
  "olck": {
    "_digits": "᠐ᠯᠠᠭᠤᠨᠠᠨᠠᠭᠤᠨ",
    "_type": "numeric"
  },
  "orya": {
    "_digits": "᠐ᠷᠢᠶᠠᠨᠠᠨᠠᠭᠤᠨ",
    "_type": "numeric"
  },
  "osma": {
    "_digits": "᠐ᠰᠤᠮᠠᠨᠠᠨᠠᠭᠤᠨ",
    "_type": "numeric"
  },
  "roman": {
    "_rules": "roman-upper",
    "_type": "algorithmic"
  },
  "romanlow": {
    "_rules": "roman-lower",
    "_type": "algorithmic"
  },
  "saur": {
    "_digits": "ᠰᠠᠭᠤᠷᠠᠨᠠᠨᠠᠭᠤᠨ",
    "_type": "numeric"
  },
  "shrd": {
    "_digits": "ᠰᠢᠷᠳᠠᠨᠠᠨᠠᠭᠤᠨ",
    "_type": "numeric"
  },
  "sind": {
    "_digits": "ᠰᠢᠨᠳᠠᠨᠠᠨᠠᠭᠤᠨ",
    "_type": "numeric"
  },
  "sinh": {
    "_digits": "ᠰᠢᠨᠠᠨᠠᠭᠤᠨᠠᠨᠠᠭᠤᠨ",
    "_type": "numeric"
  },
  "sora": {
    "_digits": "ᠰᠣᠷᠠᠨᠠᠨᠠᠭᠤᠨ",
    "_type": "numeric"
  },
  "sund": {
    "_digits": "ᠰᠤᠨᠳᠠᠨᠠᠨᠠᠭᠤᠨ",
    "_type": "numeric"
  },
  "takr": {
    "_digits": "ᠲᠠᠭᠤᠷᠠᠨᠠᠨᠠᠭᠤᠨ",
    "_type": "numeric"
  },
  "talv": {
    "_digits": "ᠲᠠᠯᠠᠨᠠᠨᠠᠭᠤᠨ",
    "_type": "numeric"
  },

```

```

    "taml": {
      "_rules": "tamil",
      "_type": "algorithmic"
    },
    "tamldec": {
      "_digits": "0௧௨௩௪௫௬௭௮௯",
      "_type": "numeric"
    },
    "telu": {
      "_digits": "౦౧౨౩౪౫౬౭౮౯",
      "_type": "numeric"
    },
    "thai": {
      "_digits": "๐๑๒๓๔๕๖๗๘๙",
      "_type": "numeric"
    },
    "tibb": {
      "_digits": "༠༡༢༣༤༥༦༧༨༩",
      "_type": "numeric"
    },
    "tinh": {
      "_digits": "0123456789",
      "_type": "numeric"
    },
    "vaih": {
      "_digits": "0123456789",
      "_type": "numeric"
    },
    "wara": {
      "_digits": "0123456789",
      "_type": "numeric"
    }
  }
}

```

### NUMBERINGSYSTEMS.JSX

```

{
  "supplemental";
  {
    "version";
    {
      "_number";
      "$Revision: 12732 $",
      "_unicodeVersion";
      "9.0.0",
      "_cldrVersion";
      "31";
    }
    "numberingSystems";
    {
      "adlm";
      {
        "_digits";

```

```

        "୧୨୩୪୫୬୭୮୯୦",
        "_type";
        "numeric";
    }
    "ahom";
    {
        "_digits";
        "ᱠᱡᱢᱣᱤᱨᱫᱽᱜᱽᱲᱤ",
        "_type";
        "numeric";
    }
    "arab";
    {
        "_digits";
        "٠١٢٣٤٥٦٧٨٩",
        "_type";
        "numeric";
    }
    "arabext";
    {
        "_digits";
        "٠١٢٣٤٥٦٧٨٩",
        "_type";
        "numeric";
    }
    "armn";
    {
        "_rules";
        "armenian-upper",
        "_type";
        "algorithmic";
    }
    "armnlow";
    {
        "_rules";
        "armenian-lower",
        "_type";
        "algorithmic";
    }
    "bali";
    {
        "_digits";
        "᭐᭄ᭅᭆᭇᭈᭉᭊᭋᭌ᭍᭎᭏",
        "_type";
        "numeric";
    }
    "beng";
    {
        "_digits";
        "୦୧୨୩୪୫୬୭୮୯",
        "_type";
        "numeric";
    }
    "bhks";
    {
        "_digits";

```

```

        "□□□□□□□□",
        "_type";
        "numeric";
    }
    "brah";
    {
        "_digits";
        "ᱠᱡᱢᱯᱤᱨᱫᱽᱜᱟᱲ",
        "_type";
        "numeric";
    }
    "cakm";
    {
        "_digits";
        "ᱵᱚᱠᱟᱨᱢᱟᱝ",
        "_type";
        "numeric";
    }
    "cham";
    {
        "_digits";
        "ᱠᱡᱢᱯᱤᱨᱫᱽᱜᱟᱲ",
        "_type";
        "numeric";
    }
    "cyr1";
    {
        "_rules";
        "cyrillic-lower",
        "_type";
        "algorithmic";
    }
    "deva";
    {
        "_digits";
        "ᱠᱡᱢᱯᱤᱨᱫᱽᱜᱟᱲ",
        "_type";
        "numeric";
    }
    "ethi";
    {
        "_rules";
        "ethiopic",
        "_type";
        "algorithmic";
    }
    "fullwide";
    {
        "_digits";
        "0 1 2 3 4 5 6 7 8 9",
        "_type";
        "numeric";
    }
    "geor";
    {
        "_rules";

```

```

        "georgian",
        "_type";
        "algorithmic";
    }
    "grek";
    {
        "_rules";
        "greek-upper",
        "_type";
        "algorithmic";
    }
    "greklow";
    {
        "_rules";
        "greek-lower",
        "_type";
        "algorithmic";
    }
    "gujr";
    {
        "_digits";
        "૦૧૨૩૪૫૬૭૮૯",
        "_type";
        "numeric";
    }
    "guru";
    {
        "_digits";
        "੦੧੨੩੪੫੬੭੮੯",
        "_type";
        "numeric";
    }
    "hanidays";
    {
        "_rules";
        "zh/SpelloutRules/spellout-numbering-days",
        "_type";
        "algorithmic";
    }
    "hanidec";
    {
        "_digits";
        "〇一二三四五六七八九",
        "_type";
        "numeric";
    }
    "hans";
    {
        "_rules";
        "zh/SpelloutRules/spellout-cardinal",
        "_type";
        "algorithmic";
    }
    "hansfin";
    {
        "_rules";

```



```

"zh/SpelloutRules/spellout-cardinal",
    "_type";
    "algorithmic";
}
"hant";
{
    "_rules";
    "zh_Hant/SpelloutRules/spellout-cardinal",
        "_type";
        "algorithmic";
}
"hantfin";
{
    "_rules";
    "zh_Hant/SpelloutRules/spellout-cardinal-financial",
        "_type";
        "algorithmic";
}
"hebr";
{
    "_rules";
    "hebrew",
        "_type";
        "algorithmic";
}
"hmng";
{
    "_digits";
    "□□□□□□□□□□",
        "_type";
        "numeric";
}
"java";
{
    "_digits";
    "᐀ᐁᐂᐃᐄᐅᐆᐇᐈᐉᐊᐋᐌᐍᐎᐏᐐᐑᐒᐓᐔᐕᐖᐗᐘᐙᐚᐛᐜᐝᐞᐟᐠᐡᐢᐣᐤᐥᐦᐧᐨᐩᐪᐫᐬᐭᐮᐯᐰᐱᐲᐳᐴᐵᐶᐷᐸᐹᐺᐻᐼᐽᐾᐿᐿ",
        "_type";
        "numeric";
}
"jpan";
{
    "_rules";
    "ja/SpelloutRules/spellout-cardinal",
        "_type";
        "algorithmic";
}
"jpanfin";
{
    "_rules";
    "ja/SpelloutRules/spellout-cardinal-financial",
        "_type";
        "algorithmic";
}
"kali";
{
    " digits";

```

```

        "□□□□□□□□",
        "_type";
        "numeric";
    }
    "khmr";
    {
        "_digits";
        "០១២៣៤៥៦៧៨៩",
        "_type";
        "numeric";
    }
    "knda";
    {
        "_digits";
        "೦೧೨೩೪೫೬೭೮೯",
        "_type";
        "numeric";
    }
    "lana";
    {
        "_digits";
        "□□□□□□□□",
        "_type";
        "numeric";
    }
    "lanatham";
    {
        "_digits";
        "□□□□□□□□",
        "_type";
        "numeric";
    }
    "laoo";
    {
        "_digits";
        "໐໑໒໓໔໕໖໗໘໙",
        "_type";
        "numeric";
    }
    "latn";
    {
        "_digits";
        "0123456789",
        "_type";
        "numeric";
    }
    "lepc";
    {
        "_digits";
        "□□□□□□□□",
        "_type";
        "numeric";
    }
    "limb";
    {
        "_digits";

```

```

        "□□□□□□□□",
        "_type";
        "numeric";
    }
    "mathbold";
    {
        "_digits";
        "0123456789",
        "_type";
        "numeric";
    }
    "mathdbl";
    {
        "_digits";
        "0123456789",
        "_type";
        "numeric";
    }
    "mathmono";
    {
        "_digits";
        "0123456789",
        "_type";
        "numeric";
    }
    "mathsanb";
    {
        "_digits";
        "0123456789",
        "_type";
        "numeric";
    }
    "mathsans";
    {
        "_digits";
        "0123456789",
        "_type";
        "numeric";
    }
    "mlym";
    {
        "_digits";
        "൦൧൨൩൪൫൬൭൮൯",
        "_type";
        "numeric";
    }
    "modi";
    {
        "_digits";
        "□□□□□□□□",
        "_type";
        "numeric";
    }
    "mong";
    {
        "_digits";

```

```

        "၀၅၂၃၂၂၅၆၇၈",
        "_type";
        "numeric";
    }
    "mroo";
    {
        "_digits";
        "၀၀၀၀၀၀၀၀၀၀",
        "_type";
        "numeric";
    }
    "mtei";
    {
        "_digits";
        "၀၅၆၇၈၉၀၁၂၃၄၅",
        "_type";
        "numeric";
    }
    "mymr";
    {
        "_digits";
        "၀၁၂၃၄၅၆၇၈၉",
        "_type";
        "numeric";
    }
    "mymrshan";
    {
        "_digits";
        "၀၁၂၃၄၅၆၇၈၉",
        "_type";
        "numeric";
    }
    "mymrtlng";
    {
        "_digits";
        "၀၁၂၃၄၅၆၇၈၉",
        "_type";
        "numeric";
    }
    "newa";
    {
        "_digits";
        "၀၀၀၀၀၀၀၀၀၀",
        "_type";
        "numeric";
    }
    "nkoo";
    {
        "_digits";
        "၀၁၂၃၄၅၆၇၈၉",
        "_type";
        "numeric";
    }
    "olck";

```

```
{
    "_digits";
    "00123456789",
    "_type";
    "numeric";
}
"orya";
{
    "_digits";
    "012345678901",
    "_type";
    "numeric";
}
"osma";
{
    "_digits";
    "01234567890123456789",
    "_type";
    "numeric";
}
"roman";
{
    "_rules";
    "roman-upper",
    "_type";
    "algorithmic";
}
"romanlow";
{
    "_rules";
    "roman-lower",
    "_type";
    "algorithmic";
}
"saur";
{
    "_digits";
    "0123456789",
    "_type";
    "numeric";
}
"shrd";
{
    "_digits";
    "0123456789",
    "_type";
    "numeric";
}
"sind";
{
    "_digits";
    "0123456789",
    "_type";
    "numeric";
}
"sinh";
```

```

    {
      "_digits";
      "൧൨൩൪൫൬൭൮൯",
      "_type";
      "numeric";
    }
    "sora";
    {
      "_digits";
      "0123456789",
      "_type";
      "numeric";
    }
    "sund";
    {
      "_digits";
      "□□□□□□□□",
      "_type";
      "numeric";
    }
    "takr";
    {
      "_digits";
      "□□□□□□□□",
      "_type";
      "numeric";
    }
    "talu";
    {
      "_digits";
      "0123456789",
      "_type";
      "numeric";
    }
    "taml";
    {
      "_rules";
      "tamil",
      "_type";
      "algorithmic";
    }
    "tamldec";
    {
      "_digits";
      "0௧௨௩௪௫௬௭௮௯",
      "_type";
      "numeric";
    }
    "telu";
    {
      "_digits";
      "0౧౨౩౪౫౬౭౮౯",
      "_type";
      "numeric";
    }
    "thai";

```

}

```
{
  "main": {
    "ar": {
      "identity": {
        "version": {
          "_number": "$Revision: 12879 $",
          "_cldrVersion": "30.0.3"
        },
        "language": "ar"
      },
      "numbers": {
        "defaultNumberingSystem": "arab",
        "otherNumberingSystems": {
          "native": "arab"
        },
        "minimumGroupingDigits": "1",

```

```

"symbols-numberSystem-arab": {
  "decimal": ",",
  "group": ",",
  "list": ":",
  "percentSign": "%",
  "plusSign": "+",
  "minusSign": "-",
  "exponential": "١٠١٥",
  "superscriptingExponent": "×",
  "perMille": "‰",
  "infinity": "∞",
  "nan": "ليس رقم",
  "timeSeparator": ":"
},
"symbols-numberSystem-latn": {
  "decimal": ".",
  "group": ",",
  "list": ";",
  "percentSign": "%",
  "plusSign": "+",
  "minusSign": "-",
  "exponential": "E",
  "superscriptingExponent": "×",
  "perMille": "‰",
  "infinity": "∞",
  "nan": "ليس رقمًا",
  "timeSeparator": ":"
},
"decimalFormats-numberSystem-arab": {
  "standard": "#,##0.###",
  "long": {
    "decimalFormat": {
      "1000-count-zero": "0 ألف",
      "1000-count-one": "0 ألف",
      "1000-count-two": "0 ألف",
      "1000-count-few": "0 آلاف",
      "1000-count-many": "0 ألف",
      "1000-count-other": "0 ألف",
      "10000-count-zero": "00 ألف",
      "10000-count-one": "00 ألف",
      "10000-count-two": "00 ألف",
      "10000-count-few": "00 ألف",
      "10000-count-many": "00 ألف",
      "10000-count-other": "00 ألف",
      "100000-count-zero": "000 ألف",
      "100000-count-one": "000 ألف",
      "100000-count-two": "000 ألف",
      "100000-count-few": "000 ألف",
      "100000-count-many": "000 ألف",
      "100000-count-other": "000 ألف",
      "1000000-count-zero": "0 مليون",
      "1000000-count-one": "0 مليون",
      "1000000-count-two": "0 مليون",
      "1000000-count-few": "0 ملايين",
      "1000000-count-many": "0 مليون",
      "1000000-count-other": "0 مليون",
      "10000000-count-zero": "00 مليون"
    }
  }
}

```



```

    "10000000-count-one": "00 مليون",
    "10000000-count-two": "00 مليون",
    "10000000-count-few": "00 ملايين",
    "10000000-count-many": "00 مليون",
    "10000000-count-other": "00 مليون",
    "100000000-count-zero": "000 مليون",
    "100000000-count-one": "000 مليون",
    "100000000-count-two": "000 مليون",
    "100000000-count-few": "000 مليون",
    "100000000-count-many": "000 مليون",
    "100000000-count-other": "000 مليون",
    "1000000000-count-zero": "0 مليار",
    "1000000000-count-one": "0 مليار",
    "1000000000-count-two": "0 مليار",
    "1000000000-count-few": "0 مليار",
    "1000000000-count-many": "0 مليار",
    "1000000000-count-other": "0 مليار",
    "10000000000-count-zero": "00 مليار",
    "10000000000-count-one": "00 مليار",
    "10000000000-count-two": "00 مليار",
    "10000000000-count-few": "00 مليار",
    "10000000000-count-many": "00 مليار",
    "10000000000-count-other": "00 مليار",
    "100000000000-count-zero": "000 مليار",
    "100000000000-count-one": "000 مليار",
    "100000000000-count-two": "000 مليار",
    "100000000000-count-few": "000 مليار",
    "100000000000-count-many": "000 مليار",
    "100000000000-count-other": "000 مليار",
    "1000000000000-count-zero": "0 تريليون",
    "1000000000000-count-one": "0 تريليون",
    "1000000000000-count-two": "0 تريليون",
    "1000000000000-count-few": "0 تريليونات",
    "1000000000000-count-many": "0 تريليون",
    "1000000000000-count-other": "0 تريليون",
    "10000000000000-count-zero": "00 تريليون",
    "10000000000000-count-one": "00 تريليون",
    "10000000000000-count-two": "00 تريليون",
    "10000000000000-count-few": "00 تريليون",
    "10000000000000-count-many": "00 تريليون",
    "10000000000000-count-other": "00 تريليون",
    "100000000000000-count-zero": "000 تريليون",
    "100000000000000-count-one": "000 تريليون",
    "100000000000000-count-two": "000 تريليون",
    "100000000000000-count-few": "000 تريليون",
    "100000000000000-count-many": "000 تريليون",
    "100000000000000-count-other": "000 تريليون"
  }
},
"short": {
  "decimalFormat": {
    "1000-count-zero": "0 ألف",
    "1000-count-one": "0 ألف",
    "1000-count-two": "0 ألف",
    "1000-count-few": "0 آلاف",
    "1000-count-many": "0 ألف",
    "1000-count-other": "0 ألف",
  }
}

```

```
"10000-count-zero": "00 ألف",
"10000-count-one": "00 ألف",
"10000-count-two": "00 ألف",
"10000-count-few": "00 ألف",
"10000-count-many": "00 ألف",
"10000-count-other": "00 ألف",
"100000-count-zero": "000 ألف",
"100000-count-one": "000 ألف",
"100000-count-two": "000 ألف",
"100000-count-few": "000 ألف",
"100000-count-many": "000 ألف",
"100000-count-other": "000 ألف",
"1000000-count-zero": "0 مليون",
"1000000-count-one": "0 مليون",
"1000000-count-two": "0 مليون",
"1000000-count-few": "0 مليون",
"1000000-count-many": "0 مليون",
"1000000-count-other": "0 مليون",
"10000000-count-zero": "00 مليون",
"10000000-count-one": "00 مليون",
"10000000-count-two": "00 مليون",
"10000000-count-few": "00 مليون",
"10000000-count-many": "00 مليون",
"10000000-count-other": "00 مليون",
"100000000-count-zero": "000 مليون",
"100000000-count-one": "000 مليون",
"100000000-count-two": "000 مليون",
"100000000-count-few": "000 مليون",
"100000000-count-many": "000 مليون",
"100000000-count-other": "000 مليون",
"1000000000-count-zero": "0 مليا",
"1000000000-count-one": "0 مليا",
"1000000000-count-two": "0 مليا",
"1000000000-count-few": "0 مليا",
"1000000000-count-many": "0 مليا",
"1000000000-count-other": "0 مليا",
"10000000000-count-zero": "00 مليا",
"10000000000-count-one": "00 مليا",
"10000000000-count-two": "00 مليا",
"10000000000-count-few": "00 مليا",
"10000000000-count-many": "00 مليا",
"10000000000-count-other": "00 مليا",
"100000000000-count-zero": "000 مليا",
"100000000000-count-one": "000 مليا",
"100000000000-count-two": "000 مليا",
"100000000000-count-few": "000 مليا",
"100000000000-count-many": "000 مليا",
"100000000000-count-other": "000 مليا",
"1000000000000-count-zero": "0 ترليو",
"1000000000000-count-one": "0 ترليو",
"1000000000000-count-two": "0 ترليو",
"1000000000000-count-few": "0 ترليو",
"1000000000000-count-many": "0 ترليو",
"1000000000000-count-other": "0 ترليو",
"10000000000000-count-zero": "00 ترليو",
"10000000000000-count-one": "00 ترليو",
"10000000000000-count-two": "00 ترليو",
```

```

        "10000000000000-count-few": "00 ترليو",
        "10000000000000-count-many": "00 ترليو",
        "10000000000000-count-other": "00 ترليو",
        "10000000000000-count-zero": "000 ترليو",
        "10000000000000-count-one": "000 ترليو",
        "10000000000000-count-two": "000 ترليو",
        "10000000000000-count-few": "000 ترليو",
        "10000000000000-count-many": "000 ترليو",
        "10000000000000-count-other": "000 ترليو"
    }
}
},
"decimalFormats-numberSystem-latn": {
    "standard": "#,##0.###",
    "long": {
        "decimalFormat": {
            "1000-count-zero": "0 ألف",
            "1000-count-one": "0 ألف",
            "1000-count-two": "0 ألف",
            "1000-count-few": "0 آلاف",
            "1000-count-many": "0 ألف",
            "1000-count-other": "0 ألف",
            "10000-count-zero": "00 ألف",
            "10000-count-one": "00 ألف",
            "10000-count-two": "00 ألف",
            "10000-count-few": "00 ألف",
            "10000-count-many": "00 ألف",
            "10000-count-other": "00 ألف",
            "100000-count-zero": "000 ألف",
            "100000-count-one": "000 ألف",
            "100000-count-two": "000 ألف",
            "100000-count-few": "000 ألف",
            "100000-count-many": "000 ألف",
            "100000-count-other": "000 ألف",
            "1000000-count-zero": "0 مليون",
            "1000000-count-one": "0 مليون",
            "1000000-count-two": "0 مليون",
            "1000000-count-few": "0 ملايين",
            "1000000-count-many": "0 مليون",
            "1000000-count-other": "0 مليون",
            "10000000-count-zero": "00 مليون",
            "10000000-count-one": "00 مليون",
            "10000000-count-two": "00 مليون",
            "10000000-count-few": "00 ملايين",
            "10000000-count-many": "00 مليون",
            "10000000-count-other": "00 مليون",
            "100000000-count-zero": "000 مليون",
            "100000000-count-one": "000 مليون",
            "100000000-count-two": "000 مليون",
            "100000000-count-few": "000 مليون",
            "100000000-count-many": "000 مليون",
            "100000000-count-other": "000 مليون",
            "1000000000-count-zero": "0 مليار",
            "1000000000-count-one": "0 مليار",
            "1000000000-count-two": "0 مليار",
            "1000000000-count-few": "0 مليار",
            "1000000000-count-many": "0 مليار",

```

```

        "1000000000-count-other": "0 مليار",
        "1000000000-count-zero": "00 مليار",
        "1000000000-count-one": "00 مليار",
        "1000000000-count-two": "00 مليار",
        "1000000000-count-few": "00 مليار",
        "1000000000-count-many": "00 مليار",
        "1000000000-count-other": "00 مليار",
        "1000000000-count-zero": "000 مليار",
        "1000000000-count-one": "000 مليار",
        "1000000000-count-two": "000 مليار",
        "1000000000-count-few": "000 مليار",
        "1000000000-count-many": "000 مليار",
        "1000000000-count-other": "000 مليار",
        "10000000000-count-zero": "0 تريليون",
        "10000000000-count-one": "0 تريليون",
        "10000000000-count-two": "0 تريليون",
        "10000000000-count-few": "0 تريليونات",
        "10000000000-count-many": "0 تريليون",
        "10000000000-count-other": "0 تريليون",
        "100000000000-count-zero": "00 تريليون",
        "100000000000-count-one": "00 تريليون",
        "100000000000-count-two": "00 تريليون",
        "100000000000-count-few": "00 تريليون",
        "100000000000-count-many": "00 تريليون",
        "100000000000-count-other": "00 تريليون",
        "1000000000000-count-zero": "000 تريليون",
        "1000000000000-count-one": "000 تريليون",
        "1000000000000-count-two": "000 تريليون",
        "1000000000000-count-few": "000 تريليون",
        "1000000000000-count-many": "000 تريليون",
        "1000000000000-count-other": "000 تريليون"
    },
    },
    "short": {
        "decimalFormat": {
            "100-count-zero": "0 ألف",
            "100-count-one": "0 ألف",
            "100-count-two": "0 ألف",
            "100-count-few": "0 آلاف",
            "100-count-many": "0 ألف",
            "100-count-other": "0 ألف",
            "1000-count-zero": "00 ألف",
            "1000-count-one": "00 ألف",
            "1000-count-two": "00 ألف",
            "1000-count-few": "00 ألف",
            "1000-count-many": "00 ألف",
            "1000-count-other": "00 ألف",
            "10000-count-zero": "000 ألف",
            "10000-count-one": "000 ألف",
            "10000-count-two": "000 ألف",
            "10000-count-few": "000 ألف",
            "10000-count-many": "000 ألف",
            "10000-count-other": "000 ألف",
            "100000-count-zero": "0 مليو",
            "100000-count-one": "0 مليو",
            "100000-count-two": "0 مليو",
            "100000-count-few": "0 مليو",

```

```

        "1000000-count-many": "0 مليو",
        "1000000-count-other": "0 مليو",
        "1000000-count-zero": "00 مليو",
        "1000000-count-one": "00 مليو",
        "1000000-count-two": "00 مليو",
        "1000000-count-few": "00 مليو",
        "1000000-count-many": "00 مليو",
        "1000000-count-other": "00 مليو",
        "10000000-count-zero": "000 مليو",
        "10000000-count-one": "000 مليو",
        "10000000-count-two": "000 مليو",
        "10000000-count-few": "000 مليو",
        "10000000-count-many": "000 مليو",
        "10000000-count-other": "000 مليو",
        "1000000000-count-zero": "0 مليا",
        "1000000000-count-one": "0 مليا",
        "1000000000-count-two": "0 مليا",
        "1000000000-count-few": "0 مليا",
        "1000000000-count-many": "0 مليا",
        "1000000000-count-other": "0 مليا",
        "10000000000-count-zero": "00 مليا",
        "10000000000-count-one": "00 مليا",
        "10000000000-count-two": "00 مليا",
        "10000000000-count-few": "00 مليا",
        "10000000000-count-many": "00 مليا",
        "10000000000-count-other": "00 مليا",
        "100000000000-count-zero": "000 مليا",
        "100000000000-count-one": "000 مليا",
        "100000000000-count-two": "000 مليا",
        "100000000000-count-few": "000 مليا",
        "100000000000-count-many": "000 مليا",
        "100000000000-count-other": "000 مليا",
        "1000000000000-count-zero": "0 ترليو",
        "1000000000000-count-one": "0 ترليو",
        "1000000000000-count-two": "0 ترليو",
        "1000000000000-count-few": "0 ترليو",
        "1000000000000-count-many": "0 ترليو",
        "1000000000000-count-other": "0 ترليو",
        "10000000000000-count-zero": "00 ترليو",
        "10000000000000-count-one": "00 ترليو",
        "10000000000000-count-two": "00 ترليو",
        "10000000000000-count-few": "00 ترليو",
        "10000000000000-count-many": "00 ترليو",
        "10000000000000-count-other": "00 ترليو",
        "100000000000000-count-zero": "000 ترليو",
        "100000000000000-count-one": "000 ترليو",
        "100000000000000-count-two": "000 ترليو",
        "100000000000000-count-few": "000 ترليو",
        "100000000000000-count-many": "000 ترليو",
        "100000000000000-count-other": "000 ترليو"
    }
}
},
"scientificFormats-numberSystem-arab": {
    "standard": "#E0"
},
"scientificFormats-numberSystem-latn": {

```

```

    "standard": "#E0"
  },
  "percentFormats-numberSystem-arab": {
    "standard": "#,##0 %"
  },
  "percentFormats-numberSystem-latn": {
    "standard": "#,##0%"
  },
  "currencyFormats-numberSystem-arab": {
    "currencySpacing": {
      "beforeCurrency": {
        "currencyMatch": "[:^S:]",
        "surroundingMatch": "[:digit:]",
        "insertBetween": " "
      },
      "afterCurrency": {
        "currencyMatch": "[:^S:]",
        "surroundingMatch": "[:digit:]",
        "insertBetween": " "
      }
    },
    "standard": "#,##0.00 ¤",
    "accounting": "#,##0.00 ¤",
    "unitPattern-count-zero": "{0} {1}",
    "unitPattern-count-one": "{0} {1}",
    "unitPattern-count-two": "{0} {1}",
    "unitPattern-count-few": "{0} {1}",
    "unitPattern-count-many": "{0} {1}",
    "unitPattern-count-other": "{0} {1}"
  },
  "currencyFormats-numberSystem-latn": {
    "currencySpacing": {
      "beforeCurrency": {
        "currencyMatch": "[:^S:]",
        "surroundingMatch": "[:digit:]",
        "insertBetween": " "
      },
      "afterCurrency": {
        "currencyMatch": "[:^S:]",
        "surroundingMatch": "[:digit:]",
        "insertBetween": " "
      }
    },
    "standard": "¤ #,##0.00",
    "accounting": "¤#,##0.00; (¤#,##0.00)",
    "short": {
      "standard": {
        "1000-count-zero": "¤ 0 ألف",
        "1000-count-one": "¤ 0 ألف",
        "1000-count-two": "¤ 0 ألف",
        "1000-count-few": "¤ 0 ألف",
        "1000-count-many": "¤ 0 ألف",
        "1000-count-other": "¤ 0 ألف",
        "10000-count-zero": "¤ 00 ألف",
        "10000-count-one": "¤ 00 ألف",
        "10000-count-two": "¤ 00 ألف",
        "10000-count-few": "¤ 00 ألف",

```

```
"10000-count-many": "ألف 00",
"10000-count-other": "ألف 00",
"100000-count-zero": "ألف 000",
"100000-count-one": "ألف 000",
"100000-count-two": "ألف 000",
"100000-count-few": "ألف 000",
"100000-count-many": "ألف 000",
"100000-count-other": "ألف 000",
"1000000-count-zero": "مليون 0",
"1000000-count-one": "مليون 0",
"1000000-count-two": "مليون 0",
"1000000-count-few": "مليون 0",
"1000000-count-many": "مليون 0",
"1000000-count-other": "مليون 0",
"10000000-count-zero": "مليون 00",
"10000000-count-one": "مليون 00",
"10000000-count-two": "مليون 00",
"10000000-count-few": "مليون 00",
"10000000-count-many": "مليون 00",
"10000000-count-other": "مليون 00",
"100000000-count-zero": "مليون 000",
"100000000-count-one": "مليون 000",
"100000000-count-two": "مليون 000",
"100000000-count-few": "مليون 000",
"100000000-count-many": "مليون 000",
"100000000-count-other": "مليون 000",
"1000000000-count-zero": "مليار 0",
"1000000000-count-one": "مليار 0",
"1000000000-count-two": "مليار 0",
"1000000000-count-few": "مليار 0",
"1000000000-count-many": "مليار 0",
"1000000000-count-other": "مليار 0",
"10000000000-count-zero": "مليار 00",
"10000000000-count-one": "مليار 00",
"10000000000-count-two": "مليار 00",
"10000000000-count-few": "مليار 00",
"10000000000-count-many": "مليار 00",
"10000000000-count-other": "مليار 00",
"100000000000-count-zero": "مليار 000",
"100000000000-count-one": "مليار 000",
"100000000000-count-two": "مليار 000",
"100000000000-count-few": "مليار 000",
"100000000000-count-many": "مليار 000",
"100000000000-count-other": "مليار 000",
"1000000000000-count-zero": "ترليون 0",
"1000000000000-count-one": "ترليون 0",
"1000000000000-count-two": "ترليون 0",
"1000000000000-count-few": "ترليون 0",
"1000000000000-count-many": "ترليون 0",
"1000000000000-count-other": "ترليون 0",
"10000000000000-count-zero": "ترليون 00",
"10000000000000-count-one": "ترليون 00",
"10000000000000-count-two": "ترليون 00",
"10000000000000-count-few": "ترليون 00",
"10000000000000-count-many": "ترليون 00",
"10000000000000-count-other": "ترليون 00",
"100000000000000-count-zero": "ترليون 000",
```

```

        "1000000000000000-count-one": "٠ ٠٠٠ ترليو",
        "1000000000000000-count-two": "٠ ٠٠٠ ترليو",
        "1000000000000000-count-few": "٠ ٠٠٠ ترليو",
        "1000000000000000-count-many": "٠ ٠٠٠ ترليو",
        "1000000000000000-count-other": "٠ ٠٠٠ ترليو"
    },
    },
    "unitPattern-count-zero": "{0} {1}",
    "unitPattern-count-one": "{0} {1}",
    "unitPattern-count-two": "{0} {1}",
    "unitPattern-count-few": "{0} {1}",
    "unitPattern-count-many": "{0} {1}",
    "unitPattern-count-other": "{0} {1}"
},
"miscPatterns-numberSystem-arab": {
    "atLeast": "+{0}",
    "range": "{0}-{1}"
},
"miscPatterns-numberSystem-latn": {
    "atLeast": "+{0}",
    "range": "{0}-{1}"
}
}
}
}
}
}

```

## NUMBERS.JSX

```

{
    "main";
    {
        "ar";
        {
            "identity";
            {
                "version";
                {
                    "_number";
                    "$Revision: 12879 $",
                    "_cldrVersion";
                    "30.0.3";
                }
                "language";
                "ar";
            }
            "numbers";
            {
                "defaultNumberingSystem";
                "arab",
                "otherNumberingSystems";
                {
                    "native";
                    "arab";
                }
                "minimumGroupingDigits";
            }
        }
    }
}

```



```

    "1",
    "symbols-numberSystem-arab";
  {
    "decimal";
    ",",
    "group";
    ", ",
    "list";
    "؛",
    "percentSign";
    "%",
    "plusSign";
    "+",
    "minusSign";
    "-",
    "exponential";
    "١٠",
    "superscriptingExponent";
    "×",
    "perMille";
    "‰",
    "infinity";
    "∞",
    "nan";
    "ليس رقم",
    "timeSeparator";
    ":";
  }
  "symbols-numberSystem-latn";
  {
    "decimal";
    ".",
    "group";
    ", ",
    "list";
    ";",
    "percentSign";
    "%",
    "plusSign";
    "+",
    "minusSign";
    "-",
    "exponential";
    "E",
    "superscriptingExponent";
    "×",
    "perMille";
    "‰",
    "infinity";
    "∞",
    "nan";
    "ليس رقمًا",
    "timeSeparator";
    ":";
  }
  "decimalFormats-numberSystem-arab";
  {

```

```

"standard";
"#,##0.###",
    "long";
{
    "decimalFormat";
    {
        "1000-count-zero";
        "0 ألف",
        "1000-count-one";
        "0 ألف",
        "1000-count-two";
        "0 ألف",
        "1000-count-few";
        "0 آلاف",
        "1000-count-many";
        "0 ألف",
        "1000-count-other";
        "0 ألف",
        "10000-count-zero";
        "00 ألف",
        "10000-count-one";
        "00 ألف",
        "10000-count-two";
        "00 ألف",
        "10000-count-few";
        "00 ألف",
        "10000-count-many";
        "00 ألف",
        "10000-count-other";
        "00 ألف",
        "100000-count-zero";
        "000 ألف",
        "100000-count-one";
        "000 ألف",
        "100000-count-two";
        "000 ألف",
        "100000-count-few";
        "000 ألف",
        "100000-count-many";
        "000 ألف",
        "100000-count-other";
        "000 ألف",
        "1000000-count-zero";
        "0 مليون",
        "1000000-count-one";
        "0 مليون",
        "1000000-count-two";
        "0 مليون",
        "1000000-count-few";
        "0 ملايين",
        "1000000-count-many";
        "0 مليون",
        "1000000-count-other";
        "0 مليون",
        "10000000-count-zero";
        "00 مليون",
        "10000000-count-one";
    }
}

```

```
"00 مليون",  
    "10000000-count-two";  
"00 مليون",  
    "10000000-count-few";  
"00 ملايين",  
    "10000000-count-many";  
"00 مليون",  
    "10000000-count-other";  
"00 مليون",  
    "100000000-count-zero";  
"000 مليون",  
    "100000000-count-one";  
"000 مليون",  
    "100000000-count-two";  
"000 مليون",  
    "100000000-count-few";  
"000 مليون",  
    "100000000-count-many";  
"000 مليون",  
    "100000000-count-other";  
"000 مليون",  
    "1000000000-count-zero";  
"0 مليار",  
    "1000000000-count-one";  
"0 مليار",  
    "1000000000-count-two";  
"0 مليار",  
    "1000000000-count-few";  
"0 مليار",  
    "1000000000-count-many";  
"0 مليار",  
    "1000000000-count-other";  
"0 مليار",  
    "10000000000-count-zero";  
"00 مليار",  
    "10000000000-count-one";  
"00 مليار",  
    "10000000000-count-two";  
"00 مليار",  
    "10000000000-count-few";  
"00 مليار",  
    "10000000000-count-many";  
"00 مليار",  
    "10000000000-count-other";  
"00 مليار",  
    "100000000000-count-zero";  
"000 مليار",  
    "100000000000-count-one";  
"000 مليار",  
    "100000000000-count-two";  
"000 مليار",  
    "100000000000-count-few";  
"000 مليار",  
    "100000000000-count-many";  
"000 مليار",  
    "100000000000-count-other";  
"000 مليار",
```

```

        "1000000000000-count-zero";
        "0 تريليون",
        "1000000000000-count-one";
        "0 تريليون",
        "1000000000000-count-two";
        "0 تريليون",
        "1000000000000-count-few";
        "0 تريليونات",
        "1000000000000-count-many";
        "0 تريليون",
        "1000000000000-count-other";
        "0 تريليون",
        "1000000000000-count-zero";
        "00 تريليون",
        "1000000000000-count-one";
        "00 تريليون",
        "1000000000000-count-two";
        "00 تريليون",
        "1000000000000-count-few";
        "00 تريليون",
        "1000000000000-count-many";
        "00 تريليون",
        "1000000000000-count-other";
        "00 تريليون",
        "1000000000000-count-zero";
        "000 تريليون",
        "1000000000000-count-one";
        "000 تريليون",
        "1000000000000-count-two";
        "000 تريليون",
        "1000000000000-count-few";
        "000 تريليون",
        "1000000000000-count-many";
        "000 تريليون",
        "1000000000000-count-other";
        "000 تريليون";
    }
}
"short";
{
    "decimalFormat";
    {
        "1000-count-zero";
        "0 ألف",
        "1000-count-one";
        "0 ألف",
        "1000-count-two";
        "0 ألف",
        "1000-count-few";
        "0 آلاف",
        "1000-count-many";
        "0 ألف",
        "1000-count-other";
        "0 ألف",
        "10000-count-zero";
        "00 ألف",
        "10000-count-one";
    }
}

```

```
"00 ألف",
    "10000-count-two";
"00 ألف",
    "10000-count-few";
"00 ألف",
    "10000-count-many";
"00 ألف",
    "10000-count-other";
"00 ألف",
    "100000-count-zero";
"000 ألف",
    "100000-count-one";
"000 ألف",
    "100000-count-two";
"000 ألف",
    "100000-count-few";
"000 ألف",
    "100000-count-many";
"000 ألف",
    "100000-count-other";
"000 ألف",
    "1000000-count-zero";
"0 مليون",
    "1000000-count-one";
"0 مليون",
    "1000000-count-two";
"0 مليون",
    "1000000-count-few";
"0 مليون",
    "1000000-count-many";
"0 مليون",
    "1000000-count-other";
"0 مليون",
    "10000000-count-zero";
"00 مليون",
    "10000000-count-one";
"00 مليون",
    "10000000-count-two";
"00 مليون",
    "10000000-count-few";
"00 مليون",
    "10000000-count-many";
"00 مليون",
    "10000000-count-other";
"00 مليون",
    "100000000-count-zero";
"000 مليون",
    "100000000-count-one";
"000 مليون",
    "100000000-count-two";
"000 مليون",
    "100000000-count-few";
"000 مليون",
    "100000000-count-many";
"000 مليون",
    "100000000-count-other";
"000 مليون",
```

```
"1000000000-count-zero";
"0 مليا",
"1000000000-count-one";
"0 مليا",
"1000000000-count-two";
"0 مليا",
"1000000000-count-few";
"0 مليا",
"1000000000-count-many";
"0 مليا",
"1000000000-count-other";
"0 مليا",
"10000000000-count-zero";
"00 مليا",
"10000000000-count-one";
"00 مليا",
"10000000000-count-two";
"00 مليا",
"10000000000-count-few";
"00 مليا",
"10000000000-count-many";
"00 مليا",
"10000000000-count-other";
"00 مليا",
"100000000000-count-zero";
"000 مليا",
"100000000000-count-one";
"000 مليا",
"100000000000-count-two";
"000 مليا",
"100000000000-count-few";
"000 مليا",
"100000000000-count-many";
"000 مليا",
"100000000000-count-other";
"000 مليا",
"1000000000000-count-zero";
"0 ترليو",
"1000000000000-count-one";
"0 ترليو",
"1000000000000-count-two";
"0 ترليو",
"1000000000000-count-few";
"0 ترليو",
"1000000000000-count-many";
"0 ترليو",
"1000000000000-count-other";
"0 ترليو",
"10000000000000-count-zero";
"00 ترليو",
"10000000000000-count-one";
"00 ترليو",
"10000000000000-count-two";
"00 ترليو",
"10000000000000-count-few";
"00 ترليو",
"10000000000000-count-many";
```

```

        "00 ترليو",
        "10000000000000-count-other";
        "00 ترليو",
        "10000000000000-count-zero";
        "000 ترليو",
        "10000000000000-count-one";
        "000 ترليو",
        "10000000000000-count-two";
        "000 ترليو",
        "10000000000000-count-few";
        "000 ترليو",
        "10000000000000-count-many";
        "000 ترليو",
        "10000000000000-count-other";
        "000 ترليو";
    }
}
}
"decimalFormats-numberSystem-latn";
{
    "standard";
    "#,##0.###",
    "long";
    {
        "decimalFormat";
        {
            "1000-count-zero";
            "0 ألف",
            "1000-count-one";
            "0 ألف",
            "1000-count-two";
            "0 ألف",
            "1000-count-few";
            "0 آلاف",
            "1000-count-many";
            "0 ألف",
            "1000-count-other";
            "0 ألف",
            "10000-count-zero";
            "00 ألف",
            "10000-count-one";
            "00 ألف",
            "10000-count-two";
            "00 ألف",
            "10000-count-few";
            "00 ألف",
            "10000-count-many";
            "00 ألف",
            "10000-count-other";
            "00 ألف",
            "100000-count-zero";
            "000 ألف",
            "100000-count-one";
            "000 ألف",
            "100000-count-two";
            "000 ألف",
            "100000-count-few";
        }
    }
}

```

```
"000 ألف",
    "100000-count-many";
"000 ألف",
    "100000-count-other";
"000 ألف",
    "1000000-count-zero";
"0 مليون",
    "1000000-count-one";
"0 مليون",
    "1000000-count-two";
"0 مليون",
    "1000000-count-few";
"0 ملايين",
    "1000000-count-many";
"0 مليون",
    "1000000-count-other";
"0 مليون",
    "10000000-count-zero";
"00 مليون",
    "10000000-count-one";
"00 مليون",
    "10000000-count-two";
"00 مليون",
    "10000000-count-few";
"00 ملايين",
    "10000000-count-many";
"00 مليون",
    "10000000-count-other";
"00 مليون",
    "100000000-count-zero";
"000 مليون",
    "100000000-count-one";
"000 مليون",
    "100000000-count-two";
"000 مليون",
    "100000000-count-few";
"000 مليون",
    "100000000-count-many";
"000 مليون",
    "100000000-count-other";
"000 مليون",
    "1000000000-count-zero";
"0 مليار",
    "1000000000-count-one";
"0 مليار",
    "1000000000-count-two";
"0 مليار",
    "1000000000-count-few";
"0 مليار",
    "1000000000-count-many";
"0 مليار",
    "1000000000-count-other";
"0 مليار",
    "10000000000-count-zero";
"00 مليار",
    "10000000000-count-one";
"00 مليار",
```



```

        "10000000000-count-two";
    "00 مليار",
    "10000000000-count-few";
    "00 مليار",
    "10000000000-count-many";
    "00 مليار",
    "10000000000-count-other";
    "00 مليار",
    "10000000000-count-zero";
    "000 مليار",
    "10000000000-count-one";
    "000 مليار",
    "10000000000-count-two";
    "000 مليار",
    "10000000000-count-few";
    "000 مليار",
    "10000000000-count-many";
    "000 مليار",
    "10000000000-count-other";
    "000 مليار",
    "10000000000-count-zero";
    "0 تريليون",
    "100000000000-count-one";
    "0 تريليون",
    "100000000000-count-two";
    "0 تريليون",
    "100000000000-count-few";
    "0 تريليونات",
    "100000000000-count-many";
    "0 تريليون",
    "100000000000-count-other";
    "0 تريليون",
    "1000000000000-count-zero";
    "00 تريليون",
    "1000000000000-count-one";
    "00 تريليون",
    "1000000000000-count-two";
    "00 تريليون",
    "1000000000000-count-few";
    "00 تريليون",
    "1000000000000-count-many";
    "00 تريليون",
    "1000000000000-count-other";
    "00 تريليون",
    "1000000000000-count-zero";
    "000 تريليون",
    "10000000000000-count-one";
    "000 تريليون",
    "10000000000000-count-two";
    "000 تريليون",
    "10000000000000-count-few";
    "000 تريليون",
    "10000000000000-count-many";
    "000 تريليون",
    "10000000000000-count-other";
    "000 تريليون";
    }

```

```
}
"short";
{
  "decimalFormat";
  {
    "1000-count-zero";
    "0 ألف",
    "1000-count-one";
    "0 ألف",
    "1000-count-two";
    "0 ألف",
    "1000-count-few";
    "0 آلاف",
    "1000-count-many";
    "0 ألف",
    "1000-count-other";
    "0 ألف",
    "10000-count-zero";
    "00 ألف",
    "10000-count-one";
    "00 ألف",
    "10000-count-two";
    "00 ألف",
    "10000-count-few";
    "00 ألف",
    "10000-count-many";
    "00 ألف",
    "10000-count-other";
    "00 ألف",
    "100000-count-zero";
    "000 ألف",
    "100000-count-one";
    "000 ألف",
    "100000-count-two";
    "000 ألف",
    "100000-count-few";
    "000 ألف",
    "100000-count-many";
    "000 ألف",
    "100000-count-other";
    "000 ألف",
    "1000000-count-zero";
    "0 مليون",
    "1000000-count-one";
    "0 مليون",
    "1000000-count-two";
    "0 مليون",
    "1000000-count-few";
    "0 مليون",
    "1000000-count-many";
    "0 مليون",
    "1000000-count-other";
    "0 مليون",
    "10000000-count-zero";
    "00 مليون",
    "10000000-count-one";
    "00 مليون",
  }
}
```

```
"10000000-count-two";
"00 مليو",
"10000000-count-few";
"00 مليو",
"10000000-count-many";
"00 مليو",
"10000000-count-other";
"00 مليو",
"100000000-count-zero";
"000 مليو",
"100000000-count-one";
"000 مليو",
"100000000-count-two";
"000 مليو",
"100000000-count-few";
"000 مليو",
"100000000-count-many";
"000 مليو",
"100000000-count-other";
"000 مليو",
"1000000000-count-zero";
"0 مليا",
"1000000000-count-one";
"0 مليا",
"1000000000-count-two";
"0 مليا",
"1000000000-count-few";
"0 مليا",
"1000000000-count-many";
"0 مليا",
"1000000000-count-other";
"0 مليا",
"10000000000-count-zero";
"00 مليا",
"10000000000-count-one";
"00 مليا",
"10000000000-count-two";
"00 مليا",
"10000000000-count-few";
"00 مليا",
"10000000000-count-many";
"00 مليا",
"10000000000-count-other";
"00 مليا",
"100000000000-count-zero";
"000 مليا",
"100000000000-count-one";
"000 مليا",
"100000000000-count-two";
"000 مليا",
"100000000000-count-few";
"000 مليا",
"100000000000-count-many";
"000 مليا",
"100000000000-count-other";
"000 مليا",
"1000000000000-count-zero";
```

```

        "0 ترلیو",
        "1000000000000-count-one";
        "0 ترلیو",
        "1000000000000-count-two";
        "0 ترلیو",
        "1000000000000-count-few";
        "0 ترلیو",
        "1000000000000-count-many";
        "0 ترلیو",
        "1000000000000-count-other";
        "0 ترلیو",
        "1000000000000-count-zero";
        "00 ترلیو",
        "1000000000000-count-one";
        "00 ترلیو",
        "1000000000000-count-two";
        "00 ترلیو",
        "1000000000000-count-few";
        "00 ترلیو",
        "1000000000000-count-many";
        "00 ترلیو",
        "1000000000000-count-other";
        "00 ترلیو",
        "1000000000000-count-zero";
        "000 ترلیو",
        "1000000000000-count-one";
        "000 ترلیو",
        "1000000000000-count-two";
        "000 ترلیو",
        "1000000000000-count-few";
        "000 ترلیو",
        "1000000000000-count-many";
        "000 ترلیو",
        "1000000000000-count-other";
        "000 ترلیو";
    }
}
}
"scientificFormats-numberSystem-arab";
{
    "standard";
    "#E0";
}
"scientificFormats-numberSystem-latn";
{
    "standard";
    "#E0";
}
"percentFormats-numberSystem-arab";
{
    "standard";
    "#,##0 %";
}
"percentFormats-numberSystem-latn";
{
    "standard";
    "#,##0%";
}

```

```

    }
    "currencyFormats-numberSystem-arab";
    {
      "currencySpacing";
      {
        "beforeCurrency";
        {
          "currencyMatch";
          "[:^S:]",
          "surroundingMatch";
          "[:digit:]",
          "insertBetween";
          " ";
        }
        "afterCurrency";
        {
          "currencyMatch";
          "[:^S:]",
          "surroundingMatch";
          "[:digit:]",
          "insertBetween";
          " ";
        }
      }
    }
    "standard";
    "#,##0.00 ¤",
    "accounting";
    "#,##0.00 ¤",
    "unitPattern-count-zero";
    "{0} {1}",
    "unitPattern-count-one";
    "{0} {1}",
    "unitPattern-count-two";
    "{0} {1}",
    "unitPattern-count-few";
    "{0} {1}",
    "unitPattern-count-many";
    "{0} {1}",
    "unitPattern-count-other";
    "{0} {1}";
  }
  "currencyFormats-numberSystem-latn";
  {
    "currencySpacing";
    {
      "beforeCurrency";
      {
        "currencyMatch";
        "[:^S:]",
        "surroundingMatch";
        "[:digit:]",
        "insertBetween";
        " ";
      }
      "afterCurrency";
      {
        "currencyMatch";

```

```

        "[:^S:]",
        "surroundingMatch";
        "[:digit:]",
        "insertBetween";
        " ";
    }
}
"standard";
"¤ #,##0.00",
    "accounting";
"¤#,##0.00; (¤#,##0.00) ",
    "short";
{
    "standard";
    {
        "1000-count-zero";
        "¤ 0 ألف",
            "1000-count-one";
        "¤ 0 ألف",
            "1000-count-two";
        "¤ 0 ألف",
            "1000-count-few";
        "¤ 0 ألف",
            "1000-count-many";
        "¤ 0 ألف",
            "1000-count-other";
        "¤ 0 ألف",
            "10000-count-zero";
        "¤ 00 ألف",
            "10000-count-one";
        "¤ 00 ألف",
            "10000-count-two";
        "¤ 00 ألف",
            "10000-count-few";
        "¤ 00 ألف",
            "10000-count-many";
        "¤ 00 ألف",
            "10000-count-other";
        "¤ 00 ألف",
            "100000-count-zero";
        "¤ 000 ألف",
            "100000-count-one";
        "¤ 000 ألف",
            "100000-count-two";
        "¤ 000 ألف",
            "100000-count-few";
        "¤ 000 ألف",
            "100000-count-many";
        "¤ 000 ألف",
            "100000-count-other";
        "¤ 000 ألف",
            "1000000-count-zero";
        "¤ 0 مليون",
            "1000000-count-one";
        "¤ 0 مليون",
            "1000000-count-two";
        "¤ 0 مليون",
    }
}

```

```
"1000000-count-few";
"۰ مليو",
"1000000-count-many";
"۰ مليو",
"1000000-count-other";
"۰ مليو",
"10000000-count-zero";
"۰۰ مليو",
"10000000-count-one";
"۰۰ مليو",
"10000000-count-two";
"۰۰ مليو",
"10000000-count-few";
"۰۰ مليو",
"10000000-count-many";
"۰۰ مليو",
"10000000-count-other";
"۰۰ مليو",
"100000000-count-zero";
"۰۰۰ مليو",
"100000000-count-one";
"۰۰۰ مليو",
"100000000-count-two";
"۰۰۰ مليو",
"100000000-count-few";
"۰۰۰ مليو",
"100000000-count-many";
"۰۰۰ مليو",
"100000000-count-other";
"۰۰۰ مليو",
"1000000000-count-zero";
"۰ مليا",
"1000000000-count-one";
"۰ مليا",
"1000000000-count-two";
"۰ مليا",
"1000000000-count-few";
"۰ مليا",
"1000000000-count-many";
"۰ مليا",
"1000000000-count-other";
"۰ مليا",
"10000000000-count-zero";
"۰۰ مليا",
"10000000000-count-one";
"۰۰ مليا",
"10000000000-count-two";
"۰۰ مليا",
"10000000000-count-few";
"۰۰ مليا",
"10000000000-count-many";
"۰۰ مليا",
"10000000000-count-other";
"۰۰ مليا",
"100000000000-count-zero";
"۰۰۰ مليا",
"100000000000-count-one";
```

```

        "٠ ٠٠٠ مليا",
        "100000000000-count-two";
        "٠ ٠٠٠ مليا",
        "100000000000-count-few";
        "٠ ٠٠٠ مليا",
        "100000000000-count-many";
        "٠ ٠٠٠ مليا",
        "100000000000-count-other";
        "٠ ٠٠٠ مليا",
        "100000000000-count-zero";
        "٠ ٠ ترليو",
        "100000000000-count-one";
        "٠ ٠ ترليو",
        "100000000000-count-two";
        "٠ ٠ ترليو",
        "100000000000-count-few";
        "٠ ٠ ترليو",
        "100000000000-count-many";
        "٠ ٠ ترليو",
        "100000000000-count-other";
        "٠ ٠ ترليو",
        "100000000000-count-zero";
        "٠ ٠٠ ترليو",
        "100000000000-count-one";
        "٠ ٠٠ ترليو",
        "100000000000-count-two";
        "٠ ٠٠ ترليو",
        "100000000000-count-few";
        "٠ ٠٠ ترليو",
        "100000000000-count-many";
        "٠ ٠٠ ترليو",
        "100000000000-count-other";
        "٠ ٠٠ ترليو",
        "100000000000-count-zero";
        "٠ ٠٠٠ ترليو",
        "100000000000-count-one";
        "٠ ٠٠٠ ترليو",
        "100000000000-count-two";
        "٠ ٠٠٠ ترليو",
        "100000000000-count-few";
        "٠ ٠٠٠ ترليو",
        "100000000000-count-many";
        "٠ ٠٠٠ ترليو",
        "100000000000-count-other";
        "٠ ٠٠٠ ترليو";
    }
}
"unitPattern-count-zero";
"{0} {1}",
    "unitPattern-count-one";
"{0} {1}",
    "unitPattern-count-two";
"{0} {1}",
    "unitPattern-count-few";
"{0} {1}",
    "unitPattern-count-many";
"{0} {1}",

```



```

        "unitPattern-count-other";
        "{0} {1}";
    }
    "miscPatterns-numberSystem-arab";
    {
        "atLeast";
        "+{0}",
        "range";
        "{0}-{1}";
    }
    "miscPatterns-numberSystem-latn";
    {
        "atLeast";
        "+{0}",
        "range";
        "{0}-{1}";
    }
    }
}

```

### TIMEZONENAMES.JSON

```

{
  "main": {
    "ar": {
      "identity": {
        "version": {
          "_number": "$Revision: 12879 $",
          "_cldrVersion": "30.0.3"
        },
        "language": "ar"
      },
      "dates": {
        "timeZoneNames": {
          "hourFormat": "+HH:mm;-HH:mm",
          "gmtFormat": "0{جرينتش",
          "gmtZeroFormat": "جرينتش",
          "regionFormat": "0{توقيت",
          "regionFormat-type-daylight": "الصيفي } 0{توقيت",
          "regionFormat-type-standard": "الرسمي } 0{توقيت",
          "fallbackFormat": "{1} ({0})",
          "zone": {
            "America": {
              "Adak": {
                "exemplarCity": "أداك"
              },
              "Anchorage": {
                "exemplarCity": "أنشوراج"
              },
              "Anguilla": {
                "exemplarCity": "أنغيلا"
              },
              "Antigua": {
                "exemplarCity": "أنتيغوا"
              }
            }
          }
        }
      }
    }
  }
}

```

```
    },  
    "Araguaina": {  
      "exemplarCity": "أروجوانيا"  
    },  
    "Argentina": {  
      "Rio_Gallegos": {  
        "exemplarCity": "ريو جالييوس"  
      },  
      "San_Juan": {  
        "exemplarCity": "سان خوان"  
      },  
      "Ushuaia": {  
        "exemplarCity": "أشوا"  
      },  
      "La_Rioja": {  
        "exemplarCity": "لا ريوجا"  
      },  
      "San_Luis": {  
        "exemplarCity": "سان لويس"  
      },  
      "Salta": {  
        "exemplarCity": "سالطا"  
      },  
      "Tucuman": {  
        "exemplarCity": "تاكمان"  
      }  
    },  
    "Aruba": {  
      "exemplarCity": "أروبا"  
    },  
    "Asuncion": {  
      "exemplarCity": "أسونسيون"  
    },  
    "Bahia": {  
      "exemplarCity": "باهيا"  
    },  
    "Bahia_Banderas": {  
      "exemplarCity": "باهيا بانديراس"  
    },  
    "Barbados": {  
      "exemplarCity": "بربادوس"  
    },  
    "Belem": {  
      "exemplarCity": "بلم"  
    },  
    "Belize": {  
      "exemplarCity": "بليز"  
    },  
    "Blanc-Sablon": {  
      "exemplarCity": "بلانك-سابلون"  
    },  
    "Boa_Vista": {  
      "exemplarCity": "باو فيستا"  
    },  
    "Bogota": {  
      "exemplarCity": "بوغوتا"  
    },  
  },  
}
```

```
"Boise": {
  "exemplarCity": "بويس"
},
"Buenos_Aires": {
  "exemplarCity": "بوينوس آيرس"
},
"Cambridge_Bay": {
  "exemplarCity": "كامبرديج باي"
},
"Campo_Grande": {
  "exemplarCity": "كومبو جراند"
},
"Cancun": {
  "exemplarCity": "كانكون"
},
"Caracas": {
  "exemplarCity": "كاراكاس"
},
"Catamarca": {
  "exemplarCity": "كاتاماركا"
},
"Cayenne": {
  "exemplarCity": "كايين"
},
"Cayman": {
  "exemplarCity": "كيمان"
},
"Chicago": {
  "exemplarCity": "شيكاغو"
},
"Chihuahua": {
  "exemplarCity": "تشيوآوا"
},
"Coral_Harbour": {
  "exemplarCity": "كoral هاربر"
},
"Cordoba": {
  "exemplarCity": "كوردوبا"
},
"Costa_Rica": {
  "exemplarCity": "كوستاريكا"
},
"Creston": {
  "exemplarCity": "كريستون"
},
"Cuiaba": {
  "exemplarCity": "كيابا"
},
"Curacao": {
  "exemplarCity": "كوراكاو"
},
"Danmarkshavn": {
  "exemplarCity": "دانمرك شافن"
},
"Dawson": {
  "exemplarCity": "داوسان"
},
```

```
"Dawson_Creek": {
  "exemplarCity": "داوسن كريك"
},
"Denver": {
  "exemplarCity": "دنفر"
},
"Detroit": {
  "exemplarCity": "ديترويت"
},
"Dominica": {
  "exemplarCity": "دومينيكا"
},
"Edmonton": {
  "exemplarCity": "ايدمونتون"
},
"Eirunepe": {
  "exemplarCity": "ايرونبي"
},
"El_Salvador": {
  "exemplarCity": "السلفادور"
},
"Fort_Nelson": {
  "exemplarCity": "فورت نيلسون"
},
"Fortaleza": {
  "exemplarCity": "فورتاليزا"
},
"Glace_Bay": {
  "exemplarCity": "جلاس باي"
},
"Godthab": {
  "exemplarCity": "غودثاب"
},
"Goose_Bay": {
  "exemplarCity": "جوس باي"
},
"Grand_Turk": {
  "exemplarCity": "غراند ترك"
},
"Grenada": {
  "exemplarCity": "جرينادا"
},
"Guadeloupe": {
  "exemplarCity": "غوادلوب"
},
"Guatemala": {
  "exemplarCity": "غواتيمالا"
},
"Guayaquil": {
  "exemplarCity": "غواياكويل"
},
"Guyana": {
  "exemplarCity": "غيانا"
},
"Halifax": {
  "exemplarCity": "هاليفاكس"
},
}
```

```
"Havana": {
  "exemplarCity": "هافانا"
},
"Hermosillo": {
  "exemplarCity": "هيرموسيلو"
},
"Indiana": {
  "Vincennes": {
    "exemplarCity": "فينسينس"
  },
  "Petersburg": {
    "exemplarCity": "بيتربيرغ"
  },
  "Tell_City": {
    "exemplarCity": "مدينة تل، إنديانا"
  },
  "Knox": {
    "exemplarCity": "كونكس"
  },
  "Winamac": {
    "exemplarCity": "ويناماك"
  },
  "Marengo": {
    "exemplarCity": "مارنجو"
  },
  "Vevay": {
    "exemplarCity": "فيفاي"
  }
},
"Indianapolis": {
  "exemplarCity": "إنديانا بوليس"
},
"Inuvik": {
  "exemplarCity": "اينوفيك"
},
"Iqaluit": {
  "exemplarCity": "اكويلت"
},
"Jamaica": {
  "exemplarCity": "جامايكا"
},
"Jujuy": {
  "exemplarCity": "جوجو"
},
"Juneau": {
  "exemplarCity": "جونى"
},
"Kentucky": {
  "Monticello": {
    "exemplarCity": "مونتيسيلو"
  }
},
"Kralendijk": {
  "exemplarCity": "كرالنديك"
},
"La_Paz": {
  "exemplarCity": "لا باز"
```

```
,
  "Lima": {
    "exemplarCity": "ليما"
  },
  "Los_Angeles": {
    "exemplarCity": "لوس انجلوس"
  },
  "Louisville": {
    "exemplarCity": "لويس فيل"
  },
  "Lower_Princes": {
    "exemplarCity": "حي الأمير السفلي"
  },
  "Maceio": {
    "exemplarCity": "ماشيو"
  },
  "Managua": {
    "exemplarCity": "ماناغوا"
  },
  "Manaus": {
    "exemplarCity": "ماناوس"
  },
  "Marigot": {
    "exemplarCity": "ماريغوت"
  },
  "Martinique": {
    "exemplarCity": "المارتينيك"
  },
  "Matamoros": {
    "exemplarCity": "ماتاموروس"
  },
  "Mazatlan": {
    "exemplarCity": "مازاتلان"
  },
  "Mendoza": {
    "exemplarCity": "ميندوزا"
  },
  "Menominee": {
    "exemplarCity": "مينوميني"
  },
  "Merida": {
    "exemplarCity": "ميريديا"
  },
  "Metlakatla": {
    "exemplarCity": "ميتلاكاتلا"
  },
  "Mexico_City": {
    "exemplarCity": "مدينة المكسيك"
  },
  "Miquelon": {
    "exemplarCity": "ميكلون"
  },
  "Moncton": {
    "exemplarCity": "وينكتون"
  },
  "Monterrey": {
    "exemplarCity": "مونتيري"
```

```

    },
    "Montevideo": {
      "exemplarCity": "مونتفيديو"
    },
    "Montserrat": {
      "exemplarCity": "مونتسيرات"
    },
    "Nassau": {
      "exemplarCity": "ناسو"
    },
    "New_York": {
      "exemplarCity": "نيويورك"
    },
    "Nipigon": {
      "exemplarCity": "نيبيجون"
    },
    "Nome": {
      "exemplarCity": "نوم"
    },
    "Noronha": {
      "exemplarCity": "نوروناه"
    },
    "North_Dakota": {
      "Beulah": {
        "exemplarCity": "بيولا، داكوتا الشمالية"
      },
      "New_Salem": {
        "exemplarCity": "نيو ساليم"
      },
      "Center": {
        "exemplarCity": "سنتر"
      }
    },
    "Ojinaga": {
      "exemplarCity": "أوجيناغا"
    },
    "Panama": {
      "exemplarCity": "بنما"
    },
    "Pangnirtung": {
      "exemplarCity": "بانجينتينج"
    },
    "Paramaribo": {
      "exemplarCity": "باراماريبو"
    },
    "Phoenix": {
      "exemplarCity": "فينكس"
    },
    "Port-au-Prince": {
      "exemplarCity": "بورت أو برنس"
    },
    "Port_of_Spain": {
      "exemplarCity": "بورت أوف سبين"
    },
    "Porto_Velho": {
      "exemplarCity": "بورتو فيلو"
    }
  },

```

```
"Puerto_Rico": {
  "exemplarCity": "بورتوريكو"
},
"Rainy_River": {
  "exemplarCity": "راني ريفر"
},
"Rankin_Inlet": {
  "exemplarCity": "رانكن انلت"
},
"Recife": {
  "exemplarCity": "ريسيف"
},
"Regina": {
  "exemplarCity": "ريجينا"
},
"Resolute": {
  "exemplarCity": "ريزولوت"
},
"Rio_Branco": {
  "exemplarCity": "ريوبرانكو"
},
"Santa_Isabel": {
  "exemplarCity": "سانتا إيزابيل"
},
"Santarem": {
  "exemplarCity": "سانتاريم"
},
"Santiago": {
  "exemplarCity": "سانتياغو"
},
"Santo_Domingo": {
  "exemplarCity": "سانتو دومينغو"
},
"Sao_Paulo": {
  "exemplarCity": "ساو باولو"
},
"Scoresbysund": {
  "exemplarCity": "سكورسبيسند"
},
"Sitka": {
  "exemplarCity": "سيتكا"
},
"St_Barthelemy": {
  "exemplarCity": "سانت بارتيليمي"
},
"St_Johns": {
  "exemplarCity": "سانت جونز"
},
"St_Kitts": {
  "exemplarCity": "سانت كيتس"
},
"St_Lucia": {
  "exemplarCity": "سانت لوشيا"
},
"St_Thomas": {
  "exemplarCity": "سانت توماس"
},
}
```



```

"St_Vincent": {
  "exemplarCity": "سانت فنسنت"
},
"Swift_Current": {
  "exemplarCity": "سوفت كارنت"
},
"Tegucigalpa": {
  "exemplarCity": "تيغوسيغالبا"
},
"Thule": {
  "exemplarCity": "ثيل"
},
"Thunder_Bay": {
  "exemplarCity": "ثندر باي"
},
"Tijuana": {
  "exemplarCity": "تيخوانا"
},
"Toronto": {
  "exemplarCity": "تورونتو"
},
"Tortola": {
  "exemplarCity": "تورتولا"
},
"Vancouver": {
  "exemplarCity": "فانكوفر"
},
"Whitehorse": {
  "exemplarCity": "وايت هورس"
},
"Winnipeg": {
  "exemplarCity": "وينيبيج"
},
"Yakutat": {
  "exemplarCity": "ياكوتات"
},
"Yellowknife": {
  "exemplarCity": "يلونيف"
}
},
"Atlantic": {
  "Azores": {
    "exemplarCity": "أزورس"
  },
  "Bermuda": {
    "exemplarCity": "برمودا"
  },
  "Canary": {
    "exemplarCity": "كناري"
  },
  "Cape_Verde": {
    "exemplarCity": "الرأس الأخضر"
  },
  "Faeroe": {
    "exemplarCity": "فارو"
  },
  "Madeira": {

```

```
    "exemplarCity": "ماديرا"  
  },  
  "Reykjavik": {  
    "exemplarCity": "ريكيافيك"  
  },  
  "South_Georgia": {  
    "exemplarCity": "جورجيا الجنوبية"  
  },  
  "St_Helena": {  
    "exemplarCity": "سانت هيلينا"  
  },  
  "Stanley": {  
    "exemplarCity": "استانلي"  
  },  
},  
"Europe": {  
  "Amsterdam": {  
    "exemplarCity": "أمستردام"  
  },  
  "Andorra": {  
    "exemplarCity": "أندورا"  
  },  
  "Astrakhan": {  
    "exemplarCity": "أستراخان"  
  },  
  "Athens": {  
    "exemplarCity": "أثينا"  
  },  
  "Belgrade": {  
    "exemplarCity": "بلغراد"  
  },  
  "Berlin": {  
    "exemplarCity": "برلين"  
  },  
  "Bratislava": {  
    "exemplarCity": "براتيسلافا"  
  },  
  "Brussels": {  
    "exemplarCity": "بروكسل"  
  },  
  "Bucharest": {  
    "exemplarCity": "بوخارست"  
  },  
  "Budapest": {  
    "exemplarCity": "بودابست"  
  },  
  "Busingen": {  
    "exemplarCity": "بوسنغن"  
  },  
  "Chisinau": {  
    "exemplarCity": "تشيسيناو"  
  },  
  "Copenhagen": {  
    "exemplarCity": "كوبنهاغن"  
  },  
  "Dublin": {  
    "long": {
```

```
    "daylight": "توقيت أيرلندا الرسمي"  
  },  
  "exemplarCity": "دبلن"  
},  
"Gibraltar": {  
  "exemplarCity": "جبل طارق"  
},  
"Guernsey": {  
  "exemplarCity": "غيرنسي"  
},  
"Helsinki": {  
  "exemplarCity": "هلسنكي"  
},  
"Isle_of_Man": {  
  "exemplarCity": "جزيرة مان"  
},  
"Istanbul": {  
  "exemplarCity": "إسطنبول"  
},  
"Jersey": {  
  "exemplarCity": "جيرسي"  
},  
"Kaliningrad": {  
  "exemplarCity": "كالينجراد"  
},  
"Kiev": {  
  "exemplarCity": "كييف"  
},  
"Kirov": {  
  "exemplarCity": "كيروف"  
},  
"Lisbon": {  
  "exemplarCity": "لشبونة"  
},  
"Ljubljana": {  
  "exemplarCity": "ليوبليانا"  
},  
"London": {  
  "long": {  
    "daylight": "توقيت بريطانيا الصيفي"  
  },  
  "exemplarCity": "لندن"  
},  
"Luxembourg": {  
  "exemplarCity": "لوكسمبورغ"  
},  
"Madrid": {  
  "exemplarCity": "مدريد"  
},  
"Malta": {  
  "exemplarCity": "مالطة"  
},  
"Mariehamn": {  
  "exemplarCity": "ماريهامن"  
},  
"Minsk": {  
  "exemplarCity": "مينسك"
```

```
    },  
    "Monaco": {  
      "exemplarCity": "موناكو"  
    },  
    "Moscow": {  
      "exemplarCity": "موسكو"  
    },  
    "Oslo": {  
      "exemplarCity": "أوسلو"  
    },  
    "Paris": {  
      "exemplarCity": "باريس"  
    },  
    "Podgorica": {  
      "exemplarCity": "بودغوريكا"  
    },  
    "Prague": {  
      "exemplarCity": "براغ"  
    },  
    "Riga": {  
      "exemplarCity": "ريغا"  
    },  
    "Rome": {  
      "exemplarCity": "روما"  
    },  
    "Samara": {  
      "exemplarCity": "سمراء"  
    },  
    "San_Marino": {  
      "exemplarCity": "سان مارينو"  
    },  
    "Sarajevo": {  
      "exemplarCity": "سراييفو"  
    },  
    "Simferopol": {  
      "exemplarCity": "سيمفروبول"  
    },  
    "Skopje": {  
      "exemplarCity": "سكوبي"  
    },  
    "Sofia": {  
      "exemplarCity": "صوفيا"  
    },  
    "Stockholm": {  
      "exemplarCity": "ستوكهولم"  
    },  
    "Tallinn": {  
      "exemplarCity": "تالين"  
    },  
    "Tirane": {  
      "exemplarCity": "تيرانا"  
    },  
    "Ulyanovsk": {  
      "exemplarCity": "أوليانوفسك"  
    },  
    "Uzhgorod": {  
      "exemplarCity": "أوزجروود"  
    },  
  },  
  "exemplarCity": "موناكو"  
}
```

```
    },
    "Vaduz": {
      "exemplarCity": "فادوز"
    },
    "Vatican": {
      "exemplarCity": "الفاتيكان"
    },
    "Vienna": {
      "exemplarCity": "فيينا"
    },
    "Vilnius": {
      "exemplarCity": "فيلنيوس"
    },
    "Volgograd": {
      "exemplarCity": "فولجوراد"
    },
    "Warsaw": {
      "exemplarCity": "وارسو"
    },
    "Zagreb": {
      "exemplarCity": "زغرب"
    },
    "Zaporozhye": {
      "exemplarCity": "زابوروزي"
    },
    "Zurich": {
      "exemplarCity": "زيورخ"
    }
  },
  "Africa": {
    "Abidjan": {
      "exemplarCity": "أبيدجان"
    },
    "Accra": {
      "exemplarCity": "أكرا"
    },
    "Addis_Ababa": {
      "exemplarCity": "أديس أبابا"
    },
    "Algiers": {
      "exemplarCity": "الجزائر"
    },
    "Asmera": {
      "exemplarCity": "أسمرّة"
    },
    "Bamako": {
      "exemplarCity": "باماكو"
    },
    "Bangui": {
      "exemplarCity": "بانغوي"
    },
    "Banjul": {
      "exemplarCity": "بانجول"
    },
    "Bissau": {
      "exemplarCity": "بيساو"
    }
  },
}
```

```
"Blantyre": {
  "exemplarCity": "بلانتيير"
},
"Brazzaville": {
  "exemplarCity": "برازافيل"
},
"Bujumbura": {
  "exemplarCity": "بوجومبورا"
},
"Cairo": {
  "exemplarCity": "القاهرة"
},
"Casablanca": {
  "exemplarCity": "الدار البيضاء"
},
"Ceuta": {
  "exemplarCity": "سيتا"
},
"Conakry": {
  "exemplarCity": "كوناكري"
},
"Dakar": {
  "exemplarCity": "داكار"
},
"Dar_es_Salaam": {
  "exemplarCity": "دار السلام"
},
"Djibouti": {
  "exemplarCity": "جيبوتي"
},
"Douala": {
  "exemplarCity": "دوالا"
},
"El_Aaiun": {
  "exemplarCity": "العيون"
},
"Freetown": {
  "exemplarCity": "فري تاون"
},
"Gaborone": {
  "exemplarCity": "غابورون"
},
"Harare": {
  "exemplarCity": "هراري"
},
"Johannesburg": {
  "exemplarCity": "جوهانسبرغ"
},
"Juba": {
  "exemplarCity": "جوبا"
},
"Kampala": {
  "exemplarCity": "كامبالا"
},
"Khartoum": {
  "exemplarCity": "الخرطوم"
},
}
```

```
"Kigali": {
  "exemplarCity": "كيغالي"
},
"Kinshasa": {
  "exemplarCity": "كينشاسا"
},
"Lagos": {
  "exemplarCity": "لاغوس"
},
"Libreville": {
  "exemplarCity": "ليبرفيل"
},
"Lome": {
  "exemplarCity": "لومي"
},
"Luanda": {
  "exemplarCity": "لواندا"
},
"Lubumbashi": {
  "exemplarCity": "لومبباشا"
},
"Lusaka": {
  "exemplarCity": "لوساكا"
},
"Malabo": {
  "exemplarCity": "مالابو"
},
"Maputo": {
  "exemplarCity": "مابوتو"
},
"Maseru": {
  "exemplarCity": "ماسيرو"
},
"Mbabane": {
  "exemplarCity": "مباباني"
},
"Mogadishu": {
  "exemplarCity": "مقديشو"
},
"Monrovia": {
  "exemplarCity": "مونروفيا"
},
"Nairobi": {
  "exemplarCity": "نيروبي"
},
"Ndjamena": {
  "exemplarCity": "نجامينا"
},
"Niamey": {
  "exemplarCity": "نيامي"
},
"Nouakchott": {
  "exemplarCity": "نواكشوط"
},
"Ouagadougou": {
  "exemplarCity": "واغادوغو"
},
}
```

```
"Porto-Novo": {
  "exemplarCity": "بورتو نوفو"
},
"Sao_Tome": {
  "exemplarCity": "ساو تومي"
},
"Tripoli": {
  "exemplarCity": "طرابلس"
},
"Tunis": {
  "exemplarCity": "تونس"
},
"Windhoek": {
  "exemplarCity": "ويندهوك"
}
},
"Asia": {
  "Aden": {
    "exemplarCity": "عدن"
  },
  "Almaty": {
    "exemplarCity": "ألماتي"
  },
  "Amman": {
    "exemplarCity": "عمان"
  },
  "Anadyr": {
    "exemplarCity": "أندير"
  },
  "Aqtau": {
    "exemplarCity": "أكتاو"
  },
  "Aqtobe": {
    "exemplarCity": "أكتوب"
  },
  "Ashgabat": {
    "exemplarCity": "عشق آباد"
  },
  "Baghdad": {
    "exemplarCity": "بغداد"
  },
  "Bahrain": {
    "exemplarCity": "البحرين"
  },
  "Baku": {
    "exemplarCity": "باكو"
  },
  "Bangkok": {
    "exemplarCity": "بانكوك"
  },
  "Barnaul": {
    "exemplarCity": "بارناول"
  },
  "Beirut": {
    "exemplarCity": "بيروت"
  },
  "Bishkek": {
```



```
    "exemplarCity": "بشكيك"  
  },  
  "Brunei": {  
    "exemplarCity": "بروناي"  
  },  
  "Calcutta": {  
    "exemplarCity": "كالكتا"  
  },  
  "Chita": {  
    "exemplarCity": "تشيتا"  
  },  
  "Choibalsan": {  
    "exemplarCity": "تشوبالسان"  
  },  
  "Colombo": {  
    "exemplarCity": "كولومبو"  
  },  
  "Damascus": {  
    "exemplarCity": "دمشق"  
  },  
  "Dhaka": {  
    "exemplarCity": "دكا"  
  },  
  "Dili": {  
    "exemplarCity": "ديلي"  
  },  
  "Dubai": {  
    "exemplarCity": "دبي"  
  },  
  "Dushanbe": {  
    "exemplarCity": "دوشانبي"  
  },  
  "Gaza": {  
    "exemplarCity": "غزة"  
  },  
  "Hebron": {  
    "exemplarCity": "هيبرون (مدينة الخليل)"  
  },  
  "Hong_Kong": {  
    "exemplarCity": "هونغ كونغ"  
  },  
  "Hovd": {  
    "exemplarCity": "هوفد"  
  },  
  "Irkutsk": {  
    "exemplarCity": "ايركيتسك"  
  },  
  "Jakarta": {  
    "exemplarCity": "جاكرتا"  
  },  
  "Jayapura": {  
    "exemplarCity": "جاياپورا"  
  },  
  "Jerusalem": {  
    "exemplarCity": "القدس"  
  },  
  "Kabul": {
```

```
    "exemplarCity": "کابل"
  },
  "Kamchatka": {
    "exemplarCity": "کامتشاتکا"
  },
  "Karachi": {
    "exemplarCity": "کراتشي"
  },
  "Katmandu": {
    "exemplarCity": "کاتماندو"
  },
  "Khandyga": {
    "exemplarCity": "خاندیجا"
  },
  "Krasnoyarsk": {
    "exemplarCity": "کراسنویارسک"
  },
  "Kuala_Lumpur": {
    "exemplarCity": "کوالا لامپور"
  },
  "Kuching": {
    "exemplarCity": "کیشینج"
  },
  "Kuwait": {
    "exemplarCity": "الکویت"
  },
  "Macau": {
    "exemplarCity": "ماکاو"
  },
  "Magadan": {
    "exemplarCity": "مجادن"
  },
  "Makassar": {
    "exemplarCity": "ماکسار"
  },
  "Manila": {
    "exemplarCity": "مانیلا"
  },
  "Muscat": {
    "exemplarCity": "مسقط"
  },
  "Nicosia": {
    "exemplarCity": "نیقوسیا"
  },
  "Novokuznetsk": {
    "exemplarCity": "نوفوکوزنتسک"
  },
  "Novosibirsk": {
    "exemplarCity": "نوفوسبیرسک"
  },
  "Omsk": {
    "exemplarCity": "أومسک"
  },
  "Oral": {
    "exemplarCity": "أورال"
  },
  "Phnom_Penh": {
```

```
    "exemplarCity": "بنوم بنه"
  },
  "Pontianak": {
    "exemplarCity": "بونتيانك"
  },
  "Pyongyang": {
    "exemplarCity": "بيونغ يانغ"
  },
  "Qatar": {
    "exemplarCity": "قطر"
  },
  "Qyzylorda": {
    "exemplarCity": "كيزيلوردا"
  },
  "Rangoon": {
    "exemplarCity": "رانغون"
  },
  "Riyadh": {
    "exemplarCity": "الرياض"
  },
  "Saigon": {
    "exemplarCity": "مدينة هو تشي منه"
  },
  "Sakhalin": {
    "exemplarCity": "سكالين"
  },
  "Samarkand": {
    "exemplarCity": "سمرقند"
  },
  "Seoul": {
    "exemplarCity": "سول"
  },
  "Shanghai": {
    "exemplarCity": "شنغهاي"
  },
  "Singapore": {
    "exemplarCity": "سنغافورة"
  },
  "Srednekolymsk": {
    "exemplarCity": "سريدنكوليمسك"
  },
  "Taipei": {
    "exemplarCity": "تايبيه"
  },
  "Tashkent": {
    "exemplarCity": "تشقند"
  },
  "Tbilisi": {
    "exemplarCity": "تبليسي"
  },
  "Tehran": {
    "exemplarCity": "تهران"
  },
  "Thimphu": {
    "exemplarCity": "تيمفو"
  },
  "Tokyo": {
```

```
    "exemplarCity": "طوكيو"
  },
  "Tomsk": {
    "exemplarCity": "تومسك"
  },
  "Ulaanbaatar": {
    "exemplarCity": "آلانباتار"
  },
  "Urumqi": {
    "exemplarCity": "أرومكي"
  },
  "Ust-Nera": {
    "exemplarCity": "أوست نيرا"
  },
  "Vientiane": {
    "exemplarCity": "فيانتيان"
  },
  "Vladivostok": {
    "exemplarCity": "فلاديفوستك"
  },
  "Yakutsk": {
    "exemplarCity": "ياكتسك"
  },
  "Yekaterinburg": {
    "exemplarCity": "يكاترنبيرج"
  },
  "Yerevan": {
    "exemplarCity": "يريفان"
  }
},
"Indian": {
  "Antananarivo": {
    "exemplarCity": "أنتاناناريفو"
  },
  "Chagos": {
    "exemplarCity": "تشاغوس"
  },
  "Christmas": {
    "exemplarCity": "كريسماس"
  },
  "Cocos": {
    "exemplarCity": "كوكوس"
  },
  "Comoro": {
    "exemplarCity": "جزر القمر"
  },
  "Kerguelen": {
    "exemplarCity": "كيرغويلين"
  },
  "Mahe": {
    "exemplarCity": "ماهي"
  },
  "Maldives": {
    "exemplarCity": "المالديف"
  },
  "Mauritius": {
    "exemplarCity": "موريشيوس"
  }
}
```

```
    },  
    "Mayotte": {  
      "exemplarCity": "مايوت"  
    },  
    "Reunion": {  
      "exemplarCity": "ريونيون"  
    }  
  },  
  "Australia": {  
    "Adelaide": {  
      "exemplarCity": "أديليد"  
    },  
    "Brisbane": {  
      "exemplarCity": "برسبان"  
    },  
    "Broken_Hill": {  
      "exemplarCity": "بروكن هيل"  
    },  
    "Currie": {  
      "exemplarCity": "كوري"  
    },  
    "Darwin": {  
      "exemplarCity": "دارون"  
    },  
    "Eucla": {  
      "exemplarCity": "أوكلأ"  
    },  
    "Hobart": {  
      "exemplarCity": "هوبارت"  
    },  
    "Lindeman": {  
      "exemplarCity": "ليندمان"  
    },  
    "Lord_Howe": {  
      "exemplarCity": "لورد هاو"  
    },  
    "Melbourne": {  
      "exemplarCity": "ميلبورن"  
    },  
    "Perth": {  
      "exemplarCity": "برثأ"  
    },  
    "Sydney": {  
      "exemplarCity": "سيدني"  
    }  
  },  
  "Pacific": {  
    "Apia": {  
      "exemplarCity": "أبيا"  
    },  
    "Auckland": {  
      "exemplarCity": "أوكلاند"  
    },  
    "Bougainville": {  
      "exemplarCity": "بوغانفيل"  
    },  
    "Chatham": {
```

```
    "exemplarCity": "تشاثام"
  },
  "Easter": {
    "exemplarCity": "استر"
  },
  "Efate": {
    "exemplarCity": "إيفات"
  },
  "Enderbury": {
    "exemplarCity": "اندربيرج"
  },
  "Fakaofu": {
    "exemplarCity": "فاكاوفو"
  },
  "Fiji": {
    "exemplarCity": "فيجي"
  },
  "Funafuti": {
    "exemplarCity": "فونافوتي"
  },
  "Galapagos": {
    "exemplarCity": "جلاباجوس"
  },
  "Gambier": {
    "exemplarCity": "جامبير"
  },
  "Guadalcanal": {
    "exemplarCity": "غواد الكانال"
  },
  "Guam": {
    "exemplarCity": "غوام"
  },
  "Honolulu": {
    "exemplarCity": "هونولولو"
  },
  "Johnston": {
    "exemplarCity": "جونستون"
  },
  "Kiritimati": {
    "exemplarCity": "كيريتي ماتي"
  },
  "Kosrae": {
    "exemplarCity": "كوسرا"
  },
  "Kwajalein": {
    "exemplarCity": "كواجالين"
  },
  "Majuro": {
    "exemplarCity": "ماجورو"
  },
  "Marquesas": {
    "exemplarCity": "ماركيساس"
  },
  "Midway": {
    "exemplarCity": "ميدواي"
  },
  "Nauru": {
```

```
    "exemplarCity": "ناورو"  
  },  
  "Niue": {  
    "exemplarCity": "نيوي"  
  },  
  "Norfolk": {  
    "exemplarCity": "نورفولك"  
  },  
  "Noumea": {  
    "exemplarCity": "نوميا"  
  },  
  "Pago_Pago": {  
    "exemplarCity": "باغو باغو"  
  },  
  "Palau": {  
    "exemplarCity": "بالاو"  
  },  
  "Pitcairn": {  
    "exemplarCity": "بيتكيرن"  
  },  
  "Ponape": {  
    "exemplarCity": "باناب"  
  },  
  "Port_Moresby": {  
    "exemplarCity": "بور مورسبي"  
  },  
  "Rarotonga": {  
    "exemplarCity": "راروتونغا"  
  },  
  "Saipan": {  
    "exemplarCity": "سايبان"  
  },  
  "Tahiti": {  
    "exemplarCity": "تاهيتي"  
  },  
  "Tarawa": {  
    "exemplarCity": "تاراوا"  
  },  
  "Tongatapu": {  
    "exemplarCity": "تونغاتابو"  
  },  
  "Truk": {  
    "exemplarCity": "ترك"  
  },  
  "Wake": {  
    "exemplarCity": "واك"  
  },  
  "Wallis": {  
    "exemplarCity": "واليس"  
  },  
  },  
  "Arctic": {  
    "Longyearbyen": {  
      "exemplarCity": "لونجيربين"  
    }  
  },  
  "Antarctica": {
```

```
"Casey": {
  "exemplarCity": "كاساي"
},
"Davis": {
  "exemplarCity": "دافيز"
},
"DumontDUrville": {
  "exemplarCity": "دي مونت دو روفيل"
},
"Macquarie": {
  "exemplarCity": "ماكواري"
},
"Mawson": {
  "exemplarCity": "ماوسون"
},
"McMurdo": {
  "exemplarCity": "ماك موردو"
},
"Palmer": {
  "exemplarCity": "بالمير"
},
"Rothera": {
  "exemplarCity": "روثيرا"
},
"Syowa": {
  "exemplarCity": "سايووا"
},
"Troll": {
  "exemplarCity": "ترول"
},
"Vostok": {
  "exemplarCity": "فوستوك"
}
},
"Etc": {
  "GMT": {
    "exemplarCity": "GMT"
  },
  "GMT1": {
    "exemplarCity": "GMT+1"
  },
  "GMT10": {
    "exemplarCity": "GMT+10"
  },
  "GMT11": {
    "exemplarCity": "GMT+11"
  },
  "GMT12": {
    "exemplarCity": "GMT+12"
  },
  "GMT2": {
    "exemplarCity": "GMT+2"
  },
  "GMT3": {
    "exemplarCity": "GMT+3"
  },
  "GMT4": {
```



```
    "exemplarCity": "GMT+4"
  },
  "GMT5": {
    "exemplarCity": "GMT+5"
  },
  "GMT6": {
    "exemplarCity": "GMT+6"
  },
  "GMT7": {
    "exemplarCity": "GMT+7"
  },
  "GMT8": {
    "exemplarCity": "GMT+8"
  },
  "GMT9": {
    "exemplarCity": "GMT+9"
  },
  "GMT-1": {
    "exemplarCity": "GMT-1"
  },
  "GMT-10": {
    "exemplarCity": "GMT-10"
  },
  "GMT-11": {
    "exemplarCity": "GMT-11"
  },
  "GMT-12": {
    "exemplarCity": "GMT-12"
  },
  "GMT-13": {
    "exemplarCity": "GMT-13"
  },
  "GMT-14": {
    "exemplarCity": "GMT-14"
  },
  "GMT-2": {
    "exemplarCity": "GMT-2"
  },
  "GMT-3": {
    "exemplarCity": "GMT-3"
  },
  "GMT-4": {
    "exemplarCity": "GMT-4"
  },
  "GMT-5": {
    "exemplarCity": "GMT-5"
  },
  "GMT-6": {
    "exemplarCity": "GMT-6"
  },
  "GMT-7": {
    "exemplarCity": "GMT-7"
  },
  "GMT-8": {
    "exemplarCity": "GMT-8"
  },
  "GMT-9": {
```

```

        "exemplarCity": "GMT-9"
    },
    "Unknown": {
        "exemplarCity": "مدينة غير معروفة"
    }
},
"metazone": {
    "Afghanistan": {
        "long": {
            "standard": "توقيت أفغانستان"
        }
    },
    "Africa_Central": {
        "long": {
            "standard": "توقيت وسط أفريقيا"
        }
    },
    "Africa_Eastern": {
        "long": {
            "standard": "توقيت شرق أفريقيا"
        }
    },
    "Africa_Southern": {
        "long": {
            "standard": "توقيت جنوب أفريقيا"
        }
    },
    "Africa_Western": {
        "long": {
            "generic": "توقيت غرب أفريقيا",
            "standard": "توقيت غرب أفريقيا الرسمي",
            "daylight": "توقيت غرب أفريقيا الصيفي"
        }
    },
    "Alaska": {
        "long": {
            "generic": "توقيت ألاسكا",
            "standard": "التوقيت الرسمي لألاسكا",
            "daylight": "توقيت ألاسكا الصيفي"
        }
    },
    "Amazon": {
        "long": {
            "generic": "توقيت الأمازون",
            "standard": "توقيت الأمازون الرسمي",
            "daylight": "توقيت الأمازون الصيفي"
        }
    },
    "America_Central": {
        "long": {
            "generic": "التوقيت المركزي لأمريكا الشمالية",
            "standard": "التوقيت الرسمي المركزي لأمريكا الشمالية",
            "daylight": "التوقيت الصيفي المركزي لأمريكا الشمالية"
        }
    },
    "America_Eastern": {

```

```

    "long": {
      "generic": "التوقيت الشرقي لأمريكا الشمالية",
      "standard": "التوقيت الرسمي الشرقي لأمريكا الشمالية",
      "daylight": "التوقيت الصيفي الشرقي لأمريكا الشمالية"
    }
  },
  "America_Mountain": {
    "long": {
      "generic": "التوقيت الجبلي لأمريكا الشمالية",
      "standard": "التوقيت الجبلي الرسمي لأمريكا الشمالية",
      "daylight": "التوقيت الجبلي الصيفي لأمريكا الشمالية"
    }
  },
  "America_Pacific": {
    "long": {
      "generic": "توقيت المحيط الهادي",
      "standard": "توقيت المحيط الهادي الرسمي",
      "daylight": "توقيت المحيط الهادي الصيفي"
    }
  },
  "Anadyr": {
    "long": {
      "generic": "توقيت أنادير",
      "standard": "توقيت أنادير الرسمي",
      "daylight": "التوقيت الصيفي لأنادير"
    }
  },
  "Apia": {
    "long": {
      "generic": "توقيت آبيا",
      "standard": "التوقيت الرسمي لآبيا",
      "daylight": "التوقيت الصيفي لآبيا"
    }
  },
  "Arabian": {
    "long": {
      "generic": "التوقيت العربي",
      "standard": "التوقيت العربي الرسمي",
      "daylight": "التوقيت العربي الصيفي"
    }
  },
  "Argentina": {
    "long": {
      "generic": "توقيت الأرجنتين",
      "standard": "توقيت الأرجنتين الرسمي",
      "daylight": "توقيت الأرجنتين الصيفي"
    }
  },
  "Argentina_Western": {
    "long": {
      "generic": "توقيت غرب الأرجنتين",
      "standard": "توقيت غرب الأرجنتين الرسمي",
      "daylight": "توقيت غرب الأرجنتين الصيفي"
    }
  },
  "Armenia": {
    "long": {

```

```

        "generic": "توقيت أرمينيا",
        "standard": "توقيت أرمينيا الرسمي",
        "daylight": "توقيت أرمينيا الصيفي"
    },
},
"Atlantic": {
    "long": {
        "generic": "توقيت الأطلسي",
        "standard": "التوقيت الرسمي الأطلسي",
        "daylight": "التوقيت الصيفي الأطلسي"
    }
},
"Australia_Central": {
    "long": {
        "generic": "توقيت وسط أستراليا",
        "standard": "توقيت وسط أستراليا الرسمي",
        "daylight": "توقيت وسط أستراليا الصيفي"
    }
},
"Australia_CentralWestern": {
    "long": {
        "generic": "توقيت غرب وسط أستراليا",
        "standard": "توقيت غرب وسط أستراليا الرسمي",
        "daylight": "توقيت غرب وسط أستراليا الصيفي"
    }
},
"Australia_Eastern": {
    "long": {
        "generic": "توقيت شرق أستراليا",
        "standard": "توقيت شرق أستراليا الرسمي",
        "daylight": "توقيت شرق أستراليا الصيفي"
    }
},
"Australia_Western": {
    "long": {
        "generic": "توقيت غرب أستراليا",
        "standard": "توقيت غرب أستراليا الرسمي",
        "daylight": "توقيت غرب أستراليا الصيفي"
    }
},
"Azerbaijan": {
    "long": {
        "generic": "توقيت أذربيجان",
        "standard": "توقيت أذربيجان الرسمي",
        "daylight": "توقيت أذربيجان الصيفي"
    }
},
"Azores": {
    "long": {
        "generic": "توقيت أزورس",
        "standard": "توقيت أزورس الرسمي",
        "daylight": "توقيت أزورس الصيفي"
    }
},
"Bangladesh": {
    "long": {
        "generic": "توقيت بنجلاديش",

```

```

        "standard": "توقيت بنجلاديش الرسمي",
        "daylight": "توقيت بنجلاديش الصيفي"
    },
    },
    "Bhutan": {
        "long": {
            "standard": "توقيت بوتان"
        }
    },
    },
    "Bolivia": {
        "long": {
            "standard": "توقيت بوليفيا"
        }
    },
    },
    "Brasilia": {
        "long": {
            "generic": "توقيت برازيليا",
            "standard": "توقيت برازيليا الرسمي",
            "daylight": "توقيت برازيليا الصيفي"
        }
    },
    },
    "Brunei": {
        "long": {
            "standard": "توقيت بروناي"
        }
    },
    },
    "Cape_Verde": {
        "long": {
            "generic": "توقيت الرأس الأخضر",
            "standard": "توقيت الرأس الأخضر الرسمي",
            "daylight": "توقيت الرأس الأخضر الصيفي"
        }
    },
    },
    "Chamorro": {
        "long": {
            "standard": "توقيت تشامورو"
        }
    },
    },
    "Chatham": {
        "long": {
            "generic": "توقيت تشاتام",
            "standard": "توقيت تشاتام الرسمي",
            "daylight": "توقيت تشاتام الصيفي"
        }
    },
    },
    "Chile": {
        "long": {
            "generic": "توقيت شيلي",
            "standard": "توقيت شيلي الرسمي",
            "daylight": "توقيت شيلي الصيفي"
        }
    },
    },
    "China": {
        "long": {
            "generic": "توقيت الصين",
            "standard": "توقيت الصين الرسمي",
            "daylight": "توقيت الصين الصيفي"
        }
    },
    },

```

```

    }
  },
  "Choibalsan": {
    "long": {
      "generic": "توقيت شويبالسان",
      "standard": "توقيت شويبالسان الرسمي",
      "daylight": "التوقيت الصيفي لشويبالسان"
    }
  },
  "Christmas": {
    "long": {
      "standard": "توقيت جزر الكريسماس"
    }
  },
  "Cocos": {
    "long": {
      "standard": "توقيت جزر كوكوس"
    }
  },
  "Colombia": {
    "long": {
      "generic": "توقيت كولومبيا",
      "standard": "توقيت كولومبيا الرسمي",
      "daylight": "توقيت كولومبيا الصيفي"
    }
  },
  "Cook": {
    "long": {
      "generic": "توقيت جزر كوك",
      "standard": "توقيت جزر كوك الرسمي",
      "daylight": "توقيت جزر كوك الصيفي"
    }
  },
  "Cuba": {
    "long": {
      "generic": "توقيت كوبا",
      "standard": "توقيت كوبا الرسمي",
      "daylight": "توقيت كوبا الصيفي"
    }
  },
  "Davis": {
    "long": {
      "standard": "توقيت دافيز"
    }
  },
  "DumontDUrville": {
    "long": {
      "standard": "توقيت دي مونت دو روفيل"
    }
  },
  "East_Timor": {
    "long": {
      "standard": "توقيت تيمور الشرقية"
    }
  },
  "Easter": {
    "long": {

```

```

        "generic": "توقيت جزيرة استر",
        "standard": "توقيت جزيرة استر الرسمي",
        "daylight": "توقيت جزيرة استر الصيفي"
    },
},
"Ecuador": {
    "long": {
        "standard": "توقيت الإكوادور"
    }
},
"Europe_Central": {
    "long": {
        "generic": "توقيت وسط أوروبا",
        "standard": "توقيت وسط أوروبا الرسمي",
        "daylight": "توقيت وسط أوروبا الصيفي"
    }
},
"Europe_Eastern": {
    "long": {
        "generic": "توقيت شرق أوروبا",
        "standard": "توقيت شرق أوروبا الرسمي",
        "daylight": "توقيت شرق أوروبا الصيفي"
    }
},
"Europe_Further_Eastern": {
    "long": {
        "standard": "التوقيت الأوروبي (أكثر شرقًا)"
    }
},
"Europe_Western": {
    "long": {
        "generic": "توقيت غرب أوروبا",
        "standard": "توقيت غرب أوروبا الرسمي",
        "daylight": "توقيت غرب أوروبا الصيفي"
    }
},
"Falkland": {
    "long": {
        "generic": "توقيت جزر فوكلاند",
        "standard": "توقيت جزر فوكلاند الرسمي",
        "daylight": "توقيت جزر فوكلاند الصيفي"
    }
},
"Fiji": {
    "long": {
        "generic": "توقيت فيجي",
        "standard": "توقيت فيجي الرسمي",
        "daylight": "توقيت فيجي الصيفي"
    }
},
"French_Guiana": {
    "long": {
        "standard": "توقيت غايانا الفرنسية"
    }
},
"French_Southern": {
    "long": {

```

```

        "standard": "توقيت المقاطعات الفرنسية الجنوبية والانتارتيكية"
    },
    "Galapagos": {
        "long": {
            "standard": "توقيت غلاباغوس"
        }
    },
    "Gambier": {
        "long": {
            "standard": "توقيت جامبير"
        }
    },
    "Georgia": {
        "long": {
            "generic": "توقيت جورجيا",
            "standard": "توقيت جورجيا الرسمي",
            "daylight": "توقيت جورجيا الصيفي"
        }
    },
    "Gilbert_Islands": {
        "long": {
            "standard": "توقيت جزر جيلبرت"
        }
    },
    "GMT": {
        "long": {
            "standard": "توقيت غرينتش"
        }
    },
    "Greenland_Eastern": {
        "long": {
            "generic": "توقيت شرق غرينلاند",
            "standard": "توقيت شرق غرينلاند الرسمي",
            "daylight": "توقيت شرق غرينلاند الصيفي"
        }
    },
    "Greenland_Western": {
        "long": {
            "generic": "توقيت غرب غرينلاند",
            "standard": "توقيت غرب غرينلاند الرسمي",
            "daylight": "توقيت غرب غرينلاند الصيفي"
        }
    },
    "Guam": {
        "long": {
            "standard": "توقيت غوام"
        }
    },
    "Gulf": {
        "long": {
            "standard": "توقيت الخليج"
        }
    },
    "Guyana": {
        "long": {
            "standard": "توقيت غيانا"
        }
    }

```



```

    }
  },
  "Hawaii_Aleutian": {
    "long": {
      "generic": "توقيت هاواي ألوتيان",
      "standard": "توقيت هاواي ألوتيان الرسمي",
      "daylight": "توقيت هاواي ألوتيان الصيفي"
    }
  },
  "Hong_Kong": {
    "long": {
      "generic": "توقيت هونغ كونغ",
      "standard": "توقيت هونغ كونغ الرسمي",
      "daylight": "توقيت هونغ كونغ الصيفي"
    }
  },
  "Hovd": {
    "long": {
      "generic": "توقيت هوفد",
      "standard": "توقيت هوفد الرسمي",
      "daylight": "توقيت هوفد الصيفي"
    }
  },
  "India": {
    "long": {
      "standard": "توقيت الهند"
    }
  },
  "Indian_Ocean": {
    "long": {
      "standard": "توقيت المحيط الهندي"
    }
  },
  "Indochina": {
    "long": {
      "standard": "توقيت الهند الصينية"
    }
  },
  "Indonesia_Central": {
    "long": {
      "standard": "توقيت وسط إندونيسيا"
    }
  },
  "Indonesia_Eastern": {
    "long": {
      "standard": "توقيت شرق إندونيسيا"
    }
  },
  "Indonesia_Western": {
    "long": {
      "standard": "توقيت غرب إندونيسيا"
    }
  },
  "Iran": {
    "long": {
      "generic": "توقيت إيران",
      "standard": "توقيت إيران الرسمي",

```

```

        "daylight": "توقيت إيران الصيفي"
    },
    },
    "Irkutsk": {
        "long": {
            "generic": "توقيت إركوتسك",
            "standard": "توقيت إركوتسك الرسمي",
            "daylight": "توقيت إركوتسك الصيفي"
        }
    },
    "Israel": {
        "long": {
            "generic": "توقيت إسرائيل",
            "standard": "توقيت إسرائيل الرسمي",
            "daylight": "توقيت إسرائيل الصيفي"
        }
    },
    "Japan": {
        "long": {
            "generic": "توقيت اليابان",
            "standard": "توقيت اليابان الرسمي",
            "daylight": "توقيت اليابان الصيفي"
        }
    },
    "Kamchatka": {
        "long": {
            "generic": "توقيت كامشاتكا",
            "standard": "توقيت بيتروبافلوفسك-كامتشاتسكي",
            "daylight": "توقيت بيتروبافلوفسك-كامتشاتسكي الصيفي"
        }
    },
    "Kazakhstan_Eastern": {
        "long": {
            "standard": "توقيت شرق كازاخستان"
        }
    },
    "Kazakhstan_Western": {
        "long": {
            "standard": "توقيت غرب كازاخستان"
        }
    },
    "Korea": {
        "long": {
            "generic": "توقيت كوريا",
            "standard": "توقيت كوريا الرسمي",
            "daylight": "توقيت كوريا الصيفي"
        }
    },
    "Kosrae": {
        "long": {
            "standard": "توقيت كوسرا"
        }
    },
    "Krasnoyarsk": {
        "long": {
            "generic": "توقيت كراسنويارسك",
            "standard": "توقيت كراسنويارسك الرسمي",

```

```

        "daylight": "التوقيت الصيفي لكراسنويارسك"
    },
    },
    "Kyrgystan": {
        "long": {
            "standard": "توقيت قرغيزستان"
        }
    },
    },
    "Line_Islands": {
        "long": {
            "standard": "توقيت جزر لاين"
        }
    },
    },
    "Lord_Howe": {
        "long": {
            "generic": "توقيت لورد هاو",
            "standard": "توقيت لورد هاو الرسمي",
            "daylight": "التوقيت الصيفي للورد هاو"
        }
    },
    },
    "Macquarie": {
        "long": {
            "standard": "توقيت ماكوارى"
        }
    },
    },
    "Magadan": {
        "long": {
            "generic": "توقيت ماغادان",
            "standard": "توقيت ماغادان الرسمي",
            "daylight": "توقيت ماغادان الصيفي"
        }
    },
    },
    "Malaysia": {
        "long": {
            "standard": "توقيت ماليزيا"
        }
    },
    },
    "Maldives": {
        "long": {
            "standard": "توقيت المالديف"
        }
    },
    },
    "Marquesas": {
        "long": {
            "standard": "توقيت ماركيساس"
        }
    },
    },
    "Marshall_Islands": {
        "long": {
            "standard": "توقيت جزر مارشال"
        }
    },
    },
    "Mauritius": {
        "long": {
            "generic": "توقيت موريشيوس",
            "standard": "توقيت موريشيوس الرسمي",
            "daylight": "توقيت موريشيوس الصيفي"
        }
    }

```

```

    }
  },
  "Mawson": {
    "long": {
      "standard": "توقيت ماوسون"
    }
  },
  "Mexico_Northwest": {
    "long": {
      "generic": "توقيت شمال غرب المكسيك",
      "standard": "التوقيت الرسمي لشمال غرب المكسيك",
      "daylight": "التوقيت الصيفي لشمال غرب المكسيك"
    }
  },
  "Mexico_Pacific": {
    "long": {
      "generic": "توقيت المحيط الهادي للمكسيك",
      "standard": "توقيت المحيط الهادي الرسمي للمكسيك",
      "daylight": "توقيت المحيط الهادي الصيفي للمكسيك"
    }
  },
  "Mongolia": {
    "long": {
      "generic": "توقيت أولان باتور",
      "standard": "توقيت أولان باتور الرسمي",
      "daylight": "توقيت أولان باتور الصيفي"
    }
  },
  "Moscow": {
    "long": {
      "generic": "توقيت موسكو",
      "standard": "توقيت موسكو الرسمي",
      "daylight": "توقيت موسكو الصيفي"
    }
  },
  "Myanmar": {
    "long": {
      "standard": "توقيت ميانمار"
    }
  },
  "Nauru": {
    "long": {
      "standard": "توقيت ناورو"
    }
  },
  "Nepal": {
    "long": {
      "standard": "توقيت نيبال"
    }
  },
  "New_Caledonia": {
    "long": {
      "generic": "توقيت كاليدونيا الجديدة",
      "standard": "توقيت كاليدونيا الجديدة الرسمي",
      "daylight": "توقيت كاليدونيا الجديدة الصيفي"
    }
  },
},

```

```
"New_Zealand": {
  "long": {
    "generic": "توقيت نيوزيلندا",
    "standard": "توقيت نيوزيلندا الرسمي",
    "daylight": "توقيت نيوزيلندا الصيفي"
  }
},
"Newfoundland": {
  "long": {
    "generic": "توقيت نيوفاوندلاند",
    "standard": "توقيت نيوفاوندلاند الرسمي",
    "daylight": "توقيت نيوفاوندلاند الصيفي"
  }
},
"Niue": {
  "long": {
    "standard": "توقيت نيوي"
  }
},
"Norfolk": {
  "long": {
    "standard": "توقيت جزيرة نورفولك"
  }
},
"Noronha": {
  "long": {
    "generic": "توقيت فيرناندو دي نورونها",
    "standard": "توقيت فرناندو دي نورونها الرسمي",
    "daylight": "توقيت فرناندو دي نورونها الصيفي"
  }
},
"North_Mariana": {
  "long": {
    "standard": "توقيت جزر ماريانا الشمالية"
  }
},
"Novosibirsk": {
  "long": {
    "generic": "توقيت نوفوسيبيرسك",
    "standard": "توقيت نوفوسيبيرسك الرسمي",
    "daylight": "توقيت نوفوسيبيرسك الصيفي"
  }
},
"Omsk": {
  "long": {
    "generic": "توقيت أومسك",
    "standard": "توقيت أومسك الرسمي",
    "daylight": "توقيت أومسك الصيفي"
  }
},
"Pakistan": {
  "long": {
    "generic": "توقيت باكستان",
    "standard": "توقيت باكستان الرسمي",
    "daylight": "توقيت باكستان الصيفي"
  }
},
}
```

```
"Palau": {
  "long": {
    "standard": "توقيت بالاو"
  }
},
"Papua_New_Guinea": {
  "long": {
    "standard": "توقيت بابوا غينيا الجديدة"
  }
},
"Paraguay": {
  "long": {
    "generic": "توقيت باراغواي",
    "standard": "توقيت باراغواي الرسمي",
    "daylight": "توقيت باراغواي الصيفي"
  }
},
"Peru": {
  "long": {
    "generic": "توقيت بيرو",
    "standard": "توقيت بيرو الرسمي",
    "daylight": "توقيت بيرو الصيفي"
  }
},
"Philippines": {
  "long": {
    "generic": "توقيت الفلبين",
    "standard": "توقيت الفلبين الرسمي",
    "daylight": "توقيت الفلبين الصيفي"
  }
},
"Phoenix_Islands": {
  "long": {
    "standard": "توقيت جزر فينكس"
  }
},
"Pierre_Miquelon": {
  "long": {
    "generic": "توقيت سانت بيير وميكلون",
    "standard": "توقيت سانت بيير وميكلون الرسمي",
    "daylight": "توقيت سانت بيير وميكلون الصيفي"
  }
},
"Pitcairn": {
  "long": {
    "standard": "توقيت بيتكيرن"
  }
},
"Ponape": {
  "long": {
    "standard": "توقيت بونابي"
  }
},
"Pyongyang": {
  "long": {
    "standard": "توقيت بيونغ يانغ"
  }
}
```

```

    },
    "Reunion": {
      "long": {
        "standard": "توقيت ريونيون"
      }
    },
    "Rothera": {
      "long": {
        "standard": "توقيت روثيرا"
      }
    },
    "Sakhalin": {
      "long": {
        "generic": "توقيت ساخالين",
        "standard": "توقيت ساخالين الرسمي",
        "daylight": "توقيت ساخالين الصيفي"
      }
    },
    "Samara": {
      "long": {
        "generic": "توقيت سامارا",
        "standard": "توقيت سامارا",
        "daylight": "توقيت سامارا الصيفي"
      }
    },
    "Samoa": {
      "long": {
        "generic": "توقيت ساموا",
        "standard": "توقيت ساموا الرسمي",
        "daylight": "توقيت ساموا الصيفي"
      }
    },
    "Seychelles": {
      "long": {
        "standard": "توقيت سيشل"
      }
    },
    "Singapore": {
      "long": {
        "standard": "توقيت سنغافورة"
      }
    },
    "Solomon": {
      "long": {
        "standard": "توقيت جزر سليمان"
      }
    },
    "South_Georgia": {
      "long": {
        "standard": "توقيت جنوب جورجيا"
      }
    },
    "Suriname": {
      "long": {
        "standard": "توقيت سورينام"
      }
    }
  },

```

```
"Syowa": {
  "long": {
    "standard": "توقيت سايووا"
  }
},
"Tahiti": {
  "long": {
    "standard": "توقيت تاهيتي"
  }
},
"Taipei": {
  "long": {
    "generic": "توقيت تايبه",
    "standard": "توقيت تايبه الرسمي",
    "daylight": "توقيت تايبه الصيفي"
  }
},
"Tajikistan": {
  "long": {
    "standard": "توقيت طاجكستان"
  }
},
"Tokelau": {
  "long": {
    "standard": "توقيت توكيلاو"
  }
},
"Tonga": {
  "long": {
    "generic": "توقيت تونغا",
    "standard": "توقيت تونغا الرسمي",
    "daylight": "توقيت تونغا الصيفي"
  }
},
"Truk": {
  "long": {
    "standard": "توقيت شوك"
  }
},
"Turkmenistan": {
  "long": {
    "generic": "توقيت تركمانستان",
    "standard": "توقيت تركمانستان الرسمي",
    "daylight": "توقيت تركمانستان الصيفي"
  }
},
"Tuvalu": {
  "long": {
    "standard": "توقيت توفالو"
  }
},
"Uruguay": {
  "long": {
    "generic": "توقيت أورغواي",
    "standard": "توقيت أورغواي الرسمي",
    "daylight": "توقيت أورغواي الصيفي"
  }
}
```



```
    },
    "Uzbekistan": {
      "long": {
        "generic": "توقيت أوزبكستان",
        "standard": "توقيت أوزبكستان الرسمي",
        "daylight": "توقيت أوزبكستان الصيفي"
      }
    },
    "Vanuatu": {
      "long": {
        "generic": "توقيت فانواتو",
        "standard": "توقيت فانواتو الرسمي",
        "daylight": "توقيت فانواتو الصيفي"
      }
    },
    "Venezuela": {
      "long": {
        "standard": "توقيت فنزويلا"
      }
    },
    "Vladivostok": {
      "long": {
        "generic": "توقيت فلاديفوستوك",
        "standard": "توقيت فلاديفوستوك الرسمي",
        "daylight": "توقيت فلاديفوستوك الصيفي"
      }
    },
    "Volgograd": {
      "long": {
        "generic": "توقيت فولغوغراد",
        "standard": "توقيت فولغوغراد الرسمي",
        "daylight": "توقيت فولغوغراد الصيفي"
      }
    },
    "Vostok": {
      "long": {
        "standard": "توقيت فوستوك"
      }
    },
    "Wake": {
      "long": {
        "standard": "توقيت جزيرة ويك"
      }
    },
    "Wallis": {
      "long": {
        "standard": "توقيت واليس و فوتونا"
      }
    },
    "Yakutsk": {
      "long": {
        "generic": "توقيت ياكوتسك",
        "standard": "توقيت ياكوتسك الرسمي",
        "daylight": "توقيت ياكوتسك الصيفي"
      }
    },
    "Yekaterinburg": {
```

```

        "long": {
          "generic": "توقيت يكاترينبورغ",
          "standard": "توقيت يكاترينبورغ الرسمي",
          "daylight": "توقيت يكاترينبورغ الصيفي"
        }
      }
    }
  }
}

```

### TIMEZONENAMES.JSX

```

{
  "main";
  {
    "ar";
    {
      "identity";
      {
        "version";
        {
          "_number";
          "$Revision: 12879 $",
          "_cldrVersion";
          "30.0.3";
        }
        "language";
        "ar";
      }
      "dates";
      {
        "timeZoneNames";
        {
          "hourFormat";
          "+HH:mm;-HH:mm",
          "gmtFormat";
          "0{جرينتش",
          "gmtZeroFormat";
          "جرينتش",
          "regionFormat";
          "0{توقيت",
          "regionFormat-type-daylight";
          "توقيت } 0{ الصيفي",
          "regionFormat-type-standard";
          "توقيت } 0{ الرسمي",
          "fallbackFormat";
          "{1} ({0})",
          "zone";
          {
            "America";
            {
              "Adak";
              {

```

```
        "exemplarCity";
        "أدراك";
    }
    "Anchorage";
    {
        "exemplarCity";
        "أنشوراج";
    }
    "Anguilla";
    {
        "exemplarCity";
        "أنغيلا";
    }
    "Antigua";
    {
        "exemplarCity";
        "أنتيغوا";
    }
    "Araguaina";
    {
        "exemplarCity";
        "أروجوانيا";
    }
    "Argentina";
    {
        "Rio_Gallegos";
        {
            "exemplarCity";
            "ريو جالييوس";
        }
        "San_Juan";
        {
            "exemplarCity";
            "سان خوان";
        }
        "Ushuaia";
        {
            "exemplarCity";
            "أشوا";
        }
        "La_Rioja";
        {
            "exemplarCity";
            "لا ريوجا";
        }
        "San_Luis";
        {
            "exemplarCity";
            "سان لويس";
        }
        "Salta";
        {
            "exemplarCity";
            "سالطا";
        }
        "Tucuman";
        {
```

```
        "exemplarCity";
        "تاکمان";
    }
}
"Aruba";
{
    "exemplarCity";
    "أروبا";
}
"Asuncion";
{
    "exemplarCity";
    "أسونسيون";
}
"Bahia";
{
    "exemplarCity";
    "باهيا";
}
"Bahia_Banderas";
{
    "exemplarCity";
    "باهيا بانديراس";
}
"Barbados";
{
    "exemplarCity";
    "بربادوس";
}
"Belem";
{
    "exemplarCity";
    "بلم";
}
"Belize";
{
    "exemplarCity";
    "بليز";
}
"Blanc-Sablon";
{
    "exemplarCity";
    "بلانك-سابلون";
}
"Boa_Vista";
{
    "exemplarCity";
    "باو فيستا";
}
"Bogota";
{
    "exemplarCity";
    "بوغوتا";
}
"Boise";
{
    "exemplarCity";
```

```
        "بويس";
    }
    "Buenos_Aires";
    {
        "exemplarCity";
        "بوينوس آيرس";
    }
    "Cambridge_Bay";
    {
        "exemplarCity";
        "كامبرديج باي";
    }
    "Campo_Grande";
    {
        "exemplarCity";
        "كومبو جراند";
    }
    "Cancun";
    {
        "exemplarCity";
        "كانكون";
    }
    "Caracas";
    {
        "exemplarCity";
        "كاراكاس";
    }
    "Catamarca";
    {
        "exemplarCity";
        "كاتاماركا";
    }
    "Cayenne";
    {
        "exemplarCity";
        "كايين";
    }
    "Cayman";
    {
        "exemplarCity";
        "كيمان";
    }
    "Chicago";
    {
        "exemplarCity";
        "شيكاغو";
    }
    "Chihuahua";
    {
        "exemplarCity";
        "تشياواوا";
    }
    "Coral_Harbour";
    {
        "exemplarCity";
        "كورانال هاربر";
    }
}
```

```
"Cordoba";
{
  "exemplarCity";
  "كوردوبا";
}
"Costa_Rica";
{
  "exemplarCity";
  "كوستاريكا";
}
"Creston";
{
  "exemplarCity";
  "كريستون";
}
"Cuiaba";
{
  "exemplarCity";
  "كيابا";
}
"Curacao";
{
  "exemplarCity";
  "كوراكاو";
}
"Danmarkshavn";
{
  "exemplarCity";
  "دانمرك شافن";
}
"Dawson";
{
  "exemplarCity";
  "داوسان";
}
"Dawson_Creek";
{
  "exemplarCity";
  "داوسن كريك";
}
"Denver";
{
  "exemplarCity";
  "دنفر";
}
"Detroit";
{
  "exemplarCity";
  "ديترويت";
}
"Dominica";
{
  "exemplarCity";
  "دومينيكا";
}
"Edmonton";
{
```

```
        "exemplarCity";
        "ايدمونتون";
    }
    "Eirunepe";
    {
        "exemplarCity";
        "ايرونبي";
    }
    "El_Salvador";
    {
        "exemplarCity";
        "السلفادور";
    }
    "Fort_Nelson";
    {
        "exemplarCity";
        "فورت نيلسون";
    }
    "Fortaleza";
    {
        "exemplarCity";
        "فورتاليزا";
    }
    "Glace_Bay";
    {
        "exemplarCity";
        "جلاس باي";
    }
    "Godthab";
    {
        "exemplarCity";
        "غودثاب";
    }
    "Goose_Bay";
    {
        "exemplarCity";
        "جوس باي";
    }
    "Grand_Turk";
    {
        "exemplarCity";
        "غراند ترك";
    }
    "Grenada";
    {
        "exemplarCity";
        "جرينادا";
    }
    "Guadeloupe";
    {
        "exemplarCity";
        "غوادلوب";
    }
    "Guatemala";
    {
        "exemplarCity";
        "غواتيمالا";
    }
}
```

```
}
"Guayaquil";
{
  "exemplarCity";
  "غواياكويل";
}
"Guyana";
{
  "exemplarCity";
  "غيانا";
}
"Halifax";
{
  "exemplarCity";
  "هاليفاكس";
}
"Havana";
{
  "exemplarCity";
  "هافانا";
}
"Hermosillo";
{
  "exemplarCity";
  "هيرموسيلو";
}
"Indiana";
{
  "Vincennes";
  {
    "exemplarCity";
    "فينسينس";
  }
  "Petersburg";
  {
    "exemplarCity";
    "بيتربيرغ";
  }
  "Tell_City";
  {
    "exemplarCity";
    "مدينة تل، إنديانا";
  }
  "Knox";
  {
    "exemplarCity";
    "كونكس";
  }
  "Winamac";
  {
    "exemplarCity";
    "ويناماك";
  }
  "Marengo";
  {
    "exemplarCity";
    "مارنجو";
  }
}
```



```
    }  
    "Vevay";  
    {  
        "exemplarCity";  
        "فيفاي";  
    }  
}  
"Indianapolis";  
{  
    "exemplarCity";  
    "إنديانا بوليس";  
}  
"Inuvik";  
{  
    "exemplarCity";  
    "اينوفيك";  
}  
"Iqaluit";  
{  
    "exemplarCity";  
    "اكويلت";  
}  
"Jamaica";  
{  
    "exemplarCity";  
    "جامايكا";  
}  
"Jujuy";  
{  
    "exemplarCity";  
    "جوجو";  
}  
"Juneau";  
{  
    "exemplarCity";  
    "جونى";  
}  
"Kentucky";  
{  
    "Monticello";  
    {  
        "exemplarCity";  
        "مونتي سيلو";  
    }  
}  
"Kralendijk";  
{  
    "exemplarCity";  
    "كرالنديك";  
}  
"La_Paz";  
{  
    "exemplarCity";  
    "لا باز";  
}  
"Lima";  
{
```

```
        "exemplarCity";
        "ليما";
    }
    "Los_Angeles";
    {
        "exemplarCity";
        "لوس انجلوس";
    }
    "Louisville";
    {
        "exemplarCity";
        "لويس فيل";
    }
    "Lower_Princes";
    {
        "exemplarCity";
        "حي الأمير السفلي";
    }
    "Maceio";
    {
        "exemplarCity";
        "ماشيو";
    }
    "Managua";
    {
        "exemplarCity";
        "ماناغوا";
    }
    "Manaus";
    {
        "exemplarCity";
        "ماناوس";
    }
    "Marigot";
    {
        "exemplarCity";
        "ماريغوت";
    }
    "Martinique";
    {
        "exemplarCity";
        "المارتينيك";
    }
    "Matamoros";
    {
        "exemplarCity";
        "ماتاموروس";
    }
    "Mazatlan";
    {
        "exemplarCity";
        "مازاتلان";
    }
    "Mendoza";
    {
        "exemplarCity";
        "ميندوزا";
    }
```

```
}
"Menominee";
{
  "exemplarCity";
  "مينوميني";
}
"Merida";
{
  "exemplarCity";
  "ميريدا";
}
"Metlakatla";
{
  "exemplarCity";
  "ميتلاكاتلا";
}
"Mexico_City";
{
  "exemplarCity";
  "مدينة المكسيك";
}
"Miquelon";
{
  "exemplarCity";
  "ميكلون";
}
"Moncton";
{
  "exemplarCity";
  "وينكتون";
}
"Monterrey";
{
  "exemplarCity";
  "مونتييري";
}
"Montevideo";
{
  "exemplarCity";
  "مونتيفيديو";
}
"Montserrat";
{
  "exemplarCity";
  "مونتسيرات";
}
"Nassau";
{
  "exemplarCity";
  "ناسو";
}
"New_York";
{
  "exemplarCity";
  "نيويورك";
}
"Nipigon";
```

```
{
  "exemplarCity";
  "نيبيجون";
}
"Nome";
{
  "exemplarCity";
  "نوم";
}
"Noronha";
{
  "exemplarCity";
  "نورونه";
}
"North_Dakota";
{
  "Beulah";
  {
    "exemplarCity";
    "بيولا، داکوتا الشمالية";
  }
  "New_Salem";
  {
    "exemplarCity";
    "نيو ساليم";
  }
  "Center";
  {
    "exemplarCity";
    "سنتر";
  }
}
"Ojinaga";
{
  "exemplarCity";
  "أوجيناغا";
}
"Panama";
{
  "exemplarCity";
  "بنما";
}
"Pangnirtung";
{
  "exemplarCity";
  "بانجينتینگ";
}
"Paramaribo";
{
  "exemplarCity";
  "باراماريبو";
}
"Phoenix";
{
  "exemplarCity";
  "فينكس";
}
}
```

```
"Port-au-Prince";
{
  "exemplarCity";
  "بورت أو برنس";
}
"Port_of_Spain";
{
  "exemplarCity";
  "بورت أوف سبين";
}
"Porto_Velho";
{
  "exemplarCity";
  "بورتو فيلو";
}
"Puerto_Rico";
{
  "exemplarCity";
  "بورتوريكو";
}
"Rainy_River";
{
  "exemplarCity";
  "راني ريفر";
}
"Rankin_Inlet";
{
  "exemplarCity";
  "رانكن انلت";
}
"Recife";
{
  "exemplarCity";
  "ريسيف";
}
"Regina";
{
  "exemplarCity";
  "ريجينا";
}
"Resolute";
{
  "exemplarCity";
  "ريزولوت";
}
"Rio_Branco";
{
  "exemplarCity";
  "ريوبرانكو";
}
"Santa_Isabel";
{
  "exemplarCity";
  "سانتا إيزابيل";
}
"Santarem";
{
```

```

        "exemplarCity";
        "سانتاريم";
    }
    "Santiago";
    {
        "exemplarCity";
        "سانتياغو";
    }
    "Santo_Domingo";
    {
        "exemplarCity";
        "سانتو دومينغو";
    }
    "Sao_Paulo";
    {
        "exemplarCity";
        "ساو باولو";
    }
    "Scoresbysund";
    {
        "exemplarCity";
        "سكورسبيسند";
    }
    "Sitka";
    {
        "exemplarCity";
        "سيتكا";
    }
    "St_Barthelemy";
    {
        "exemplarCity";
        "سانت بارتيليمي";
    }
    "St_Johns";
    {
        "exemplarCity";
        "سانت جونز";
    }
    "St_Kitts";
    {
        "exemplarCity";
        "سانت كيتس";
    }
    "St_Lucia";
    {
        "exemplarCity";
        "سانت لوشيا";
    }
    "St_Thomas";
    {
        "exemplarCity";
        "سانت توماس";
    }
    "St_Vincent";
    {
        "exemplarCity";
        "سانت فنسنت";
    }

```

```
}
"Swift_Current";
{
  "exemplarCity";
  "سوفت كارنت";
}
"Tegucigalpa";
{
  "exemplarCity";
  "تيغوسيغالبا";
}
"Thule";
{
  "exemplarCity";
  "ثيل";
}
"Thunder_Bay";
{
  "exemplarCity";
  "ثندر باي";
}
"Tijuana";
{
  "exemplarCity";
  "تيخوانا";
}
"Toronto";
{
  "exemplarCity";
  "تورونتو";
}
"Tortola";
{
  "exemplarCity";
  "تورتولا";
}
"Vancouver";
{
  "exemplarCity";
  "فانكوفر";
}
"Whitehorse";
{
  "exemplarCity";
  "وايت هورس";
}
"Winnipeg";
{
  "exemplarCity";
  "وينيبيج";
}
"Yakutat";
{
  "exemplarCity";
  "ياكوتات";
}
"Yellowknife";
```

```
{
    "exemplarCity";
    "يلونيف";
}
}
"Atlantic";
{
    "Azores";
    {
        "exemplarCity";
        "أزورس";
    }
    "Bermuda";
    {
        "exemplarCity";
        "برمودا";
    }
    "Canary";
    {
        "exemplarCity";
        "كناري";
    }
    "Cape_Verde";
    {
        "exemplarCity";
        "الرأس الأخضر";
    }
    "Faeroe";
    {
        "exemplarCity";
        "فارو";
    }
    "Madeira";
    {
        "exemplarCity";
        "ماديرا";
    }
    "Reykjavik";
    {
        "exemplarCity";
        "ريكيافيك";
    }
    "South_Georgia";
    {
        "exemplarCity";
        "جورجيا الجنوبية";
    }
    "St_Helena";
    {
        "exemplarCity";
        "سانت هيلينا";
    }
    "Stanley";
    {
        "exemplarCity";
        "استانلي";
    }
}
```



```
}
"Europe";
{
  "Amsterdam";
  {
    "exemplarCity";
    "أمستردام";
  }
  "Andorra";
  {
    "exemplarCity";
    "أندورا";
  }
  "Astrakhan";
  {
    "exemplarCity";
    "أستراخان";
  }
  "Athens";
  {
    "exemplarCity";
    "أثينا";
  }
  "Belgrade";
  {
    "exemplarCity";
    "بلغراد";
  }
  "Berlin";
  {
    "exemplarCity";
    "برلين";
  }
  "Bratislava";
  {
    "exemplarCity";
    "براتيسلافا";
  }
  "Brussels";
  {
    "exemplarCity";
    "بروكسل";
  }
  "Bucharest";
  {
    "exemplarCity";
    "بوخارست";
  }
  "Budapest";
  {
    "exemplarCity";
    "بودابست";
  }
  "Busingen";
  {
    "exemplarCity";
    "بوسنغن";
  }
}
```

```
}
"Chisinau";
{
  "exemplarCity";
  "تشيسيناو";
}
"Copenhagen";
{
  "exemplarCity";
  "كوبنهاغن";
}
"Dublin";
{
  "long";
  {
    "daylight";
    "توقيت أيرلندا الرسمي";
  }
  "exemplarCity";
  "دبلن";
}
"Gibraltar";
{
  "exemplarCity";
  "جبل طارق";
}
"Guernsey";
{
  "exemplarCity";
  "غيرنسي";
}
"Helsinki";
{
  "exemplarCity";
  "هلسنكي";
}
"Isle_of_Man";
{
  "exemplarCity";
  "جزيرة مان";
}
"Istanbul";
{
  "exemplarCity";
  "إسطنبول";
}
"Jersey";
{
  "exemplarCity";
  "جيرسي";
}
"Kaliningrad";
{
  "exemplarCity";
  "كالينجراد";
}
"Kiev";
```

```
{
  "exemplarCity";
  "كييف";
}
"Kirov";
{
  "exemplarCity";
  "كيروف";
}
"Lisbon";
{
  "exemplarCity";
  "لشبونة";
}
"Ljubljana";
{
  "exemplarCity";
  "ليوبليانا";
}
"London";
{
  "long";
  {
    "daylight";
    "توقيت بريطانيا الصيفي";
  }
  "exemplarCity";
  "لندن";
}
"Luxembourg";
{
  "exemplarCity";
  "لوكسمبورغ";
}
"Madrid";
{
  "exemplarCity";
  "مدريد";
}
"Malta";
{
  "exemplarCity";
  "مالطة";
}
"Mariehamn";
{
  "exemplarCity";
  "ماريهامن";
}
"Minsk";
{
  "exemplarCity";
  "مينسك";
}
"Monaco";
{
  "exemplarCity";
```

```
        "موناكو";
    }
    "Moscow";
    {
        "exemplarCity";
        "موسكو";
    }
    "Oslo";
    {
        "exemplarCity";
        "أوسلو";
    }
    "Paris";
    {
        "exemplarCity";
        "باريس";
    }
    "Podgorica";
    {
        "exemplarCity";
        "بودغوريكا";
    }
    "Prague";
    {
        "exemplarCity";
        "براغ";
    }
    "Riga";
    {
        "exemplarCity";
        "ريغا";
    }
    "Rome";
    {
        "exemplarCity";
        "روما";
    }
    "Samara";
    {
        "exemplarCity";
        "سمراء";
    }
    "San_Marino";
    {
        "exemplarCity";
        "سان مارينو";
    }
    "Sarajevo";
    {
        "exemplarCity";
        "سراييفو";
    }
    "Simferopol";
    {
        "exemplarCity";
        "سيمفروبول";
    }
}
```

```
"Skopje";
{
  "exemplarCity";
  "سكوبي";
}
"Sofia";
{
  "exemplarCity";
  "صوفيا";
}
"Stockholm";
{
  "exemplarCity";
  "ستوكهولم";
}
"Tallinn";
{
  "exemplarCity";
  "تالين";
}
"Tirane";
{
  "exemplarCity";
  "تيرانا";
}
"Ulyanovsk";
{
  "exemplarCity";
  "أوليانوفسك";
}
"Uzhgorod";
{
  "exemplarCity";
  "أوزجروود";
}
"Vaduz";
{
  "exemplarCity";
  "فادوز";
}
"Vatican";
{
  "exemplarCity";
  "الفاتيكان";
}
"Vienna";
{
  "exemplarCity";
  "فيينا";
}
"Vilnius";
{
  "exemplarCity";
  "فيلنيوس";
}
"Volgograd";
{
```

```
        "exemplarCity";
        "فولوجراد";
    }
    "Warsaw";
    {
        "exemplarCity";
        "وارسو";
    }
    "Zagreb";
    {
        "exemplarCity";
        "زغرب";
    }
    "Zaporozhye";
    {
        "exemplarCity";
        "زابوروزي";
    }
    "Zurich";
    {
        "exemplarCity";
        "زيورخ";
    }
}
"Africa";
{
    "Abidjan";
    {
        "exemplarCity";
        "أبيدجان";
    }
    "Accra";
    {
        "exemplarCity";
        "أكرا";
    }
    "Addis_Ababa";
    {
        "exemplarCity";
        "أديس أبابا";
    }
    "Algiers";
    {
        "exemplarCity";
        "الجزائر";
    }
    "Asmera";
    {
        "exemplarCity";
        "أسمره";
    }
    "Bamako";
    {
        "exemplarCity";
        "باماكو";
    }
    "Bangui";
```

```
{
  "exemplarCity";
  "بانغوي";
}
"Banjul";
{
  "exemplarCity";
  "بانجول";
}
"Bissau";
{
  "exemplarCity";
  "بيساو";
}
"Blantyre";
{
  "exemplarCity";
  "بلانتيير";
}
"Brazzaville";
{
  "exemplarCity";
  "برازافيل";
}
"Bujumbura";
{
  "exemplarCity";
  "بوجومبورا";
}
"Cairo";
{
  "exemplarCity";
  "القاهرة";
}
"Casablanca";
{
  "exemplarCity";
  "الدار البيضاء";
}
"Ceuta";
{
  "exemplarCity";
  "سيتا";
}
"Conakry";
{
  "exemplarCity";
  "كوناكري";
}
"Dakar";
{
  "exemplarCity";
  "داكار";
}
"Dar_es_Salaam";
{
  "exemplarCity";
```

```
        "دار السلام";
    }
    "Djibouti";
    {
        "exemplarCity";
        "جيبوتي";
    }
    "Douala";
    {
        "exemplarCity";
        "دوالا";
    }
    "El_Aaiun";
    {
        "exemplarCity";
        "العيون";
    }
    "Freetown";
    {
        "exemplarCity";
        "فري تاون";
    }
    "Gaborone";
    {
        "exemplarCity";
        "غابورون";
    }
    "Harare";
    {
        "exemplarCity";
        "هراري";
    }
    "Johannesburg";
    {
        "exemplarCity";
        "جوهانسبرغ";
    }
    "Juba";
    {
        "exemplarCity";
        "جوبا";
    }
    "Kampala";
    {
        "exemplarCity";
        "كامبالا";
    }
    "Khartoum";
    {
        "exemplarCity";
        "الخرطوم";
    }
    "Kigali";
    {
        "exemplarCity";
        "كيغالي";
    }
}
```



```
"Kinshasa";
{
  "exemplarCity";
  "كينشاسا";
}
"Lagos";
{
  "exemplarCity";
  "لاغوس";
}
"Libreville";
{
  "exemplarCity";
  "ليبرفيل";
}
"Lome";
{
  "exemplarCity";
  "لومي";
}
"Luanda";
{
  "exemplarCity";
  "لواندا";
}
"Lubumbashi";
{
  "exemplarCity";
  "لومبباشا";
}
"Lusaka";
{
  "exemplarCity";
  "لوساكا";
}
"Malabo";
{
  "exemplarCity";
  "مالابو";
}
"Maputo";
{
  "exemplarCity";
  "مابوتو";
}
"Maseru";
{
  "exemplarCity";
  "ماسيرو";
}
"Mbabane";
{
  "exemplarCity";
  "مباباني";
}
"Mogadishu";
{
```

```
        "exemplarCity";
        "مقديشو";
    }
    "Monrovia";
    {
        "exemplarCity";
        "مونروفيا";
    }
    "Nairobi";
    {
        "exemplarCity";
        "نيروبي";
    }
    "Ndjamena";
    {
        "exemplarCity";
        "نجامينا";
    }
    "Niamey";
    {
        "exemplarCity";
        "نيامي";
    }
    "Nouakchott";
    {
        "exemplarCity";
        "نواكشوط";
    }
    "Ouagadougou";
    {
        "exemplarCity";
        "واغادوغو";
    }
    "Porto-Novo";
    {
        "exemplarCity";
        "بورتو نوفو";
    }
    "Sao_Tome";
    {
        "exemplarCity";
        "ساو تومي";
    }
    "Tripoli";
    {
        "exemplarCity";
        "طرابلس";
    }
    "Tunis";
    {
        "exemplarCity";
        "تونس";
    }
    "Windhoek";
    {
        "exemplarCity";
        "ويندهوك";
    }
```

```
    }  
  }  
  "Asia";  
  {  
    "Aden";  
    {  
      "exemplarCity";  
      "عدن";  
    }  
    "Almaty";  
    {  
      "exemplarCity";  
      "ألماتي";  
    }  
    "Amman";  
    {  
      "exemplarCity";  
      "عمان";  
    }  
    "Anadyr";  
    {  
      "exemplarCity";  
      "أندير";  
    }  
    "Aqtau";  
    {  
      "exemplarCity";  
      "أكتاو";  
    }  
    "Aqtobe";  
    {  
      "exemplarCity";  
      "أکتوب";  
    }  
    "Ashgabat";  
    {  
      "exemplarCity";  
      "عشق آباد";  
    }  
    "Baghdad";  
    {  
      "exemplarCity";  
      "بغداد";  
    }  
    "Bahrain";  
    {  
      "exemplarCity";  
      "البحرين";  
    }  
    "Baku";  
    {  
      "exemplarCity";  
      "باکو";  
    }  
    "Bangkok";  
    {  
      "exemplarCity";  
    }  
  }
```

```
        "بانكوك";
    }
    "Barnaul";
    {
        "exemplarCity";
        "بارناول";
    }
    "Beirut";
    {
        "exemplarCity";
        "بيروت";
    }
    "Bishkek";
    {
        "exemplarCity";
        "بشكيك";
    }
    "Brunei";
    {
        "exemplarCity";
        "بروناي";
    }
    "Calcutta";
    {
        "exemplarCity";
        "كالكتا";
    }
    "Chita";
    {
        "exemplarCity";
        "تشيتا";
    }
    "Choibalsan";
    {
        "exemplarCity";
        "تشوبالسان";
    }
    "Colombo";
    {
        "exemplarCity";
        "كولومبو";
    }
    "Damascus";
    {
        "exemplarCity";
        "دمشق";
    }
    "Dhaka";
    {
        "exemplarCity";
        "دكا";
    }
    "Dili";
    {
        "exemplarCity";
        "ديلي";
    }
}
```

```
"Dubai";
{
  "exemplarCity";
  "دبي";
}
"Dushanbe";
{
  "exemplarCity";
  "دوشانبی";
}
"Gaza";
{
  "exemplarCity";
  "غزة";
}
"Hebron";
{
  "exemplarCity";
  "هيبرون (مدينة الخليل)";
}
"Hong_Kong";
{
  "exemplarCity";
  "هونغ كونغ";
}
"Hovd";
{
  "exemplarCity";
  "هوفد";
}
"Irkutsk";
{
  "exemplarCity";
  "ايركيتسك";
}
"Jakarta";
{
  "exemplarCity";
  "جاكرتا";
}
"Jayapura";
{
  "exemplarCity";
  "جاياپورا";
}
"Jerusalem";
{
  "exemplarCity";
  "القدس";
}
"Kabul";
{
  "exemplarCity";
  "كابول";
}
"Kamchatka";
{
```

```
        "exemplarCity";
        "کامٹشاتکا";
    }
    "Karachi";
    {
        "exemplarCity";
        "کراتشي";
    }
    "Katmandu";
    {
        "exemplarCity";
        "کاتماندو";
    }
    "Khandyga";
    {
        "exemplarCity";
        "خاندیجا";
    }
    "Krasnoyarsk";
    {
        "exemplarCity";
        "کراسنویارسک";
    }
    "Kuala_Lumpur";
    {
        "exemplarCity";
        "کوالا لامپور";
    }
    "Kuching";
    {
        "exemplarCity";
        "کیشینج";
    }
    "Kuwait";
    {
        "exemplarCity";
        "الکویت";
    }
    "Macau";
    {
        "exemplarCity";
        "ماکاو";
    }
    "Magadan";
    {
        "exemplarCity";
        "مجادن";
    }
    "Makassar";
    {
        "exemplarCity";
        "ماکسار";
    }
    "Manila";
    {
        "exemplarCity";
        "مانیلا";
    }
```

```
}
"Muscat";
{
  "exemplarCity";
  "مسقط";
}
"Nicosia";
{
  "exemplarCity";
  "نيقوسيا";
}
"Novokuznetsk";
{
  "exemplarCity";
  "نوفوكوزنتسك";
}
"Novosibirsk";
{
  "exemplarCity";
  "نوفوسبيرسك";
}
"Omsk";
{
  "exemplarCity";
  "أومسك";
}
"Oral";
{
  "exemplarCity";
  "أورال";
}
"Phnom_Penh";
{
  "exemplarCity";
  "بنوم بنه";
}
"Pontianak";
{
  "exemplarCity";
  "بونتيانك";
}
"Pyongyang";
{
  "exemplarCity";
  "بيونغ يانغ";
}
"Qatar";
{
  "exemplarCity";
  "قطر";
}
"Qyzylorda";
{
  "exemplarCity";
  "كيزيلوردا";
}
"Rangoon";
```

```
{
  "exemplarCity";
  "رانغون";
}
"Riyadh";
{
  "exemplarCity";
  "الرياض";
}
"Saigon";
{
  "exemplarCity";
  "مدينة هو تشي منه";
}
"Sakhalin";
{
  "exemplarCity";
  "سكالين";
}
"Samarkand";
{
  "exemplarCity";
  "سمرقند";
}
"Seoul";
{
  "exemplarCity";
  "سول";
}
"Shanghai";
{
  "exemplarCity";
  "شنغهاي";
}
"Singapore";
{
  "exemplarCity";
  "سنغافورة";
}
"Srednekolymsk";
{
  "exemplarCity";
  "سريدنكوليمسك";
}
"Taipei";
{
  "exemplarCity";
  "تايبيه";
}
"Tashkent";
{
  "exemplarCity";
  "طشقند";
}
"Tbilisi";
{
  "exemplarCity";
```



```
        "تبلیسی";
    }
    "Tehran";
    {
        "exemplarCity";
        "تهران";
    }
    "Thimphu";
    {
        "exemplarCity";
        "تیمفو";
    }
    "Tokyo";
    {
        "exemplarCity";
        "طوکیو";
    }
    "Tomsk";
    {
        "exemplarCity";
        "تومسک";
    }
    "Ulaanbaatar";
    {
        "exemplarCity";
        "آلانباتار";
    }
    "Urumqi";
    {
        "exemplarCity";
        "أرومکی";
    }
    "Ust-Nera";
    {
        "exemplarCity";
        "أوست نیرا";
    }
    "Vientiane";
    {
        "exemplarCity";
        "فیانتیان";
    }
    "Vladivostok";
    {
        "exemplarCity";
        "فلادیفوستک";
    }
    "Yakutsk";
    {
        "exemplarCity";
        "یاکتسک";
    }
    "Yekaterinburg";
    {
        "exemplarCity";
        "یکاترنبرج";
    }
}
```

```
        "Yerevan";
        {
            "exemplarCity";
            "یريفان";
        }
    }
    "Indian";
    {
        "Antananarivo";
        {
            "exemplarCity";
            "أنتاناناريفو";
        }
        "Chagos";
        {
            "exemplarCity";
            "تشاغوس";
        }
        "Christmas";
        {
            "exemplarCity";
            "كريسماس";
        }
        "Cocos";
        {
            "exemplarCity";
            "كوكوس";
        }
        "Comoro";
        {
            "exemplarCity";
            "جزر القمر";
        }
        "Kerguelen";
        {
            "exemplarCity";
            "كيرغويلين";
        }
        "Mahe";
        {
            "exemplarCity";
            "ماهي";
        }
        "Maldives";
        {
            "exemplarCity";
            "المالديف";
        }
        "Mauritius";
        {
            "exemplarCity";
            "موريشيوس";
        }
        "Mayotte";
        {
            "exemplarCity";
            "مايوت";
        }
    }
```

```
}
  "Reunion";
  {
    "exemplarCity";
    "ريونيون";
  }
}
"Australia";
{
  "Adelaide";
  {
    "exemplarCity";
    "أديليد";
  }
  "Brisbane";
  {
    "exemplarCity";
    "برسبان";
  }
  "Broken_Hill";
  {
    "exemplarCity";
    "بروكن هيل";
  }
  "Currie";
  {
    "exemplarCity";
    "كوري";
  }
  "Darwin";
  {
    "exemplarCity";
    "دارون";
  }
  "Eucla";
  {
    "exemplarCity";
    "أوكل";
  }
  "Hobart";
  {
    "exemplarCity";
    "هوبارت";
  }
  "Lindeman";
  {
    "exemplarCity";
    "ليندمان";
  }
  "Lord_Howe";
  {
    "exemplarCity";
    "لورد هاو";
  }
  "Melbourne";
  {
    "exemplarCity";
```

```
        "ميلبورن";
    }
    "Perth";
    {
        "exemplarCity";
        "برثا";
    }
    "Sydney";
    {
        "exemplarCity";
        "سيدني";
    }
}
"Pacific";
{
    "Apia";
    {
        "exemplarCity";
        "أبيا";
    }
    "Auckland";
    {
        "exemplarCity";
        "أوكلاند";
    }
    "Bougainville";
    {
        "exemplarCity";
        "بوغانفيل";
    }
    "Chatham";
    {
        "exemplarCity";
        "تشاثام";
    }
    "Easter";
    {
        "exemplarCity";
        "استر";
    }
    "Efate";
    {
        "exemplarCity";
        "إيفات";
    }
    "Enderbury";
    {
        "exemplarCity";
        "اندربيرج";
    }
    "Fakaofu";
    {
        "exemplarCity";
        "فاكاوفو";
    }
    "Fiji";
    {
```

```
        "exemplarCity";
        "فيجي";
    }
    "Funafuti";
    {
        "exemplarCity";
        "فونافوتي";
    }
    "Galapagos";
    {
        "exemplarCity";
        "جلاپاجوس";
    }
    "Gambier";
    {
        "exemplarCity";
        "جامبير";
    }
    "Guadalcanal";
    {
        "exemplarCity";
        "غواد الكانال";
    }
    "Guam";
    {
        "exemplarCity";
        "غوام";
    }
    "Honolulu";
    {
        "exemplarCity";
        "هونولولو";
    }
    "Johnston";
    {
        "exemplarCity";
        "جونستون";
    }
    "Kiritimati";
    {
        "exemplarCity";
        "كيريتي ماتي";
    }
    "Kosrae";
    {
        "exemplarCity";
        "كوسرا";
    }
    "Kwajalein";
    {
        "exemplarCity";
        "كواجالين";
    }
    "Majuro";
    {
        "exemplarCity";
        "ماجورو";
    }
```

```
}
"Marquesas";
{
  "exemplarCity";
  "ماركيساس";
}
"Midway";
{
  "exemplarCity";
  "ميدواي";
}
"Nauru";
{
  "exemplarCity";
  "ناورو";
}
"Niue";
{
  "exemplarCity";
  "نيوي";
}
"Norfolk";
{
  "exemplarCity";
  "نورفولك";
}
"Noumea";
{
  "exemplarCity";
  "نوميا";
}
"Pago_Pago";
{
  "exemplarCity";
  "باغو باغو";
}
"Palau";
{
  "exemplarCity";
  "بالاو";
}
"Pitcairn";
{
  "exemplarCity";
  "بيتكيرن";
}
"Ponape";
{
  "exemplarCity";
  "باناب";
}
"Port_Moresby";
{
  "exemplarCity";
  "بور مورسبي";
}
"Rarotonga";
```

```
{
  "exemplarCity";
  "راروتونغا";
}
"Saipan";
{
  "exemplarCity";
  "سايبان";
}
"Tahiti";
{
  "exemplarCity";
  "تاهيتي";
}
"Tarawa";
{
  "exemplarCity";
  "تاراوا";
}
"Tongatapu";
{
  "exemplarCity";
  "تونغاتابو";
}
"Truk";
{
  "exemplarCity";
  "ترك";
}
"Wake";
{
  "exemplarCity";
  "واك";
}
"Wallis";
{
  "exemplarCity";
  "واليس";
}
}
"Arctic";
{
  "Longyearbyen";
  {
    "exemplarCity";
    "لونغجيربين";
  }
}
"Antarctica";
{
  "Casey";
  {
    "exemplarCity";
    "كاساي";
  }
}
"Davis";
{
```

```
        "exemplarCity";
        "دافيز";
    }
    "DumontDUrville";
    {
        "exemplarCity";
        "دي مونت دو روفيل";
    }
    "Macquarie";
    {
        "exemplarCity";
        "ماكوارى";
    }
    "Mawson";
    {
        "exemplarCity";
        "ماوسون";
    }
    "McMurdo";
    {
        "exemplarCity";
        "ماك موردو";
    }
    "Palmer";
    {
        "exemplarCity";
        "بالمير";
    }
    "Rothera";
    {
        "exemplarCity";
        "روثيرا";
    }
    "Syowa";
    {
        "exemplarCity";
        "سايووا";
    }
    "Troll";
    {
        "exemplarCity";
        "ترول";
    }
    "Vostok";
    {
        "exemplarCity";
        "فوستوك";
    }
}
"Etc";
{
    "GMT";
    {
        "exemplarCity";
        "GMT";
    }
    "GMT1";
}
```



```
{
    "exemplarCity";
    "GMT+1";
}
"GMT10";
{
    "exemplarCity";
    "GMT+10";
}
"GMT11";
{
    "exemplarCity";
    "GMT+11";
}
"GMT12";
{
    "exemplarCity";
    "GMT+12";
}
"GMT2";
{
    "exemplarCity";
    "GMT+2";
}
"GMT3";
{
    "exemplarCity";
    "GMT+3";
}
"GMT4";
{
    "exemplarCity";
    "GMT+4";
}
"GMT5";
{
    "exemplarCity";
    "GMT+5";
}
"GMT6";
{
    "exemplarCity";
    "GMT+6";
}
"GMT7";
{
    "exemplarCity";
    "GMT+7";
}
"GMT8";
{
    "exemplarCity";
    "GMT+8";
}
"GMT9";
{
    "exemplarCity";
```

```
        "GMT+9";
    }
    "GMT-1";
    {
        "exemplarCity";
        "GMT-1";
    }
    "GMT-10";
    {
        "exemplarCity";
        "GMT-10";
    }
    "GMT-11";
    {
        "exemplarCity";
        "GMT-11";
    }
    "GMT-12";
    {
        "exemplarCity";
        "GMT-12";
    }
    "GMT-13";
    {
        "exemplarCity";
        "GMT-13";
    }
    "GMT-14";
    {
        "exemplarCity";
        "GMT-14";
    }
    "GMT-2";
    {
        "exemplarCity";
        "GMT-2";
    }
    "GMT-3";
    {
        "exemplarCity";
        "GMT-3";
    }
    "GMT-4";
    {
        "exemplarCity";
        "GMT-4";
    }
    "GMT-5";
    {
        "exemplarCity";
        "GMT-5";
    }
    "GMT-6";
    {
        "exemplarCity";
        "GMT-6";
    }
}
```

```
        "GMT-7";
        {
            "exemplarCity";
            "GMT-7";
        }
        "GMT-8";
        {
            "exemplarCity";
            "GMT-8";
        }
        "GMT-9";
        {
            "exemplarCity";
            "GMT-9";
        }
        "Unknown";
        {
            "exemplarCity";
            "مدينة غير معروفة";
        }
    }
}
"metazone";
{
    "Afghanistan";
    {
        "long";
        {
            "standard";
            "توقيت أفغانستان";
        }
    }
    "Africa_Central";
    {
        "long";
        {
            "standard";
            "توقيت وسط أفريقيا";
        }
    }
    "Africa_Eastern";
    {
        "long";
        {
            "standard";
            "توقيت شرق أفريقيا";
        }
    }
    "Africa_Southern";
    {
        "long";
        {
            "standard";
            "توقيت جنوب أفريقيا";
        }
    }
    "Africa_Western";
```

```
{
  "long";
  {
    "generic";
    "توقيت غرب أفريقيا",
    "standard";
    "توقيت غرب أفريقيا الرسمي",
    "daylight";
    "توقيت غرب أفريقيا الصيفي";
  }
}
"Alaska";
{
  "long";
  {
    "generic";
    "توقيت ألاسكا",
    "standard";
    "التوقيت الرسمي لألاسكا",
    "daylight";
    "توقيت ألاسكا الصيفي";
  }
}
"Amazon";
{
  "long";
  {
    "generic";
    "توقيت الأمازون",
    "standard";
    "توقيت الأمازون الرسمي",
    "daylight";
    "توقيت الأمازون الصيفي";
  }
}
"America_Central";
{
  "long";
  {
    "generic";
    "التوقيت المركزي لأمريكا الشمالية",
    "standard";
    "التوقيت الرسمي المركزي لأمريكا الشمالية",
    "daylight";
    "التوقيت الصيفي المركزي لأمريكا الشمالية";
  }
}
"America_Eastern";
{
  "long";
  {
    "generic";
    "التوقيت الشرقي لأمريكا الشمالية",
    "standard";
    "التوقيت الرسمي الشرقي لأمريكا الشمالية",
    "daylight";
    "التوقيت الصيفي الشرقي لأمريكا الشمالية";
  }
}
```

```

    }
  }
  "America_Mountain";
  {
    "long";
    {
      "generic";
      "التوقيت الجبلي لأمريكا الشمالية",
      "standard";
      "التوقيت الجبلي الرسمي لأمريكا الشمالية",
      "daylight";
      "التوقيت الجبلي الصيفي لأمريكا الشمالية";
    }
  }
  "America_Pacific";
  {
    "long";
    {
      "generic";
      "توقيت المحيط الهادي",
      "standard";
      "توقيت المحيط الهادي الرسمي",
      "daylight";
      "توقيت المحيط الهادي الصيفي";
    }
  }
  "Anadyr";
  {
    "long";
    {
      "generic";
      "توقيت أنادير",
      "standard";
      "توقيت أنادير الرسمي",
      "daylight";
      "التوقيت الصيفي لأنادير";
    }
  }
  "Apia";
  {
    "long";
    {
      "generic";
      "توقيت آبيا",
      "standard";
      "التوقيت الرسمي لآبيا",
      "daylight";
      "التوقيت الصيفي لآبيا";
    }
  }
  "Arabian";
  {
    "long";
    {
      "generic";
      "التوقيت العربي",
      "standard";

```

```

        "التوقيت العربي الرسمي",
        "daylight";
        "التوقيت العربي الصيفي";
    }
}
"Argentina";
{
    "long";
    {
        "generic";
        "توقيت الأرجنتين",
        "standard";
        "توقيت الأرجنتين الرسمي",
        "daylight";
        "توقيت الأرجنتين الصيفي";
    }
}
"Argentina_Western";
{
    "long";
    {
        "generic";
        "توقيت غرب الأرجنتين",
        "standard";
        "توقيت غرب الأرجنتين الرسمي",
        "daylight";
        "توقيت غرب الأرجنتين الصيفي";
    }
}
"Armenia";
{
    "long";
    {
        "generic";
        "توقيت أرمينيا",
        "standard";
        "توقيت أرمينيا الرسمي",
        "daylight";
        "توقيت أرمينيا الصيفي";
    }
}
"Atlantic";
{
    "long";
    {
        "generic";
        "توقيت الأطلسي",
        "standard";
        "التوقيت الرسمي الأطلسي",
        "daylight";
        "التوقيت الصيفي الأطلسي";
    }
}
"Australia_Central";
{
    "long";
    {

```

```

        "generic";
        "توقيت وسط أستراليا",
        "standard";
        "توقيت وسط أستراليا الرسمي",
        "daylight";
        "توقيت وسط أستراليا الصيفي";
    }
}
"Australia_CentralWestern";
{
    "long";
    {
        "generic";
        "توقيت غرب وسط أستراليا",
        "standard";
        "توقيت غرب وسط أستراليا الرسمي",
        "daylight";
        "توقيت غرب وسط أستراليا الصيفي";
    }
}
"Australia_Eastern";
{
    "long";
    {
        "generic";
        "توقيت شرق أستراليا",
        "standard";
        "توقيت شرق أستراليا الرسمي",
        "daylight";
        "توقيت شرق أستراليا الصيفي";
    }
}
"Australia_Western";
{
    "long";
    {
        "generic";
        "توقيت غرب أستراليا",
        "standard";
        "توقيت غرب أستراليا الرسمي",
        "daylight";
        "توقيت غرب أستراليا الصيفي";
    }
}
"Azerbaijan";
{
    "long";
    {
        "generic";
        "توقيت أذربيجان",
        "standard";
        "توقيت أذربيجان الرسمي",
        "daylight";
        "توقيت أذربيجان الصيفي";
    }
}
"Azores";

```

```
{
  "long":
  {
    "generic":
      "توقيت أزورس",
    "standard":
      "توقيت أزورس الرسمي",
    "daylight":
      "توقيت أزورس الصيفي";
  }
}
"Bangladesh";
{
  "long":
  {
    "generic":
      "توقيت بنجلاديش",
    "standard":
      "توقيت بنجلاديش الرسمي",
    "daylight":
      "توقيت بنجلاديش الصيفي";
  }
}
"Bhutan";
{
  "long":
  {
    "standard":
      "توقيت بوتان";
  }
}
"Bolivia";
{
  "long":
  {
    "standard":
      "توقيت بوليفيا";
  }
}
"Brasilia";
{
  "long":
  {
    "generic":
      "توقيت برازيليا",
    "standard":
      "توقيت برازيليا الرسمي",
    "daylight":
      "توقيت برازيليا الصيفي";
  }
}
"Brunei";
{
  "long":
  {
    "standard":
      "توقيت بروناي";
  }
}
```



```
    }  
  }  
  "Cape_Verde";  
  {  
    "long";  
    {  
      "generic";  
      "توقيت الرأس الأخضر",  
      "standard";  
      "توقيت الرأس الأخضر الرسمي",  
      "daylight";  
      "توقيت الرأس الأخضر الصيفي";  
    }  
  }  
  "Chamorro";  
  {  
    "long";  
    {  
      "standard";  
      "توقيت تشامورو";  
    }  
  }  
  "Chatham";  
  {  
    "long";  
    {  
      "generic";  
      "توقيت تشاتام",  
      "standard";  
      "توقيت تشاتام الرسمي",  
      "daylight";  
      "توقيت تشاتام الصيفي";  
    }  
  }  
  "Chile";  
  {  
    "long";  
    {  
      "generic";  
      "توقيت شيلي",  
      "standard";  
      "توقيت شيلي الرسمي",  
      "daylight";  
      "توقيت شيلي الصيفي";  
    }  
  }  
  "China";  
  {  
    "long";  
    {  
      "generic";  
      "توقيت الصين",  
      "standard";  
      "توقيت الصين الرسمي",  
      "daylight";  
      "توقيت الصين الصيفي";  
    }  
  }
```

```
}
"Choibalsan";
{
  "long";
  {
    "generic";
    "توقيت شويبالسان",
    "standard";
    "توقيت شويبالسان الرسمي",
    "daylight";
    "التوقيت الصيفي لشويبالسان";
  }
}
"Christmas";
{
  "long";
  {
    "standard";
    "توقيت جزر الكريسماس";
  }
}
"Cocos";
{
  "long";
  {
    "standard";
    "توقيت جزر كوكوس";
  }
}
"Colombia";
{
  "long";
  {
    "generic";
    "توقيت كولومبيا",
    "standard";
    "توقيت كولومبيا الرسمي",
    "daylight";
    "توقيت كولومبيا الصيفي";
  }
}
"Cook";
{
  "long";
  {
    "generic";
    "توقيت جزر كوك",
    "standard";
    "توقيت جزر كوك الرسمي",
    "daylight";
    "توقيت جزر كوك الصيفي";
  }
}
"Cuba";
{
  "long";
  {
```

```

        "generic";
        "توقيت كوبا",
        "standard";
        "توقيت كوبا الرسمي",
        "daylight";
        "توقيت كوبا الصيفي";
    }
}
"Davis";
{
    "long";
    {
        "standard";
        "توقيت دافيز";
    }
}
"DumontDUrville";
{
    "long";
    {
        "standard";
        "توقيت دي مونت دو روفيل";
    }
}
"East_Timor";
{
    "long";
    {
        "standard";
        "توقيت تيمور الشرقية";
    }
}
"Easter";
{
    "long";
    {
        "generic";
        "توقيت جزيرة استر",
        "standard";
        "توقيت جزيرة استر الرسمي",
        "daylight";
        "توقيت جزيرة استر الصيفي";
    }
}
"Ecuador";
{
    "long";
    {
        "standard";
        "توقيت الإكوادور";
    }
}
"Europe_Central";
{
    "long";
    {
        "generic";

```

```

        "توقيت وسط أوروبا",
        "standard";
        "توقيت وسط أوروبا الرسمي",
        "daylight";
        "توقيت وسط أوروبا الصيفي";
    }
}
"Europe_Eastern";
{
    "long";
    {
        "generic";
        "توقيت شرق أوروبا",
        "standard";
        "توقيت شرق أوروبا الرسمي",
        "daylight";
        "توقيت شرق أوروبا الصيفي";
    }
}
"Europe_Further_Eastern";
{
    "long";
    {
        "standard";
        "التوقيت الأوروبي (أكثر شرقًا)";
    }
}
"Europe_Western";
{
    "long";
    {
        "generic";
        "توقيت غرب أوروبا",
        "standard";
        "توقيت غرب أوروبا الرسمي",
        "daylight";
        "توقيت غرب أوروبا الصيفي";
    }
}
"Falkland";
{
    "long";
    {
        "generic";
        "توقيت جزر فوكلاند",
        "standard";
        "توقيت جزر فوكلاند الرسمي",
        "daylight";
        "توقيت جزر فوكلاند الصيفي";
    }
}
"Fiji";
{
    "long";
    {
        "generic";
        "توقيت فيجي",

```

```

        "standard";
        "توقيت فيجي الرسمي",
        "daylight";
        "توقيت فيجي الصيفي";
    }
}
"French_Guiana";
{
    "long";
    {
        "standard";
        "توقيت غايانا الفرنسية";
    }
}
"French_Southern";
{
    "long";
    {
        "standard";
        "توقيت المقاطعات الفرنسية الجنوبية";
    }
}
"Galapagos";
{
    "long";
    {
        "standard";
        "توقيت غلاباغوس";
    }
}
"Gambier";
{
    "long";
    {
        "standard";
        "توقيت جامبير";
    }
}
"Georgia";
{
    "long";
    {
        "generic";
        "توقيت جورجيا",
        "standard";
        "توقيت جورجيا الرسمي",
        "daylight";
        "توقيت جورجيا الصيفي";
    }
}
"Gilbert_Islands";
{
    "long";
    {
        "standard";
        "توقيت جزر جيلبرت";
    }
}

```

```

    }
  }
  "GMT";
  {
    "long";
    {
      "standard";
      "توقيت غرينتش";
    }
  }
  "Greenland_Eastern";
  {
    "long";
    {
      "generic";
      "توقيت شرق غرينلاند",
      "standard";
      "توقيت شرق غرينلاند الرسمي",
      "daylight";
      "توقيت شرق غرينلاند الصيفي";
    }
  }
  "Greenland_Western";
  {
    "long";
    {
      "generic";
      "توقيت غرب غرينلاند",
      "standard";
      "توقيت غرب غرينلاند الرسمي",
      "daylight";
      "توقيت غرب غرينلاند الصيفي";
    }
  }
  "Guam";
  {
    "long";
    {
      "standard";
      "توقيت غوام";
    }
  }
  "Gulf";
  {
    "long";
    {
      "standard";
      "توقيت الخليج";
    }
  }
  "Guyana";
  {
    "long";
    {
      "standard";
      "توقيت غيانا";
    }
  }

```

```
}
"Hawaii_Aleutian";
{
  "long";
  {
    "generic";
    "توقيت هاواي ألوتيان",
    "standard";
    "توقيت هاواي ألوتيان الرسمي",
    "daylight";
    "توقيت هاواي ألوتيان الصيفي";
  }
}
"Hong_Kong";
{
  "long";
  {
    "generic";
    "توقيت هونغ كونغ",
    "standard";
    "توقيت هونغ كونغ الرسمي",
    "daylight";
    "توقيت هونغ كونغ الصيفي";
  }
}
"Hovd";
{
  "long";
  {
    "generic";
    "توقيت هوفد",
    "standard";
    "توقيت هوفد الرسمي",
    "daylight";
    "توقيت هوفد الصيفي";
  }
}
"India";
{
  "long";
  {
    "standard";
    "توقيت الهند";
  }
}
"Indian_Ocean";
{
  "long";
  {
    "standard";
    "توقيت المحيط الهندي";
  }
}
"Indochina";
{
  "long";
  {
```

```

        "standard";
        "توقيت الهند الصينية";
    }
}
"Indonesia_Central";
{
    "long";
    {
        "standard";
        "توقيت وسط إندونيسيا";
    }
}
"Indonesia_Eastern";
{
    "long";
    {
        "standard";
        "توقيت شرق إندونيسيا";
    }
}
"Indonesia_Western";
{
    "long";
    {
        "standard";
        "توقيت غرب إندونيسيا";
    }
}
"Iran";
{
    "long";
    {
        "generic";
        "توقيت إيران",
        "standard";
        "توقيت إيران الرسمي",
        "daylight";
        "توقيت إيران الصيفي";
    }
}
"Irkutsk";
{
    "long";
    {
        "generic";
        "توقيت إركوتسك",
        "standard";
        "توقيت إركوتسك الرسمي",
        "daylight";
        "توقيت إركوتسك الصيفي";
    }
}
"Israel";
{
    "long";
    {
        "generic";

```



```

        "توقيت إسرائيل",
        "standard";
        "توقيت إسرائيل الرسمي",
        "daylight";
        "توقيت إسرائيل الصيفي";
    }
}
"Japan";
{
    "long";
    {
        "generic";
        "توقيت اليابان",
        "standard";
        "توقيت اليابان الرسمي",
        "daylight";
        "توقيت اليابان الصيفي";
    }
}
"Kamchatka";
{
    "long";
    {
        "generic";
        "توقيت كامشاتكا",
        "standard";
        "توقيت بيتروبافلوفسك-كامتشاتسكي",
        "daylight";
        "توقيت بيتروبافلوفسك-كامتشاتسكي الصيفي";
    }
}
"Kazakhstan_Eastern";
{
    "long";
    {
        "standard";
        "توقيت شرق كازاخستان";
    }
}
"Kazakhstan_Western";
{
    "long";
    {
        "standard";
        "توقيت غرب كازاخستان";
    }
}
"Korea";
{
    "long";
    {
        "generic";
        "توقيت كوريا",
        "standard";
        "توقيت كوريا الرسمي",
        "daylight";
        "توقيت كوريا الصيفي";
    }
}

```

```

    }
  }
  "Kosrae";
  {
    "long";
    {
      "standard";
      "توقيت كوسرا";
    }
  }
  "Krasnoyarsk";
  {
    "long";
    {
      "generic";
      "توقيت كراسنويارسك",
      "standard";
      "توقيت كراسنويارسك الرسمي",
      "daylight";
      "التوقيت الصيفي لكراسنويارسك";
    }
  }
  "Kyrgystan";
  {
    "long";
    {
      "standard";
      "توقيت قرغيزستان";
    }
  }
  "Line_Islands";
  {
    "long";
    {
      "standard";
      "توقيت جزر لاين";
    }
  }
  "Lord_Howe";
  {
    "long";
    {
      "generic";
      "توقيت لورد هاو",
      "standard";
      "توقيت لورد هاو الرسمي",
      "daylight";
      "التوقيت الصيفي للورد هاو";
    }
  }
  "Macquarie";
  {
    "long";
    {
      "standard";
      "توقيت ماكوارى";
    }
  }

```

```
}
"Magadan";
{
  "long";
  {
    "generic";
    "توقيت ماغادان",
    "standard";
    "توقيت ماغادان الرسمي",
    "daylight";
    "توقيت ماغادان الصيفي";
  }
}
"Malaysia";
{
  "long";
  {
    "standard";
    "توقيت ماليزيا";
  }
}
"Maldives";
{
  "long";
  {
    "standard";
    "توقيت المالديف";
  }
}
"Marquesas";
{
  "long";
  {
    "standard";
    "توقيت ماركيساس";
  }
}
"Marshall_Islands";
{
  "long";
  {
    "standard";
    "توقيت جزر مارشال";
  }
}
"Mauritius";
{
  "long";
  {
    "generic";
    "توقيت موريشيوس",
    "standard";
    "توقيت موريشيوس الرسمي",
    "daylight";
    "توقيت موريشيوس الصيفي";
  }
}
}
```

```
"Mawson";
{
  "long";
  {
    "standard";
    "توقيت ماوسون";
  }
}
"Mexico_Northwest";
{
  "long";
  {
    "generic";
    "توقيت شمال غرب المكسيك",
    "standard";
    "التوقيت الرسمي لشمال غرب المكسيك",
    "daylight";
    "التوقيت الصيفي لشمال غرب المكسيك";
  }
}
"Mexico_Pacific";
{
  "long";
  {
    "generic";
    "توقيت المحيط الهادي للمكسيك",
    "standard";
    "توقيت المحيط الهادي الرسمي للمكسيك",
    "daylight";
    "توقيت المحيط الهادي الصيفي للمكسيك";
  }
}
"Mongolia";
{
  "long";
  {
    "generic";
    "توقيت أولان باتور",
    "standard";
    "توقيت أولان باتور الرسمي",
    "daylight";
    "توقيت أولان باتور الصيفي";
  }
}
"Moscow";
{
  "long";
  {
    "generic";
    "توقيت موسكو",
    "standard";
    "توقيت موسكو الرسمي",
    "daylight";
    "توقيت موسكو الصيفي";
  }
}
"Myanmar";
```

```

{
  "long";
  {
    "standard";
    "توقيت میانمار";
  }
}
"Nauru";
{
  "long";
  {
    "standard";
    "توقيت ناورو";
  }
}
"Nepal";
{
  "long";
  {
    "standard";
    "توقيت نيبال";
  }
}
"New_Caledonia";
{
  "long";
  {
    "generic";
    "توقيت كاليدونيا الجديدة",
    "standard";
    "توقيت كاليدونيا الجديدة الرسمي",
    "daylight";
    "توقيت كاليدونيا الجديدة الصيفي";
  }
}
"New_Zealand";
{
  "long";
  {
    "generic";
    "توقيت نيوزيلندا",
    "standard";
    "توقيت نيوزيلندا الرسمي",
    "daylight";
    "توقيت نيوزيلندا الصيفي";
  }
}
"Newfoundland";
{
  "long";
  {
    "generic";
    "توقيت نيوفاوندلاند",
    "standard";
    "توقيت نيوفاوندلاند الرسمي",
    "daylight";
    "توقيت نيوفاوندلاند الصيفي";
  }
}

```

```

    }
  }
  "Niue";
  {
    "long";
    {
      "standard";
      "توقيت نيوي";
    }
  }
  "Norfolk";
  {
    "long";
    {
      "standard";
      "توقيت جزيرة نورفولك";
    }
  }
  "Noronha";
  {
    "long";
    {
      "generic";
      "توقيت فيرناندو دي نورونها",
      "standard";
      "توقيت فرناندو دي نورونها الرسمي",
      "daylight";
      "توقيت فرناندو دي نورونها الصيفي";
    }
  }
  "North_Mariana";
  {
    "long";
    {
      "standard";
      "توقيت جزر ماريانا الشمالية";
    }
  }
  "Novosibirsk";
  {
    "long";
    {
      "generic";
      "توقيت نوفوسيبيرسك",
      "standard";
      "توقيت نوفوسيبيرسك الرسمي",
      "daylight";
      "توقيت نوفوسيبيرسك الصيفي";
    }
  }
  "Omsk";
  {
    "long";
    {
      "generic";
      "توقيت أومسك",
      "standard";

```

```

        "توقيت أومسك الرسمي",
        "daylight";
        "توقيت أومسك الصيفي";
    }
}
"Pakistan";
{
    "long";
    {
        "generic";
        "توقيت باكستان",
        "standard";
        "توقيت باكستان الرسمي",
        "daylight";
        "توقيت باكستان الصيفي";
    }
}
"Palau";
{
    "long";
    {
        "standard";
        "توقيت بالاو";
    }
}
"Papua_New_Guinea";
{
    "long";
    {
        "standard";
        "توقيت بابوا غينيا الجديدة";
    }
}
"Paraguay";
{
    "long";
    {
        "generic";
        "توقيت باراغواي",
        "standard";
        "توقيت باراغواي الرسمي",
        "daylight";
        "توقيت باراغواي الصيفي";
    }
}
"Peru";
{
    "long";
    {
        "generic";
        "توقيت بيرو",
        "standard";
        "توقيت بيرو الرسمي",
        "daylight";
        "توقيت بيرو الصيفي";
    }
}
}

```

```
"Philippines";
{
  "long";
  {
    "generic";
    "توقيت الفيلبين",
    "standard";
    "توقيت الفيلبين الرسمي",
    "daylight";
    "توقيت الفيلبين الصيفي";
  }
}
"Phoenix_Islands";
{
  "long";
  {
    "standard";
    "توقيت جزر فينكس";
  }
}
"Pierre_Miquelon";
{
  "long";
  {
    "generic";
    "توقيت سانت بيير وميكلون",
    "standard";
    "توقيت سانت بيير وميكلون الرسمي",
    "daylight";
    "توقيت سانت بيير وميكلون الصيفي";
  }
}
"Pitcairn";
{
  "long";
  {
    "standard";
    "توقيت بيتكيرن";
  }
}
"Ponape";
{
  "long";
  {
    "standard";
    "توقيت بونابي";
  }
}
"Pyongyang";
{
  "long";
  {
    "standard";
    "توقيت بيونغ يانغ";
  }
}
"Reunion";
```



```

{
  "long";
  {
    "standard";
    "توقيت ريونيون";
  }
}
"Rothera";
{
  "long";
  {
    "standard";
    "توقيت روثيرا";
  }
}
"Sakhalin";
{
  "long";
  {
    "generic";
    "توقيت ساخالين",
    "standard";
    "توقيت ساخالين الرسمي",
    "daylight";
    "توقيت ساخالين الصيفي";
  }
}
"Samara";
{
  "long";
  {
    "generic";
    "توقيت سامارا",
    "standard";
    "توقيت سامارا",
    "daylight";
    "توقيت سامارا الصيفي";
  }
}
"Samoa";
{
  "long";
  {
    "generic";
    "توقيت ساموا",
    "standard";
    "توقيت ساموا الرسمي",
    "daylight";
    "توقيت ساموا الصيفي";
  }
}
"Seychelles";
{
  "long";
  {
    "standard";
    "توقيت سيشل";
  }
}

```

```
    }  
  }  
  "Singapore";  
  {  
    "long";  
    {  
      "standard";  
      "توقيت سنغافورة";  
    }  
  }  
  "Solomon";  
  {  
    "long";  
    {  
      "standard";  
      "توقيت جزر سليمان";  
    }  
  }  
  "South_Georgia";  
  {  
    "long";  
    {  
      "standard";  
      "توقيت جنوب جورجيا";  
    }  
  }  
  "Suriname";  
  {  
    "long";  
    {  
      "standard";  
      "توقيت سورينام";  
    }  
  }  
  "Syowa";  
  {  
    "long";  
    {  
      "standard";  
      "توقيت سايووا";  
    }  
  }  
  "Tahiti";  
  {  
    "long";  
    {  
      "standard";  
      "توقيت تاهيتي";  
    }  
  }  
  "Taipei";  
  {  
    "long";  
    {  
      "generic";  
      "توقيت تايبيه",  
      "standard";  
    }  
  }
```

```

        "توقيت تايبيه الرسمي",
        "daylight";
        "توقيت تايبيه الصيفي";
    }
}
"Tajikistan";
{
    "long";
    {
        "standard";
        "توقيت طاجكستان";
    }
}
"Tokelau";
{
    "long";
    {
        "standard";
        "توقيت توكيلاو";
    }
}
"Tonga";
{
    "long";
    {
        "generic";
        "توقيت تونغا",
        "standard";
        "توقيت تونغا الرسمي",
        "daylight";
        "توقيت تونغا الصيفي";
    }
}
"Truk";
{
    "long";
    {
        "standard";
        "توقيت شوك";
    }
}
"Turkmenistan";
{
    "long";
    {
        "generic";
        "توقيت تركمانستان",
        "standard";
        "توقيت تركمانستان الرسمي",
        "daylight";
        "توقيت تركمانستان الصيفي";
    }
}
"Tuvalu";
{
    "long";
    {

```

```
        "standard";
        "توقيت توفالو";
    }
}
"Uruguay";
{
    "long";
    {
        "generic";
        "توقيت أورغواي",
        "standard";
        "توقيت أورغواي الرسمي",
        "daylight";
        "توقيت أورغواي الصيفي";
    }
}
"Uzbekistan";
{
    "long";
    {
        "generic";
        "توقيت أوزبكستان",
        "standard";
        "توقيت أوزبكستان الرسمي",
        "daylight";
        "توقيت أوزبكستان الصيفي";
    }
}
"Vanuatu";
{
    "long";
    {
        "generic";
        "توقيت فانواتو",
        "standard";
        "توقيت فانواتو الرسمي",
        "daylight";
        "توقيت فانواتو الصيفي";
    }
}
"Venezuela";
{
    "long";
    {
        "standard";
        "توقيت فنزويلا";
    }
}
"Vladivostok";
{
    "long";
    {
        "generic";
        "توقيت فلاديفوستوك",
        "standard";
        "توقيت فلاديفوستوك الرسمي",
        "daylight";
    }
}
```

```

        "توقيت فلاديفوستوك الصيفي";
    }
}
"Volgograd";
{
    "long";
    {
        "generic";
        "توقيت فولغوغراد",
        "standard";
        "توقيت فولغوغراد الرسمي",
        "daylight";
        "توقيت فولغوغراد الصيفي";
    }
}
"Vostok";
{
    "long";
    {
        "standard";
        "توقيت فوستوك";
    }
}
"Wake";
{
    "long";
    {
        "standard";
        "توقيت جزيرة ويك";
    }
}
"Wallis";
{
    "long";
    {
        "standard";
        "توقيت واليس و فوتونا";
    }
}
"Yakutsk";
{
    "long";
    {
        "generic";
        "توقيت ياكوتسك",
        "standard";
        "توقيت ياكوتسك الرسمي",
        "daylight";
        "توقيت ياكوتسك الصيفي";
    }
}
"Yekaterinburg";
{
    "long";
    {
        "generic";
        "توقيت يكاترينبورغ",

```

```
        "standard";
        "توقيت يكاترينبورغ الرسمي",
        "daylight";
        "توقيت يكاترينبورغ الصيفي";
    }
}
}
}
}
```

[Functional-component]

## APP.COMPONENT.JSX

## APP.COMPONENT.TS

--

## APP.MODULE.JSX

```

var __decorate = (this && this.__decorate) || function (decorators, target,
key, desc) {
    var c = arguments.length, r = c < 3 ? target : desc === null ? desc =
Object.getOwnPropertyDescriptor(target, key) : desc, d;
    if (typeof Reflect === "object" && typeof Reflect.decorate ===
"function") r = Reflect.decorate(decorators, target, key, desc);
    else for (var i = decorators.length - 1; i >= 0; i--) if (d =
decorators[i]) r = (c < 3 ? d(r) : c > 3 ? d(target, key, r) : d(target,
key)) || r;
    return c > 3 && r && Object.defineProperty(target, key, r), r;
};
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { TimePickerModule } from '@syncfusion/ej2-ng-calendars';
import { AppComponent } from './app.component';
/**
 * Module
 */
let AppModule = class AppModule {
};
AppModule = __decorate([
    NgModule({
        imports: [
            BrowserModule,
            TimePickerModule
        ],
        declarations: [AppComponent],
        bootstrap: [AppComponent]
    })
]);

```

```
], AppModule);
export { AppModule };
```

### APP.MODULE.TS

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { TimePickerModule } from '@syncfusion/ej2-ng-calendars';
import { AppComponent } from './app.component';
/**
 * Module
 */
@NgModule({
  imports: [
    BrowserModule,
    TimePickerModule
  ],
  declarations: [AppComponent],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

### CA-GREGORIAN.JSON

```
{
  "main": {
    "ar": {
      "identity": {
        "version": {
          "_number": "$Revision: 12879 $",
          "_cldrVersion": "30.0.3"
        },
        "language": "ar"
      },
      "dates": {
        "calendars": {
          "gregorian": {
            "months": {
              "format": {
                "abbreviated": {
                  "1": "يناير",
                  "2": "فبراير",
                  "3": "مارس",
                  "4": "أبريل",
                  "5": "مايو",
                  "6": "يونيو",
                  "7": "يوليو",
                  "8": "أغسطس",
                  "9": "سبتمبر",
                  "10": "أكتوبر",
                  "11": "نوفمبر",
                  "12": "ديسمبر"
                },
                "narrow": {
                  "1": "ي",
                  "2": "ف",
```

```

        "3": "م",
        "4": "أ",
        "5": "و",
        "6": "ن",
        "7": "ل",
        "8": "غ",
        "9": "س",
        "10": "ك",
        "11": "ب",
        "12": "د"
    },
    "wide": {
        "1": "يناير",
        "2": "فبراير",
        "3": "مارس",
        "4": "أبريل",
        "5": "مايو",
        "6": "يونيو",
        "7": "يوليو",
        "8": "أغسطس",
        "9": "سبتمبر",
        "10": "أكتوبر",
        "11": "نوفمبر",
        "12": "ديسمبر"
    }
},
"stand-alone": {
    "abbreviated": {
        "1": "يناير",
        "2": "فبراير",
        "3": "مارس",
        "4": "أبريل",
        "5": "مايو",
        "6": "يونيو",
        "7": "يوليو",
        "8": "أغسطس",
        "9": "سبتمبر",
        "10": "أكتوبر",
        "11": "نوفمبر",
        "12": "ديسمبر"
    },
    "narrow": {
        "1": "ي",
        "2": "ف",
        "3": "م",
        "4": "أ",
        "5": "و",
        "6": "ن",
        "7": "ل",
        "8": "غ",
        "9": "س",
        "10": "ك",
        "11": "ب",
        "12": "د"
    },
    "wide": {
        "1": "يناير",

```



```

        "2": "فبراير",
        "3": "مارس",
        "4": "أبريل",
        "5": "مايو",
        "6": "يونيو",
        "7": "يوليو",
        "8": "أغسطس",
        "9": "سبتمبر",
        "10": "أكتوبر",
        "11": "نوفمبر",
        "12": "ديسمبر"
    }
}
},
"days": {
    "format": {
        "abbreviated": {
            "sun": "الأحد",
            "mon": "الاثنين",
            "tue": "الثلاثاء",
            "wed": "الأربعاء",
            "thu": "الخميس",
            "fri": "الجمعة",
            "sat": "السبت"
        },
        "narrow": {
            "sun": "ح",
            "mon": "ن",
            "tue": "ث",
            "wed": "ر",
            "thu": "خ",
            "fri": "ج",
            "sat": "س"
        },
        "short": {
            "sun": "الأحد",
            "mon": "الاثنين",
            "tue": "الثلاثاء",
            "wed": "الأربعاء",
            "thu": "الخميس",
            "fri": "الجمعة",
            "sat": "السبت"
        },
        "wide": {
            "sun": "الأحد",
            "mon": "الاثنين",
            "tue": "الثلاثاء",
            "wed": "الأربعاء",
            "thu": "الخميس",
            "fri": "الجمعة",
            "sat": "السبت"
        }
    },
    "stand-alone": {
        "abbreviated": {
            "sun": "الأحد",
            "mon": "الاثنين",

```

```

        "tue": "الثلاثاء",
        "wed": "الأربعاء",
        "thu": "الخميس",
        "fri": "الجمعة",
        "sat": "السبت"
    },
    "narrow": {
        "sun": "ح",
        "mon": "ن",
        "tue": "ث",
        "wed": "ر",
        "thu": "خ",
        "fri": "ج",
        "sat": "س"
    },
    "short": {
        "sun": "الأحد",
        "mon": "الاثنين",
        "tue": "الثلاثاء",
        "wed": "الأربعاء",
        "thu": "الخميس",
        "fri": "الجمعة",
        "sat": "السبت"
    },
    "wide": {
        "sun": "الأحد",
        "mon": "الاثنين",
        "tue": "الثلاثاء",
        "wed": "الأربعاء",
        "thu": "الخميس",
        "fri": "الجمعة",
        "sat": "السبت"
    }
},
"quarters": {
    "format": {
        "abbreviated": {
            "1": "الربع الأول",
            "2": "الربع الثاني",
            "3": "الربع الثالث",
            "4": "الربع الرابع"
        },
        "narrow": {
            "1": "١",
            "2": "٢",
            "3": "٣",
            "4": "٤"
        },
        "wide": {
            "1": "الربع الأول",
            "2": "الربع الثاني",
            "3": "الربع الثالث",
            "4": "الربع الرابع"
        }
    }
},
"stand-alone": {

```

```

    "abbreviated": {
      "1": "الربع الأول",
      "2": "الربع الثاني",
      "3": "الربع الثالث",
      "4": "الربع الرابع"
    },
    "narrow": {
      "1": "١",
      "2": "٢",
      "3": "٣",
      "4": "٤"
    },
    "wide": {
      "1": "الربع الأول",
      "2": "الربع الثاني",
      "3": "الربع الثالث",
      "4": "الربع الرابع"
    }
  },
  "dayPeriods": {
    "format": {
      "abbreviated": {
        "am": "ص",
        "pm": "م",
        "morning1": "فجراً",
        "morning2": "ص",
        "afternoon1": "ظهراً",
        "afternoon2": "بعد الظهر",
        "evening1": "مساءً",
        "night1": "منتصف الليل",
        "night2": "ل"
      },
      "narrow": {
        "am": "ص",
        "pm": "م",
        "morning1": "فجراً",
        "morning2": "صباحاً",
        "afternoon1": "ظهراً",
        "afternoon2": "بعد الظهر",
        "evening1": "مساءً",
        "night1": "منتصف الليل",
        "night2": "ليلاً"
      },
      "wide": {
        "am": "ص",
        "pm": "م",
        "morning1": "فجراً",
        "morning2": "صباحاً",
        "afternoon1": "ظهراً",
        "afternoon2": "بعد الظهر",
        "evening1": "مساءً",
        "night1": "منتصف الليل",
        "night2": "ليلاً"
      }
    },
    "stand-alone": {

```

```

    "abbreviated": {
      "am": "ص",
      "pm": "م",
      "morning1": "فجراً",
      "morning2": "ص",
      "afternoon1": "ظهراً",
      "afternoon2": "بعد الظهر",
      "evening1": "مساءً",
      "night1": "منتصف الليل",
      "night2": "ليلاً"
    },
    "narrow": {
      "am": "ص",
      "pm": "م",
      "morning1": "فجراً",
      "morning2": "صباحاً",
      "afternoon1": "ظهراً",
      "afternoon2": "بعد الظهر",
      "evening1": "مساءً",
      "night1": "منتصف الليل",
      "night2": "ليلاً"
    },
    "wide": {
      "am": "صباحاً",
      "pm": "مساءً",
      "morning1": "فجراً",
      "morning2": "صباحاً",
      "afternoon1": "ظهراً",
      "afternoon2": "بعد الظهر",
      "evening1": "مساءً",
      "night1": "منتصف الليل",
      "night2": "ليلاً"
    }
  },
  "eras": {
    "eraNames": {
      "0": "قبل الميلاد",
      "0-alt-variant": "BCE",
      "1": "ميلادي",
      "1-alt-variant": "بعد الميلاد"
    },
    "eraAbbr": {
      "0": "ق.م",
      "0-alt-variant": "BCE",
      "1": "م",
      "1-alt-variant": "ب.م"
    },
    "eraNarrow": {
      "0": "ق.م",
      "0-alt-variant": "BCE",
      "1": "م",
      "1-alt-variant": "ب.م"
    }
  },
  "dateFormats": {
    "full": "EEEE، d MMMM، y",

```

```

    "long": "d MMMM, y",
    "medium": "dd/MM/y",
    "short": "d/M/y"
  },
  "timeFormats": {
    "full": "h:mm:ss a zzzz",
    "long": "h:mm:ss a z",
    "medium": "h:mm:ss a",
    "short": "h:mm a"
  },
  "dateTimeFormats": {
    "full": "{1} {0}",
    "long": "{1} {0}",
    "medium": "{1} {0}",
    "short": "{1} {0}",
    "availableFormats": {
      "d": "d",
      "E": "ccc",
      "Ed": "E, d",
      "Ehm": "E h:mm a",
      "EHm": "E HH:mm",
      "Ehms": "E h:mm:ss a",
      "EHms": "E HH:mm:ss",
      "Gy": "y G",
      "GyMMM": "MMM y G",
      "GyMMMd": "d MMM, y G",
      "GyMMMED": "E, d MMM, y G",
      "h": "h a",
      "H": "HH",
      "hm": "h:mm a",
      "Hm": "HH:mm",
      "hms": "h:mm:ss a",
      "Hms": "HH:mm:ss",
      "hmsv": "h:mm:ss a v",
      "Hmsv": "HH:mm:ss v",
      "hmv": "h:mm a v",
      "Hmv": "HH:mm v",
      "M": "L",
      "Md": "d/M",
      "MEd": "E, d/M",
      "MMdd": "dd/MM",
      "MMM": "LLL",
      "MMMd": "d MMM",
      "MMMED": "E, d MMM",
      "MMMMd": "d MMMM",
      "MMMMEd": "E, d MMMM",
      "MMMMW": "الأسبوع W من MMM",
      "MMMMW": "الأسبوع W من MMM",
      "MMMMW": "الأسبوع W من MMM",
      "MMMMW": "الأسبوع W من MMM",
      "MMMMW": "الأسبوع W من MMM",
      "MMMMW": "الأسبوع W من MMM",
      "ms": "mm:ss",
      "y": "y",
      "yM": "M/y",
      "yMd": "d/M/y",
      "yMEd": "E, d/M/y",

```

```

        "yMM": "MM/y",
        "yMMM": "MMM y",
        "yMMMd": "d MMM, y",
        "yMMMED": "E, d MMM, y",
        "yMMMM": "MMMM y",
        "yQQQ": "QQQ y",
        "yQQQQ": "QQQQ y",
        "yw": "الأسبوع w من سنة y",
        "yw": "الأسبوع w من سنة y",
        "yw": "الأسبوع w من سنة y",
        "yw": "الأسبوع w من سنة y",
        "yw": "الأسبوع w من سنة y",
        "yw": "الأسبوع w من سنة y",
    },
    "appendItems": {
        "Day": "{0} ({2}: {1})",
        "Day-Of-Week": "{0} {1}",
        "Era": "{1} {0}",
        "Hour": "{0} ({2}: {1})",
        "Minute": "{0} ({2}: {1})",
        "Month": "{0} ({2}: {1})",
        "Quarter": "{0} ({2}: {1})",
        "Second": "{0} ({2}: {1})",
        "Timezone": "{0} {1}",
        "Week": "{0} ({2}: {1})",
        "Year": "{1} {0}"
    },
    "intervalFormats": {
        "intervalFormatFallback": "{0} - {1}",
        "d": {
            "d": "d-d"
        },
        "h": {
            "a": "h a - h a",
            "h": "h-h a"
        },
        "H": {
            "H": "HH-HH"
        },
        "hm": {
            "a": "h:mm a - h:mm a",
            "h": "h:mm-h:mm a",
            "m": "h:mm-h:mm a"
        },
        "Hm": {
            "H": "HH:mm-HH:mm",
            "m": "HH:mm-HH:mm"
        },
        "hmv": {
            "a": "h:mm a - h:mm a v",
            "h": "h:mm-h:mm a v",
            "m": "h:mm-h:mm a v"
        },
        "Hmv": {
            "H": "HH:mm-HH:mm v",
            "m": "HH:mm-HH:mm v"
        }
    },

```

```

    "hv": {
      "a": "h a - h a v",
      "h": "h-h a v"
    },
    "Hv": {
      "H": "HH-HH v"
    },
    "M": {
      "M": "M-M"
    },
    "Md": {
      "d": "M/d - M/d",
      "M": "M/d - M/d"
    },
    "MEd": {
      "d": "E, d/M - E, d/M",
      "M": "E, d/M - E, d/M"
    },
    "MMM": {
      "M": "MMM-MMM"
    },
    "MMMd": {
      "d": "d-d MMM",
      "M": "d MMM - d MMM"
    },
    "MMMED": {
      "d": "E, d - E, d MMM",
      "M": "E, d MMM - E, d MMM"
    },
    "MMMM": {
      "M": "LLLL-LLLL"
    },
    "Y": {
      "Y": "Y-Y"
    },
    "yM": {
      "M": "M/y - M/y",
      "Y": "M/y - M/y"
    },
    "yMd": {
      "d": "d/M/y - d/M/y",
      "M": "d/M/y - d/M/y",
      "Y": "d/M/y - d/M/y"
    },
    "yMEd": {
      "d": "E, dd/MM/y - E, dd/MM/y",
      "M": "E, d/M/y - E, d/M/y",
      "Y": "E, d/M/y - E, d/M/y"
    },
    "yMMM": {
      "M": "MMM - MMM, y",
      "Y": "MMM, y - MMM, y"
    },
    "yMMMd": {
      "d": "d-d MMM, y",
      "M": "d MMM - d MMM, y",
      "Y": "d MMM, y - d MMM, y"
    }
  }

```

```
    },  
    "yMMMEd": {  
      "d": "E, d - E, d MMM, y",  
      "M": "E, d MMM - E, d MMM, y",  
      "Y": "E, d MMM, y - E, d MMM, y"  
    },  
    "yMMMM": {  
      "M": "MMMM - MMMM, y",  
      "Y": "MMMM, y - MMMM, y"  
    }  
  }  
}  
  
}  
  
}  
  
}  
  
}
```

## CA-GREGORIAN.JSX

```
{
  "main";
  {
    "ar";
    {
      "identity";
      {
        "version";
        {
          "_number";
          "$Revision: 12879 $",
          "_cldrVersion";
          "30.0.3";
        }
        "language";
        "ar";
      }
    }
    "dates";
    {
      "calendars";
      {
        "gregorian";
        {
          "months";
          {
            "format";
            {
              "abbreviated";
              {
                "1";
                "يناير",
                "2";
                "فبراير",
                "3";
                "مارس",

```



```
        "4";
        "أبريل",
        "5";
        "مايو",
        "6";
        "يونيو",
        "7";
        "يوليو",
        "8";
        "أغسطس",
        "9";
        "سبتمبر",
        "10";
        "أكتوبر",
        "11";
        "نوفمبر",
        "12";
        "ديسمبر";
    }
    "narrow";
    {
        "1";
        "ي",
        "2";
        "ف",
        "3";
        "م",
        "4";
        "أ",
        "5";
        "و",
        "6";
        "ن",
        "7";
        "ل",
        "8";
        "غ",
        "9";
        "س",
        "10";
        "ك",
        "11";
        "ب",
        "12";
        "د";
    }
    "wide";
    {
        "1";
        "يناير",
        "2";
        "فبراير",
        "3";
        "مارس",
        "4";
        "أبريل",
        "5";
```

```

        "مايو",
        "6";
        "يونيو",
        "7";
        "يوليو",
        "8";
        "أغسطس",
        "9";
        "سبتمبر",
        "10";
        "أكتوبر",
        "11";
        "نوفمبر",
        "12";
        "ديسمبر";
    }
}
"stand-alone";
{
    "abbreviated";
    {
        "1";
        "يناير",
        "2";
        "فبراير",
        "3";
        "مارس",
        "4";
        "أبريل",
        "5";
        "مايو",
        "6";
        "يونيو",
        "7";
        "يوليو",
        "8";
        "أغسطس",
        "9";
        "سبتمبر",
        "10";
        "أكتوبر",
        "11";
        "نوفمبر",
        "12";
        "ديسمبر";
    }
}
"narrow";
{
    "1";
    "ي",
    "2";
    "ف",
    "3";
    "م",
    "4";
    "أ",
    "5";
}

```

```

        "و",
        "6";
        "ن",
        "7";
        "ل",
        "8";
        "غ",
        "9";
        "س",
        "10";
        "ك",
        "11";
        "ب",
        "12";
        "د";
    }
    "wide";
    {
        "1";
        "يناير",
        "2";
        "فبراير",
        "3";
        "مارس",
        "4";
        "أبريل",
        "5";
        "مايو",
        "6";
        "يونيو",
        "7";
        "يوليو",
        "8";
        "أغسطس",
        "9";
        "سبتمبر",
        "10";
        "أكتوبر",
        "11";
        "نوفمبر",
        "12";
        "ديسمبر";
    }
}
"days";
{
    "format";
    {
        "abbreviated";
        {
            "sun";
            "الأحد",
            "mon";
            "الاثنين",
            "tue";
            "الثلاثاء",

```

```
        "wed";
        "الأربعاء",
        "thu";
        "الخميس",
        "fri";
        "الجمعة",
        "sat";
        "السبت";
    }
    "narrow";
    {
        "sun";
        "ح",
        "mon";
        "ن",
        "tue";
        "ث",
        "wed";
        "ر",
        "thu";
        "خ",
        "fri";
        "ج",
        "sat";
        "س";
    }
    "short";
    {
        "sun";
        "الأحد",
        "mon";
        "الاثنين",
        "tue";
        "الثلاثاء",
        "wed";
        "الأربعاء",
        "thu";
        "الخميس",
        "fri";
        "الجمعة",
        "sat";
        "السبت";
    }
    "wide";
    {
        "sun";
        "الأحد",
        "mon";
        "الاثنين",
        "tue";
        "الثلاثاء",
        "wed";
        "الأربعاء",
        "thu";
        "الخميس",
        "fri";
        "الجمعة",
```

```
        "sat";
        "السبت";
    }
}
"stand-alone";
{
    "abbreviated";
    {
        "sun";
        "الأحد",
        "mon";
        "الاثنين",
        "tue";
        "الثلاثاء",
        "wed";
        "الأربعاء",
        "thu";
        "الخميس",
        "fri";
        "الجمعة",
        "sat";
        "السبت";
    }
    "narrow";
    {
        "sun";
        "ح",
        "mon";
        "ن",
        "tue";
        "ث",
        "wed";
        "ر",
        "thu";
        "خ",
        "fri";
        "ج",
        "sat";
        "س";
    }
    "short";
    {
        "sun";
        "الأحد",
        "mon";
        "الاثنين",
        "tue";
        "الثلاثاء",
        "wed";
        "الأربعاء",
        "thu";
        "الخميس",
        "fri";
        "الجمعة",
        "sat";
        "السبت";
    }
}
```

```

        "wide";
        {
            "sun";
            "الأحد",
            "mon";
            "الاثنين",
            "tue";
            "الثلاثاء",
            "wed";
            "الأربعاء",
            "thu";
            "الخميس",
            "fri";
            "الجمعة",
            "sat";
            "السبت";
        }
    }
}
"quarters";
{
    "format";
    {
        "abbreviated";
        {
            "1";
            "الربع الأول",
            "2";
            "الربع الثاني",
            "3";
            "الربع الثالث",
            "4";
            "الربع الرابع";
        }
        "narrow";
        {
            "1";
            "١",
            "2";
            "٢",
            "3";
            "٣",
            "4";
            "٤";
        }
        "wide";
        {
            "1";
            "الربع الأول",
            "2";
            "الربع الثاني",
            "3";
            "الربع الثالث",
            "4";
            "الربع الرابع";
        }
    }
}

```

```
    "stand-alone";
    {
      "abbreviated";
      {
        "1";
        "الربع الأول",
        "2";
        "الربع الثاني",
        "3";
        "الربع الثالث",
        "4";
        "الربع الرابع";
      }
      "narrow";
      {
        "1";
        "١",
        "2";
        "٢",
        "3";
        "٣",
        "4";
        "٤";
      }
      "wide";
      {
        "1";
        "الربع الأول",
        "2";
        "الربع الثاني",
        "3";
        "الربع الثالث",
        "4";
        "الربع الرابع";
      }
    }
  }
  "dayPeriods";
  {
    "format";
    {
      "abbreviated";
      {
        "am";
        "ص",
        "pm";
        "م",
        "morning1";
        "فجراً",
        "morning2";
        "ص",
        "afternoon1";
        "ظهراً",
        "afternoon2";
        "بعد الظهر",
        "evening1";
        "مساءً",

```

```

        "night1";
        "منتصف الليل",
        "night2";
        "ل";
    }
    "narrow";
    {
        "am";
        "ص",
        "pm";
        "م",
        "morning1";
        "فجرًا",
        "morning2";
        "صباحًا",
        "afternoon1";
        "ظهرًا",
        "afternoon2";
        "بعد الظهر",
        "evening1";
        "مساءً",
        "night1";
        "منتصف الليل",
        "night2";
        "ليلاً";
    }
    "wide";
    {
        "am";
        "ص",
        "pm";
        "م",
        "morning1";
        "فجرًا",
        "morning2";
        "صباحًا",
        "afternoon1";
        "ظهرًا",
        "afternoon2";
        "بعد الظهر",
        "evening1";
        "مساءً",
        "night1";
        "منتصف الليل",
        "night2";
        "ليلاً";
    }
}
"stand-alone";
{
    "abbreviated";
    {
        "am";
        "ص",
        "pm";
        "م",
        "morning1";

```



```

        "فجراً",
        "morning2";
        "ص",
        "afternoon1";
        "ظهراً",
        "afternoon2";
        "بعد الظهر",
        "evening1";
        "مساءً",
        "night1";
        "منتصف الليل",
        "night2";
        "ليلاً";
    }
    "narrow";
    {
        "am";
        "ص",
        "pm";
        "م",
        "morning1";
        "فجراً",
        "morning2";
        "صباحاً",
        "afternoon1";
        "ظهراً",
        "afternoon2";
        "بعد الظهر",
        "evening1";
        "مساءً",
        "night1";
        "منتصف الليل",
        "night2";
        "ليلاً";
    }
    "wide";
    {
        "am";
        "صباحاً",
        "pm";
        "مساءً",
        "morning1";
        "فجراً",
        "morning2";
        "صباحاً",
        "afternoon1";
        "ظهراً",
        "afternoon2";
        "بعد الظهر",
        "evening1";
        "مساءً",
        "night1";
        "منتصف الليل",
        "night2";
        "ليلاً";
    }
}

```

```

    }
    "eras";
    {
      "eraNames";
      {
        "0";
        "قبل الميلاد",
        "0-alt-variant";
        "BCE",
        "1";
        "ميلادي",
        "1-alt-variant";
        "بعد الميلاد";
      }
      "eraAbbr";
      {
        "0";
        "ق.م",
        "0-alt-variant";
        "BCE",
        "1";
        "م",
        "1-alt-variant";
        "ب.م";
      }
      "eraNarrow";
      {
        "0";
        "ق.م",
        "0-alt-variant";
        "BCE",
        "1";
        "م",
        "1-alt-variant";
        "ب.م";
      }
    }
    "dateFormats";
    {
      "full";
      "EEEE, d MMMM, y",
      "long";
      "d MMMM, y",
      "medium";
      "dd/MM/y",
      "short";
      "d/M/y";
    }
    "timeFormats";
    {
      "full";
      "h:mm:ss a zzzz",
      "long";
      "h:mm:ss a z",
      "medium";
      "h:mm:ss a",
      "short";
    }
  }

```

```

        "h:mm a";
    }
    "dateTimeFormats";
    {
        "full";
        "{1} {0}",
        "long";
        "{1} {0}",
        "medium";
        "{1} {0}",
        "short";
        "{1} {0}",
        "availableFormats";
        {
            "d";
            "d",
            "E";
            "ccc",
            "Ed";
            "E, d",
            "Ehm";
            "E h:mm a",
            "EHm";
            "E HH:mm",
            "Ehms";
            "E h:mm:ss a",
            "EHms";
            "E HH:mm:ss",
            "Gy";
            "y G",
            "GyMMM";
            "MMM y G",
            "GyMMMd";
            "d MMM, y G",
            "GyMMMED";
            "E, d MMM, y G",
            "h";
            "h a",
            "H";
            "HH",
            "hm";
            "h:mm a",
            "Hm";
            "HH:mm",
            "hms";
            "h:mm:ss a",
            "Hms";
            "HH:mm:ss",
            "hmsv";
            "h:mm:ss a v",
            "Hmsv";
            "HH:mm:ss v",
            "hmv";
            "h:mm a v",
            "Hmv";
            "HH:mm v",
            "M";
        }
    }
}

```

```

"L",
  "Md";
"d/M",
  "MEd";
"E, d/M",
  "MMdd";
"dd/MM",
  "MMM";
"LLL",
  "MMMd";
"d MMM",
  "MMMEd";
"E, d MMM",
  "MMMMd";
"d MMMM",
  "MMMMEd";
"E, d MMMM",
  "MMMMW";
"من الاسبوع W MMM",
  "MMMMW";
"من الاسبوع W MMM",
  "MMMMW";
"من الاسبوع W MMM",
  "MMMMW";
"من الاسبوع W MMM",
  "MMMMW";
"من الاسبوع W MMM",
  "MMMMW";
"من الاسبوع W MMM",
  "MMMMW";
"من الاسبوع W MMM",
  "ms";
"mm:ss",
  "y";
"Y",
  "yM";
"M/Y",
  "yMd";
"d/M/Y",
  "yMEd";
"E, d/M/Y",
  "yMM";
"MM/Y",
  "yMMM";
"MMM Y",
  "yMMMd";
"d MMM, Y",
  "yMMMEd";
"E, d MMM, Y",
  "yMMMM";
"MMMM Y",
  "yQQQ";
"QQQ Y",
  "yQQQQ";
"QQQQ Y",
  "yw";
"Y من سنة w الاسبوع",
  "yw";
"Y من سنة w الاسبوع",

```

```

        "yw";
        "y من سنة w الأسبوع",
        "yw";
        "y من سنة w الأسبوع",
        "yw";
        "y من سنة w الأسبوع",
        "yw";
        "y من سنة w الأسبوع";
    }
    "appendItems";
    {
        "Day";
        "{0} ({2}: {1})",
        "Day-Of-Week";
        "{0} {1}",
        "Era";
        "{1} {0}",
        "Hour";
        "{0} ({2}: {1})",
        "Minute";
        "{0} ({2}: {1})",
        "Month";
        "{0} ({2}: {1})",
        "Quarter";
        "{0} ({2}: {1})",
        "Second";
        "{0} ({2}: {1})",
        "Timezone";
        "{0} {1}",
        "Week";
        "{0} ({2}: {1})",
        "Year";
        "{1} {0}";
    }
    "intervalFormats";
    {
        "intervalFormatFallback";
        "{0} - {1}",
        "d";
        {
            "d";
            "d-d";
        }
        "h";
        {
            "a";
            "h a - h a",
            "h";
            "h-h a";
        }
        "H";
        {
            "H";
            "HH-HH";
        }
        "hm";
        {

```

```

        "a";
        "h:mm a - h:mm a",
        "h";
        "h:mm-h:mm a",
        "m";
        "h:mm-h:mm a";
    }
    "Hm";
    {
        "H";
        "HH:mm-HH:mm",
        "m";
        "HH:mm-HH:mm";
    }
    "hmv";
    {
        "a";
        "h:mm a - h:mm a v",
        "h";
        "h:mm-h:mm a v",
        "m";
        "h:mm-h:mm a v";
    }
    "Hmv";
    {
        "H";
        "HH:mm-HH:mm v",
        "m";
        "HH:mm-HH:mm v";
    }
    "hv";
    {
        "a";
        "h a - h a v",
        "h";
        "h-h a v";
    }
    "Hv";
    {
        "H";
        "HH-HH v";
    }
    "M";
    {
        "M";
        "M-M";
    }
    "Md";
    {
        "d";
        "M/d - M/d",
        "M";
        "M/d - M/d";
    }
    "MEd";
    {
        "d";

```

```

        "E, d/M - E, d/M",
        "M";
        "E, d/M - E, d/M";
    }
    "MMM";
    {
        "M";
        "MMM-MMM";
    }
    "MMMd";
    {
        "d";
        "d-d MMM",
        "M";
        "d MMM - d MMM";
    }
    "MMMEd";
    {
        "d";
        "E, d - E, d MMM",
        "M";
        "E, d MMM - E, d MMM";
    }
    "MMMM";
    {
        "M";
        "LLLL-LLLL";
    }
    "Y";
    {
        "Y";
        "Y-Y";
    }
    "yM";
    {
        "M";
        "M/y - M/y",
        "Y";
        "M/y - M/y";
    }
    "yMd";
    {
        "d";
        "d/M/y - d/M/y",
        "M";
        "d/M/y - d/M/y",
        "Y";
        "d/M/y - d/M/y";
    }
    "yMEd";
    {
        "d";
        "E, dd/MM/y - E, dd/MM/y",
        "M";
        "E, d/M/y - E, d/M/y",
        "Y";
        "E, d/M/y - E, d/M/y";
    }

```

```

}
"yMMM";
{
    "M";
    "MMM - MMM. y",
    "Y";
    "MMM. y - MMM. y";
}
"yMMMd";
{
    "d";
    "d-d MMM. y",
    "M";
    "d MMM - d MMM. y",
    "Y";
    "d MMM. y - d MMM. y";
}
"yMMMEd";
{
    "d";
    "E. d - E. d MMM. y",
    "M";
    "E. d MMM - E. d MMM. y",
    "Y";
    "E. d MMM. y - E. d MMM. y";
}
"yMMMM";
{
    "M";
    "MMMM - MMMM. y",
    "Y";
    "MMMM. y - MMMM. y";
}
}
}
}
}
}
}
}
}
}

```

## CURRENCIES.JSON

```
{
  "main": {
    "ar": {
      "identity": {
        "version": {
          "_number": "$Revision: 12879 $",
          "_cldrVersion": "30.0.3"
        },
        "language": "ar"
      },
      "numbers": {
        "currencies": {
```



```

"ADP": {
  "displayName": "بيستا أندوري",
  "symbol": "ADP"
},
"AED": {
  "displayName": "درهم إماراتي",
  "displayName-count-zero": "درهم إماراتي",
  "displayName-count-one": "درهم إماراتي",
  "displayName-count-two": "درهم إماراتي",
  "displayName-count-few": "درهم إماراتي",
  "displayName-count-many": "درهم إماراتي",
  "displayName-count-other": "درهم إماراتي",
  "symbol": "د.إ."
},
"AFA": {
  "displayName": "أفغاني - 1927-2002",
  "symbol": "AFA"
},
"AFN": {
  "displayName": "أفغاني",
  "displayName-count-zero": "أفغاني أفغانستان",
  "displayName-count-one": "أفغاني أفغانستان",
  "displayName-count-two": "أفغاني أفغانستان",
  "displayName-count-few": "أفغاني أفغانستان",
  "displayName-count-many": "أفغاني أفغانستان",
  "displayName-count-other": "أفغاني أفغانستان",
  "symbol": "AFN"
},
"ALL": {
  "displayName": "ليك ألباني",
  "displayName-count-zero": "ليك ألباني",
  "displayName-count-one": "ليك ألباني",
  "displayName-count-two": "ليك ألباني",
  "displayName-count-few": "ليك ألباني",
  "displayName-count-many": "ليك ألباني",
  "displayName-count-other": "ليك ألباني",
  "symbol": "ALL"
},
"AMD": {
  "displayName": "درام أرميني",
  "displayName-count-zero": "درام أرميني",
  "displayName-count-one": "درام أرميني",
  "displayName-count-two": "درام أرميني",
  "displayName-count-few": "درام أرميني",
  "displayName-count-many": "درام أرميني",
  "displayName-count-other": "درام أرميني",
  "symbol": "AMD"
},
"ANG": {
  "displayName": "غيلدر أنتيلي هولندي",
  "displayName-count-zero": "غيلدر أنتيلي هولندي",
  "displayName-count-one": "غيلدر أنتيلي هولندي",
  "displayName-count-two": "غيلدر أنتيلي هولندي",
  "displayName-count-few": "غيلدر أنتيلي هولندي",
  "displayName-count-many": "غيلدر أنتيلي هولندي",
  "displayName-count-other": "غيلدر أنتيلي هولندي",
  "symbol": "ANG"
}

```

```

    },
    "AOA": {
      "displayName": "كوانزا أنجولي",
      "displayName-count-zero": "كوانزا أنجولي",
      "displayName-count-one": "كوانزا أنجولي",
      "displayName-count-two": "كوانزا أنجولي",
      "displayName-count-few": "كوانزا أنجولي",
      "displayName-count-many": "كوانزا أنجولي",
      "displayName-count-other": "كوانزا أنجولي",
      "symbol": "AOA",
      "symbol-alt-narrow": "Kz"
    },
    "AOK": {
      "displayName": "كوانزا أنجولي - 1990-1977",
      "symbol": "AOK"
    },
    "AON": {
      "displayName": "كوانزا أنجولي جديدة - 2000-1990",
      "symbol": "AON"
    },
    "AOR": {
      "displayName": "كوانزا أنجولي معدلة - 1999 - 1995",
      "symbol": "AOR"
    },
    "ARA": {
      "displayName": "استرال أرجنتيني",
      "symbol": "ARA"
    },
    "ARL": {
      "displayName": "ARL",
      "symbol": "ARL"
    },
    "ARM": {
      "displayName": "ARM",
      "symbol": "ARM"
    },
    "ARP": {
      "displayName": "بيزو أرجنتيني - 1985-1983",
      "symbol": "ARP"
    },
    "ARS": {
      "displayName": "بيزو أرجنتيني",
      "displayName-count-zero": "بيزو أرجنتيني",
      "displayName-count-one": "بيزو أرجنتيني",
      "displayName-count-two": "بيزو أرجنتيني",
      "displayName-count-few": "بيزو أرجنتيني",
      "displayName-count-many": "بيزو أرجنتيني",
      "displayName-count-other": "بيزو أرجنتيني",
      "symbol": "ARS",
      "symbol-alt-narrow": "AR$"
    },
    "ATS": {
      "displayName": "شلن نمساوي",
      "symbol": "ATS"
    },
    "AUD": {
      "displayName": "دولار أسترالي",

```

```

    "displayName-count-zero": "دولار أسترالي",
    "displayName-count-one": "دولار أسترالي",
    "displayName-count-two": "دولار أسترالي",
    "displayName-count-few": "دولار أسترالي",
    "displayName-count-many": "دولار أسترالي",
    "displayName-count-other": "دولار أسترالي",
    "symbol": "AU$",
    "symbol-alt-narrow": "AU$"
  },
  "AWG": {
    "displayName": "فلورن أروبي",
    "displayName-count-zero": "فلورن أروبي",
    "displayName-count-one": "فلورن أروبي",
    "displayName-count-two": "فلورن أروبي",
    "displayName-count-few": "فلورن أروبي",
    "displayName-count-many": "فلورن أروبي",
    "displayName-count-other": "فلورن أروبي",
    "symbol": "AWG"
  },
  "AZM": {
    "displayName": "مانات أذربيجاني",
    "symbol": "AZM"
  },
  "AZN": {
    "displayName": "مانات أذربيجان",
    "displayName-count-zero": "مانت أذربيجاني",
    "displayName-count-one": "مانت أذربيجاني",
    "displayName-count-two": "مانت أذربيجاني",
    "displayName-count-few": "مانت أذربيجاني",
    "displayName-count-many": "مانت أذربيجاني",
    "displayName-count-other": "مانت أذربيجاني",
    "symbol": "AZN"
  },
  "BAD": {
    "displayName": "دينار البوسنة والهرسك",
    "symbol": "BAD"
  },
  "BAM": {
    "displayName": "مارك البوسنة والهرسك قابل للتحويل",
    "displayName-count-zero": "مارك البوسنة والهرسك قابل للتحويل",
    "displayName-count-one": "مارك البوسنة والهرسك قابل للتحويل",
    "displayName-count-two": "مارك البوسنة والهرسك قابل للتحويل",
    "displayName-count-few": "مارك البوسنة والهرسك قابل للتحويل",
    "displayName-count-many": "مارك البوسنة والهرسك قابل للتحويل",
    "displayName-count-other": "مارك البوسنة والهرسك قابل للتحويل",
    "symbol": "BAM",
    "symbol-alt-narrow": "KM"
  },
  "BAN": {
    "displayName": "BAN",
    "symbol": "BAN"
  },
  "BBD": {
    "displayName": "دولار بربادوسي",
    "displayName-count-zero": "دولار بربادوسي",
    "displayName-count-one": "دولار بربادوسي",
    "displayName-count-two": "دولار بربادوسي",

```

```

    "displayName-count-few": "دولار بربادوسي",
    "displayName-count-many": "دولار بربادوسي",
    "displayName-count-other": "دولار بربادوسي",
    "symbol": "BBD",
    "symbol-alt-narrow": "BB$"
  },
  "BDT": {
    "displayName": "তাকা بنجلاديشي",
    "displayName-count-zero": "তাকা بنجلاديشي",
    "displayName-count-one": "তাকা بنجلاديشي",
    "displayName-count-two": "তাকা بنجلاديشي",
    "displayName-count-few": "তাকা بنجلاديشي",
    "displayName-count-many": "তাকা بنجلاديشي",
    "displayName-count-other": "তাকা بنجلاديشي",
    "symbol": "BDT",
    "symbol-alt-narrow": "₳"
  },
  "BEC": {
    "displayName": "فرنك بلجيكي قابل للتحويل",
    "symbol": "BEC"
  },
  "BEF": {
    "displayName": "فرنك بلجيكي",
    "symbol": "BEF"
  },
  "BEL": {
    "displayName": "فرنك بلجيكي مالي",
    "symbol": "BEL"
  },
  "BGL": {
    "displayName": "BGL",
    "symbol": "BGL"
  },
  "BGM": {
    "displayName": "BGM",
    "symbol": "BGM"
  },
  "BGN": {
    "displayName": "ليف بلغاري",
    "displayName-count-zero": "ليف بلغاري",
    "displayName-count-one": "ليف بلغاري",
    "displayName-count-two": "ليف بلغاري",
    "displayName-count-few": "ليف بلغاري",
    "displayName-count-many": "ليف بلغاري",
    "displayName-count-other": "ليف بلغاري",
    "symbol": "BGN"
  },
  "BGO": {
    "displayName": "BGO",
    "symbol": "BGO"
  },
  "BHD": {
    "displayName": "دينار بحريني",
    "displayName-count-zero": "دينار بحريني",
    "displayName-count-one": "دينار بحريني",
    "displayName-count-two": "دينار بحريني",

```

```

    "displayName-count-few": "دينار بحريني",
    "displayName-count-many": "دينار بحريني",
    "displayName-count-other": "دينار بحريني",
    "symbol": "د.ب."
  },
  "BIF": {
    "displayName": "فرنك بروندي",
    "displayName-count-zero": "فرنك بروندي",
    "displayName-count-one": "فرنك بروندي",
    "displayName-count-two": "فرنك بروندي",
    "displayName-count-few": "فرنك بروندي",
    "displayName-count-many": "فرنك بروندي",
    "displayName-count-other": "فرنك بروندي",
    "symbol": "BIF"
  },
  "BMD": {
    "displayName": "دولار برمودي",
    "displayName-count-zero": "دولار برمودي",
    "displayName-count-one": "دولار برمودي",
    "displayName-count-two": "دولار برمودي",
    "displayName-count-few": "دولار برمودي",
    "displayName-count-many": "دولار برمودي",
    "displayName-count-other": "دولار برمودي",
    "symbol": "BMD",
    "symbol-alt-narrow": "BM$"
  },
  "BND": {
    "displayName": "دولار بروناي",
    "displayName-count-zero": "دولار بروناي",
    "displayName-count-one": "دولار بروناي",
    "displayName-count-two": "دولار بروناي",
    "displayName-count-few": "دولار بروناي",
    "displayName-count-many": "دولار بروناي",
    "displayName-count-other": "دولار بروناي",
    "symbol": "BND",
    "symbol-alt-narrow": "BN$"
  },
  "BOB": {
    "displayName": "بوليفيانو بوليفي",
    "displayName-count-zero": "بوليفيانو بوليفي",
    "displayName-count-one": "بوليفيانو بوليفي",
    "displayName-count-two": "بوليفيانو بوليفي",
    "displayName-count-few": "بوليفيانو بوليفي",
    "displayName-count-many": "بوليفيانو بوليفي",
    "displayName-count-other": "بوليفيانو بوليفي",
    "symbol": "BOB",
    "symbol-alt-narrow": "Bs"
  },
  "BOL": {
    "displayName": "BOL",
    "symbol": "BOL"
  },
  "BOP": {
    "displayName": "بيزو بوليفي",
    "symbol": "BOP"
  },
  "BOV": {

```

```

    "displayName": "مفدول بوليڤي",
    "symbol": "BOV"
  },
  "BRB": {
    "displayName": "نوفو كروزايرو برازيلى - 1986-1967",
    "symbol": "BRB"
  },
  "BRC": {
    "displayName": "كروزادو برازيلى",
    "symbol": "BRC"
  },
  "BRE": {
    "displayName": "كروزايرو برازيلى - 1993-1990",
    "symbol": "BRE"
  },
  "BRL": {
    "displayName": "ريال برازيلى",
    "displayName-count-zero": "ريال برازيلى",
    "displayName-count-one": "ريال برازيلى",
    "displayName-count-two": "ريال برازيلى",
    "displayName-count-few": "ريال برازيلى",
    "displayName-count-many": "ريال برازيلى",
    "displayName-count-other": "ريال برازيلى",
    "symbol": "R$",
    "symbol-alt-narrow": "R$"
  },
  "BRN": {
    "displayName": "BRN",
    "symbol": "BRN"
  },
  "BRR": {
    "displayName": "BRR",
    "symbol": "BRR"
  },
  "BRZ": {
    "displayName": "BRZ",
    "symbol": "BRZ"
  },
  "BSD": {
    "displayName": "دولار باهامى",
    "displayName-count-zero": "دولار باهامى",
    "displayName-count-one": "دولار باهامى",
    "displayName-count-two": "دولار باهامى",
    "displayName-count-few": "دولار باهامى",
    "displayName-count-many": "دولار باهامى",
    "displayName-count-other": "دولار باهامى",
    "symbol": "BSD",
    "symbol-alt-narrow": "BS$"
  },
  "BTN": {
    "displayName": "نولتوم بوتانى",
    "displayName-count-zero": "نولتوم بوتانى",
    "displayName-count-one": "نولتوم بوتانى",
    "displayName-count-two": "نولتوم بوتانى",
    "displayName-count-few": "نولتوم بوتانى",
    "displayName-count-many": "نولتوم بوتانى",
    "displayName-count-other": "نولتوم بوتانى",

```

```

    "symbol": "BTN"
  },
  "BUK": {
    "displayName": "کیات بورمی",
    "symbol": "BUK"
  },
  "BWP": {
    "displayName": "بولا بتسوانی",
    "displayName-count-zero": "بولا بتسوانی",
    "displayName-count-one": "بولا بتسوانی",
    "displayName-count-two": "بولا بتسوانی",
    "displayName-count-few": "بولا بتسوانی",
    "displayName-count-many": "بولا بتسوانی",
    "displayName-count-other": "بولا بتسوانی",
    "symbol": "BWP",
    "symbol-alt-narrow": "P"
  },
  "BYB": {
    "displayName": "روبل بیلاروسی جدید - 1999-1994",
    "symbol": "BYB"
  },
  "BYN": {
    "displayName": "روبل بیلاروسی",
    "displayName-count-zero": "روبل بیلاروسی",
    "displayName-count-one": "روبل بیلاروسی",
    "displayName-count-two": "روبل بیلاروسی",
    "displayName-count-few": "روبل بیلاروسی",
    "displayName-count-many": "روبل بیلاروسی",
    "displayName-count-other": "روبل بیلاروسی",
    "symbol": "BYN",
    "symbol-alt-narrow": "p."
  },
  "BYR": {
    "displayName": "روبل بیلاروسی (۲۰۱۶-۲۰۰۰)",
    "displayName-count-zero": "روبل بیلاروسی (۲۰۱۶-۲۰۰۰)",
    "displayName-count-one": "روبل بیلاروسی (۲۰۱۶-۲۰۰۰)",
    "displayName-count-two": "روبل بیلاروسی (۲۰۱۶-۲۰۰۰)",
    "displayName-count-few": "روبل بیلاروسی (۲۰۱۶-۲۰۰۰)",
    "displayName-count-many": "روبل بیلاروسی (۲۰۱۶-۲۰۰۰)",
    "displayName-count-other": "روبل بیلاروسی (۲۰۱۶-۲۰۰۰)",
    "symbol": "BYR"
  },
  "BZD": {
    "displayName": "دولار بلیزی",
    "displayName-count-zero": "دولار بلیزی",
    "displayName-count-one": "دولار بلیزی",
    "displayName-count-two": "دولار ان بلیزیان",
    "displayName-count-few": "دولار بلیزی",
    "displayName-count-many": "دولار بلیزی",
    "displayName-count-other": "دولار بلیزی",
    "symbol": "BZD",
    "symbol-alt-narrow": "BZ$"
  },
  "CAD": {
    "displayName": "دولار کندي",
    "displayName-count-zero": "دولار کندي",
    "displayName-count-one": "دولار کندي",

```

```

    "displayName-count-two": "دولار كندي",
    "displayName-count-few": "دولار كندي",
    "displayName-count-many": "دولار كندي",
    "displayName-count-other": "دولار كندي",
    "symbol": "CA$",
    "symbol-alt-narrow": "CA$"
  },
  "CDF": {
    "displayName": "فرنك كونغولي",
    "displayName-count-zero": "فرنك كونغولي",
    "displayName-count-one": "فرنك كونغولي",
    "displayName-count-two": "فرنك كونغولي",
    "displayName-count-few": "فرنك كونغولي",
    "displayName-count-many": "فرنك كونغولي",
    "displayName-count-other": "فرنك كونغولي",
    "symbol": "CDF"
  },
  "CHE": {
    "displayName": "CHE",
    "symbol": "CHE"
  },
  "CHF": {
    "displayName": "فرنك سويسري",
    "displayName-count-zero": "فرنك سويسري",
    "displayName-count-one": "فرنك سويسري",
    "displayName-count-two": "فرنك سويسري",
    "displayName-count-few": "فرنك سويسري",
    "displayName-count-many": "فرنك سويسري",
    "displayName-count-other": "فرنك سويسري",
    "symbol": "CHF"
  },
  "CHW": {
    "displayName": "CHW",
    "symbol": "CHW"
  },
  "CLE": {
    "displayName": "CLE",
    "symbol": "CLE"
  },
  "CLF": {
    "displayName": "CLF",
    "symbol": "CLF"
  },
  "CLP": {
    "displayName": "بيزو شيلي",
    "displayName-count-zero": "بيزو شيلي",
    "displayName-count-one": "بيزو شيلي",
    "displayName-count-two": "بيزو شيلي",
    "displayName-count-few": "بيزو شيلي",
    "displayName-count-many": "بيزو شيلي",
    "displayName-count-other": "بيزو شيلي",
    "symbol": "CLP",
    "symbol-alt-narrow": "CL$"
  },
  "CNY": {
    "displayName": "يوان صيني",
    "displayName-count-zero": "يوان صيني",

```



```

    "displayName-count-one": "يوان صيني",
    "displayName-count-two": "يوان صيني",
    "displayName-count-few": "يوان صيني",
    "displayName-count-many": "يوان صيني",
    "displayName-count-other": "يوان صيني",
    "symbol": "CN¥",
    "symbol-alt-narrow": "CN¥"
  },
  "COP": {
    "displayName": "بيزو كولومبي",
    "displayName-count-zero": "بيزو كولومبي",
    "displayName-count-one": "بيزو كولومبي",
    "displayName-count-two": "بيزو كولومبي",
    "displayName-count-few": "بيزو كولومبي",
    "displayName-count-many": "بيزو كولومبي",
    "displayName-count-other": "بيزو كولومبي",
    "symbol": "COP",
    "symbol-alt-narrow": "CO$"
  },
  "COU": {
    "displayName": "COU",
    "symbol": "COU"
  },
  "CRC": {
    "displayName": "كولن كوستا ريكي",
    "displayName-count-zero": "كولن كوستا ريكي",
    "displayName-count-one": "كولن كوستا ريكي",
    "displayName-count-two": "كولن كوستا ريكي",
    "displayName-count-few": "كولن كوستا ريكي",
    "displayName-count-many": "كولن كوستا ريكي",
    "displayName-count-other": "كولن كوستا ريكي",
    "symbol": "CRC",
    "symbol-alt-narrow": "₡"
  },
  "CSD": {
    "displayName": "دينار صربي قديم",
    "symbol": "CSD"
  },
  "CSK": {
    "displayName": "كرونه تشيكوسلوفاكيا",
    "symbol": "CSK"
  },
  "CUC": {
    "displayName": "بيزو كوبي قابل للتحويل",
    "displayName-count-zero": "بيزو كوبي قابل للتحويل",
    "displayName-count-one": "بيزو كوبي قابل للتحويل",
    "displayName-count-two": "بيزو كوبي قابل للتحويل",
    "displayName-count-few": "بيزو كوبي قابل للتحويل",
    "displayName-count-many": "بيزو كوبي قابل للتحويل",
    "displayName-count-other": "بيزو كوبي قابل للتحويل",
    "symbol": "CUC",
    "symbol-alt-narrow": "$"
  },
  "CUP": {
    "displayName": "بيزو كوبي",
    "displayName-count-zero": "بيزو كوبي",
    "displayName-count-one": "بيزو كوبي",

```

```

    "displayName-count-two": "بيزو كوبي",
    "displayName-count-few": "بيزو كوبي",
    "displayName-count-many": "بيزو كوبي",
    "displayName-count-other": "بيزو كوبي",
    "symbol": "CUP",
    "symbol-alt-narrow": "CU$"
  },
  "CVE": {
    "displayName": "اسكودو الرأس الخضراء",
    "displayName-count-zero": "اسكودو الرأس الخضراء",
    "displayName-count-one": "اسكودو الرأس الخضراء",
    "displayName-count-two": "اسكودو الرأس الخضراء",
    "displayName-count-few": "اسكودو الرأس الخضراء",
    "displayName-count-many": "اسكودو الرأس الخضراء",
    "displayName-count-other": "اسكودو الرأس الخضراء",
    "symbol": "CVE"
  },
  "CYP": {
    "displayName": "جنيه قبرصي",
    "symbol": "CYP"
  },
  "CZK": {
    "displayName": "كرونة تشيكية",
    "displayName-count-zero": "كرونة تشيكية",
    "displayName-count-one": "كرونة تشيكية",
    "displayName-count-two": "كرونة تشيكية",
    "displayName-count-few": "كرونة تشيكية",
    "displayName-count-many": "كرونة تشيكية",
    "displayName-count-other": "كرونة تشيكية",
    "symbol": "CZK",
    "symbol-alt-narrow": "Kč"
  },
  "DDM": {
    "displayName": "أوستمارك ألماني شرقي",
    "symbol": "DDM"
  },
  "DEM": {
    "displayName": "مارك ألماني",
    "symbol": "DEM"
  },
  "DJF": {
    "displayName": "فرنك جيبوتي",
    "displayName-count-zero": "فرنك جيبوتي",
    "displayName-count-one": "فرنك جيبوتي",
    "displayName-count-two": "فرنك جيبوتي",
    "displayName-count-few": "فرنك جيبوتي",
    "displayName-count-many": "فرنك جيبوتي",
    "displayName-count-other": "فرنك جيبوتي",
    "symbol": "DJF"
  },
  "DKK": {
    "displayName": "كرونة دانماركي",
    "displayName-count-zero": "كرونة دانماركي",
    "displayName-count-one": "كرونة دانماركي",
    "displayName-count-two": "كرونة دانماركي",
    "displayName-count-few": "كرونة دانماركي",
    "displayName-count-many": "كرونة دانماركي",

```

```

    "displayName-count-other": "كرونه دانماركي",
    "symbol": "DKK",
    "symbol-alt-narrow": "kr"
  },
  "DOP": {
    "displayName": "بيزو الدومنيكان",
    "displayName-count-zero": "بيزو الدومنيكان",
    "displayName-count-one": "بيزو الدومنيكان",
    "displayName-count-two": "بيزو الدومنيكان",
    "displayName-count-few": "بيزو الدومنيكان",
    "displayName-count-many": "بيزو الدومنيكان",
    "displayName-count-other": "بيزو الدومنيكان",
    "symbol": "DOP",
    "symbol-alt-narrow": "DO$"
  },
  "DZD": {
    "displayName": "دينار جزائري",
    "displayName-count-zero": "دينار جزائري",
    "displayName-count-one": "دينار جزائري",
    "displayName-count-two": "ديناران جزائريان",
    "displayName-count-few": "دينارات جزائرية",
    "displayName-count-many": "دينارًا جزائريًا",
    "displayName-count-other": "دينار جزائري",
    "symbol": "د.ج."
  },
  "ECS": {
    "displayName": "ECS",
    "symbol": "ECS"
  },
  "ECV": {
    "displayName": "ECV",
    "symbol": "ECV"
  },
  "EEK": {
    "displayName": "كرونه استونية",
    "symbol": "EEK"
  },
  "EGP": {
    "displayName": "جنيه مصري",
    "displayName-count-zero": "جنيه مصري",
    "displayName-count-one": "جنيه مصري",
    "displayName-count-two": "جنيهان مصريان",
    "displayName-count-few": "جنيهات مصرية",
    "displayName-count-many": "جنيهاً مصريًا",
    "displayName-count-other": "جنيه مصري",
    "symbol": "ج.م.",
    "symbol-alt-narrow": "E£"
  },
  "ERN": {
    "displayName": "ناكفا أريتري",
    "displayName-count-zero": "ناكفا أريتري",
    "displayName-count-one": "ناكفا أريتري",
    "displayName-count-two": "ناكفا أريتري",
    "displayName-count-few": "ناكفا أريتري",
    "displayName-count-many": "ناكفا أريتري",
    "displayName-count-other": "ناكفا أريتري",
    "symbol": "ERN"
  }

```

```

    },
    "ESA": {
      "displayName": "ESA",
      "symbol": "ESA"
    },
    "ESB": {
      "displayName": "ESB",
      "symbol": "ESB"
    },
    "ESP": {
      "displayName": "بیزیتا إسبانی",
      "symbol": "ESP",
      "symbol-alt-narrow": "₧"
    },
    "ETB": {
      "displayName": "بیر اثیوبی",
      "displayName-count-zero": "بیر اثیوبی",
      "displayName-count-one": "بیر اثیوبی",
      "displayName-count-two": "بیر اثیوبی",
      "displayName-count-few": "بیر اثیوبی",
      "displayName-count-many": "بیر اثیوبی",
      "displayName-count-other": "بیر اثیوبی",
      "symbol": "ETB"
    },
    "EUR": {
      "displayName": "یورو",
      "displayName-count-zero": "یورو",
      "displayName-count-one": "یورو",
      "displayName-count-two": "یورو",
      "displayName-count-few": "یورو",
      "displayName-count-many": "یورو",
      "displayName-count-other": "یورو",
      "symbol": "€",
      "symbol-alt-narrow": "€"
    },
    "FIM": {
      "displayName": "مارکا فنلندی",
      "symbol": "FIM"
    },
    "FJD": {
      "displayName": "دولار فیجی",
      "displayName-count-zero": "دولار فیجی",
      "displayName-count-one": "دولار فیجی",
      "displayName-count-two": "دولار فیجی",
      "displayName-count-few": "دولار فیجی",
      "displayName-count-many": "دولار فیجی",
      "displayName-count-other": "دولار فیجی",
      "symbol": "FJD",
      "symbol-alt-narrow": "FJ$"
    },
    "FKP": {
      "displayName": "جنيه جزر فوکلاند",
      "displayName-count-zero": "جنيه جزر فوکلاند",
      "displayName-count-one": "جنيه جزر فوکلاند",
      "displayName-count-two": "جنيه جزر فوکلاند",
      "displayName-count-few": "جنيه جزر فوکلاند",
      "displayName-count-many": "جنيه جزر فوکلاند",

```

```

    "displayName-count-other": "جنيه جزر فوكلاند",
    "symbol": "FKP",
    "symbol-alt-narrow": "£"
  },
  "FRF": {
    "displayName": "فرنك فرنسي",
    "symbol": "FRF"
  },
  "GBP": {
    "displayName": "جنيه إسترليني",
    "displayName-count-zero": "جنيه إسترليني",
    "displayName-count-one": "جنيه إسترليني",
    "displayName-count-two": "جنيه إسترليني",
    "displayName-count-few": "جنيه إسترليني",
    "displayName-count-many": "جنيه إسترليني",
    "displayName-count-other": "جنيه إسترليني",
    "symbol": "£",
    "symbol-alt-narrow": "UK£"
  },
  "GEK": {
    "displayName": "GEK",
    "symbol": "GEK"
  },
  "GEL": {
    "displayName": "لاري جورجي",
    "displayName-count-zero": "لاري جورجي",
    "displayName-count-one": "لاري جورجي",
    "displayName-count-two": "لاري جورجي",
    "displayName-count-few": "لاري جورجي",
    "displayName-count-many": "لاري جورجي",
    "displayName-count-other": "لاري جورجي",
    "symbol": "GEL",
    "symbol-alt-narrow": "ლ",
    "symbol-alt-variant": "ლ"
  },
  "GHC": {
    "displayName": "سيدي غاني",
    "symbol": "GHC"
  },
  "GHS": {
    "displayName": "سيدي غانا",
    "displayName-count-zero": "سيدي غانا",
    "displayName-count-one": "سيدي غانا",
    "displayName-count-two": "سيدي غانا",
    "displayName-count-few": "سيدي غانا",
    "displayName-count-many": "سيدي غانا",
    "displayName-count-other": "سيدي غانا",
    "symbol": "GHS"
  },
  "GIP": {
    "displayName": "جنيه جبل طارق",
    "displayName-count-zero": "جنيه جبل طارق",
    "displayName-count-one": "جنيه جبل طارق",
    "displayName-count-two": "جنيه جبل طارق",
    "displayName-count-few": "جنيه جبل طارق",
    "displayName-count-many": "جنيه جبل طارق",
    "displayName-count-other": "جنيه جبل طارق",

```

```

    "symbol": "GIP",
    "symbol-alt-narrow": "£"
  },
  "GMD": {
    "displayName": "دلّاسي جامبي",
    "displayName-count-zero": "دلّاسي جامبي",
    "displayName-count-one": "دلّاسي جامبي",
    "displayName-count-two": "دلّاسي جامبي",
    "displayName-count-few": "دلّاسي جامبي",
    "displayName-count-many": "دلّاسي جامبي",
    "displayName-count-other": "دلّاسي جامبي",
    "symbol": "GMD"
  },
  "GNF": {
    "displayName": "فرنك غينيا",
    "displayName-count-zero": "فرنك غينيا",
    "displayName-count-one": "فرنك غينيا",
    "displayName-count-two": "فرنك غينيا",
    "displayName-count-few": "فرنك غينيا",
    "displayName-count-many": "فرنك غينيا",
    "displayName-count-other": "فرنك غينيا",
    "symbol": "GNF",
    "symbol-alt-narrow": "FG"
  },
  "GNS": {
    "displayName": "سيلي غينيا",
    "symbol": "GNS"
  },
  "GQE": {
    "displayName": "اكويل جونيّنا غينيّا الاستوائيّة",
    "symbol": "GQE"
  },
  "GRD": {
    "displayName": "دراخما يوناني",
    "symbol": "GRD"
  },
  "GTQ": {
    "displayName": "كوتزال جواتيمالا",
    "displayName-count-zero": "كوتزال جواتيمالا",
    "displayName-count-one": "كوتزال جواتيمالا",
    "displayName-count-two": "كوتزال جواتيمالا",
    "displayName-count-few": "كوتزال جواتيمالا",
    "displayName-count-many": "كوتزال جواتيمالا",
    "displayName-count-other": "كوتزال جواتيمالا",
    "symbol": "GTQ",
    "symbol-alt-narrow": "Q"
  },
  "GWE": {
    "displayName": "اسكود برتغالي غينيا",
    "symbol": "GWE"
  },
  "GWP": {
    "displayName": "بيزو غينيا بيساو",
    "symbol": "GWP"
  },
  "GYD": {
    "displayName": "دولار غيانا",

```

```

    "displayName-count-zero": "دولار غيانا",
    "displayName-count-one": "دولار غيانا",
    "displayName-count-two": "دولار غيانا",
    "displayName-count-few": "دولار غيانا",
    "displayName-count-many": "دولار غيانا",
    "displayName-count-other": "دولار غيانا",
    "symbol": "GYD",
    "symbol-alt-narrow": "GY$"
  },
  "HKD": {
    "displayName": "دولار هونغ كونغ",
    "displayName-count-zero": "دولار هونغ كونغ",
    "displayName-count-one": "دولار هونغ كونغ",
    "displayName-count-two": "دولار هونغ كونغ",
    "displayName-count-few": "دولار هونغ كونغ",
    "displayName-count-many": "دولار هونغ كونغ",
    "displayName-count-other": "دولار هونغ كونغ",
    "symbol": "HK$",
    "symbol-alt-narrow": "HK$"
  },
  "HNL": {
    "displayName": "ليمبيرا هندوراس",
    "displayName-count-zero": "ليمبيرا هندوراس",
    "displayName-count-one": "ليمبيرا هندوراس",
    "displayName-count-two": "ليمبيرا هندوراس",
    "displayName-count-few": "ليمبيرا هندوراس",
    "displayName-count-many": "ليمبيرا هندوراس",
    "displayName-count-other": "ليمبيرا هندوراس",
    "symbol": "HNL",
    "symbol-alt-narrow": "L"
  },
  "HRD": {
    "displayName": "دينار كرواتي",
    "symbol": "HRD"
  },
  "HRK": {
    "displayName": "كونا كرواتي",
    "displayName-count-zero": "كونا كرواتي",
    "displayName-count-one": "كونا كرواتي",
    "displayName-count-two": "كونا كرواتي",
    "displayName-count-few": "كونا كرواتي",
    "displayName-count-many": "كونا كرواتي",
    "displayName-count-other": "كونا كرواتي",
    "symbol": "HRK",
    "symbol-alt-narrow": "kn"
  },
  "HTG": {
    "displayName": "جوردي هاييتي",
    "displayName-count-zero": "جوردي هاييتي",
    "displayName-count-one": "جوردي هاييتي",
    "displayName-count-two": "جوردي هاييتي",
    "displayName-count-few": "جوردي هاييتي",
    "displayName-count-many": "جوردي هاييتي",
    "displayName-count-other": "جوردي هاييتي",
    "symbol": "HTG"
  },
  "HUF": {

```

```

    "displayName": "فورينت مجري",
    "displayName-count-zero": "فورينت مجري",
    "displayName-count-one": "فورينت مجري",
    "displayName-count-two": "فورينت مجري",
    "displayName-count-few": "فورينت مجري",
    "displayName-count-many": "فورينت مجري",
    "displayName-count-other": "فورينت مجري",
    "symbol": "HUF",
    "symbol-alt-narrow": "Ft"
  },
  "IDR": {
    "displayName": "روبية إندونيسية",
    "displayName-count-zero": "روبية إندونيسية",
    "displayName-count-one": "روبية إندونيسية",
    "displayName-count-two": "روبية إندونيسية",
    "displayName-count-few": "روبية إندونيسية",
    "displayName-count-many": "روبية إندونيسية",
    "displayName-count-other": "روبية إندونيسية",
    "symbol": "IDR",
    "symbol-alt-narrow": "Rp"
  },
  "IEP": {
    "displayName": "جنيه إيرلندي",
    "symbol": "IEP"
  },
  "ILP": {
    "displayName": "جنيه إسرائيلي",
    "symbol": "ILP"
  },
  "ILS": {
    "displayName": "شيكل إسرائيلي جديد",
    "displayName-count-zero": "شيكل إسرائيلي جديد",
    "displayName-count-one": "شيكل إسرائيلي جديد",
    "displayName-count-two": "شيكل إسرائيلي جديد",
    "displayName-count-few": "شيكل إسرائيلي جديد",
    "displayName-count-many": "شيكل إسرائيلي جديد",
    "displayName-count-other": "شيكل إسرائيلي جديد",
    "symbol": "₪",
    "symbol-alt-narrow": "₪"
  },
  "INR": {
    "displayName": "روبية هندي",
    "displayName-count-zero": "روبية هندي",
    "displayName-count-one": "روبية هندي",
    "displayName-count-two": "روبية هندي",
    "displayName-count-few": "روبية هندي",
    "displayName-count-many": "روبية هندي",
    "displayName-count-other": "روبية هندي",
    "symbol": "₹",
    "symbol-alt-narrow": "₹"
  },
  "IQD": {
    "displayName": "دينار عراقي",
    "displayName-count-zero": "دينار عراقي",
    "displayName-count-one": "دينار عراقي",
    "displayName-count-two": "دينار عراقي",
    "displayName-count-few": "دينار عراقي",

```



```

    "displayName-count-many": "دينار عراقي",
    "displayName-count-other": "دينار عراقي",
    "symbol": "د.ع."
  },
  "IRR": {
    "displayName": "ریال ایرانی",
    "displayName-count-zero": "ریال ایرانی",
    "displayName-count-one": "ریال ایرانی",
    "displayName-count-two": "ریال ایرانی",
    "displayName-count-few": "ریال ایرانی",
    "displayName-count-many": "ریال ایرانی",
    "displayName-count-other": "ریال ایرانی",
    "symbol": "ر.ا."
  },
  "ISK": {
    "displayName": "کرونة ايسلندي",
    "displayName-count-zero": "کرونة ايسلندي",
    "displayName-count-one": "کرونة ايسلندي",
    "displayName-count-two": "کرونة ايسلندي",
    "displayName-count-few": "کرونة ايسلندي",
    "displayName-count-many": "کرونة ايسلندي",
    "displayName-count-other": "کرونة ايسلندي",
    "symbol": "ISK",
    "symbol-alt-narrow": "kr"
  },
  "ITL": {
    "displayName": "ليرة إيطالية",
    "symbol": "ITL"
  },
  "JMD": {
    "displayName": "دولار جامايكي",
    "displayName-count-zero": "دولار جامايكي",
    "displayName-count-one": "دولار جامايكي",
    "displayName-count-two": "دولار جامايكي",
    "displayName-count-few": "دولار جامايكي",
    "displayName-count-many": "دولار جامايكي",
    "displayName-count-other": "دولار جامايكي",
    "symbol": "JMD",
    "symbol-alt-narrow": "JM$"
  },
  "JOD": {
    "displayName": "دينار أردني",
    "displayName-count-zero": "دينار أردني",
    "displayName-count-one": "دينار أردني",
    "displayName-count-two": "دينار أردني",
    "displayName-count-few": "دينار أردني",
    "displayName-count-many": "دينار أردني",
    "displayName-count-other": "دينار أردني",
    "symbol": "د.أ."
  },
  "JPY": {
    "displayName": "ين ياباني",
    "displayName-count-zero": "ين ياباني",
    "displayName-count-one": "ين ياباني",
    "displayName-count-two": "ين ياباني",
    "displayName-count-few": "ين ياباني",
    "displayName-count-many": "ين ياباني",

```

```

    "displayName-count-other": "ين ياباني",
    "symbol": "JP¥",
    "symbol-alt-narrow": "JP¥"
  },
  "KES": {
    "displayName": "شلن كينيي",
    "displayName-count-zero": "شلن كينيي",
    "displayName-count-one": "شلن كينيي",
    "displayName-count-two": "شلن كينيي",
    "displayName-count-few": "شلن كينيي",
    "displayName-count-many": "شلن كينيي",
    "displayName-count-other": "شلن كينيي",
    "symbol": "KES"
  },
  "KGS": {
    "displayName": "سوم قيرغستاني",
    "displayName-count-zero": "سوم قيرغستاني",
    "displayName-count-one": "سوم قيرغستاني",
    "displayName-count-two": "سوم قيرغستاني",
    "displayName-count-few": "سوم قيرغستاني",
    "displayName-count-many": "سوم قيرغستاني",
    "displayName-count-other": "سوم قيرغستاني",
    "symbol": "KGS"
  },
  "KHR": {
    "displayName": "ريال كمبودي",
    "displayName-count-zero": "ريال كمبودي",
    "displayName-count-one": "ريال كمبودي",
    "displayName-count-two": "ريال كمبودي",
    "displayName-count-few": "ريال كمبودي",
    "displayName-count-many": "ريال كمبودي",
    "displayName-count-other": "ريال كمبودي",
    "symbol": "KHR",
    "symbol-alt-narrow": "៛"
  },
  "KMF": {
    "displayName": "فرنك جزر القمر",
    "displayName-count-zero": "فرنك جزر القمر",
    "displayName-count-one": "فرنك جزر القمر",
    "displayName-count-two": "فرنك جزر القمر",
    "displayName-count-few": "فرنك جزر القمر",
    "displayName-count-many": "فرنك جزر القمر",
    "displayName-count-other": "فرنك جزر القمر",
    "symbol": "ف.ق.ق.",
    "symbol-alt-narrow": "CF"
  },
  "KPW": {
    "displayName": "وون كوريا الشمالية",
    "displayName-count-zero": "وون كوريا الشمالية",
    "displayName-count-one": "وون كوريا الشمالية",
    "displayName-count-two": "وون كوريا الشمالية",
    "displayName-count-few": "وون كوريا الشمالية",
    "displayName-count-many": "وون كوريا الشمالية",
    "displayName-count-other": "وون كوريا الشمالية",
    "symbol": "KPW",
    "symbol-alt-narrow": "₩"
  }

```

```

    },
    "KRH": {
      "displayName": "KRH",
      "symbol": "KRH"
    },
    },
    "KRO": {
      "displayName": "KRO",
      "symbol": "KRO"
    },
    },
    "KRW": {
      "displayName": "₩",
      "displayName-count-zero": "₩",
      "displayName-count-one": "₩",
      "displayName-count-two": "₩",
      "displayName-count-few": "₩",
      "displayName-count-many": "₩",
      "displayName-count-other": "₩",
      "symbol": "₩",
      "symbol-alt-narrow": "₩"
    },
    },
    "KWD": {
      "displayName": "دينار كويتي",
      "displayName-count-zero": "دينار كويتي",
      "displayName-count-one": "دينار كويتي",
      "displayName-count-two": "دينار كويتي",
      "displayName-count-few": "دينار كويتي",
      "displayName-count-many": "دينار كويتي",
      "displayName-count-other": "دينار كويتي",
      "symbol": "د.ك."
    },
    },
    "KYD": {
      "displayName": "دولار جزر كيمن",
      "displayName-count-zero": "دولار جزر كيمن",
      "displayName-count-one": "دولار جزر كيمن",
      "displayName-count-two": "دولار جزر كيمن",
      "displayName-count-few": "دولار جزر كيمن",
      "displayName-count-many": "دولار جزر كيمن",
      "displayName-count-other": "دولار جزر كيمن",
      "symbol": "KYD",
      "symbol-alt-narrow": "KY$"
    },
    },
    "KZT": {
      "displayName": "تينغ كازاخستاني",
      "displayName-count-zero": "تينغ كازاخستاني",
      "displayName-count-one": "تينغ كازاخستاني",
      "displayName-count-two": "تينغ كازاخستاني",
      "displayName-count-few": "تينغ كازاخستاني",
      "displayName-count-many": "تينغ كازاخستاني",
      "displayName-count-other": "تينغ كازاخستاني",
      "symbol": "KZT",
      "symbol-alt-narrow": "Т"
    },
    },
    "LAK": {
      "displayName": "كيب لاوسي",
      "displayName-count-zero": "كيب لاوسي",
      "displayName-count-one": "كيب لاوسي",
      "displayName-count-two": "كيب لاوسي",

```

```

    "displayName-count-few": "كيب لاوسي",
    "displayName-count-many": "كيب لاوسي",
    "displayName-count-other": "كيب لاوسي",
    "symbol": "LAK",
    "symbol-alt-narrow": "₭"
  },
  "LBP": {
    "displayName": "جنيه لبناني",
    "displayName-count-zero": "جنيه لبناني",
    "displayName-count-one": "جنيه لبناني",
    "displayName-count-two": "جنيه لبناني",
    "displayName-count-few": "جنيه لبناني",
    "displayName-count-many": "جنيه لبناني",
    "displayName-count-other": "جنيه لبناني",
    "symbol": "ل.ل.",
    "symbol-alt-narrow": "L£"
  },
  "LKR": {
    "displayName": "روبية سريلانكية",
    "displayName-count-zero": "روبية سريلانكية",
    "displayName-count-one": "روبية سريلانكية",
    "displayName-count-two": "روبية سريلانكية",
    "displayName-count-few": "روبية سريلانكية",
    "displayName-count-many": "روبية سريلانكية",
    "displayName-count-other": "روبية سريلانكية",
    "symbol": "LKR",
    "symbol-alt-narrow": "Rs"
  },
  "LRD": {
    "displayName": "دولار ليبيري",
    "displayName-count-zero": "دولار ليبيري",
    "displayName-count-one": "دولار ليبيري",
    "displayName-count-two": "دولاران ليبيريان",
    "displayName-count-few": "دولارات ليبيرية",
    "displayName-count-many": "دولارًا ليبيريًا",
    "displayName-count-other": "دولار ليبيري",
    "symbol": "LRD",
    "symbol-alt-narrow": "$"
  },
  "LSL": {
    "displayName": "لوتي ليسوتو",
    "symbol": "LSL"
  },
  "LTL": {
    "displayName": "ليتة ليتوانية",
    "displayName-count-zero": "ليتة ليتوانية",
    "displayName-count-one": "ليتة ليتوانية",
    "displayName-count-two": "ليتة ليتوانية",
    "displayName-count-few": "ليتة ليتوانية",
    "displayName-count-many": "ليتة ليتوانية",
    "displayName-count-other": "ليتة ليتوانية",
    "symbol": "LTL",
    "symbol-alt-narrow": "Lt"
  },
  "LTT": {
    "displayName": "تالوناس ليتواني",
    "symbol": "LTT"
  }

```

```

    },
    "LUC": {
      "displayName": "فرنك لوكسمبرج قابل للتحويل",
      "symbol": "LUC"
    },
    "LUF": {
      "displayName": "فرنك لوكسمبرج",
      "symbol": "LUF"
    },
    "LUL": {
      "displayName": "فرنك لوكسمبرج المالي",
      "symbol": "LUL"
    },
    "LVL": {
      "displayName": "لاتس لاتفيا",
      "displayName-count-zero": "لاتس لاتفي",
      "displayName-count-one": "لاتس لاتفي",
      "displayName-count-two": "لاتس لاتفي",
      "displayName-count-few": "لاتس لاتفي",
      "displayName-count-many": "لاتس لاتفي",
      "displayName-count-other": "لاتس لاتفي",
      "symbol": "LVL",
      "symbol-alt-narrow": "Ls"
    },
    "LVR": {
      "displayName": "روبل لاتفيا",
      "symbol": "LVR"
    },
    "LYD": {
      "displayName": "دينار ليبي",
      "displayName-count-zero": "دينار ليبي",
      "displayName-count-one": "دينار ليبي",
      "displayName-count-two": "ديناران ليبيان",
      "displayName-count-few": "دينارات ليبية",
      "displayName-count-many": "دينارًا ليبيا",
      "displayName-count-other": "دينار ليبي",
      "symbol": "د.ل."
    },
    "MAD": {
      "displayName": "درهم مغربي",
      "displayName-count-zero": "درهم مغربي",
      "displayName-count-one": "درهم مغربي",
      "displayName-count-two": "درهمان مغربيان",
      "displayName-count-few": "دراهم مغربية",
      "displayName-count-many": "درهمًا مغربيًا",
      "displayName-count-other": "درهم مغربي",
      "symbol": "د.م."
    },
    "MAF": {
      "displayName": "فرنك مغربي",
      "symbol": "MAF"
    },
    "MCF": {
      "displayName": "MCF",
      "symbol": "MCF"
    },
    "MDC": {

```

```

    "displayName": "MDC",
    "symbol": "MDC"
  },
  "MDL": {
    "displayName": "ليو مولدوفي",
    "displayName-count-zero": "ليو مولدوفي",
    "displayName-count-one": "ليو مولدوفي",
    "displayName-count-two": "ليو مولدوفي",
    "displayName-count-few": "ليو مولدوفي",
    "displayName-count-many": "ليو مولدوفي",
    "displayName-count-other": "ليو مولدوفي",
    "symbol": "MDL"
  },
  "MGA": {
    "displayName": "أرياري مدغشقر",
    "displayName-count-zero": "أرياري مدغشقر",
    "displayName-count-one": "أرياري مدغشقر",
    "displayName-count-two": "أرياري مدغشقر",
    "displayName-count-few": "أرياري مدغشقر",
    "displayName-count-many": "أرياري مدغشقر",
    "displayName-count-other": "أرياري مدغشقر",
    "symbol": "MGA",
    "symbol-alt-narrow": "Ar"
  },
  "MGF": {
    "displayName": "فرنك مدغشقر",
    "symbol": "MGF"
  },
  "MKD": {
    "displayName": "دينار مقدوني",
    "displayName-count-zero": "دينار مقدوني",
    "displayName-count-one": "دينار مقدوني",
    "displayName-count-two": "ديناران مقدونيان",
    "displayName-count-few": "دينارات مقدونية",
    "displayName-count-many": "دينارًا مقدونيًا",
    "displayName-count-other": "دينار مقدوني",
    "symbol": "MKD"
  },
  "MKN": {
    "displayName": "MKN",
    "symbol": "MKN"
  },
  "MLF": {
    "displayName": "فرنك مالي",
    "symbol": "MLF"
  },
  "MMK": {
    "displayName": "كيات ميانمار",
    "displayName-count-zero": "كيات ميانمار",
    "displayName-count-one": "كيات ميانمار",
    "displayName-count-two": "كيات ميانمار",
    "displayName-count-few": "كيات ميانمار",
    "displayName-count-many": "كيات ميانمار",
    "displayName-count-other": "كيات ميانمار",
    "symbol": "MMK",
    "symbol-alt-narrow": "K"
  },
},

```

```

"MNT": {
  "displayName": "توغروغ منغولي",
  "displayName-count-zero": "توغروغ منغولي",
  "displayName-count-one": "توغروغ منغولي",
  "displayName-count-two": "توغروغ منغولي",
  "displayName-count-few": "توغروغ منغولي",
  "displayName-count-many": "توغروغ منغولي",
  "displayName-count-other": "توغروغ منغولي",
  "symbol": "MNT",
  "symbol-alt-narrow": "₮"
},
"MOP": {
  "displayName": "باتاكا مكاوي",
  "displayName-count-zero": "باتاكا مكاوي",
  "displayName-count-one": "باتاكا مكاوي",
  "displayName-count-two": "باتاكا مكاوي",
  "displayName-count-few": "باتاكا مكاوي",
  "displayName-count-many": "باتاكا مكاوي",
  "displayName-count-other": "باتاكا مكاوي",
  "symbol": "MOP"
},
"MRO": {
  "displayName": "أوقية موريتانية",
  "displayName-count-zero": "أوقية موريتانية",
  "displayName-count-one": "أوقية موريتانية",
  "displayName-count-two": "أوقية موريتانية",
  "displayName-count-few": "أوقية موريتانية",
  "displayName-count-many": "أوقية موريتانية",
  "displayName-count-other": "أوقية موريتانية",
  "symbol": "أ.م."
},
"MTL": {
  "displayName": "ليرة مالطية",
  "symbol": "MTL"
},
"MTP": {
  "displayName": "جنيه مالطي",
  "symbol": "MTP"
},
"MUR": {
  "displayName": "روبية موريشيوسية",
  "displayName-count-zero": "روبية موريشيوسية",
  "displayName-count-one": "روبية موريشيوسية",
  "displayName-count-two": "روبية موريشيوسية",
  "displayName-count-few": "روبية موريشيوسية",
  "displayName-count-many": "روبية موريشيوسية",
  "displayName-count-other": "روبية موريشيوسية",
  "symbol": "MUR",
  "symbol-alt-narrow": "Rs"
},
"MVR": {
  "displayName": "روفيه جزر المالديف",
  "displayName-count-zero": "روفيه جزر المالديف",
  "displayName-count-one": "روفيه جزر المالديف",
  "displayName-count-two": "روفيه جزر المالديف",
  "displayName-count-few": "روفيه جزر المالديف",
  "displayName-count-many": "روفيه جزر المالديف",

```

```

    "displayName-count-other": "روفيه جزر المالديف",
    "symbol": "MVR"
  },
  "MWK": {
    "displayName": "كواشا مالاوي",
    "displayName-count-zero": "كواشا مالاوي",
    "displayName-count-one": "كواشا مالاوي",
    "displayName-count-two": "كواشا مالاوي",
    "displayName-count-few": "كواشا مالاوي",
    "displayName-count-many": "كواشا مالاوي",
    "displayName-count-other": "كواشا مالاوي",
    "symbol": "MWK"
  },
  "MXN": {
    "displayName": "بيزو مكسيكي",
    "displayName-count-zero": "بيزو مكسيكي",
    "displayName-count-one": "بيزو مكسيكي",
    "displayName-count-two": "بيزو مكسيكي",
    "displayName-count-few": "بيزو مكسيكي",
    "displayName-count-many": "بيزو مكسيكي",
    "displayName-count-other": "بيزو مكسيكي",
    "symbol": "MX$",
    "symbol-alt-narrow": "MX$"
  },
  "MXP": {
    "displayName": "بيزو فضي مكسيكي - 1992-1861",
    "symbol": "MXP"
  },
  "MXV": {
    "displayName": "MXV",
    "symbol": "MXV"
  },
  "MYR": {
    "displayName": "رينغيت ماليزي",
    "displayName-count-zero": "رينغيت ماليزي",
    "displayName-count-one": "رينغيت ماليزي",
    "displayName-count-two": "رينغيت ماليزي",
    "displayName-count-few": "رينغيت ماليزي",
    "displayName-count-many": "رينغيت ماليزي",
    "displayName-count-other": "رينغيت ماليزي",
    "symbol": "MYR",
    "symbol-alt-narrow": "RM"
  },
  "MZE": {
    "displayName": "اسكود موزمبيقي",
    "symbol": "MZE"
  },
  "MZM": {
    "displayName": "MZM",
    "symbol": "MZM"
  },
  "MZN": {
    "displayName": "متكال موزمبيقي",
    "displayName-count-zero": "متكال موزمبيقي",
    "displayName-count-one": "متكال موزمبيقي",
    "displayName-count-two": "متكال موزمبيقي",
    "displayName-count-few": "متكال موزمبيقي",

```



```

    "displayName-count-many": "متكال موزمبيقى",
    "displayName-count-other": "متكال موزمبيقى",
    "symbol": "MZN"
  },
  "NAD": {
    "displayName": "دولار ناميبى",
    "displayName-count-zero": "دولار ناميبى",
    "displayName-count-one": "دولار ناميبى",
    "displayName-count-two": "دولار ناميبى",
    "displayName-count-few": "دولار ناميبى",
    "displayName-count-many": "دولار ناميبى",
    "displayName-count-other": "دولار ناميبى",
    "symbol": "NAD",
    "symbol-alt-narrow": "$"
  },
  "NGN": {
    "displayName": "نايرا نيجيرى",
    "displayName-count-zero": "نايرا نيجيرى",
    "displayName-count-one": "نايرا نيجيرى",
    "displayName-count-two": "نايرا نيجيرى",
    "displayName-count-few": "نايرا نيجيرى",
    "displayName-count-many": "نايرا نيجيرى",
    "displayName-count-other": "نايرا نيجيرى",
    "symbol": "NGN",
    "symbol-alt-narrow": "₦"
  },
  "NIC": {
    "displayName": "كوردوبه نيكاراغوا",
    "symbol": "NIC"
  },
  "NIO": {
    "displayName": "قرطبة نيكاراغوا",
    "displayName-count-zero": "قرطبة نيكاراغوا",
    "displayName-count-one": "قرطبة نيكاراغوا",
    "displayName-count-two": "قرطبة نيكاراغوا",
    "displayName-count-few": "قرطبة نيكاراغوا",
    "displayName-count-many": "قرطبة نيكاراغوا",
    "displayName-count-other": "قرطبة نيكاراغوا",
    "symbol": "NIO",
    "symbol-alt-narrow": "C$"
  },
  "NLG": {
    "displayName": "جلدر هولندي",
    "symbol": "NLG"
  },
  "NOK": {
    "displayName": "كرونه نرويجية",
    "displayName-count-zero": "كرونه نرويجية",
    "displayName-count-one": "كرونه نرويجية",
    "displayName-count-two": "كرونه نرويجية",
    "displayName-count-few": "كرونه نرويجية",
    "displayName-count-many": "كرونه نرويجية",
    "displayName-count-other": "كرونه نرويجية",
    "symbol": "NOK",
    "symbol-alt-narrow": "kr"
  },
  "NPR": {

```

```

    "displayName": "روبية نيپالي",
    "displayName-count-zero": "روبية نيپالي",
    "displayName-count-one": "روبية نيپالي",
    "displayName-count-two": "روبية نيپالي",
    "displayName-count-few": "روبية نيپالي",
    "displayName-count-many": "روبية نيپالي",
    "displayName-count-other": "روبية نيپالي",
    "symbol": "NPR",
    "symbol-alt-narrow": "Rs"
  },
  "NZD": {
    "displayName": "دولار نيوزيلندي",
    "displayName-count-zero": "دولار نيوزيلندي",
    "displayName-count-one": "دولار نيوزيلندي",
    "displayName-count-two": "دولار نيوزيلندي",
    "displayName-count-few": "دولار نيوزيلندي",
    "displayName-count-many": "دولار نيوزيلندي",
    "displayName-count-other": "دولار نيوزيلندي",
    "symbol": "NZ$",
    "symbol-alt-narrow": "NZ$"
  },
  "OMR": {
    "displayName": "رول عماني",
    "displayName-count-zero": "رول عماني",
    "displayName-count-one": "رول عماني",
    "displayName-count-two": "رول عماني",
    "displayName-count-few": "رول عماني",
    "displayName-count-many": "رول عماني",
    "displayName-count-other": "رول عماني",
    "symbol": "ر.ع."
  },
  "PAB": {
    "displayName": "بالبوا بنمي",
    "displayName-count-zero": "بالبوا بنمي",
    "displayName-count-one": "بالبوا بنمي",
    "displayName-count-two": "بالبوا بنمي",
    "displayName-count-few": "بالبوا بنمي",
    "displayName-count-many": "بالبوا بنمي",
    "displayName-count-other": "بالبوا بنمي",
    "symbol": "PAB"
  },
  "PEI": {
    "displayName": "PEI",
    "symbol": "PEI"
  },
  "PEN": {
    "displayName": "سول جديد البيرو",
    "displayName-count-zero": "سول جديد البيرو",
    "displayName-count-one": "سول جديد البيرو",
    "displayName-count-two": "سول جديد البيرو",
    "displayName-count-few": "سول جديد البيرو",
    "displayName-count-many": "سول جديد البيرو",
    "displayName-count-other": "سول جديد البيرو",
    "symbol": "PEN"
  },
  "PES": {
    "displayName": "PES",

```

```

    "symbol": "PES"
  },
  "PGK": {
    "displayName": "كيننا بابوا غينيا الجديدة",
    "displayName-count-zero": "كيننا بابوا غينيا الجديدة",
    "displayName-count-one": "كيننا بابوا غينيا الجديدة",
    "displayName-count-two": "كيننا بابوا غينيا الجديدة",
    "displayName-count-few": "كيننا بابوا غينيا الجديدة",
    "displayName-count-many": "كيننا بابوا غينيا الجديدة",
    "displayName-count-other": "كيننا بابوا غينيا الجديدة",
    "symbol": "PGK"
  },
  "PHP": {
    "displayName": "بيزو فلبيني",
    "displayName-count-zero": "بيزو فلبيني",
    "displayName-count-one": "بيزو فلبيني",
    "displayName-count-two": "بيزو فلبيني",
    "displayName-count-few": "بيزو فلبيني",
    "displayName-count-many": "بيزو فلبيني",
    "displayName-count-other": "بيزو فلبيني",
    "symbol": "PHP",
    "symbol-alt-narrow": "₱"
  },
  "PKR": {
    "displayName": "روبية باكستاني",
    "displayName-count-zero": "روبية باكستاني",
    "displayName-count-one": "روبية باكستاني",
    "displayName-count-two": "روبية باكستاني",
    "displayName-count-few": "روبية باكستاني",
    "displayName-count-many": "روبية باكستاني",
    "displayName-count-other": "روبية باكستاني",
    "symbol": "PKR",
    "symbol-alt-narrow": "Rs"
  },
  "PLN": {
    "displayName": "زلوتي بولندي",
    "displayName-count-zero": "زلوتي بولندي",
    "displayName-count-one": "زلوتي بولندي",
    "displayName-count-two": "زلوتي بولندي",
    "displayName-count-few": "زلوتي بولندي",
    "displayName-count-many": "زلوتي بولندي",
    "displayName-count-other": "زلوتي بولندي",
    "symbol": "PLN",
    "symbol-alt-narrow": "zł"
  },
  "PLZ": {
    "displayName": "زلوتي بولندي - 1995-1950",
    "symbol": "PLZ"
  },
  "PTE": {
    "displayName": "اسكود برتغالي",
    "symbol": "PTE"
  },
  "PYG": {
    "displayName": "جواراني باراجواي",
    "displayName-count-zero": "جواراني باراجواي",
    "displayName-count-one": "جواراني باراجواي",

```

```

    "displayName-count-two": "جواراني باراجوای",
    "displayName-count-few": "جواراني باراجوای",
    "displayName-count-many": "جواراني باراجوای",
    "displayName-count-other": "جواراني باراجوای",
    "symbol": "PYG",
    "symbol-alt-narrow": "₲"
  },
  "QAR": {
    "displayName": "ڤل قطري",
    "displayName-count-zero": "ڤل قطري",
    "displayName-count-one": "ڤل قطري",
    "displayName-count-two": "ڤل قطري",
    "displayName-count-few": "ڤل قطري",
    "displayName-count-many": "ڤل قطري",
    "displayName-count-other": "ڤل قطري",
    "symbol": "ر.ق."
  },
  "RHD": {
    "displayName": "دولار روديسي",
    "symbol": "RHD"
  },
  "ROL": {
    "displayName": "ليو روماني قديم",
    "symbol": "ROL"
  },
  "RON": {
    "displayName": "ليو روماني",
    "displayName-count-zero": "ليو روماني",
    "displayName-count-one": "ليو روماني",
    "displayName-count-two": "ليو روماني",
    "displayName-count-few": "ليو روماني",
    "displayName-count-many": "ليو روماني",
    "displayName-count-other": "ليو روماني",
    "symbol": "RON",
    "symbol-alt-narrow": "lei"
  },
  "RSD": {
    "displayName": "دينار صربي",
    "displayName-count-zero": "دينار صربي",
    "displayName-count-one": "دينار صربي",
    "displayName-count-two": "ديناران صربيان",
    "displayName-count-few": "دينارات صربية",
    "displayName-count-many": "دينارًا صربيًا",
    "displayName-count-other": "دينار صربي",
    "symbol": "RSD"
  },
  "RUB": {
    "displayName": "روبل روسي",
    "displayName-count-zero": "روبل روسي",
    "displayName-count-one": "روبل روسي",
    "displayName-count-two": "روبل روسي",
    "displayName-count-few": "روبل روسي",
    "displayName-count-many": "روبل روسي",
    "displayName-count-other": "روبل روسي",
    "symbol": "RUB",
    "symbol-alt-narrow": "₽"
  },

```

```

"RUR": {
  "displayName": "1998-1991 - روبل روسي",
  "symbol": "RUR",
  "symbol-alt-narrow": "p."
},
"RWF": {
  "displayName": "فرنك رواندي",
  "displayName-count-zero": "فرنك رواندي",
  "displayName-count-one": "فرنك رواندي",
  "displayName-count-two": "فرنك رواندي",
  "displayName-count-few": "فرنك رواندي",
  "displayName-count-many": "فرنك رواندي",
  "displayName-count-other": "فرنك رواندي",
  "symbol": "RWF",
  "symbol-alt-narrow": "RF"
},
"SAR": {
  "displayName": "ريال سعودي",
  "displayName-count-zero": "ريال سعودي",
  "displayName-count-one": "ريال سعودي",
  "displayName-count-two": "ريال سعودي",
  "displayName-count-few": "ريال سعودي",
  "displayName-count-many": "ريال سعودي",
  "displayName-count-other": "ريال سعودي",
  "symbol": "ر.س."
},
"SBD": {
  "displayName": "دولار جزر سليمان",
  "displayName-count-zero": "دولار جزر سليمان",
  "displayName-count-one": "دولار جزر سليمان",
  "displayName-count-two": "دولار جزر سليمان",
  "displayName-count-few": "دولار جزر سليمان",
  "displayName-count-many": "دولار جزر سليمان",
  "displayName-count-other": "دولار جزر سليمان",
  "symbol": "SBD",
  "symbol-alt-narrow": "SB$"
},
"SCR": {
  "displayName": "روبية سيشيلية",
  "displayName-count-zero": "روبية سيشيلية",
  "displayName-count-one": "روبية سيشيلية",
  "displayName-count-two": "روبية سيشيلية",
  "displayName-count-few": "روبية سيشيلية",
  "displayName-count-many": "روبية سيشيلية",
  "displayName-count-other": "روبية سيشيلية",
  "symbol": "SCR"
},
"SDD": {
  "displayName": "دينار سوداني",
  "symbol": "د.س."
},
"SDG": {
  "displayName": "جنيه سوداني",
  "displayName-count-zero": "جنيه سوداني",
  "displayName-count-one": "جنيه سوداني",
  "displayName-count-two": "جنيه سوداني",
  "displayName-count-few": "جنيهات سودانية",

```

```

    "displayName-count-many": "جنيهاً سودانيًا",
    "displayName-count-other": "جنيه سوداني",
    "symbol": "ج.س."
  },
  "SDP": {
    "displayName": "جنيه سوداني قديم",
    "symbol": "SDP"
  },
  "SEK": {
    "displayName": "كرونة سويدية",
    "displayName-count-zero": "كرونة سويدية",
    "displayName-count-one": "كرونة سويدية",
    "displayName-count-two": "كرونة سويدية",
    "displayName-count-few": "كرونة سويدية",
    "displayName-count-many": "كرونة سويدية",
    "displayName-count-other": "كرونة سويدية",
    "symbol": "SEK",
    "symbol-alt-narrow": "kr"
  },
  "SGD": {
    "displayName": "دولار سنغافوري",
    "displayName-count-zero": "دولار سنغافوري",
    "displayName-count-one": "دولار سنغافوري",
    "displayName-count-two": "دولار سنغافوري",
    "displayName-count-few": "دولار سنغافوري",
    "displayName-count-many": "دولار سنغافوري",
    "displayName-count-other": "دولار سنغافوري",
    "symbol": "SGD",
    "symbol-alt-narrow": "$"
  },
  "SHP": {
    "displayName": "جنيه سانت هيلين",
    "displayName-count-zero": "جنيه سانت هيلين",
    "displayName-count-one": "جنيه سانت هيلين",
    "displayName-count-two": "جنيه سانت هيلين",
    "displayName-count-few": "جنيه سانت هيلين",
    "displayName-count-many": "جنيه سانت هيلين",
    "displayName-count-other": "جنيه سانت هيلين",
    "symbol": "SHP",
    "symbol-alt-narrow": "£"
  },
  "SIT": {
    "displayName": "تولار سلوفيني",
    "symbol": "SIT"
  },
  "SKK": {
    "displayName": "كرونة سلوفاكية",
    "symbol": "SKK"
  },
  "SLL": {
    "displayName": "ليون سيراليوني",
    "displayName-count-zero": "ليون سيراليوني",
    "displayName-count-one": "ليون سيراليوني",
    "displayName-count-two": "ليون سيراليوني",
    "displayName-count-few": "ليون سيراليوني",
    "displayName-count-many": "ليون سيراليوني",
    "displayName-count-other": "ليون سيراليوني",

```

```

    "symbol": "SLL"
  },
  "SOS": {
    "displayName": "شلن صومالي",
    "displayName-count-zero": "شلن صومالي",
    "displayName-count-one": "شلن صومالي",
    "displayName-count-two": "شلن صومالي",
    "displayName-count-few": "شلن صومالي",
    "displayName-count-many": "شلن صومالي",
    "displayName-count-other": "شلن صومالي",
    "symbol": "SOS"
  },
  "SRD": {
    "displayName": "دولار سورينامي",
    "displayName-count-zero": "دولار سورينامي",
    "displayName-count-one": "دولار سورينامي",
    "displayName-count-two": "دولار سورينامي",
    "displayName-count-few": "دولار سورينامي",
    "displayName-count-many": "دولار سورينامي",
    "displayName-count-other": "دولار سورينامي",
    "symbol": "SRD",
    "symbol-alt-narrow": "SR$"
  },
  "SRG": {
    "displayName": "جلدر سورينامي",
    "symbol": "SRG"
  },
  "SSP": {
    "displayName": "جنيه جنوب السودان",
    "displayName-count-zero": "جنيه جنوب السودان",
    "displayName-count-one": "جنيه جنوب السودان",
    "displayName-count-two": "جنيهان جنوب السودان",
    "displayName-count-few": "جنيهات جنوب السودان",
    "displayName-count-many": "جنيها جنوب السودان",
    "displayName-count-other": "جنيه جنوب السودان",
    "symbol": "SSP",
    "symbol-alt-narrow": "£"
  },
  "STD": {
    "displayName": "دوبرا ساو تومي وبرينسيبي",
    "displayName-count-zero": "دوبرا ساو تومي وبرينسيبي",
    "displayName-count-one": "دوبرا ساو تومي وبرينسيبي",
    "displayName-count-two": "دوبرا ساو تومي وبرينسيبي",
    "displayName-count-few": "دوبرا ساو تومي وبرينسيبي",
    "displayName-count-many": "دوبرا ساو تومي وبرينسيبي",
    "displayName-count-other": "دوبرا ساو تومي وبرينسيبي",
    "symbol": "STD",
    "symbol-alt-narrow": "Db"
  },
  "SUR": {
    "displayName": "روبل سوفيتي",
    "symbol": "SUR"
  },
  "SVC": {
    "displayName": "كولون سلفادوري",
    "symbol": "SVC"
  },
}

```

```

"SYP": {
  "displayName": "ليرة سورية",
  "displayName-count-zero": "ليرة سورية",
  "displayName-count-one": "ليرة سورية",
  "displayName-count-two": "ليرة سورية",
  "displayName-count-few": "ليرة سورية",
  "displayName-count-many": "ليرة سورية",
  "displayName-count-other": "ليرة سورية",
  "symbol": "ل.س.",
  "symbol-alt-narrow": "£"
},
"SZL": {
  "displayName": "ليلانجيني سوازيلندي",
  "displayName-count-zero": "ليلانجيني سوازيلندي",
  "displayName-count-one": "ليلانجيني سوازيلندي",
  "displayName-count-two": "ليلانجيني سوازيلندي",
  "displayName-count-few": "ليلانجيني سوازيلندي",
  "displayName-count-many": "ليلانجيني سوازيلندي",
  "displayName-count-other": "ليلانجيني سوازيلندي",
  "symbol": "SZL"
},
"THB": {
  "displayName": "باخت تايلاندي",
  "displayName-count-zero": "باخت تايلاندي",
  "displayName-count-one": "باخت تايلاندي",
  "displayName-count-two": "باخت تايلاندي",
  "displayName-count-few": "باخت تايلاندي",
  "displayName-count-many": "باخت تايلاندي",
  "displayName-count-other": "باخت تايلاندي",
  "symbol": "฿",
  "symbol-alt-narrow": "฿"
},
"TJR": {
  "displayName": "روبل طاجيكستاني",
  "symbol": "TJR"
},
"TJS": {
  "displayName": "سوموني طاجيكستاني",
  "displayName-count-zero": "سوموني طاجيكستاني",
  "displayName-count-one": "سوموني طاجيكستاني",
  "displayName-count-two": "سوموني طاجيكستاني",
  "displayName-count-few": "سوموني طاجيكستاني",
  "displayName-count-many": "سوموني طاجيكستاني",
  "displayName-count-other": "سوموني طاجيكستاني",
  "symbol": "TJS"
},
"TMM": {
  "displayName": "مانات تركمنستاني",
  "symbol": "TMM"
},
"TMT": {
  "displayName": "مانات تركمانستان",
  "displayName-count-zero": "مانات تركمانستان",
  "displayName-count-one": "مانات تركمانستان",
  "displayName-count-two": "مانات تركمانستان",
  "displayName-count-few": "مانات تركمانستان",
  "displayName-count-many": "مانات تركمانستان",

```



```

    "displayName-count-other": "مانات تركمانستان",
    "symbol": "TMT"
  },
  "TND": {
    "displayName": "دينار تونسي",
    "displayName-count-zero": "دينار تونسي",
    "displayName-count-one": "دينار تونسي",
    "displayName-count-two": "ديناران تونسيان",
    "displayName-count-few": "دينارات تونسية",
    "displayName-count-many": "دينارًا تونسيًا",
    "displayName-count-other": "دينار تونسي",
    "symbol": "د.ت."
  },
  "TOP": {
    "displayName": "بانغا تونغا",
    "displayName-count-zero": "بانغا تونغا",
    "displayName-count-one": "بانغا تونغا",
    "displayName-count-two": "بانغا تونغا",
    "displayName-count-few": "بانغا تونغا",
    "displayName-count-many": "بانغا تونغا",
    "displayName-count-other": "بانغا تونغا",
    "symbol": "TOP",
    "symbol-alt-narrow": "T$"
  },
  "TPE": {
    "displayName": "اسكود تيموري",
    "symbol": "TPE"
  },
  "TRL": {
    "displayName": "ليرة تركي",
    "symbol": "TRL"
  },
  "TRY": {
    "displayName": "ليرة تركية",
    "displayName-count-zero": "ليرة تركية",
    "displayName-count-one": "ليرة تركية",
    "displayName-count-two": "ليرة تركية",
    "displayName-count-few": "ليرة تركية",
    "displayName-count-many": "ليرة تركية",
    "displayName-count-other": "ليرة تركية",
    "symbol": "TRY",
    "symbol-alt-narrow": "₺",
    "symbol-alt-variant": "TL"
  },
  "TTD": {
    "displayName": "دولار ترينداد وتوباغو",
    "displayName-count-zero": "دولار ترينداد وتوباغو",
    "displayName-count-one": "دولار ترينداد وتوباغو",
    "displayName-count-two": "دولار ترينداد وتوباغو",
    "displayName-count-few": "دولار ترينداد وتوباغو",
    "displayName-count-many": "دولار ترينداد وتوباغو",
    "displayName-count-other": "دولار ترينداد وتوباغو",
    "symbol": "TTD",
    "symbol-alt-narrow": "TT$"
  },
  "TWD": {
    "displayName": "دولار تايواني",

```

```

    "displayName-count-zero": "دولار تايواني",
    "displayName-count-one": "دولار تايواني",
    "displayName-count-two": "دولار تايواني",
    "displayName-count-few": "دولار تايواني",
    "displayName-count-many": "دولار تايواني",
    "displayName-count-other": "دولار تايواني",
    "symbol": "NT$",
    "symbol-alt-narrow": "NT$"
  },
  "TZS": {
    "displayName": "شلن تنزاني",
    "displayName-count-zero": "شلن تنزاني",
    "displayName-count-one": "شلن تنزاني",
    "displayName-count-two": "شلن تنزاني",
    "displayName-count-few": "شلن تنزاني",
    "displayName-count-many": "شلن تنزاني",
    "displayName-count-other": "شلن تنزاني",
    "symbol": "TZS"
  },
  "UAH": {
    "displayName": "هريفنيا اوكراني",
    "displayName-count-zero": "هريفنيا اوكراني",
    "displayName-count-one": "هريفنيا اوكراني",
    "displayName-count-two": "هريفنيا اوكراني",
    "displayName-count-few": "هريفنيا اوكراني",
    "displayName-count-many": "هريفنيا اوكراني",
    "displayName-count-other": "هريفنيا اوكراني",
    "symbol": "UAH",
    "symbol-alt-narrow": "₴"
  },
  "UAK": {
    "displayName": "UAK",
    "symbol": "UAK"
  },
  "UGS": {
    "displayName": "شلن اوغندي - 1966-1987",
    "symbol": "UGS"
  },
  "UGX": {
    "displayName": "شلن اوغندي",
    "displayName-count-zero": "شلن اوغندي",
    "displayName-count-one": "شلن اوغندي",
    "displayName-count-two": "شلن اوغندي",
    "displayName-count-few": "شلن اوغندي",
    "displayName-count-many": "شلن اوغندي",
    "displayName-count-other": "شلن اوغندي",
    "symbol": "UGX"
  },
  "USD": {
    "displayName": "دولار امريكي",
    "displayName-count-zero": "دولار امريكي",
    "displayName-count-one": "دولار امريكي",
    "displayName-count-two": "دولار امريكي",
    "displayName-count-few": "دولار امريكي",
    "displayName-count-many": "دولار امريكي",
    "displayName-count-other": "دولار امريكي",
    "symbol": "US$",

```

```

    "symbol-alt-narrow": "US$"
  },
  "USN": {
    "displayName": "(دولار أمريكي) اليوم التالي",
    "symbol": "USN"
  },
  "USS": {
    "displayName": "(دولار أمريكي) نفس اليوم",
    "symbol": "USS"
  },
  "UYI": {
    "displayName": "UYI",
    "symbol": "UYI"
  },
  "UYU": {
    "displayName": "بيزو أوروجواي - 1993-1975",
    "symbol": "UYU"
  },
  "UYU": {
    "displayName": "بيزو أوروجواي",
    "displayName-count-zero": "بيزو أوروجواي",
    "displayName-count-one": "بيزو أوروجواي",
    "displayName-count-two": "بيزو أوروجواي",
    "displayName-count-few": "بيزو أوروجواي",
    "displayName-count-many": "بيزو أوروجواي",
    "displayName-count-other": "بيزو أوروجواي",
    "symbol": "UYU",
    "symbol-alt-narrow": "UY$"
  },
  "UZS": {
    "displayName": "سوم أوزبكستاني",
    "displayName-count-zero": "سوم أوزبكستاني",
    "displayName-count-one": "سوم أوزبكستاني",
    "displayName-count-two": "سوم أوزبكستاني",
    "displayName-count-few": "سوم أوزبكستاني",
    "displayName-count-many": "سوم أوزبكستاني",
    "displayName-count-other": "سوم أوزبكستاني",
    "symbol": "UZS"
  },
  "VEB": {
    "displayName": "بوليفار فنزويلي - 2008-1871",
    "symbol": "VEB"
  },
  "VEF": {
    "displayName": "بوليفار فنزويلي",
    "displayName-count-zero": "بوليفار فنزويلي",
    "displayName-count-one": "بوليفار فنزويلي",
    "displayName-count-two": "بوليفار فنزويلي",
    "displayName-count-few": "بوليفار فنزويلي",
    "displayName-count-many": "بوليفار فنزويلي",
    "displayName-count-other": "بوليفار فنزويلي",
    "symbol": "VEF",
    "symbol-alt-narrow": "Bs"
  },
  "VND": {
    "displayName": "دونج فيتنامي",
    "displayName-count-zero": "دونج فيتنامي",

```

```

    "displayName-count-one": "دونج فيتنامي",
    "displayName-count-two": "دونج فيتنامي",
    "displayName-count-few": "دونج فيتنامي",
    "displayName-count-many": "دونج فيتنامي",
    "displayName-count-other": "دونج فيتنامي",
    "symbol": "₫",
    "symbol-alt-narrow": "₫"
  },
  "VNN": {
    "displayName": "VNN",
    "symbol": "VNN"
  },
  "VUV": {
    "displayName": "فاتو فانواتو",
    "displayName-count-zero": "فاتو فانواتو",
    "displayName-count-one": "فاتو فانواتو",
    "displayName-count-two": "فاتو فانواتو",
    "displayName-count-few": "فاتو فانواتو",
    "displayName-count-many": "فاتو فانواتو",
    "displayName-count-other": "فاتو فانواتو",
    "symbol": "VUV"
  },
  "WST": {
    "displayName": "تالا ساموا",
    "displayName-count-zero": "تالا ساموا",
    "displayName-count-one": "تالا ساموا",
    "displayName-count-two": "تالا ساموا",
    "displayName-count-few": "تالا ساموا",
    "displayName-count-many": "تالا ساموا",
    "displayName-count-other": "تالا ساموا",
    "symbol": "WST"
  },
  "XAF": {
    "displayName": "فرنك وسط أفريقي",
    "displayName-count-zero": "فرنك وسط أفريقي",
    "displayName-count-one": "فرنك وسط أفريقي",
    "displayName-count-two": "فرنك وسط أفريقي",
    "displayName-count-few": "فرنك وسط أفريقي",
    "displayName-count-many": "فرنك وسط أفريقي",
    "displayName-count-other": "فرنك وسط أفريقي",
    "symbol": "FCFA"
  },
  "XAG": {
    "displayName": "فضة",
    "symbol": "XAG"
  },
  "XAU": {
    "displayName": "ذهب",
    "symbol": "XAU"
  },
  "XBA": {
    "displayName": "الوحدة الأوروبية المركبة",
    "symbol": "XBA"
  },
  "XBB": {
    "displayName": "الوحدة المالية الأوروبية",
    "symbol": "XBB"
  }

```

```

    },
    "XBC": {
      "displayName": "الوحدة الحسابية الأوروبية",
      "symbol": "XBC"
    },
    },
    "XBD": {
      "displayName": "وحدة الحساب الأوروبية (XBD)",
      "symbol": "XBD"
    },
    },
    "XCD": {
      "displayName": "دولار شرق الكاريبي",
      "displayName-count-zero": "دولار شرق الكاريبي",
      "displayName-count-one": "دولار شرق الكاريبي",
      "displayName-count-two": "دولار شرق الكاريبي",
      "displayName-count-few": "دولار شرق الكاريبي",
      "displayName-count-many": "دولار شرق الكاريبي",
      "displayName-count-other": "دولار شرق الكاريبي",
      "symbol": "EC$",
      "symbol-alt-narrow": "$"
    },
    },
    "XDR": {
      "displayName": "حقوق السحب الخاصة",
      "symbol": "XDR"
    },
    },
    "XEU": {
      "displayName": "وحدة النقد الأوروبية",
      "symbol": "XEU"
    },
    },
    "XFO": {
      "displayName": "فرنك فرنسي ذهبي",
      "symbol": "XFO"
    },
    },
    "XFU": {
      "displayName": "فرنك فرنسي (UIC)",
      "symbol": "XFU"
    },
    },
    "XOF": {
      "displayName": "فرنك غرب أفريقي",
      "displayName-count-zero": "فرنك غرب أفريقي",
      "displayName-count-one": "فرنك غرب أفريقي",
      "displayName-count-two": "فرنك غرب أفريقي",
      "displayName-count-few": "فرنك غرب أفريقي",
      "displayName-count-many": "فرنك غرب أفريقي",
      "displayName-count-other": "فرنك غرب أفريقي",
      "symbol": "CFA"
    },
    },
    "XPD": {
      "displayName": "بالاديوم",
      "symbol": "XPD"
    },
    },
    "XPF": {
      "displayName": "فرنك سي إف بي",
      "displayName-count-zero": "فرنك سي إف بي",
      "displayName-count-one": "فرنك سي إف بي",
      "displayName-count-two": "فرنك سي إف بي",
      "displayName-count-few": "فرنك سي إف بي",
      "displayName-count-many": "فرنك سي إف بي",

```

```

    "displayName-count-other": "فرنك سي إف بي",
    "symbol": "CFPF"
  },
  "XPT": {
    "displayName": "البلاطين",
    "symbol": "XPT"
  },
  "XRE": {
    "displayName": "XRE",
    "symbol": "XRE"
  },
  "XSU": {
    "displayName": "XSU",
    "symbol": "XSU"
  },
  "XTS": {
    "displayName": "كود اختبار العملة",
    "symbol": "XTS"
  },
  "XUA": {
    "displayName": "XUA",
    "symbol": "XUA"
  },
  "XXX": {
    "displayName": "عملة غير معروفة",
    "displayName-count-zero": "(عملة غير معروفة)",
    "displayName-count-one": "(عملة غير معروفة)",
    "displayName-count-two": "(عملة غير معروفة)",
    "displayName-count-few": "(عملة غير معروفة)",
    "displayName-count-many": "(عملة غير معروفة)",
    "displayName-count-other": "(عملة غير معروفة)",
    "symbol": "****"
  },
  "YDD": {
    "displayName": "دينار يمني",
    "symbol": "YDD"
  },
  "YER": {
    "displayName": "ر.ي.",
    "displayName-count-zero": "ر.ي.",
    "displayName-count-one": "ر.ي.",
    "displayName-count-two": "ر.ي.",
    "displayName-count-few": "ر.ي.",
    "displayName-count-many": "ر.ي.",
    "displayName-count-other": "ر.ي.",
    "symbol": "ر.ي."
  },
  "YUD": {
    "displayName": "دينار يوغسلافي",
    "symbol": "YUD"
  },
  "YUM": {
    "displayName": "YUM",
    "symbol": "YUM"
  },
  "YUN": {
    "displayName": "دينار يوغسلافي قابل للتحويل",

```

```

    "symbol": "YUN"
  },
  "YUR": {
    "displayName": "YUR",
    "symbol": "YUR"
  },
  "ZAL": {
    "displayName": "راند جنوب أفريقيا -مالي",
    "symbol": "ZAL"
  },
  "ZAR": {
    "displayName": "راند جنوب أفريقيا",
    "displayName-count-zero": "راند جنوب أفريقيا",
    "displayName-count-one": "راند جنوب أفريقيا",
    "displayName-count-two": "راند جنوب أفريقيا",
    "displayName-count-few": "راند جنوب أفريقيا",
    "displayName-count-many": "راند جنوب أفريقيا",
    "displayName-count-other": "راند جنوب أفريقيا",
    "symbol": "ZAR",
    "symbol-alt-narrow": "R"
  },
  "ZMK": {
    "displayName": "كواشا زامبي - 2012-1968",
    "displayName-count-zero": "كواشا زامبي - 2012-1968",
    "displayName-count-one": "كواشا زامبي - 2012-1968",
    "displayName-count-two": "كواشا زامبي - 2012-1968",
    "displayName-count-few": "كواشا زامبي - 2012-1968",
    "displayName-count-many": "كواشا زامبي - 2012-1968",
    "displayName-count-other": "كواشا زامبي - 2012-1968",
    "symbol": "ZMK"
  },
  "ZMW": {
    "displayName": "كواشا زامبي",
    "displayName-count-zero": "كواشا زامبي",
    "displayName-count-one": "كواشا زامبي",
    "displayName-count-two": "كواشا زامبي",
    "displayName-count-few": "كواشا زامبي",
    "displayName-count-many": "كواشا زامبي",
    "displayName-count-other": "كواشا زامبي",
    "symbol": "ZMW",
    "symbol-alt-narrow": "ZK"
  },
  "ZRN": {
    "displayName": "زائير زائيري جديد",
    "symbol": "ZRN"
  },
  "ZRZ": {
    "displayName": "زائير زائيري",
    "symbol": "ZRZ"
  },
  "ZWD": {
    "displayName": "دولار زمبابوي",
    "symbol": "ZWD"
  },
  "ZWL": {
    "displayName": "دولار زمبابوي 2009",
    "symbol": "ZWL"
  }

```

```
    },  
    "ZWR": {  
      "displayName": "ZWR",  
      "symbol": "ZWR"  
    }  
  }  
}  
  
}
```

## CURRENCIES.JSX

```
{
  "main";
  {
    "ar";
    {
      "identity";
      {
        "version";
        {
          "_number";
          "$Revision: 12879 $",
          "_cldrVersion";
          "30.0.3";
        }
        "language";
        "ar";
      }
    }
    "numbers";
    {
      "currencies";
      {
        "ADP";
        {
          "displayName";
          "بيسٲا أندوري",
          "symbol";
          "ADP";
        }
      }
      "AED";
      {
        "displayName";
        "درهم إماراتي",
        "displayName-count-zero";
        "درهم إماراتي",
        "displayName-count-one";
        "درهم إماراتي",
        "displayName-count-two";
        "درهم إماراتي",
        "displayName-count-few";
        "درهم إماراتي",
        "displayName-count-many";
        "درهم إماراتي",
        "displayName-count-other";
      }
    }
  }
}
```



```

        "درهم إماراتي",
        "symbol";
        "د.إ.";
    }
    "AFA";
    {
        "displayName";
        "2002-1927 - أفغاني",
        "symbol";
        "AFA";
    }
    "AFN";
    {
        "displayName";
        "أفغاني",
        "displayName-count-zero";
        "أفغاني أفغانستان",
        "displayName-count-one";
        "أفغاني أفغانستان",
        "displayName-count-two";
        "أفغاني أفغانستان",
        "displayName-count-few";
        "أفغاني أفغانستان",
        "displayName-count-many";
        "أفغاني أفغانستان",
        "displayName-count-other";
        "أفغاني أفغانستان",
        "symbol";
        "AFN";
    }
    "ALL";
    {
        "displayName";
        "ليك ألباني",
        "displayName-count-zero";
        "ليك ألباني",
        "displayName-count-one";
        "ليك ألباني",
        "displayName-count-two";
        "ليك ألباني",
        "displayName-count-few";
        "ليك ألباني",
        "displayName-count-many";
        "ليك ألباني",
        "displayName-count-other";
        "ليك ألباني",
        "symbol";
        "ALL";
    }
    "AMD";
    {
        "displayName";
        "درام أرميني",
        "displayName-count-zero";
        "درام أرميني",
        "displayName-count-one";
        "درام أرميني",

```

```

        "displayName-count-two";
        "درام آرميني",
        "displayName-count-few";
        "درام آرميني",
        "displayName-count-many";
        "درام آرميني",
        "displayName-count-other";
        "درام آرميني",
        "symbol";
        "AMD";
    }
    "ANG";
    {
        "displayName";
        "غيلدر آنتيلي هولندي",
        "displayName-count-zero";
        "غيلدر آنتيلي هولندي",
        "displayName-count-one";
        "غيلدر آنتيلي هولندي",
        "displayName-count-two";
        "غيلدر آنتيلي هولندي",
        "displayName-count-few";
        "غيلدر آنتيلي هولندي",
        "displayName-count-many";
        "غيلدر آنتيلي هولندي",
        "displayName-count-other";
        "غيلدر آنتيلي هولندي",
        "symbol";
        "ANG";
    }
    "AOA";
    {
        "displayName";
        "كوانزا أنجولي",
        "displayName-count-zero";
        "كوانزا أنجولي",
        "displayName-count-one";
        "كوانزا أنجولي",
        "displayName-count-two";
        "كوانزا أنجولي",
        "displayName-count-few";
        "كوانزا أنجولي",
        "displayName-count-many";
        "كوانزا أنجولي",
        "displayName-count-other";
        "كوانزا أنجولي",
        "symbol";
        "AOA",
        "symbol-alt-narrow";
        "Kz";
    }
    "AOK";
    {
        "displayName";
        "كوانزا أنجولي - 1990-1977",
        "symbol";
        "AOK";
    }

```

```

    }
    "AON";
    {
        "displayName";
        "كوانزا أنجولي جديدة - 2000-1990",
        "symbol";
        "AON";
    }
    "AOR";
    {
        "displayName";
        "كوانزا أنجولي معدلة - 1999 - 1995",
        "symbol";
        "AOR";
    }
    "ARA";
    {
        "displayName";
        "استرال أرجنتيني",
        "symbol";
        "ARA";
    }
    "ARL";
    {
        "displayName";
        "ARL",
        "symbol";
        "ARL";
    }
    "ARM";
    {
        "displayName";
        "ARM",
        "symbol";
        "ARM";
    }
    "ARP";
    {
        "displayName";
        "1985-1983 - أرجنتيني بيزو",
        "symbol";
        "ARP";
    }
    "ARS";
    {
        "displayName";
        "بيزو أرجنتيني",
        "displayName-count-zero";
        "بيزو أرجنتيني",
        "displayName-count-one";
        "بيزو أرجنتيني",
        "displayName-count-two";
        "بيزو أرجنتيني",
        "displayName-count-few";
        "بيزو أرجنتيني",
        "displayName-count-many";
        "بيزو أرجنتيني",
    }

```

```

        "displayName-count-other";
        "بیزو أرجنتینی",
        "symbol";
        "ARS",
        "symbol-alt-narrow";
        "AR$";
    }
    "ATS";
    {
        "displayName";
        "شلن نمساوی",
        "symbol";
        "ATS";
    }
    "AUD";
    {
        "displayName";
        "دولار أسترالي",
        "displayName-count-zero";
        "دولار أسترالي",
        "displayName-count-one";
        "دولار أسترالي",
        "displayName-count-two";
        "دولار أسترالي",
        "displayName-count-few";
        "دولار أسترالي",
        "displayName-count-many";
        "دولار أسترالي",
        "displayName-count-other";
        "دولار أسترالي",
        "symbol";
        "AU$";
        "symbol-alt-narrow";
        "AU$";
    }
    "AWG";
    {
        "displayName";
        "فلورن أروبي",
        "displayName-count-zero";
        "فلورن أروبي",
        "displayName-count-one";
        "فلورن أروبي",
        "displayName-count-two";
        "فلورن أروبي",
        "displayName-count-few";
        "فلورن أروبي",
        "displayName-count-many";
        "فلورن أروبي",
        "displayName-count-other";
        "فلورن أروبي",
        "symbol";
        "AWG";
    }
    "AZM";
    {
        "displayName";

```

```

        "مانات أذربيجاني",
        "symbol";
        "AZM";
    }
    "AZN";
    {
        "displayName";
        "مانات أذربيجان",
        "displayName-count-zero";
        "مانت أذربيجاني",
        "displayName-count-one";
        "مانت أذربيجاني",
        "displayName-count-two";
        "مانت أذربيجاني",
        "displayName-count-few";
        "مانت أذربيجاني",
        "displayName-count-many";
        "مانت أذربيجاني",
        "displayName-count-other";
        "مانت أذربيجاني",
        "symbol";
        "AZN";
    }
    "BAD";
    {
        "displayName";
        "دينار البوسنة والهرسك",
        "symbol";
        "BAD";
    }
    "BAM";
    {
        "displayName";
        "مارك البوسنة والهرسك قابل للتحويل",
        "displayName-count-zero";
        "مارك البوسنة والهرسك قابل للتحويل",
        "displayName-count-one";
        "مارك البوسنة والهرسك قابل للتحويل",
        "displayName-count-two";
        "مارك البوسنة والهرسك قابل للتحويل",
        "displayName-count-few";
        "مارك البوسنة والهرسك قابل للتحويل",
        "displayName-count-many";
        "مارك البوسنة والهرسك قابل للتحويل",
        "displayName-count-other";
        "مارك البوسنة والهرسك قابل للتحويل",
        "symbol";
        "BAM",
        "symbol-alt-narrow";
        "KM";
    }
    "BAN";
    {
        "displayName";
        "BAN",
        "symbol";
        "BAN";
    }

```

```

    }
    "BBD";
    {
      "displayName";
      "دولار بربادوسي",
      "displayName-count-zero";
      "دولار بربادوسي",
      "displayName-count-one";
      "دولار بربادوسي",
      "displayName-count-two";
      "دولار بربادوسي",
      "displayName-count-few";
      "دولار بربادوسي",
      "displayName-count-many";
      "دولار بربادوسي",
      "displayName-count-other";
      "دولار بربادوسي",
      "symbol";
      "BBD",
      "symbol-alt-narrow";
      "BB$";
    }
    "BDT";
    {
      "displayName";
      "তাকা بنجلاديشي",
      "displayName-count-zero";
      "তাকা بنجلاديشي",
      "displayName-count-one";
      "তাকা بنجلاديشي",
      "displayName-count-two";
      "তাকা بنجلاديشي",
      "displayName-count-few";
      "তাকা بنجلاديشي",
      "displayName-count-many";
      "তাকা بنجلاديشي",
      "displayName-count-other";
      "তাকা بنجلاديشي",
      "symbol";
      "BDT",
      "symbol-alt-narrow";
      "৳";
    }
    "BEC";
    {
      "displayName";
      "فرنك بلجيكي قابل للتحويل",
      "symbol";
      "BEC";
    }
    "BEF";
    {
      "displayName";
      "فرنك بلجيكي",
      "symbol";
      "BEF";
    }

```

```
}
"BEL";
{
  "displayName";
  "فرنك بلجيكي مالي",
  "symbol";
  "BEL";
}
"BGL";
{
  "displayName";
  "BGL",
  "symbol";
  "BGL";
}
"BGM";
{
  "displayName";
  "BGM",
  "symbol";
  "BGM";
}
"BGN";
{
  "displayName";
  "ليف بلغاري",
  "displayName-count-zero";
  "ليف بلغاري",
  "displayName-count-one";
  "ليف بلغاري",
  "displayName-count-two";
  "ليف بلغاري",
  "displayName-count-few";
  "ليف بلغاري",
  "displayName-count-many";
  "ليف بلغاري",
  "displayName-count-other";
  "ليف بلغاري",
  "symbol";
  "BGN";
}
"BGO";
{
  "displayName";
  "BGO",
  "symbol";
  "BGO";
}
"BHD";
{
  "displayName";
  "دينار بحريني",
  "displayName-count-zero";
  "دينار بحريني",
  "displayName-count-one";
  "دينار بحريني",
  "displayName-count-two";
```

```

        "دينار بحريني",
        "displayName-count-few";
        "دينار بحريني",
        "displayName-count-many";
        "دينار بحريني",
        "displayName-count-other";
        "دينار بحريني",
        "symbol";
        "د.ب.";
    }
    "BIF";
    {
        "displayName";
        "فرنك بروندي",
        "displayName-count-zero";
        "فرنك بروندي",
        "displayName-count-one";
        "فرنك بروندي",
        "displayName-count-two";
        "فرنك بروندي",
        "displayName-count-few";
        "فرنك بروندي",
        "displayName-count-many";
        "فرنك بروندي",
        "displayName-count-other";
        "فرنك بروندي",
        "symbol";
        "BIF";
    }
    "BMD";
    {
        "displayName";
        "دولار برمودي",
        "displayName-count-zero";
        "دولار برمودي",
        "displayName-count-one";
        "دولار برمودي",
        "displayName-count-two";
        "دولار برمودي",
        "displayName-count-few";
        "دولار برمودي",
        "displayName-count-many";
        "دولار برمودي",
        "displayName-count-other";
        "دولار برمودي",
        "symbol";
        "BMD",
        "symbol-alt-narrow";
        "BM$";
    }
    "BND";
    {
        "displayName";
        "دولار بروناي",
        "displayName-count-zero";
        "دولار بروناي",
        "displayName-count-one";
    }

```



```

        "دولار برونای",
        "displayName-count-two";
        "دولار برونای",
        "displayName-count-few";
        "دولار برونای",
        "displayName-count-many";
        "دولار برونای",
        "displayName-count-other";
        "دولار برونای",
        "symbol";
        "BND",
        "symbol-alt-narrow";
        "BN$";
    }
    "BOB";
    {
        "displayName";
        "بولیفيانو بوليفي",
        "displayName-count-zero";
        "بولیفيانو بوليفي",
        "displayName-count-one";
        "بولیفيانو بوليفي",
        "displayName-count-two";
        "بولیفيانو بوليفي",
        "displayName-count-few";
        "بولیفيانو بوليفي",
        "displayName-count-many";
        "بولیفيانو بوليفي",
        "displayName-count-other";
        "بولیفيانو بوليفي",
        "symbol";
        "BOB",
        "symbol-alt-narrow";
        "Bs";
    }
    "BOL";
    {
        "displayName";
        "BOL",
        "symbol";
        "BOL";
    }
    "BOP";
    {
        "displayName";
        "بیزو بوليفي",
        "symbol";
        "BOP";
    }
    "BOV";
    {
        "displayName";
        "مفدول بوليفي",
        "symbol";
        "BOV";
    }
    "BRB";

```

```

    {
      "displayName";
      "نوفو كروزايرو برازيلى - 1986-1967",
      "symbol";
      "BRB";
    }
    "BRC";
    {
      "displayName";
      "كروزالادو برازيلى",
      "symbol";
      "BRC";
    }
    "BRE";
    {
      "displayName";
      "كروزالايرو برازيلى - 1993-1990",
      "symbol";
      "BRE";
    }
    "BRL";
    {
      "displayName";
      "ريال برازيلى",
      "displayName-count-zero";
      "ريال برازيلى",
      "displayName-count-one";
      "ريال برازيلى",
      "displayName-count-two";
      "ريال برازيلى",
      "displayName-count-few";
      "ريال برازيلى",
      "displayName-count-many";
      "ريال برازيلى",
      "displayName-count-other";
      "ريال برازيلى",
      "symbol";
      "R$",
      "symbol-alt-narrow";
      "R$";
    }
    "BRN";
    {
      "displayName";
      "BRN",
      "symbol";
      "BRN";
    }
    "BRR";
    {
      "displayName";
      "BRR",
      "symbol";
      "BRR";
    }
    "BRZ";
    {

```

```

        "displayName";
        "BRZ",
        "symbol";
        "BRZ";
    }
    "BSD";
    {
        "displayName";
        "دولار باهامي",
        "displayName-count-zero";
        "دولار باهامي",
        "displayName-count-one";
        "دولار باهامي",
        "displayName-count-two";
        "دولار باهامي",
        "displayName-count-few";
        "دولار باهامي",
        "displayName-count-many";
        "دولار باهامي",
        "displayName-count-other";
        "دولار باهامي",
        "symbol";
        "BSD",
        "symbol-alt-narrow";
        "BS$";
    }
    "BTN";
    {
        "displayName";
        "نولتوم بوتاني",
        "displayName-count-zero";
        "نولتوم بوتاني",
        "displayName-count-one";
        "نولتوم بوتاني",
        "displayName-count-two";
        "نولتوم بوتاني",
        "displayName-count-few";
        "نولتوم بوتاني",
        "displayName-count-many";
        "نولتوم بوتاني",
        "displayName-count-other";
        "نولتوم بوتاني",
        "symbol";
        "BTN";
    }
    "BUK";
    {
        "displayName";
        "کيات بورمي",
        "symbol";
        "BUK";
    }
    "BWP";
    {
        "displayName";
        "بولا بتسواني",
        "displayName-count-zero";
    }

```

```

        "بولا بتسواني",
        "displayName-count-one";
        "بولا بتسواني",
        "displayName-count-two";
        "بولا بتسواني",
        "displayName-count-few";
        "بولا بتسواني",
        "displayName-count-many";
        "بولا بتسواني",
        "displayName-count-other";
        "بولا بتسواني",
        "symbol";
        "BWP",
        "symbol-alt-narrow";
        "P";
    }
    "BYB";
    {
        "displayName";
        "روبل بيلاروسي جديد - 1999-1994",
        "symbol";
        "BYB";
    }
    "BYN";
    {
        "displayName";
        "روبل بيلاروسي",
        "displayName-count-zero";
        "روبل بيلاروسي",
        "displayName-count-one";
        "روبل بيلاروسي",
        "displayName-count-two";
        "روبل بيلاروسي",
        "displayName-count-few";
        "روبل بيلاروسي",
        "displayName-count-many";
        "روبل بيلاروسي",
        "displayName-count-other";
        "روبل بيلاروسي",
        "symbol";
        "BYN",
        "symbol-alt-narrow";
        "p.";
    }
    "BYR";
    {
        "displayName";
        "روبل بيلاروسي (۲۰۱۶-۲۰۰۰)",
        "displayName-count-zero";
        "روبل بيلاروسي (۲۰۱۶-۲۰۰۰)",
        "displayName-count-one";
        "روبل بيلاروسي (۲۰۱۶-۲۰۰۰)",
        "displayName-count-two";
        "روبل بيلاروسي (۲۰۱۶-۲۰۰۰)",
        "displayName-count-few";
        "روبل بيلاروسي (۲۰۱۶-۲۰۰۰)",
        "displayName-count-many";
    }

```

```

        "۲۰۱۶-۲۰۰۰ ( روبل بیلاروسي)",
        "displayName-count-other";
        "۲۰۱۶-۲۰۰۰ ( روبل بیلاروسي)",
        "symbol";
        "BYR";
    }
    "BZD";
    {
        "displayName";
        "دولار بليزي",
        "displayName-count-zero";
        "دولار بليزي",
        "displayName-count-one";
        "دولار بليزي",
        "displayName-count-two";
        "دولاران بليزيان",
        "displayName-count-few";
        "دولار بليزي",
        "displayName-count-many";
        "دولار بليزي",
        "displayName-count-other";
        "دولار بليزي",
        "symbol";
        "BZD",
        "symbol-alt-narrow";
        "BZ$";
    }
    "CAD";
    {
        "displayName";
        "دولار كندي",
        "displayName-count-zero";
        "دولار كندي",
        "displayName-count-one";
        "دولار كندي",
        "displayName-count-two";
        "دولار كندي",
        "displayName-count-few";
        "دولار كندي",
        "displayName-count-many";
        "دولار كندي",
        "displayName-count-other";
        "دولار كندي",
        "symbol";
        "CA$",
        "symbol-alt-narrow";
        "CA$";
    }
    "CDF";
    {
        "displayName";
        "فرنك كونغولي",
        "displayName-count-zero";
        "فرنك كونغولي",
        "displayName-count-one";
        "فرنك كونغولي",
        "displayName-count-two";
    }

```

```

        "فرنك كونغولي",
        "displayName-count-few";
        "فرنك كونغولي",
        "displayName-count-many";
        "فرنك كونغولي",
        "displayName-count-other";
        "فرنك كونغولي",
        "symbol";
        "CDF";
    }
    "CHE";
    {
        "displayName";
        "CHE",
        "symbol";
        "CHE";
    }
    "CHF";
    {
        "displayName";
        "فرنك سويسري",
        "displayName-count-zero";
        "فرنك سويسري",
        "displayName-count-one";
        "فرنك سويسري",
        "displayName-count-two";
        "فرنك سويسري",
        "displayName-count-few";
        "فرنك سويسري",
        "displayName-count-many";
        "فرنك سويسري",
        "displayName-count-other";
        "فرنك سويسري",
        "symbol";
        "CHF";
    }
    "CHW";
    {
        "displayName";
        "CHW",
        "symbol";
        "CHW";
    }
    "CLE";
    {
        "displayName";
        "CLE",
        "symbol";
        "CLE";
    }
    "CLF";
    {
        "displayName";
        "CLF",
        "symbol";
        "CLF";
    }
}

```

```

"CLP";
{
  "displayName";
  "بیزو شیلی",
  "displayName-count-zero";
  "بیزو شیلی",
  "displayName-count-one";
  "بیزو شیلی",
  "displayName-count-two";
  "بیزو شیلی",
  "displayName-count-few";
  "بیزو شیلی",
  "displayName-count-many";
  "بیزو شیلی",
  "displayName-count-other";
  "بیزو شیلی",
  "symbol";
  "CLP",
  "symbol-alt-narrow";
  "CL$";
}
"CNY";
{
  "displayName";
  "یوان صینی",
  "displayName-count-zero";
  "یوان صینی",
  "displayName-count-one";
  "یوان صینی",
  "displayName-count-two";
  "یوان صینی",
  "displayName-count-few";
  "یوان صینی",
  "displayName-count-many";
  "یوان صینی",
  "displayName-count-other";
  "یوان صینی",
  "symbol";
  "CN¥",
  "symbol-alt-narrow";
  "CN¥";
}
"COP";
{
  "displayName";
  "بیزو کولومبی",
  "displayName-count-zero";
  "بیزو کولومبی",
  "displayName-count-one";
  "بیزو کولومبی",
  "displayName-count-two";
  "بیزو کولومبی",
  "displayName-count-few";
  "بیزو کولومبی",
  "displayName-count-many";
  "بیزو کولومبی",
  "displayName-count-other";
}

```

```

        "بيزو كولومبي",
        "symbol";
        "COP",
        "symbol-alt-narrow";
        "CO$";
    }
    "COU";
    {
        "displayName";
        "COU",
        "symbol";
        "COU";
    }
    "CRC";
    {
        "displayName";
        "كولن كوستا ريكي",
        "displayName-count-zero";
        "كولن كوستا ريكي",
        "displayName-count-one";
        "كولن كوستا ريكي",
        "displayName-count-two";
        "كولن كوستا ريكي",
        "displayName-count-few";
        "كولن كوستا ريكي",
        "displayName-count-many";
        "كولن كوستا ريكي",
        "displayName-count-other";
        "كولن كوستا ريكي",
        "symbol";
        "CRC",
        "symbol-alt-narrow";
        "₡";
    }
    "CSD";
    {
        "displayName";
        "دينار صربي قديم",
        "symbol";
        "CSD";
    }
    "CSK";
    {
        "displayName";
        "كرونة تشيكوسلوفاكيا",
        "symbol";
        "CSK";
    }
    "CUC";
    {
        "displayName";
        "بيزو كوبي قابل للتحويل",
        "displayName-count-zero";
        "بيزو كوبي قابل للتحويل",
        "displayName-count-one";
        "بيزو كوبي قابل للتحويل",
        "displayName-count-two";
    }

```



```

        "بيزو كوبي قابل للتحويل",
        "displayName-count-few";
        "بيزو كوبي قابل للتحويل",
        "displayName-count-many";
        "بيزو كوبي قابل للتحويل",
        "displayName-count-other";
        "بيزو كوبي قابل للتحويل",
        "symbol";
        "CUC",
        "symbol-alt-narrow";
        "$";
    }
    "CUP";
    {
        "displayName";
        "بيزو كوبي",
        "displayName-count-zero";
        "بيزو كوبي",
        "displayName-count-one";
        "بيزو كوبي",
        "displayName-count-two";
        "بيزو كوبي",
        "displayName-count-few";
        "بيزو كوبي",
        "displayName-count-many";
        "بيزو كوبي",
        "displayName-count-other";
        "بيزو كوبي",
        "symbol";
        "CUP",
        "symbol-alt-narrow";
        "CU$";
    }
    "CVE";
    {
        "displayName";
        "اسكودو الرأس الخضراء",
        "displayName-count-zero";
        "اسكودو الرأس الخضراء",
        "displayName-count-one";
        "اسكودو الرأس الخضراء",
        "displayName-count-two";
        "اسكودو الرأس الخضراء",
        "displayName-count-few";
        "اسكودو الرأس الخضراء",
        "displayName-count-many";
        "اسكودو الرأس الخضراء",
        "displayName-count-other";
        "اسكودو الرأس الخضراء",
        "symbol";
        "CVE";
    }
    "CYP";
    {
        "displayName";
        "جنيه قبرصي",
        "symbol";
    }

```

```

        "CYP";
    }
    "CZK";
    {
        "displayName";
        "كرونة تشيكية",
        "displayName-count-zero";
        "كرونة تشيكية",
        "displayName-count-one";
        "كرونة تشيكية",
        "displayName-count-two";
        "كرونة تشيكية",
        "displayName-count-few";
        "كرونة تشيكية",
        "displayName-count-many";
        "كرونة تشيكية",
        "displayName-count-other";
        "كرونة تشيكية",
        "symbol";
        "CZK",
        "symbol-alt-narrow";
        "Kč";
    }
    "DDM";
    {
        "displayName";
        "أوستمارك ألماني شرقي",
        "symbol";
        "DDM";
    }
    "DEM";
    {
        "displayName";
        "مارك ألماني",
        "symbol";
        "DEM";
    }
    "DJF";
    {
        "displayName";
        "فرنك جيبوتي",
        "displayName-count-zero";
        "فرنك جيبوتي",
        "displayName-count-one";
        "فرنك جيبوتي",
        "displayName-count-two";
        "فرنك جيبوتي",
        "displayName-count-few";
        "فرنك جيبوتي",
        "displayName-count-many";
        "فرنك جيبوتي",
        "displayName-count-other";
        "فرنك جيبوتي",
        "symbol";
        "DJF";
    }
    "DKK";

```

```

    {
      "displayName";
      "كرونة دانماركي",
      "displayName-count-zero";
      "كرونة دانماركي",
      "displayName-count-one";
      "كرونة دانماركي",
      "displayName-count-two";
      "كرونة دانماركي",
      "displayName-count-few";
      "كرونة دانماركي",
      "displayName-count-many";
      "كرونة دانماركي",
      "displayName-count-other";
      "كرونة دانماركي",
      "symbol";
      "DKK",
      "symbol-alt-narrow";
      "kr";
    }
    "DOP";
    {
      "displayName";
      "بيزو الدومنيكان",
      "displayName-count-zero";
      "بيزو الدومنيكان",
      "displayName-count-one";
      "بيزو الدومنيكان",
      "displayName-count-two";
      "بيزو الدومنيكان",
      "displayName-count-few";
      "بيزو الدومنيكان",
      "displayName-count-many";
      "بيزو الدومنيكان",
      "displayName-count-other";
      "بيزو الدومنيكان",
      "symbol";
      "DOP",
      "symbol-alt-narrow";
      "DO$";
    }
    "DZD";
    {
      "displayName";
      "دينار جزائري",
      "displayName-count-zero";
      "دينار جزائري",
      "displayName-count-one";
      "دينار جزائري",
      "displayName-count-two";
      "ديناران جزائريان",
      "displayName-count-few";
      "دينارات جزائرية",
      "displayName-count-many";
      "دينارًا جزائريًا",
      "displayName-count-other";
      "دينار جزائري",
    }

```

```

        "symbol";
        "د.ج.";
    }
    "ECS";
    {
        "displayName";
        "ECS",
        "symbol";
        "ECS";
    }
    "ECV";
    {
        "displayName";
        "ECV",
        "symbol";
        "ECV";
    }
    "EEK";
    {
        "displayName";
        "كرونة استونية",
        "symbol";
        "EEK";
    }
    "EGP";
    {
        "displayName";
        "جنيه مصري",
        "displayName-count-zero";
        "جنيه مصري",
        "displayName-count-one";
        "جنيه مصري",
        "displayName-count-two";
        "جنيهان مصريان",
        "displayName-count-few";
        "جنيهات مصرية",
        "displayName-count-many";
        "جنيهاً مصرياً",
        "displayName-count-other";
        "جنيه مصري",
        "symbol";
        "ج.م.",
        "symbol-alt-narrow";
        "£";
    }
    "ERN";
    {
        "displayName";
        "ناكفا أريتري",
        "displayName-count-zero";
        "ناكفا أريتري",
        "displayName-count-one";
        "ناكفا أريتري",
        "displayName-count-two";
        "ناكفا أريتري",
        "displayName-count-few";
        "ناكفا أريتري",
    }

```

```

        "displayName-count-many";
        "ناكفا أريتري",
        "displayName-count-other";
        "ناكفا أريتري",
        "symbol";
        "ERN";
    }
    "ESA";
    {
        "displayName";
        "ESA",
        "symbol";
        "ESA";
    }
    "ESB";
    {
        "displayName";
        "ESB",
        "symbol";
        "ESB";
    }
    "ESP";
    {
        "displayName";
        "بيزيتا إسباني",
        "symbol";
        "ESP",
        "symbol-alt-narrow";
        "₱";
    }
    "ETB";
    {
        "displayName";
        "بیر اثیوبی",
        "displayName-count-zero";
        "بیر اثیوبی",
        "displayName-count-one";
        "بیر اثیوبی",
        "displayName-count-two";
        "بیر اثیوبی",
        "displayName-count-few";
        "بیر اثیوبی",
        "displayName-count-many";
        "بیر اثیوبی",
        "displayName-count-other";
        "بیر اثیوبی",
        "symbol";
        "ETB";
    }
    "EUR";
    {
        "displayName";
        "یورو",
        "displayName-count-zero";
        "یورو",
        "displayName-count-one";
        "یورو",
    }

```

```

        "displayName-count-two";
        "یورو",
        "displayName-count-few";
        "یورو",
        "displayName-count-many";
        "یورو",
        "displayName-count-other";
        "یورو",
        "symbol";
        "€",
        "symbol-alt-narrow";
        "€";
    }
    "FIM";
    {
        "displayName";
        "مارکا فنلندی",
        "symbol";
        "FIM";
    }
    "FJD";
    {
        "displayName";
        "دولار فیجی",
        "displayName-count-zero";
        "دولار فیجی",
        "displayName-count-one";
        "دولار فیجی",
        "displayName-count-two";
        "دولار فیجی",
        "displayName-count-few";
        "دولار فیجی",
        "displayName-count-many";
        "دولار فیجی",
        "displayName-count-other";
        "دولار فیجی",
        "symbol";
        "FJD",
        "symbol-alt-narrow";
        "FJ$";
    }
    "FKP";
    {
        "displayName";
        "جنيه جزر فوکلاند",
        "displayName-count-zero";
        "جنيه جزر فوکلاند",
        "displayName-count-one";
        "جنيه جزر فوکلاند",
        "displayName-count-two";
        "جنيه جزر فوکلاند",
        "displayName-count-few";
        "جنيه جزر فوکلاند",
        "displayName-count-many";
        "جنيه جزر فوکلاند",
        "displayName-count-other";
        "جنيه جزر فوکلاند",
    }

```

```

        "symbol";
        "FKP",
        "symbol-alt-narrow";
        "£";
    }
    "FRF";
    {
        "displayName";
        "فرنك فرنسي",
        "symbol";
        "FRF";
    }
    "GBP";
    {
        "displayName";
        "جنيه إسترليني",
        "displayName-count-zero";
        "جنيه إسترليني",
        "displayName-count-one";
        "جنيه إسترليني",
        "displayName-count-two";
        "جنيه إسترليني",
        "displayName-count-few";
        "جنيه إسترليني",
        "displayName-count-many";
        "جنيه إسترليني",
        "displayName-count-other";
        "جنيه إسترليني",
        "symbol";
        "£",
        "symbol-alt-narrow";
        "UK£";
    }
    "GEK";
    {
        "displayName";
        "GEK",
        "symbol";
        "GEK";
    }
    "GEL";
    {
        "displayName";
        "ლარი جورج",
        "displayName-count-zero";
        "ლარი جورج",
        "displayName-count-one";
        "ლარი جورج",
        "displayName-count-two";
        "ლარი جورج",
        "displayName-count-few";
        "ლარი جورج",
        "displayName-count-many";
        "ლარი جورج",
        "displayName-count-other";
        "ლარი جورج",
        "symbol";
    }

```

```

        "GEL",
        "symbol-alt-narrow";
    "₵",
        "symbol-alt-variant";
    "₵";
}
"GHC";
{
    "displayName";
    "سيدي غاني",
        "symbol";
    "GHC";
}
"GHS";
{
    "displayName";
    "سيدي غانا",
        "displayName-count-zero";
    "سيدي غانا",
        "displayName-count-one";
    "سيدي غانا",
        "displayName-count-two";
    "سيدي غانا",
        "displayName-count-few";
    "سيدي غانا",
        "displayName-count-many";
    "سيدي غانا",
        "displayName-count-other";
    "سيدي غانا",
        "symbol";
    "GHS";
}
"GIP";
{
    "displayName";
    "جنيه جبل طارق",
        "displayName-count-zero";
    "جنيه جبل طارق",
        "displayName-count-one";
    "جنيه جبل طارق",
        "displayName-count-two";
    "جنيه جبل طارق",
        "displayName-count-few";
    "جنيه جبل طارق",
        "displayName-count-many";
    "جنيه جبل طارق",
        "displayName-count-other";
    "جنيه جبل طارق",
        "symbol";
    "GIP",
        "symbol-alt-narrow";
    "£";
}
"GMD";
{
    "displayName";
    "دلاسي جامبي",

```



```

        "displayName-count-zero";
        "دلّاسي جامبي",
        "displayName-count-one";
        "دلّاسي جامبي",
        "displayName-count-two";
        "دلّاسي جامبي",
        "displayName-count-few";
        "دلّاسي جامبي",
        "displayName-count-many";
        "دلّاسي جامبي",
        "displayName-count-other";
        "دلّاسي جامبي",
        "symbol";
        "GMD";
    }
    "GNF";
    {
        "displayName";
        "فرنك غينيا",
        "displayName-count-zero";
        "فرنك غينيا",
        "displayName-count-one";
        "فرنك غينيا",
        "displayName-count-two";
        "فرنك غينيا",
        "displayName-count-few";
        "فرنك غينيا",
        "displayName-count-many";
        "فرنك غينيا",
        "displayName-count-other";
        "فرنك غينيا",
        "symbol";
        "GNF",
        "symbol-alt-narrow";
        "FG";
    }
    "GNS";
    {
        "displayName";
        "سيلي غينيا",
        "symbol";
        "GNS";
    }
    "GQE";
    {
        "displayName";
        "اكويل جونيّنا غينيّنا الاستوائيّة",
        "symbol";
        "GQE";
    }
    "GRD";
    {
        "displayName";
        "دراخما يوناني",
        "symbol";
        "GRD";
    }
}

```

```

"GTQ";
{
  "displayName";
  "كوتزال جواتيمالا",
  "displayName-count-zero";
  "كوتزال جواتيمالا",
  "displayName-count-one";
  "كوتزال جواتيمالا",
  "displayName-count-two";
  "كوتزال جواتيمالا",
  "displayName-count-few";
  "كوتزال جواتيمالا",
  "displayName-count-many";
  "كوتزال جواتيمالا",
  "displayName-count-other";
  "كوتزال جواتيمالا",
  "symbol";
  "GTQ",
  "symbol-alt-narrow";
  "Q";
}
"GWE";
{
  "displayName";
  "اسكود برتغالي غينيا",
  "symbol";
  "GWE";
}
"GWP";
{
  "displayName";
  "بيزو غينيا بيساو",
  "symbol";
  "GWP";
}
"GYD";
{
  "displayName";
  "دولار غيانا",
  "displayName-count-zero";
  "دولار غيانا",
  "displayName-count-one";
  "دولار غيانا",
  "displayName-count-two";
  "دولار غيانا",
  "displayName-count-few";
  "دولار غيانا",
  "displayName-count-many";
  "دولار غيانا",
  "displayName-count-other";
  "دولار غيانا",
  "symbol";
  "GYD",
  "symbol-alt-narrow";
  "GY$";
}
"HKD";

```

```

{
  "displayName";
  "دولار هونغ كونغ",
  "displayName-count-zero";
  "دولار هونغ كونغ",
  "displayName-count-one";
  "دولار هونغ كونغ",
  "displayName-count-two";
  "دولار هونغ كونغ",
  "displayName-count-few";
  "دولار هونغ كونغ",
  "displayName-count-many";
  "دولار هونغ كونغ",
  "displayName-count-other";
  "دولار هونغ كونغ",
  "symbol";
  "HK$",
  "symbol-alt-narrow";
  "HK$";
}
"HNL";
{
  "displayName";
  "ليمبيرا هندوراس",
  "displayName-count-zero";
  "ليمبيرا هندوراس",
  "displayName-count-one";
  "ليمبيرا هندوراس",
  "displayName-count-two";
  "ليمبيرا هندوراس",
  "displayName-count-few";
  "ليمبيرا هندوراس",
  "displayName-count-many";
  "ليمبيرا هندوراس",
  "displayName-count-other";
  "ليمبيرا هندوراس",
  "symbol";
  "HNL",
  "symbol-alt-narrow";
  "L";
}
"HRD";
{
  "displayName";
  "دينار كرواتي",
  "symbol";
  "HRD";
}
"HRK";
{
  "displayName";
  "كونا كرواتي",
  "displayName-count-zero";
  "كونا كرواتي",
  "displayName-count-one";
  "كونا كرواتي",
  "displayName-count-two";
}

```

```

        "كونا كرواتي",
        "displayName-count-few";
        "كونا كرواتي",
        "displayName-count-many";
        "كونا كرواتي",
        "displayName-count-other";
        "كونا كرواتي",
        "symbol";
        "HRK",
        "symbol-alt-narrow";
        "kn";
    }
    "HTG";
    {
        "displayName";
        "جوردی هایتي",
        "displayName-count-zero";
        "جوردی هایتي",
        "displayName-count-one";
        "جوردی هایتي",
        "displayName-count-two";
        "جوردی هایتي",
        "displayName-count-few";
        "جوردی هایتي",
        "displayName-count-many";
        "جوردی هایتي",
        "displayName-count-other";
        "جوردی هایتي",
        "symbol";
        "HTG";
    }
    "HUF";
    {
        "displayName";
        "فورينت مجري",
        "displayName-count-zero";
        "فورينت مجري",
        "displayName-count-one";
        "فورينت مجري",
        "displayName-count-two";
        "فورينت مجري",
        "displayName-count-few";
        "فورينت مجري",
        "displayName-count-many";
        "فورينت مجري",
        "displayName-count-other";
        "فورينت مجري",
        "symbol";
        "HUF",
        "symbol-alt-narrow";
        "Ft";
    }
    "IDR";
    {
        "displayName";
        "روبية إندونيسية",
        "displayName-count-zero";
    }

```

```

        "روبية إندونيسية",
        "displayName-count-one";
        "روبية إندونيسية",
        "displayName-count-two";
        "روبية إندونيسية",
        "displayName-count-few";
        "روبية إندونيسية",
        "displayName-count-many";
        "روبية إندونيسية",
        "displayName-count-other";
        "روبية إندونيسية",
        "symbol";
        "IDR",
        "symbol-alt-narrow";
        "Rp";
    }
    "IEP";
    {
        "displayName";
        "جنيه إيرلندي",
        "symbol";
        "IEP";
    }
    "ILP";
    {
        "displayName";
        "جنيه إسرائيلي",
        "symbol";
        "ILP";
    }
    "ILS";
    {
        "displayName";
        "شيكل إسرائيلي جديد",
        "displayName-count-zero";
        "شيكل إسرائيلي جديد",
        "displayName-count-one";
        "شيكل إسرائيلي جديد",
        "displayName-count-two";
        "شيكل إسرائيلي جديد",
        "displayName-count-few";
        "شيكل إسرائيلي جديد",
        "displayName-count-many";
        "شيكل إسرائيلي جديد",
        "displayName-count-other";
        "شيكل إسرائيلي جديد",
        "symbol";
        "₪",
        "symbol-alt-narrow";
        "₪";
    }
    "INR";
    {
        "displayName";
        "روبية هندي",
        "displayName-count-zero";
        "روبية هندي",

```

```

        "displayName-count-one";
        "روبية هندي",
        "displayName-count-two";
        "روبية هندي",
        "displayName-count-few";
        "روبية هندي",
        "displayName-count-many";
        "روبية هندي",
        "displayName-count-other";
        "روبية هندي",
        "symbol";
        "₹",
        "symbol-alt-narrow";
        "₹";
    }
    "IQD";
    {
        "displayName";
        "دينار عراقي",
        "displayName-count-zero";
        "دينار عراقي",
        "displayName-count-one";
        "دينار عراقي",
        "displayName-count-two";
        "دينار عراقي",
        "displayName-count-few";
        "دينار عراقي",
        "displayName-count-many";
        "دينار عراقي",
        "displayName-count-other";
        "دينار عراقي",
        "symbol";
        "د.ع.";
    }
    "IRR";
    {
        "displayName";
        "ریال ایرانی",
        "displayName-count-zero";
        "ریال ایرانی",
        "displayName-count-one";
        "ریال ایرانی",
        "displayName-count-two";
        "ریال ایرانی",
        "displayName-count-few";
        "ریال ایرانی",
        "displayName-count-many";
        "ریال ایرانی",
        "displayName-count-other";
        "ریال ایرانی",
        "symbol";
        "ر.ا.";
    }
    "ISK";
    {
        "displayName";
        "کرونة آيسلندي",

```

```

        "displayName-count-zero";
        "كرونه أيسلندي",
        "displayName-count-one";
        "كرونه أيسلندي",
        "displayName-count-two";
        "كرونه أيسلندي",
        "displayName-count-few";
        "كرونه أيسلندي",
        "displayName-count-many";
        "كرونه أيسلندي",
        "displayName-count-other";
        "كرونه أيسلندي",
        "symbol";
        "ISK",
        "symbol-alt-narrow";
        "kr";
    }
    "ITL";
    {
        "displayName";
        "ليرة إيطالية",
        "symbol";
        "ITL";
    }
    "JMD";
    {
        "displayName";
        "دولار جامايكي",
        "displayName-count-zero";
        "دولار جامايكي",
        "displayName-count-one";
        "دولار جامايكي",
        "displayName-count-two";
        "دولار جامايكي",
        "displayName-count-few";
        "دولار جامايكي",
        "displayName-count-many";
        "دولار جامايكي",
        "displayName-count-other";
        "دولار جامايكي",
        "symbol";
        "JMD",
        "symbol-alt-narrow";
        "JM$";
    }
    "JOD";
    {
        "displayName";
        "دينار أردني",
        "displayName-count-zero";
        "دينار أردني",
        "displayName-count-one";
        "دينار أردني",
        "displayName-count-two";
        "دينار أردني",
        "displayName-count-few";
        "دينار أردني",
    }

```

```

        "displayName-count-many";
        "دينار أردني",
        "displayName-count-other";
        "دينار أردني",
        "symbol";
        "د.أ.";
    }
    "JPY";
    {
        "displayName";
        "ين ياباني",
        "displayName-count-zero";
        "ين ياباني",
        "displayName-count-one";
        "ين ياباني",
        "displayName-count-two";
        "ين ياباني",
        "displayName-count-few";
        "ين ياباني",
        "displayName-count-many";
        "ين ياباني",
        "displayName-count-other";
        "ين ياباني",
        "symbol";
        "JP¥",
        "symbol-alt-narrow";
        "JP¥";
    }
    "KES";
    {
        "displayName";
        "شلن كينيي",
        "displayName-count-zero";
        "شلن كينيي",
        "displayName-count-one";
        "شلن كينيي",
        "displayName-count-two";
        "شلن كينيي",
        "displayName-count-few";
        "شلن كينيي",
        "displayName-count-many";
        "شلن كينيي",
        "displayName-count-other";
        "شلن كينيي",
        "symbol";
        "KES";
    }
    "KGS";
    {
        "displayName";
        "سوم قيرغستاني",
        "displayName-count-zero";
        "سوم قيرغستاني",
        "displayName-count-one";
        "سوم قيرغستاني",
        "displayName-count-two";
        "سوم قيرغستاني",
    }

```



```

        "displayName-count-few";
        "سوم قيرغستاني",
        "displayName-count-many";
        "سوم قيرغستاني",
        "displayName-count-other";
        "سوم قيرغستاني",
        "symbol";
        "KGS";
    }
    "KHR";
    {
        "displayName";
        "ريال كمبودي",
        "displayName-count-zero";
        "ريال كمبودي",
        "displayName-count-one";
        "ريال كمبودي",
        "displayName-count-two";
        "ريال كمبودي",
        "displayName-count-few";
        "ريال كمبودي",
        "displayName-count-many";
        "ريال كمبودي",
        "displayName-count-other";
        "ريال كمبودي",
        "symbol";
        "KHR",
        "symbol-alt-narrow";
        "៛";
    }
    "KMF";
    {
        "displayName";
        "فرنك جزر القمر",
        "displayName-count-zero";
        "فرنك جزر القمر",
        "displayName-count-one";
        "فرنك جزر القمر",
        "displayName-count-two";
        "فرنك جزر القمر",
        "displayName-count-few";
        "فرنك جزر القمر",
        "displayName-count-many";
        "فرنك جزر القمر",
        "displayName-count-other";
        "فرنك جزر القمر",
        "symbol";
        "ف.ج.ق.",
        "symbol-alt-narrow";
        "CF";
    }
    "KPW";
    {
        "displayName";
        "وون كوريا الشمالية",
        "displayName-count-zero";

```

```

        "وون كوريا الشمالية",
        "displayName-count-one";
        "وون كوريا الشمالية",
        "displayName-count-two";
        "وون كوريا الشمالية",
        "displayName-count-few";
        "وون كوريا الشمالية",
        "displayName-count-many";
        "وون كوريا الشمالية",
        "displayName-count-other";
        "وون كوريا الشمالية",
        "symbol";
        "KPW",
        "symbol-alt-narrow";
        "₩";
    }
    "KRH";
    {
        "displayName";
        "KRH",
        "symbol";
        "KRH";
    }
    "KRO";
    {
        "displayName";
        "KRO",
        "symbol";
        "KRO";
    }
    "KRW";
    {
        "displayName";
        "وون كوريا الجنوبية",
        "displayName-count-zero";
        "وون كوريا الجنوبية",
        "displayName-count-one";
        "وون كوريا الجنوبية",
        "displayName-count-two";
        "وون كوريا الجنوبية",
        "displayName-count-few";
        "وون كوريا الجنوبية",
        "displayName-count-many";
        "وون كوريا الجنوبية",
        "displayName-count-other";
        "وون كوريا الجنوبية",
        "symbol";
        "₩",
        "symbol-alt-narrow";
        "₩";
    }
    "KWD";
    {
        "displayName";
        "دينار كويتي",
        "displayName-count-zero";
        "دينار كويتي",

```

```

        "displayName-count-one";
        "دينار كويتي",
        "displayName-count-two";
        "دينار كويتي",
        "displayName-count-few";
        "دينار كويتي",
        "displayName-count-many";
        "دينار كويتي",
        "displayName-count-other";
        "دينار كويتي",
        "symbol";
        "د.ك.";
    }
    "KYD";
    {
        "displayName";
        "دولار جزر كيمن",
        "displayName-count-zero";
        "دولار جزر كيمن",
        "displayName-count-one";
        "دولار جزر كيمن",
        "displayName-count-two";
        "دولار جزر كيمن",
        "displayName-count-few";
        "دولار جزر كيمن",
        "displayName-count-many";
        "دولار جزر كيمن",
        "displayName-count-other";
        "دولار جزر كيمن",
        "symbol";
        "KYD",
        "symbol-alt-narrow";
        "KY$";
    }
    "KZT";
    {
        "displayName";
        "تينغ كازاخستاني",
        "displayName-count-zero";
        "تينغ كازاخستاني",
        "displayName-count-one";
        "تينغ كازاخستاني",
        "displayName-count-two";
        "تينغ كازاخستاني",
        "displayName-count-few";
        "تينغ كازاخستاني",
        "displayName-count-many";
        "تينغ كازاخستاني",
        "displayName-count-other";
        "تينغ كازاخستاني",
        "symbol";
        "KZT",
        "symbol-alt-narrow";
        "Т";
    }
    "LAK";
    {

```

```

        "displayName";
        "کیب لاوسی",
        "displayName-count-zero";
        "کیب لاوسی",
        "displayName-count-one";
        "کیب لاوسی",
        "displayName-count-two";
        "کیب لاوسی",
        "displayName-count-few";
        "کیب لاوسی",
        "displayName-count-many";
        "کیب لاوسی",
        "displayName-count-other";
        "کیب لاوسی",
        "symbol";
        "LAK",
        "symbol-alt-narrow";
        "₭";
    }
    "LBP";
    {
        "displayName";
        "جنيه لبناني",
        "displayName-count-zero";
        "جنيه لبناني",
        "displayName-count-one";
        "جنيه لبناني",
        "displayName-count-two";
        "جنيه لبناني",
        "displayName-count-few";
        "جنيه لبناني",
        "displayName-count-many";
        "جنيه لبناني",
        "displayName-count-other";
        "جنيه لبناني",
        "symbol";
        "ل.ل.",
        "symbol-alt-narrow";
        "L£";
    }
    "LKR";
    {
        "displayName";
        "روبية سريلانكية",
        "displayName-count-zero";
        "روبية سريلانكية",
        "displayName-count-one";
        "روبية سريلانكية",
        "displayName-count-two";
        "روبية سريلانكية",
        "displayName-count-few";
        "روبية سريلانكية",
        "displayName-count-many";
        "روبية سريلانكية",
        "displayName-count-other";
        "روبية سريلانكية",
        "symbol";
    }

```

```

        "LKR",
        "symbol-alt-narrow";
        "Rs";
    }
    "LRD";
    {
        "displayName";
        "دولار ليبيري",
        "displayName-count-zero";
        "دولار ليبيري",
        "displayName-count-one";
        "دولار ليبيري",
        "displayName-count-two";
        "دولاران ليبيريان",
        "displayName-count-few";
        "دولارات ليبيرية",
        "displayName-count-many";
        "دولارًا ليبيريًا",
        "displayName-count-other";
        "دولار ليبيري",
        "symbol";
        "LRD",
        "symbol-alt-narrow";
        "$";
    }
    "LSL";
    {
        "displayName";
        "لوتي ليسوتو",
        "symbol";
        "LSL";
    }
    "LTL";
    {
        "displayName";
        "ليتأ ليتوانية",
        "displayName-count-zero";
        "ليتأ ليتوانية",
        "displayName-count-one";
        "ليتأ ليتوانية",
        "displayName-count-two";
        "ليتأ ليتوانية",
        "displayName-count-few";
        "ليتأ ليتوانية",
        "displayName-count-many";
        "ليتأ ليتوانية",
        "displayName-count-other";
        "ليتأ ليتوانية",
        "symbol";
        "LTL",
        "symbol-alt-narrow";
        "Lt";
    }
    "LTT";
    {
        "displayName";
        "تالوناس ليتواني",

```

```

        "symbol";
        "LTT";
    }
    "LUC";
    {
        "displayName";
        "فرنك لوكسمبرج قابل للتحويل",
        "symbol";
        "LUC";
    }
    "LUF";
    {
        "displayName";
        "فرنك لوكسمبرج",
        "symbol";
        "LUF";
    }
    "LUL";
    {
        "displayName";
        "فرنك لوكسمبرج المالي",
        "symbol";
        "LUL";
    }
    "LVL";
    {
        "displayName";
        "لاتس لاتفيا",
        "displayName-count-zero";
        "لاتس لاتفي",
        "displayName-count-one";
        "لاتس لاتفي",
        "displayName-count-two";
        "لاتس لاتفي",
        "displayName-count-few";
        "لاتس لاتفي",
        "displayName-count-many";
        "لاتس لاتفي",
        "displayName-count-other";
        "لاتس لاتفي",
        "symbol";
        "LVL",
        "symbol-alt-narrow";
        "Ls";
    }
    "LVR";
    {
        "displayName";
        "روبل لاتفيا",
        "symbol";
        "LVR";
    }
    "LYD";
    {
        "displayName";
        "دينار ليبي",
        "displayName-count-zero";

```

```

        "دينار ليبي",
        "displayName-count-one";
        "دينار ليبي",
        "displayName-count-two";
        "ديناران ليبيان",
        "displayName-count-few";
        "دينارات ليبية",
        "displayName-count-many";
        "دينارًا ليبيًا",
        "displayName-count-other";
        "دينار ليبي",
        "symbol";
        "د.ل.";
    }
    "MAD";
    {
        "displayName";
        "درهم مغربي",
        "displayName-count-zero";
        "درهم مغربي",
        "displayName-count-one";
        "درهم مغربي",
        "displayName-count-two";
        "درهمان مغربيان",
        "displayName-count-few";
        "دراهم مغربية",
        "displayName-count-many";
        "درهما مغربيًا",
        "displayName-count-other";
        "درهم مغربي",
        "symbol";
        "د.م.";
    }
    "MAF";
    {
        "displayName";
        "فرنك مغربي",
        "symbol";
        "MAF";
    }
    "MCF";
    {
        "displayName";
        "MCF",
        "symbol";
        "MCF";
    }
    "MDC";
    {
        "displayName";
        "MDC",
        "symbol";
        "MDC";
    }
    "MDL";
    {
        "displayName";

```

```

        "ليو مولدوفي",
        "displayName-count-zero";
        "ليو مولدوفي",
        "displayName-count-one";
        "ليو مولدوفي",
        "displayName-count-two";
        "ليو مولدوفي",
        "displayName-count-few";
        "ليو مولدوفي",
        "displayName-count-many";
        "ليو مولدوفي",
        "displayName-count-other";
        "ليو مولدوفي",
        "symbol";
        "MDL";
    }
    "MGA";
    {
        "displayName";
        "أرياري مدغشقر",
        "displayName-count-zero";
        "أرياري مدغشقر",
        "displayName-count-one";
        "أرياري مدغشقر",
        "displayName-count-two";
        "أرياري مدغشقر",
        "displayName-count-few";
        "أرياري مدغشقر",
        "displayName-count-many";
        "أرياري مدغشقر",
        "displayName-count-other";
        "أرياري مدغشقر",
        "symbol";
        "MGA",
        "symbol-alt-narrow";
        "Ar";
    }
    "MGF";
    {
        "displayName";
        "فرنك مدغشقر",
        "symbol";
        "MGF";
    }
    "MKD";
    {
        "displayName";
        "دينار مقدوني",
        "displayName-count-zero";
        "دينار مقدوني",
        "displayName-count-one";
        "دينار مقدوني",
        "displayName-count-two";
        "ديناران مقدونيان",
        "displayName-count-few";
        "دينارات مقدونية",
        "displayName-count-many";
    }

```



```

        "دينارًا مقدونيا",
        "displayName-count-other";
        "دينار مقدوني",
        "symbol";
        "MKD";
    }
    "MKN";
    {
        "displayName";
        "MKN",
        "symbol";
        "MKN";
    }
    "MLF";
    {
        "displayName";
        "فرنك مالي",
        "symbol";
        "MLF";
    }
    "MMK";
    {
        "displayName";
        "كيات ميانمار",
        "displayName-count-zero";
        "كيات ميانمار",
        "displayName-count-one";
        "كيات ميانمار",
        "displayName-count-two";
        "كيات ميانمار",
        "displayName-count-few";
        "كيات ميانمار",
        "displayName-count-many";
        "كيات ميانمار",
        "displayName-count-other";
        "كيات ميانمار",
        "symbol";
        "MMK",
        "symbol-alt-narrow";
        "K";
    }
    "MNT";
    {
        "displayName";
        "توغروغ منغولي",
        "displayName-count-zero";
        "توغروغ منغولي",
        "displayName-count-one";
        "توغروغ منغولي",
        "displayName-count-two";
        "توغروغ منغولي",
        "displayName-count-few";
        "توغروغ منغولي",
        "displayName-count-many";
        "توغروغ منغولي",
        "displayName-count-other";
        "توغروغ منغولي",
    }

```

```

        "symbol";
        "MNT",
        "symbol-alt-narrow";
        "₮";
    }
    "MOP";
    {
        "displayName";
        "باتاكا ماكاوي",
        "displayName-count-zero";
        "باتاكا ماكاوي",
        "displayName-count-one";
        "باتاكا ماكاوي",
        "displayName-count-two";
        "باتاكا ماكاوي",
        "displayName-count-few";
        "باتاكا ماكاوي",
        "displayName-count-many";
        "باتاكا ماكاوي",
        "displayName-count-other";
        "باتاكا ماكاوي",
        "symbol";
        "MOP";
    }
    "MRO";
    {
        "displayName";
        "أوقية موريتانية",
        "displayName-count-zero";
        "أوقية موريتانية",
        "displayName-count-one";
        "أوقية موريتانية",
        "displayName-count-two";
        "أوقية موريتانية",
        "displayName-count-few";
        "أوقية موريتانية",
        "displayName-count-many";
        "أوقية موريتانية",
        "displayName-count-other";
        "أوقية موريتانية",
        "symbol";
        "أ.م.";
    }
    "MTL";
    {
        "displayName";
        "ليرة مالطية",
        "symbol";
        "MTL";
    }
    "MTP";
    {
        "displayName";
        "جنيه مالطي",
        "symbol";
        "MTP";
    }
}

```

```

"MUR";
{
  "displayName";
  "روبية موريشيوسية",
  "displayName-count-zero";
  "روبية موريشيوسية",
  "displayName-count-one";
  "روبية موريشيوسية",
  "displayName-count-two";
  "روبية موريشيوسية",
  "displayName-count-few";
  "روبية موريشيوسية",
  "displayName-count-many";
  "روبية موريشيوسية",
  "displayName-count-other";
  "روبية موريشيوسية",
  "symbol";
  "MUR",
  "symbol-alt-narrow";
  "Rs";
}
"MVR";
{
  "displayName";
  "روفيه جزر المالديف",
  "displayName-count-zero";
  "روفيه جزر المالديف",
  "displayName-count-one";
  "روفيه جزر المالديف",
  "displayName-count-two";
  "روفيه جزر المالديف",
  "displayName-count-few";
  "روفيه جزر المالديف",
  "displayName-count-many";
  "روفيه جزر المالديف",
  "displayName-count-other";
  "روفيه جزر المالديف",
  "symbol";
  "MVR";
}
"MWK";
{
  "displayName";
  "كواشا مالاوي",
  "displayName-count-zero";
  "كواشا مالاوي",
  "displayName-count-one";
  "كواشا مالاوي",
  "displayName-count-two";
  "كواشا مالاوي",
  "displayName-count-few";
  "كواشا مالاوي",
  "displayName-count-many";
  "كواشا مالاوي",
  "displayName-count-other";
  "كواشا مالاوي",
  "symbol";
}

```

```

        "MWK";
    }
    "MXN";
    {
        "displayName";
        "بيزو مكسيكي",
        "displayName-count-zero";
        "بيزو مكسيكي",
        "displayName-count-one";
        "بيزو مكسيكي",
        "displayName-count-two";
        "بيزو مكسيكي",
        "displayName-count-few";
        "بيزو مكسيكي",
        "displayName-count-many";
        "بيزو مكسيكي",
        "displayName-count-other";
        "بيزو مكسيكي",
        "symbol";
        "MX$",
        "symbol-alt-narrow";
        "MX$";
    }
    "MXP";
    {
        "displayName";
        "بيزو فضي مكسيكي - 1992-1861",
        "symbol";
        "MXP";
    }
    "MXV";
    {
        "displayName";
        "MXV",
        "symbol";
        "MXV";
    }
    "MYR";
    {
        "displayName";
        "رينغيت ماليزي",
        "displayName-count-zero";
        "رينغيت ماليزي",
        "displayName-count-one";
        "رينغيت ماليزي",
        "displayName-count-two";
        "رينغيت ماليزي",
        "displayName-count-few";
        "رينغيت ماليزي",
        "displayName-count-many";
        "رينغيت ماليزي",
        "displayName-count-other";
        "رينغيت ماليزي",
        "symbol";
        "MYR",
        "symbol-alt-narrow";
        "RM";
    }

```

```

    }
    "MZE";
    {
        "displayName";
        "اسكود موزمبيقي",
        "symbol";
        "MZE";
    }
    "MZM";
    {
        "displayName";
        "MZM",
        "symbol";
        "MZM";
    }
    "MZN";
    {
        "displayName";
        "متكال موزمبيقي",
        "displayName-count-zero";
        "متكال موزمبيقي",
        "displayName-count-one";
        "متكال موزمبيقي",
        "displayName-count-two";
        "متكال موزمبيقي",
        "displayName-count-few";
        "متكال موزمبيقي",
        "displayName-count-many";
        "متكال موزمبيقي",
        "displayName-count-other";
        "متكال موزمبيقي",
        "symbol";
        "MZN";
    }
    "NAD";
    {
        "displayName";
        "دولار ناميبي",
        "displayName-count-zero";
        "دولار ناميبي",
        "displayName-count-one";
        "دولار ناميبي",
        "displayName-count-two";
        "دولار ناميبي",
        "displayName-count-few";
        "دولار ناميبي",
        "displayName-count-many";
        "دولار ناميبي",
        "displayName-count-other";
        "دولار ناميبي",
        "symbol";
        "NAD",
        "symbol-alt-narrow";
        "$";
    }
    "NGN";
    {

```

```

        "displayName";
        "نايرا نيجيري",
        "displayName-count-zero";
        "نايرا نيجيري",
        "displayName-count-one";
        "نايرا نيجيري",
        "displayName-count-two";
        "نايرا نيجيري",
        "displayName-count-few";
        "نايرا نيجيري",
        "displayName-count-many";
        "نايرا نيجيري",
        "displayName-count-other";
        "نايرا نيجيري",
        "symbol";
        "NGN",
        "symbol-alt-narrow";
        "₦";
    }
    "NIC";
    {
        "displayName";
        "كوردوبه نيكاراغوا",
        "symbol";
        "NIC";
    }
    "NIO";
    {
        "displayName";
        "قرطبة نيكاراغوا",
        "displayName-count-zero";
        "قرطبة نيكاراغوا",
        "displayName-count-one";
        "قرطبة نيكاراغوا",
        "displayName-count-two";
        "قرطبة نيكاراغوا",
        "displayName-count-few";
        "قرطبة نيكاراغوا",
        "displayName-count-many";
        "قرطبة نيكاراغوا",
        "displayName-count-other";
        "قرطبة نيكاراغوا",
        "symbol";
        "NIO",
        "symbol-alt-narrow";
        "C$";
    }
    "NLG";
    {
        "displayName";
        "جلدر هولندي",
        "symbol";
        "NLG";
    }
    "NOK";
    {
        "displayName";

```

```

        "كرونة نرويجية",
        "displayName-count-zero";
        "كرونة نرويجية",
        "displayName-count-one";
        "كرونة نرويجية",
        "displayName-count-two";
        "كرونة نرويجية",
        "displayName-count-few";
        "كرونة نرويجية",
        "displayName-count-many";
        "كرونة نرويجية",
        "displayName-count-other";
        "كرونة نرويجية",
        "symbol";
        "NOK",
        "symbol-alt-narrow";
        "kr";
    }
    "NPR";
    {
        "displayName";
        "روبية نيبالي",
        "displayName-count-zero";
        "روبية نيبالي",
        "displayName-count-one";
        "روبية نيبالي",
        "displayName-count-two";
        "روبية نيبالي",
        "displayName-count-few";
        "روبية نيبالي",
        "displayName-count-many";
        "روبية نيبالي",
        "displayName-count-other";
        "روبية نيبالي",
        "symbol";
        "NPR",
        "symbol-alt-narrow";
        "Rs";
    }
    "NZD";
    {
        "displayName";
        "دولار نيوزيلندي",
        "displayName-count-zero";
        "دولار نيوزيلندي",
        "displayName-count-one";
        "دولار نيوزيلندي",
        "displayName-count-two";
        "دولار نيوزيلندي",
        "displayName-count-few";
        "دولار نيوزيلندي",
        "displayName-count-many";
        "دولار نيوزيلندي",
        "displayName-count-other";
        "دولار نيوزيلندي",
        "symbol";
        "NZ$";
    }

```

```

        "symbol-alt-narrow";
        "NZ$";
    }
    "OMR";
    {
        "displayName";
        "ول عماني",
        "displayName-count-zero";
        "ول عماني",
        "displayName-count-one";
        "ول عماني",
        "displayName-count-two";
        "ول عماني",
        "displayName-count-few";
        "ول عماني",
        "displayName-count-many";
        "ول عماني",
        "displayName-count-other";
        "ول عماني",
        "symbol";
        "ر.ع.";
    }
    "PAB";
    {
        "displayName";
        "بالبوا بنمي",
        "displayName-count-zero";
        "بالبوا بنمي",
        "displayName-count-one";
        "بالبوا بنمي",
        "displayName-count-two";
        "بالبوا بنمي",
        "displayName-count-few";
        "بالبوا بنمي",
        "displayName-count-many";
        "بالبوا بنمي",
        "displayName-count-other";
        "بالبوا بنمي",
        "symbol";
        "PAB";
    }
    "PEI";
    {
        "displayName";
        "PEI",
        "symbol";
        "PEI";
    }
    "PEN";
    {
        "displayName";
        "سول جديد البيرو",
        "displayName-count-zero";
        "سول جديد البيرو",
        "displayName-count-one";
        "سول جديد البيرو",
        "displayName-count-two";
    }

```



```

        "سول جديد البيرو",
        "displayName-count-few";
        "سول جديد البيرو",
        "displayName-count-many";
        "سول جديد البيرو",
        "displayName-count-other";
        "سول جديد البيرو",
        "symbol";
        "PEN";
    }
    "PES";
    {
        "displayName";
        "PES",
        "symbol";
        "PES";
    }
    "PGK";
    {
        "displayName";
        "كيننا بابوا غينيا الجديدة",
        "displayName-count-zero";
        "كيننا بابوا غينيا الجديدة",
        "displayName-count-one";
        "كيننا بابوا غينيا الجديدة",
        "displayName-count-two";
        "كيننا بابوا غينيا الجديدة",
        "displayName-count-few";
        "كيننا بابوا غينيا الجديدة",
        "displayName-count-many";
        "كيننا بابوا غينيا الجديدة",
        "displayName-count-other";
        "كيننا بابوا غينيا الجديدة",
        "symbol";
        "PGK";
    }
    "PHP";
    {
        "displayName";
        "بيزو فلبيني",
        "displayName-count-zero";
        "بيزو فلبيني",
        "displayName-count-one";
        "بيزو فلبيني",
        "displayName-count-two";
        "بيزو فلبيني",
        "displayName-count-few";
        "بيزو فلبيني",
        "displayName-count-many";
        "بيزو فلبيني",
        "displayName-count-other";
        "بيزو فلبيني",
        "symbol";
        "PHP",
        "symbol-alt-narrow";
        "₱";
    }
}

```

```

"PKR";
{
  "displayName";
  "روبية باستاني",
  "displayName-count-zero";
  "روبية باستاني",
  "displayName-count-one";
  "روبية باستاني",
  "displayName-count-two";
  "روبية باستاني",
  "displayName-count-few";
  "روبية باستاني",
  "displayName-count-many";
  "روبية باستاني",
  "displayName-count-other";
  "روبية باستاني",
  "symbol";
  "PKR",
  "symbol-alt-narrow";
  "Rs";
}
"PLN";
{
  "displayName";
  "زلوتي بولندي",
  "displayName-count-zero";
  "زلوتي بولندي",
  "displayName-count-one";
  "زلوتي بولندي",
  "displayName-count-two";
  "زلوتي بولندي",
  "displayName-count-few";
  "زلوتي بولندي",
  "displayName-count-many";
  "زلوتي بولندي",
  "displayName-count-other";
  "زلوتي بولندي",
  "symbol";
  "PLN",
  "symbol-alt-narrow";
  "zł";
}
"PLZ";
{
  "displayName";
  "1995-1950 - زلوتي بولندي",
  "symbol";
  "PLZ";
}
"PTE";
{
  "displayName";
  "اسكود برتغالي",
  "symbol";
  "PTE";
}
"PYG";

```

```

{
  "displayName";
  "جواراني باراجوای",
  "displayName-count-zero";
  "جواراني باراجوای",
  "displayName-count-one";
  "جواراني باراجوای",
  "displayName-count-two";
  "جواراني باراجوای",
  "displayName-count-few";
  "جواراني باراجوای",
  "displayName-count-many";
  "جواراني باراجوای",
  "displayName-count-other";
  "جواراني باراجوای",
  "symbol";
  "PYG",
  "symbol-alt-narrow";
  "₲";
}
"QAR";
{
  "displayName";
  "ڤڊ قطري",
  "displayName-count-zero";
  "ڤڊ قطري",
  "displayName-count-one";
  "ڤڊ قطري",
  "displayName-count-two";
  "ڤڊ قطري",
  "displayName-count-few";
  "ڤڊ قطري",
  "displayName-count-many";
  "ڤڊ قطري",
  "displayName-count-other";
  "ڤڊ قطري",
  "symbol";
  "ر.ق.";
}
"RHD";
{
  "displayName";
  "ڊولار روڊيسي",
  "symbol";
  "RHD";
}
"ROL";
{
  "displayName";
  "ليو رومانى قديم",
  "symbol";
  "ROL";
}
"RON";
{
  "displayName";
  "ليو رومانى",

```

```

        "displayName-count-zero";
        "ليو روماني",
        "displayName-count-one";
        "ليو روماني",
        "displayName-count-two";
        "ليو روماني",
        "displayName-count-few";
        "ليو روماني",
        "displayName-count-many";
        "ليو روماني",
        "displayName-count-other";
        "ليو روماني",
        "symbol";
        "RON",
        "symbol-alt-narrow";
        "lei";
    }
    "RSD";
    {
        "displayName";
        "دينار صربي",
        "displayName-count-zero";
        "دينار صربي",
        "displayName-count-one";
        "دينار صربي",
        "displayName-count-two";
        "ديناران صربيان",
        "displayName-count-few";
        "دينارات صربية",
        "displayName-count-many";
        "دينارًا صربيًا",
        "displayName-count-other";
        "دينار صربي",
        "symbol";
        "RSD";
    }
    "RUB";
    {
        "displayName";
        "روبل روسي",
        "displayName-count-zero";
        "روبل روسي",
        "displayName-count-one";
        "روبل روسي",
        "displayName-count-two";
        "روبل روسي",
        "displayName-count-few";
        "روبل روسي",
        "displayName-count-many";
        "روبل روسي",
        "displayName-count-other";
        "روبل روسي",
        "symbol";
        "RUB",
        "symbol-alt-narrow";
        "₽";
    }
}

```

```

    "RUR";
    {
      "displayName";
      "1998-1991 - روبل روسي",
      "symbol";
      "RUR",
      "symbol-alt-narrow";
      "p.";
    }
    "RWF";
    {
      "displayName";
      "فرنك رواندي",
      "displayName-count-zero";
      "فرنك رواندي",
      "displayName-count-one";
      "فرنك رواندي",
      "displayName-count-two";
      "فرنك رواندي",
      "displayName-count-few";
      "فرنك رواندي",
      "displayName-count-many";
      "فرنك رواندي",
      "displayName-count-other";
      "فرنك رواندي",
      "symbol";
      "RWF",
      "symbol-alt-narrow";
      "RF";
    }
    "SAR";
    {
      "displayName";
      "رول سعودي",
      "displayName-count-zero";
      "رول سعودي",
      "displayName-count-one";
      "رول سعودي",
      "displayName-count-two";
      "رول سعودي",
      "displayName-count-few";
      "رول سعودي",
      "displayName-count-many";
      "رول سعودي",
      "displayName-count-other";
      "رول سعودي",
      "symbol";
      "ر.س.";
    }
    "SBD";
    {
      "displayName";
      "دولار جزر سليمان",
      "displayName-count-zero";
      "دولار جزر سليمان",
      "displayName-count-one";
      "دولار جزر سليمان",

```

```

        "displayName-count-two";
        "دولار جزر سليمان",
        "displayName-count-few";
        "دولار جزر سليمان",
        "displayName-count-many";
        "دولار جزر سليمان",
        "displayName-count-other";
        "دولار جزر سليمان",
        "symbol";
        "SBD",
        "symbol-alt-narrow";
        "SB$";
    }
    "SCR";
    {
        "displayName";
        "روبية سيشيلية",
        "displayName-count-zero";
        "روبية سيشيلية",
        "displayName-count-one";
        "روبية سيشيلية",
        "displayName-count-two";
        "روبية سيشيلية",
        "displayName-count-few";
        "روبية سيشيلية",
        "displayName-count-many";
        "روبية سيشيلية",
        "displayName-count-other";
        "روبية سيشيلية",
        "symbol";
        "SCR";
    }
    "SDD";
    {
        "displayName";
        "دينار سوداني",
        "symbol";
        "د.س.";
    }
    "SDG";
    {
        "displayName";
        "جنيه سوداني",
        "displayName-count-zero";
        "جنيه سوداني",
        "displayName-count-one";
        "جنيه سوداني",
        "displayName-count-two";
        "جنيه سوداني",
        "displayName-count-few";
        "جنيهات سودانية",
        "displayName-count-many";
        "جنيهاً سودانياً",
        "displayName-count-other";
        "جنيه سوداني",
        "symbol";
        "ج.س.";
    }

```

```

    }
    "SDP";
    {
        "displayName";
        "جنيه سوداني قديم",
        "symbol";
        "SDP";
    }
    "SEK";
    {
        "displayName";
        "كرونة سويدية",
        "displayName-count-zero";
        "كرونة سويدية",
        "displayName-count-one";
        "كرونة سويدية",
        "displayName-count-two";
        "كرونة سويدية",
        "displayName-count-few";
        "كرونة سويدية",
        "displayName-count-many";
        "كرونة سويدية",
        "displayName-count-other";
        "كرونة سويدية",
        "symbol";
        "SEK",
        "symbol-alt-narrow";
        "kr";
    }
    "SGD";
    {
        "displayName";
        "دولار سنغافوري",
        "displayName-count-zero";
        "دولار سنغافوري",
        "displayName-count-one";
        "دولار سنغافوري",
        "displayName-count-two";
        "دولار سنغافوري",
        "displayName-count-few";
        "دولار سنغافوري",
        "displayName-count-many";
        "دولار سنغافوري",
        "displayName-count-other";
        "دولار سنغافوري",
        "symbol";
        "SGD",
        "symbol-alt-narrow";
        "$";
    }
    "SHP";
    {
        "displayName";
        "جنيه سانت هيلين",
        "displayName-count-zero";
        "جنيه سانت هيلين",
        "displayName-count-one";
    }

```

```

        "جنيه سانت هيلين",
        "displayName-count-two";
        "جنيه سانت هيلين",
        "displayName-count-few";
        "جنيه سانت هيلين",
        "displayName-count-many";
        "جنيه سانت هيلين",
        "displayName-count-other";
        "جنيه سانت هيلين",
        "symbol";
        "SHP",
        "symbol-alt-narrow";
        "£";
    }
    "SIT";
    {
        "displayName";
        "تولار سلوفيني",
        "symbol";
        "SIT";
    }
    "SKK";
    {
        "displayName";
        "كرونه سلوفاكية",
        "symbol";
        "SKK";
    }
    "SLL";
    {
        "displayName";
        "ليون سيراليوني",
        "displayName-count-zero";
        "ليون سيراليوني",
        "displayName-count-one";
        "ليون سيراليوني",
        "displayName-count-two";
        "ليون سيراليوني",
        "displayName-count-few";
        "ليون سيراليوني",
        "displayName-count-many";
        "ليون سيراليوني",
        "displayName-count-other";
        "ليون سيراليوني",
        "symbol";
        "SLL";
    }
    "SOS";
    {
        "displayName";
        "شلن صومالي",
        "displayName-count-zero";
        "شلن صومالي",
        "displayName-count-one";
        "شلن صومالي",
        "displayName-count-two";
        "شلن صومالي",
    }

```



```

        "displayName-count-few";
        "شلن صومالي",
        "displayName-count-many";
        "شلن صومالي",
        "displayName-count-other";
        "شلن صومالي",
        "symbol";
        "SOS";
    }
    "SRD";
    {
        "displayName";
        "دولار سورينامي",
        "displayName-count-zero";
        "دولار سورينامي",
        "displayName-count-one";
        "دولار سورينامي",
        "displayName-count-two";
        "دولار سورينامي",
        "displayName-count-few";
        "دولار سورينامي",
        "displayName-count-many";
        "دولار سورينامي",
        "displayName-count-other";
        "دولار سورينامي",
        "symbol";
        "SRD",
        "symbol-alt-narrow";
        "SR$";
    }
    "SRG";
    {
        "displayName";
        "جلدر سورينامي",
        "symbol";
        "SRG";
    }
    "SSP";
    {
        "displayName";
        "جنيه جنوب السودان",
        "displayName-count-zero";
        "جنيه جنوب السودان",
        "displayName-count-one";
        "جنيه جنوب السودان",
        "displayName-count-two";
        "جنيهان جنوب السودان",
        "displayName-count-few";
        "جنيهات جنوب السودان",
        "displayName-count-many";
        "جنيها جنوب السودان",
        "displayName-count-other";
        "جنيه جنوب السودان",
        "symbol";
        "SSP",
        "symbol-alt-narrow";
        "£";
    }

```

```

    }
    "STD";
    {
        "displayName";
        "دوبرا ساو تومي وبرينسيبي",
        "displayName-count-zero";
        "دوبرا ساو تومي وبرينسيبي",
        "displayName-count-one";
        "دوبرا ساو تومي وبرينسيبي",
        "displayName-count-two";
        "دوبرا ساو تومي وبرينسيبي",
        "displayName-count-few";
        "دوبرا ساو تومي وبرينسيبي",
        "displayName-count-many";
        "دوبرا ساو تومي وبرينسيبي",
        "displayName-count-other";
        "دوبرا ساو تومي وبرينسيبي",
        "symbol";
        "STD",
        "symbol-alt-narrow";
        "Db";
    }
    "SUR";
    {
        "displayName";
        "روبل سوفيتي",
        "symbol";
        "SUR";
    }
    "SVC";
    {
        "displayName";
        "كولون سلفادوري",
        "symbol";
        "SVC";
    }
    "SYP";
    {
        "displayName";
        "ليرة سورية",
        "displayName-count-zero";
        "ليرة سورية",
        "displayName-count-one";
        "ليرة سورية",
        "displayName-count-two";
        "ليرة سورية",
        "displayName-count-few";
        "ليرة سورية",
        "displayName-count-many";
        "ليرة سورية",
        "displayName-count-other";
        "ليرة سورية",
        "symbol";
        "ل.س.",
        "symbol-alt-narrow";
        "£";
    }
}

```

```

"SZL";
{
  "displayName";
  "ليلانجيني سوازيلندي",
  "displayName-count-zero";
  "ليلانجيني سوازيلندي",
  "displayName-count-one";
  "ليلانجيني سوازيلندي",
  "displayName-count-two";
  "ليلانجيني سوازيلندي",
  "displayName-count-few";
  "ليلانجيني سوازيلندي",
  "displayName-count-many";
  "ليلانجيني سوازيلندي",
  "displayName-count-other";
  "ليلانجيني سوازيلندي",
  "symbol";
  "SZL";
}
"THB";
{
  "displayName";
  "باخت تايلاندي",
  "displayName-count-zero";
  "باخت تايلاندي",
  "displayName-count-one";
  "باخت تايلاندي",
  "displayName-count-two";
  "باخت تايلاندي",
  "displayName-count-few";
  "باخت تايلاندي",
  "displayName-count-many";
  "باخت تايلاندي",
  "displayName-count-other";
  "باخت تايلاندي",
  "symbol";
  "฿",
  "symbol-alt-narrow";
  "฿";
}
"TJR";
{
  "displayName";
  "روبل طاجيكستاني",
  "symbol";
  "TJR";
}
"TJS";
{
  "displayName";
  "سوموني طاجيكستاني",
  "displayName-count-zero";
  "سوموني طاجيكستاني",
  "displayName-count-one";
  "سوموني طاجيكستاني",
  "displayName-count-two";
  "سوموني طاجيكستاني",

```

```

        "displayName-count-few";
        "سوموني طاجيكستاني",
        "displayName-count-many";
        "سوموني طاجيكستاني",
        "displayName-count-other";
        "سوموني طاجيكستاني",
        "symbol";
        "TJS";
    }
    "TMM";
    {
        "displayName";
        "مانات تركمنستاني",
        "symbol";
        "TMM";
    }
    "TMT";
    {
        "displayName";
        "مانات تركمانستان",
        "displayName-count-zero";
        "مانات تركمانستان",
        "displayName-count-one";
        "مانات تركمانستان",
        "displayName-count-two";
        "مانات تركمانستان",
        "displayName-count-few";
        "مانات تركمانستان",
        "displayName-count-many";
        "مانات تركمانستان",
        "displayName-count-other";
        "مانات تركمانستان",
        "symbol";
        "TMT";
    }
    "TND";
    {
        "displayName";
        "دينار تونسي",
        "displayName-count-zero";
        "دينار تونسي",
        "displayName-count-one";
        "دينار تونسي",
        "displayName-count-two";
        "ديناران تونسيان",
        "displayName-count-few";
        "دينارات تونسية",
        "displayName-count-many";
        "دينارًا تونسيًا",
        "displayName-count-other";
        "دينار تونسي",
        "symbol";
        "د.ت.";
    }
    "TOP";
    {
        "displayName";

```

```

        "بانغا تونغ",
        "displayName-count-zero";
        "بانغا تونغ",
        "displayName-count-one";
        "بانغا تونغ",
        "displayName-count-two";
        "بانغا تونغ",
        "displayName-count-few";
        "بانغا تونغ",
        "displayName-count-many";
        "بانغا تونغ",
        "displayName-count-other";
        "بانغا تونغ",
        "symbol";
        "TOP",
        "symbol-alt-narrow";
        "T$";
    }
    "TPE";
    {
        "displayName";
        "اسكود تيموري",
        "symbol";
        "TPE";
    }
    "TRL";
    {
        "displayName";
        "ليرة تركي",
        "symbol";
        "TRL";
    }
    "TRY";
    {
        "displayName";
        "ليرة تركية",
        "displayName-count-zero";
        "ليرة تركية",
        "displayName-count-one";
        "ليرة تركية",
        "displayName-count-two";
        "ليرة تركية",
        "displayName-count-few";
        "ليرة تركية",
        "displayName-count-many";
        "ليرة تركية",
        "displayName-count-other";
        "ليرة تركية",
        "symbol";
        "TRY",
        "symbol-alt-narrow";
        "₺",
        "symbol-alt-variant";
        "TL";
    }
    "TTD";
    {

```

```

        "displayName";
        "دولار ترينداد وتوباجو",
        "displayName-count-zero";
        "دولار ترينداد وتوباجو",
        "displayName-count-one";
        "دولار ترينداد وتوباجو",
        "displayName-count-two";
        "دولار ترينداد وتوباجو",
        "displayName-count-few";
        "دولار ترينداد وتوباجو",
        "displayName-count-many";
        "دولار ترينداد وتوباجو",
        "displayName-count-other";
        "دولار ترينداد وتوباجو",
        "symbol";
        "TTD",
        "symbol-alt-narrow";
        "TT$";
    }
    "TWD";
    {
        "displayName";
        "دولار تايواني",
        "displayName-count-zero";
        "دولار تايواني",
        "displayName-count-one";
        "دولار تايواني",
        "displayName-count-two";
        "دولار تايواني",
        "displayName-count-few";
        "دولار تايواني",
        "displayName-count-many";
        "دولار تايواني",
        "displayName-count-other";
        "دولار تايواني",
        "symbol";
        "NT$";
        "symbol-alt-narrow";
        "NT$";
    }
    "TZS";
    {
        "displayName";
        "شلن تنزاني",
        "displayName-count-zero";
        "شلن تنزاني",
        "displayName-count-one";
        "شلن تنزاني",
        "displayName-count-two";
        "شلن تنزاني",
        "displayName-count-few";
        "شلن تنزاني",
        "displayName-count-many";
        "شلن تنزاني",
        "displayName-count-other";
        "شلن تنزاني",
        "symbol";
    }

```

```

        "TZS";
    }
    "UAH";
    {
        "displayName";
        "هريفنيا أوكراني",
        "displayName-count-zero";
        "هريفنيا أوكراني",
        "displayName-count-one";
        "هريفنيا أوكراني",
        "displayName-count-two";
        "هريفنيا أوكراني",
        "displayName-count-few";
        "هريفنيا أوكراني",
        "displayName-count-many";
        "هريفنيا أوكراني",
        "displayName-count-other";
        "هريفنيا أوكراني",
        "symbol";
        "UAH",
        "symbol-alt-narrow";
        "₴";
    }
    "UAK";
    {
        "displayName";
        "UAK",
        "symbol";
        "UAK";
    }
    "UGS";
    {
        "displayName";
        "شلن أوغندي - 1966-1987",
        "symbol";
        "UGS";
    }
    "UGX";
    {
        "displayName";
        "شلن أوغندي",
        "displayName-count-zero";
        "شلن أوغندي",
        "displayName-count-one";
        "شلن أوغندي",
        "displayName-count-two";
        "شلن أوغندي",
        "displayName-count-few";
        "شلن أوغندي",
        "displayName-count-many";
        "شلن أوغندي",
        "displayName-count-other";
        "شلن أوغندي",
        "symbol";
        "UGX";
    }
    "USD";

```

```

{
  "displayName";
  "دولار أمريكي",
  "displayName-count-zero";
  "دولار أمريكي",
  "displayName-count-one";
  "دولار أمريكي",
  "displayName-count-two";
  "دولار أمريكي",
  "displayName-count-few";
  "دولار أمريكي",
  "displayName-count-many";
  "دولار أمريكي",
  "displayName-count-other";
  "دولار أمريكي",
  "symbol";
  "US$";
  "symbol-alt-narrow";
  "US$";
}
"USN";
{
  "displayName";
  "دولار أمريكي (اليوم التالي)",
  "symbol";
  "USN";
}
"USS";
{
  "displayName";
  "دولار أمريكي (نفس اليوم)",
  "symbol";
  "USS";
}
"UYI";
{
  "displayName";
  "UYI",
  "symbol";
  "UYI";
}
"UYP";
{
  "displayName";
  "بیزو أوروغواي - 1993-1975",
  "symbol";
  "UYP";
}
"UYU";
{
  "displayName";
  "بیزو اوروغواي",
  "displayName-count-zero";
  "بیزو اوروغواي",
  "displayName-count-one";
  "بیزو اوروغواي",
  "displayName-count-two";
}

```



```

        "بیزو اوروغوای",
        "displayName-count-few";
        "بیزو اوروغوای",
        "displayName-count-many";
        "بیزو اوروغوای",
        "displayName-count-other";
        "بیزو اوروغوای",
        "symbol";
        "UYU",
        "symbol-alt-narrow";
        "UY$";
    }
    "UZS";
    {
        "displayName";
        "سوم اۆزبکستانی",
        "displayName-count-zero";
        "سوم اۆزبکستانی",
        "displayName-count-one";
        "سوم اۆزبکستانی",
        "displayName-count-two";
        "سوم اۆزبکستانی",
        "displayName-count-few";
        "سوم اۆزبکستانی",
        "displayName-count-many";
        "سوم اۆزبکستانی",
        "displayName-count-other";
        "سوم اۆزبکستانی",
        "symbol";
        "UZS";
    }
    "VEB";
    {
        "displayName";
        "بولیفار فنزویلی - 2008-1871",
        "symbol";
        "VEB";
    }
    "VEF";
    {
        "displayName";
        "بولیفار فنزویلی",
        "displayName-count-zero";
        "بولیفار فنزویلی",
        "displayName-count-one";
        "بولیفار فنزویلی",
        "displayName-count-two";
        "بولیفار فنزویلی",
        "displayName-count-few";
        "بولیفار فنزویلی",
        "displayName-count-many";
        "بولیفار فنزویلی",
        "displayName-count-other";
        "بولیفار فنزویلی",
        "symbol";
        "VEF",
        "symbol-alt-narrow";
    }

```

```

        "Bs";
    }
    "VND";
    {
        "displayName";
        "دونج فيتنامي",
        "displayName-count-zero";
        "دونج فيتنامي",
        "displayName-count-one";
        "دونج فيتنامي",
        "displayName-count-two";
        "دونج فيتنامي",
        "displayName-count-few";
        "دونج فيتنامي",
        "displayName-count-many";
        "دونج فيتنامي",
        "displayName-count-other";
        "دونج فيتنامي",
        "symbol";
        "₫",
        "symbol-alt-narrow";
        "₫";
    }
    "VNN";
    {
        "displayName";
        "VNN",
        "symbol";
        "VNN";
    }
    "VUV";
    {
        "displayName";
        "فاتو فانواتو",
        "displayName-count-zero";
        "فاتو فانواتو",
        "displayName-count-one";
        "فاتو فانواتو",
        "displayName-count-two";
        "فاتو فانواتو",
        "displayName-count-few";
        "فاتو فانواتو",
        "displayName-count-many";
        "فاتو فانواتو",
        "displayName-count-other";
        "فاتو فانواتو",
        "symbol";
        "VUV";
    }
    "WST";
    {
        "displayName";
        "تالا ساموا",
        "displayName-count-zero";
        "تالا ساموا",
        "displayName-count-one";
        "تالا ساموا",
    }

```

```

        "displayName-count-two";
        "تالا ساموا",
        "displayName-count-few";
        "تالا ساموا",
        "displayName-count-many";
        "تالا ساموا",
        "displayName-count-other";
        "تالا ساموا",
        "symbol";
        "WST";
    }
    "XAF";
    {
        "displayName";
        "فرنك وسط أفريقي",
        "displayName-count-zero";
        "فرنك وسط أفريقي",
        "displayName-count-one";
        "فرنك وسط أفريقي",
        "displayName-count-two";
        "فرنك وسط أفريقي",
        "displayName-count-few";
        "فرنك وسط أفريقي",
        "displayName-count-many";
        "فرنك وسط أفريقي",
        "displayName-count-other";
        "فرنك وسط أفريقي",
        "symbol";
        "FCFA";
    }
    "XAG";
    {
        "displayName";
        "فضة",
        "symbol";
        "XAG";
    }
    "XAU";
    {
        "displayName";
        "ذهب",
        "symbol";
        "XAU";
    }
    "XBA";
    {
        "displayName";
        "الوحدة الأوروبية المركبة",
        "symbol";
        "XBA";
    }
    "XBB";
    {
        "displayName";
        "الوحدة المالية الأوروبية",
        "symbol";
        "XBB";
    }

```

```

    }
    "XBC";
    {
      "displayName";
      "الوحدة الحسابية الأوروبية",
      "symbol";
      "XBC";
    }
    "XBD";
    {
      "displayName";
      "وحدة الحساب الأوروبية (XBD)",
      "symbol";
      "XBD";
    }
    "XCD";
    {
      "displayName";
      "دولار شرق الكاريبي",
      "displayName-count-zero";
      "دولار شرق الكاريبي",
      "displayName-count-one";
      "دولار شرق الكاريبي",
      "displayName-count-two";
      "دولار شرق الكاريبي",
      "displayName-count-few";
      "دولار شرق الكاريبي",
      "displayName-count-many";
      "دولار شرق الكاريبي",
      "displayName-count-other";
      "دولار شرق الكاريبي",
      "symbol";
      "EC$";
      "symbol-alt-narrow";
      "$";
    }
    "XDR";
    {
      "displayName";
      "حقوق السحب الخاصة",
      "symbol";
      "XDR";
    }
    "XEU";
    {
      "displayName";
      "وحدة النقد الأوروبية",
      "symbol";
      "XEU";
    }
    "XFO";
    {
      "displayName";
      "فرنك فرنسي ذهبي",
      "symbol";
      "XFO";
    }
  }

```

```

"XFU";
{
  "displayName";
  "فرنك فرنسي (UIC)",
  "symbol";
  "XFU";
}
"XOF";
{
  "displayName";
  "فرنك غرب أفريقي",
  "displayName-count-zero";
  "فرنك غرب أفريقي",
  "displayName-count-one";
  "فرنك غرب أفريقي",
  "displayName-count-two";
  "فرنك غرب أفريقي",
  "displayName-count-few";
  "فرنك غرب أفريقي",
  "displayName-count-many";
  "فرنك غرب أفريقي",
  "displayName-count-other";
  "فرنك غرب أفريقي",
  "symbol";
  "CFA";
}
"XPD";
{
  "displayName";
  "بالاديوم",
  "symbol";
  "XPD";
}
"XPF";
{
  "displayName";
  "فرنك سي إف بي",
  "displayName-count-zero";
  "فرنك سي إف بي",
  "displayName-count-one";
  "فرنك سي إف بي",
  "displayName-count-two";
  "فرنك سي إف بي",
  "displayName-count-few";
  "فرنك سي إف بي",
  "displayName-count-many";
  "فرنك سي إف بي",
  "displayName-count-other";
  "فرنك سي إف بي",
  "symbol";
  "CFPF";
}
"XPT";
{
  "displayName";
  "البلاتين",
  "symbol";

```

```

        "XPT";
    }
    "XRE";
    {
        "displayName";
        "XRE",
        "symbol";
        "XRE";
    }
    "XSU";
    {
        "displayName";
        "XSU",
        "symbol";
        "XSU";
    }
    "XTS";
    {
        "displayName";
        "كود اختبار العملة",
        "symbol";
        "XTS";
    }
    "XUA";
    {
        "displayName";
        "XUA",
        "symbol";
        "XUA";
    }
    "XXX";
    {
        "displayName";
        "عملة غير معروفة",
        "displayName-count-zero";
        "(عملة غير معروفة)",
        "displayName-count-one";
        "(عملة غير معروفة)",
        "displayName-count-two";
        "(عملة غير معروفة)",
        "displayName-count-few";
        "(عملة غير معروفة)",
        "displayName-count-many";
        "(عملة غير معروفة)",
        "displayName-count-other";
        "(عملة غير معروفة)",
        "symbol";
        "***";
    }
    "YDD";
    {
        "displayName";
        "دينار يمني",
        "symbol";
        "YDD";
    }
    "YER";

```

```

    {
      "displayName";
      "ول يمني",
      "displayName-count-zero";
      "ول يمني",
      "displayName-count-one";
      "ول يمني",
      "displayName-count-two";
      "ول يمني",
      "displayName-count-few";
      "ول يمني",
      "displayName-count-many";
      "ول يمني",
      "displayName-count-other";
      "ول يمني",
      "symbol";
      "ر.ي.";
    }
    "YUD";
    {
      "displayName";
      "دينار يوغسلافي",
      "symbol";
      "YUD";
    }
    "YUM";
    {
      "displayName";
      "YUM",
      "symbol";
      "YUM";
    }
    "YUN";
    {
      "displayName";
      "دينار يوغسلافي قابل للتحويل",
      "symbol";
      "YUN";
    }
    "YUR";
    {
      "displayName";
      "YUR",
      "symbol";
      "YUR";
    }
    "ZAL";
    {
      "displayName";
      "راند جنوب أفريقيا -مالي",
      "symbol";
      "ZAL";
    }
    "ZAR";
    {
      "displayName";
      "راند جنوب أفريقيا",

```

```

        "displayName-count-zero";
        "راند جنوب أفريقيا",
        "displayName-count-one";
        "راند جنوب أفريقيا",
        "displayName-count-two";
        "راند جنوب أفريقيا",
        "displayName-count-few";
        "راند جنوب أفريقيا",
        "displayName-count-many";
        "راند جنوب أفريقيا",
        "displayName-count-other";
        "راند جنوب أفريقيا",
        "symbol";
        "ZAR",
        "symbol-alt-narrow";
        "R";
    }
    "ZMK";
    {
        "displayName";
        "2012-1968 - زامبي",
        "displayName-count-zero";
        "2012-1968 - زامبي",
        "displayName-count-one";
        "2012-1968 - زامبي",
        "displayName-count-two";
        "2012-1968 - زامبي",
        "displayName-count-few";
        "2012-1968 - زامبي",
        "displayName-count-many";
        "2012-1968 - زامبي",
        "displayName-count-other";
        "2012-1968 - زامبي",
        "symbol";
        "ZMK";
    }
    "ZMW";
    {
        "displayName";
        "زامبي",
        "displayName-count-zero";
        "زامبي",
        "displayName-count-one";
        "زامبي",
        "displayName-count-two";
        "زامبي",
        "displayName-count-few";
        "زامبي",
        "displayName-count-many";
        "زامبي",
        "displayName-count-other";
        "زامبي",
        "symbol";
        "ZMW",
        "symbol-alt-narrow";
        "ZK";
    }
}

```



```

        "ZRN";
    {
        "displayName";
        "زائير زائييري جديد",
        "symbol";
        "ZRN";
    }
    "ZRZ";
    {
        "displayName";
        "زائير زائييري",
        "symbol";
        "ZRZ";
    }
    "ZWD";
    {
        "displayName";
        "دولار زمبابوي",
        "symbol";
        "ZWD";
    }
    "ZWL";
    {
        "displayName";
        "دولار زمبابوي 2009",
        "symbol";
        "ZWL";
    }
    "ZWR";
    {
        "displayName";
        "ZWR",
        "symbol";
        "ZWR";
    }
}

}

}

}

```

## INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
//Load the L10n, loadCldr from ej2-base
import { loadCldr, L10n } from '@syncfusion/ej2-base';
//import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
//import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
// enable ripple effect
enableRipple(true);
//load the CLDR data files.
// Here we have referred local json files for preview purpose
import * as numberingSystems from './numberingSystems.json';
```

```
import * as gregorian from './ca-gregorian.json';
import * as numbers from './numbers.json';
loadCldr(numberingSystems, gregorian, numbers);
L10n.load({
  'ar': {
    'timepicker': { placeholder: 'حدد الوقت' }
  }
});
//import the timepickercomponent
function App() {
  let rtl = true;
  return <TimePickerComponent id="time" locale='ar' enableRtl={rtl}/>;
}
;
ReactDOM.render(<App />, document.getElementById('timer'));
```

**INDEX.TSX**

```
import * as React from "react";
import * as ReactDOM from "react-dom";
//Load the L10n, loadCldr from ej2-base
import { loadCldr, L10n } from '@syncfusion/ej2-base';
//import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
//import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
// enable ripple effect
enableRipple(true);
//load the CLDR data files.
// Here we have referred local json files for preview purpose
import * as numberingSystems from './numberingSystems.json';
import * as gregorian from './ca-gregorian.json';
import * as numbers from './numbers.json';
loadCldr(numberingSystems, gregorian, numbers);
L10n.load({
  'ar': {
    'timepicker': { placeholder: 'حدد الوقت' }
  }
});
//import the timepickercomponent
function App() {
  let rtl:boolean=true;
  return <TimePickerComponent id="time" locale='ar' enableRtl={rtl} />
};
ReactDOM.render(<App />, document.getElementById('timer'));
```

**MAIN.JSX**

```
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
import { enableProdMode } from '@angular/core';
import { AppModule } from './app.module';
enableProdMode();
platformBrowserDynamic().bootstrapModule(AppModule);
```

**MAIN.TS**

```
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
import { enableProdMode } from '@angular/core';
import { AppModule } from './app.module';
enableProdMode();
platformBrowserDynamic().bootstrapModule(AppModule);
```

### NUMBERINGSYSTEMS.JSON

```
{
  "supplemental": {
    "version": {
      "_number": "$Revision: 12732 $",
      "_unicodeVersion": "9.0.0",
      "_cldrVersion": "31"
    },
    "numberingSystems": {
      "adlm": {
        "_digits": "ᲐᲑᲒᲓᲔᲕᲖᲗᲘᲙ",
        "_type": "numeric"
      },
      "ahom": {
        "_digits": "ᩌᩍᩎᩏᩐᩑᩒᩓᩔᩕ",
        "_type": "numeric"
      },
      "arab": {
        "_digits": "٠١٢٣٤٥٦٧٨٩",
        "_type": "numeric"
      },
      "arabext": {
        "_digits": "٠١٢٣٤٥٦٧٨٩",
        "_type": "numeric"
      },
      "armn": {
        "_rules": "armenian-upper",
        "_type": "algorithmic"
      },
      "armnlow": {
        "_rules": "armenian-lower",
        "_type": "algorithmic"
      },
      "bali": {
        "_digits": "ᬂᬃᬄᬅᬆᬇᬈᬉᬊᬋ",
        "_type": "numeric"
      },
      "beng": {
        "_digits": "০১২৩৪৫৬৭৮৯",
        "_type": "numeric"
      },
      "bhks": {
        "_digits": "᩠ᩡᩢᩣᩤᩥᩦᩧᩨᩩ",
        "_type": "numeric"
      },
      "brah": {
        "_digits": "᱀᱁᱂᱃᱄᱅᱆᱇᱈᱉᱊",
        "_type": "numeric"
      }
    }
  }
}
```

```
,
  "cakm": {
    "_digits": "᠐ᠸᠲ᠐ᠨᠠᠨᠢᠨᠠᠨᠢ",
    "_type": "numeric"
  },
  "cham": {
    "_digits": "᠐᠐᠐᠐᠐᠐᠐᠐᠐᠐",
    "_type": "numeric"
  },
  "cyr1": {
    "_rules": "cyrillic-lower",
    "_type": "algorithmic"
  },
  "deva": {
    "_digits": "०१२३४५६७८९",
    "_type": "numeric"
  },
  "ethi": {
    "_rules": "ethiopic",
    "_type": "algorithmic"
  },
  "fullwide": {
    "_digits": "0 1 2 3 4 5 6 7 8 9",
    "_type": "numeric"
  },
  "geor": {
    "_rules": "georgian",
    "_type": "algorithmic"
  },
  "grek": {
    "_rules": "greek-upper",
    "_type": "algorithmic"
  },
  "greklow": {
    "_rules": "greek-lower",
    "_type": "algorithmic"
  },
  "gujr": {
    "_digits": "૦૧૨૩૪૫૬૭૮૯",
    "_type": "numeric"
  },
  "guru": {
    "_digits": "੦੧੨੩੪੫੬੭੮੯",
    "_type": "numeric"
  },
  "hanidays": {
    "_rules": "zh/SpelloutRules/spellout-numbering-days",
    "_type": "algorithmic"
  },
  "hanidec": {
    "_digits": "〇一二三四五六七八九",
    "_type": "numeric"
  },
  "hans": {
    "_rules": "zh/SpelloutRules/spellout-cardinal",
```

```

    "_type": "algorithmic"
  },
  "hansfin": {
    "_rules": "zh/SpelloutRules/spellout-cardinal-financial",
    "_type": "algorithmic"
  },
  "hant": {
    "_rules": "zh_Hant/SpelloutRules/spellout-cardinal",
    "_type": "algorithmic"
  },
  "hantfin": {
    "_rules": "zh_Hant/SpelloutRules/spellout-cardinal-financial",
    "_type": "algorithmic"
  },
  "hebr": {
    "_rules": "hebrew",
    "_type": "algorithmic"
  },
  "hmng": {
    "_digits": "□□□□□□□□□□",
    "_type": "numeric"
  },
  "java": {
    "_digits": "᐀ᐁᐂᐃᐄᐅᐆᐇᐈᐉᐊᐋᐌᐍᐎᐏᐐᐑᐒᐓᐔᐕᐖᐗᐘᐙᐚᐛᐜᐝᐞᐟᐠᐡᐢᐣᐤᐥᐦᐧᐨᐩᐪᐫᐬᐭᐮᐯᐰᐱᐲᐳᐴᐵᐶᐷᐸᐹᐺᐻᐼᐽᐾᐿᑀᑁᑂᑃᑄᑅᑆᑇᑈᑉᑊᑋᑌᑍᑎᑏᑐᑑᑒᑓᑔᑕᑖᑗᑘᑙᑚᑛᑜᑝᑞᑟᑠᑡᑢᑣᑤᑥᑦᑧᑨᑩᑪᑫᑬᑭᑮᑯᑰᑱᑲᑳᑴᑵᑶᑷᑸᑹᑺᑻᑼᑾᑿ᐀ᐁᐂᐃᐄᐅᐆᐇᐈᐉᐊᐋᐌᐍᐎᐏᐐᐑᐒᐓᐔᐕᐖᐗᐘᐙᐚᐛᐜᐝᐞᐟᐠᐡᐢᐣᐤᐥᐦᐧᐨᐩᐪᐫᐬᐭᐮᐯᐰᐱᐲᐳᐴᐵᐶᐷᐸᐹᐺᐻᐼᐽᐾᐿᑀᑁᑂᑃᑄᑅᑆᑇᑈᑉᑊᑋᑌᑍᑎᑏᑐᑑᑒᑓᑔᑕᑖᑗᑘᑙᑚᑛᑜᑝᑞᑟᑠᑡᑢᑣᑤᑥᑦᑧᑨᑩᑪᑫᑬᑭᑮᑯᑰᑱᑲᑳᑴᑵᑶᑷᑸᑹᑺᑻᑼᑾᑿ",
    "_type": "numeric"
  },
  "jpan": {
    "_rules": "ja/SpelloutRules/spellout-cardinal",
    "_type": "algorithmic"
  },
  "jpanfin": {
    "_rules": "ja/SpelloutRules/spellout-cardinal-financial",
    "_type": "algorithmic"
  },
  "kali": {
    "_digits": "□□□□□□□□□□",
    "_type": "numeric"
  },
  "khmr": {
    "_digits": "០១២៣៤៥៦៧៨៩",
    "_type": "numeric"
  },
  "knda": {
    "_digits": "೦೧೨೩೪೫೬೭೮೯",
    "_type": "numeric"
  },
  "lana": {
    "_digits": "□□□□□□□□□□",
    "_type": "numeric"
  },
  "lanatham": {
    "_digits": "□□□□□□□□□□",
    "_type": "numeric"
  },
  "lao": {

```

```

    "_digits": "൦൧൨൩൪൫൬൭൮൯",
    "_type": "numeric"
  },
  "latn": {
    "_digits": "0123456789",
    "_type": "numeric"
  },
  "lepc": {
    "_digits": "□□□□□□□□□□",
    "_type": "numeric"
  },
  "limb": {
    "_digits": "□□□□□□□□□□",
    "_type": "numeric"
  },
  "mathbold": {
    "_digits": "0123456789",
    "_type": "numeric"
  },
  "mathdbl": {
    "_digits": "0123456789",
    "_type": "numeric"
  },
  "mathmono": {
    "_digits": "0123456789",
    "_type": "numeric"
  },
  "mathsanb": {
    "_digits": "0123456789",
    "_type": "numeric"
  },
  "mathsans": {
    "_digits": "0123456789",
    "_type": "numeric"
  },
  "mlym": {
    "_digits": "൦൧൨൩൪൫൬൭൮൯",
    "_type": "numeric"
  },
  "modi": {
    "_digits": "□□□□□□□□□□",
    "_type": "numeric"
  },
  "mong": {
    "_digits": "᠐᠑᠒᠓᠐ᠠᠨᠨᠠᠭ",
    "_type": "numeric"
  },
  "mroo": {
    "_digits": "□□□□□□□□□□",
    "_type": "numeric"
  },
  "mtei": {
    "_digits": "ႤႤႤႤႤႤႤႤႤႤ",
    "_type": "numeric"
  },
},

```

```

    "mymr": {
      "_digits": "၀၁၂၃၄၅၆၇၈၉",
      "_type": "numeric"
    },
    "mymrshan": {
      "_digits": "၀၁၂၃၄၅၆၇၈၉",
      "_type": "numeric"
    },
    "mymrtlng": {
      "_digits": "၀၁၂၃၄၅၆၇၈၉",
      "_type": "numeric"
    },
    "newa": {
      "_digits": "၀၁၂၃၄၅၆၇၈၉",
      "_type": "numeric"
    },
    "nkoo": {
      "_digits": "၀၁၂၃၄၅၆၇၈၉",
      "_type": "numeric"
    },
    "olck": {
      "_digits": "၀၁၂၃၄၅၆၇၈၉",
      "_type": "numeric"
    },
    "orya": {
      "_digits": "၀၁၂၃၄၅၆၇၈၉",
      "_type": "numeric"
    },
    "osma": {
      "_digits": "၀၁၂၃၄၅၆၇၈၉",
      "_type": "numeric"
    },
    "roman": {
      "_rules": "roman-upper",
      "_type": "algorithmic"
    },
    "romanlow": {
      "_rules": "roman-lower",
      "_type": "algorithmic"
    },
    "saur": {
      "_digits": "၀၁၂၃၄၅၆၇၈၉",
      "_type": "numeric"
    },
    "shrd": {
      "_digits": "၀၁၂၃၄၅၆၇၈၉",
      "_type": "numeric"
    },
    "sind": {
      "_digits": "၀၁၂၃၄၅၆၇၈၉",
      "_type": "numeric"
    },
    "sinh": {
      "_digits": "၀၁၂၃၄၅၆၇၈၉",

```

```

    "_type": "numeric"
  },
  "sora": {
    "_digits": "ᱵᱚᱠᱟᱨᱢᱟ",
    "_type": "numeric"
  },
  "sund": {
    "_digits": "ᱵᱚᱠᱟᱨᱢᱟ",
    "_type": "numeric"
  },
  "takr": {
    "_digits": "ᱵᱚᱠᱟᱨᱢᱟ",
    "_type": "numeric"
  },
  "talv": {
    "_digits": "ᱵᱚᱠᱟᱨᱢᱟ",
    "_type": "numeric"
  },
  "taml": {
    "_rules": "tamil",
    "_type": "algorithmic"
  },
  "tamldec": {
    "_digits": "0௧௨௩௪௫௬௭௮௯",
    "_type": "numeric"
  },
  "telu": {
    "_digits": "౦౧౨౩౪౫౬౭౮౯",
    "_type": "numeric"
  },
  "thai": {
    "_digits": "๐๑๒๓๔๕๖๗๘๙",
    "_type": "numeric"
  },
  "tibb": {
    "_digits": "༠༡༢༣༤༥༦༧༨༩",
    "_type": "numeric"
  },
  "tirh": {
    "_digits": "ᱵᱚᱠᱟᱨᱢᱟ",
    "_type": "numeric"
  },
  "vaih": {
    "_digits": "ᱠᱚᱱᱚᱛᱚᱠᱚᱡᱚᱢᱚᱣᱚ",
    "_type": "numeric"
  },
  "wara": {
    "_digits": "ᱵᱚᱠᱟᱨᱢᱟ",
    "_type": "numeric"
  }
}
}
}

```



**NUMBERINGSYSTEMS.JSX**

```

{
  "supplemental";
  {
    "version";
    {
      "_number";
      "$Revision: 12732 $",
      "_unicodeVersion";
      "9.0.0",
      "_cldrVersion";
      "31";
    }
    "numberingSystems";
    {
      "adlm";
      {
        "_digits";
        "ᲐᲑᲒᲓᲔᲕᲖᲗᲘᲙ",
        "_type";
        "numeric";
      }
      "ahom";
      {
        "_digits";
        "ᩌᩍᩎᩏᩐᩑᩒᩓᩔᩕᩖᩗᩘᩙᩚᩛᩜᩝᩞ᩟᩠ᩡᩢᩣᩤᩥᩦᩧᩨᩩᩪᩫᩬᩭᩮᩯᩰᩱᩲᩳᩴ᩵᩶᩷᩸᩹᩺᩻᩼᩽᩾᩿",
        "_type";
        "numeric";
      }
      "arab";
      {
        "_digits";
        "٠١٢٣٤٥٦٧٨٩",
        "_type";
        "numeric";
      }
      "arabext";
      {
        "_digits";
        "٠١٢٣٤٥٦٧٨٩",
        "_type";
        "numeric";
      }
      "armn";
      {
        "_rules";
        "armenian-upper",
        "_type";
        "algorithmic";
      }
      "armnlow";
      {
        "_rules";
        "armenian-lower",
        "_type";
        "algorithmic";
      }
    }
  }
}

```

```

    }
    "bali";
    {
        "_digits";
        "□□□□□□□□",
        "_type";
        "numeric";
    }
    "beng";
    {
        "_digits";
        "০১২৩৪৫৬৭৮৯",
        "_type";
        "numeric";
    }
    "bhks";
    {
        "_digits";
        "□□□□□□□□",
        "_type";
        "numeric";
    }
    "brah";
    {
        "_digits";
        "·ᱠᱡᱢᱫᱷᱚᱴᱚᱨ",
        "_type";
        "numeric";
    }
    "cakm";
    {
        "_digits";
        "ႤႬႤႬႬႬႬႬႬႬ",
        "_type";
        "numeric";
    }
    "cham";
    {
        "_digits";
        "□□□□□□□□",
        "_type";
        "numeric";
    }
    "cyr1";
    {
        "_rules";
        "cyrillic-lower",
        "_type";
        "algorithmic";
    }
    "deva";
    {
        "_digits";
        "०१२३४५६७८९",
        "_type";
        "numeric";
    }

```

```
}
"ethi";
{
  "_rules";
  "ethiopic",
  "_type";
  "algorithmic";
}
"fullwide";
{
  "_digits";
  "0 1 2 3 4 5 6 7 8 9",
  "_type";
  "numeric";
}
"geor";
{
  "_rules";
  "georgian",
  "_type";
  "algorithmic";
}
"grek";
{
  "_rules";
  "greek-upper",
  "_type";
  "algorithmic";
}
"greklow";
{
  "_rules";
  "greek-lower",
  "_type";
  "algorithmic";
}
"gujr";
{
  "_digits";
  "૦ ૧ ૨ ૩ ૪ ૫ ૬ ૭ ૮ ૯",
  "_type";
  "numeric";
}
"guru";
{
  "_digits";
  "੦ ੧ ੨ ੩ ੪ ੫ ੬ ੭ ੮ ੯",
  "_type";
  "numeric";
}
"hanidays";
{
  "_rules";
  "zh/SpelloutRules/spellout-numbering-days",
  "_type";
  "algorithmic";
}
```

```

    }
    "hanidec";
    {
        "_digits";
        "〇一二三四五六七八九",
        "_type";
        "numeric";
    }
    "hans";
    {
        "_rules";
        "zh/SpelloutRules/spellout-cardinal",
        "_type";
        "algorithmic";
    }
    "hansfin";
    {
        "_rules";
        "zh/SpelloutRules/spellout-cardinal-financial",
        "_type";
        "algorithmic";
    }
    "hant";
    {
        "_rules";
        "zh_Hant/SpelloutRules/spellout-cardinal",
        "_type";
        "algorithmic";
    }
    "hantfin";
    {
        "_rules";
        "zh_Hant/SpelloutRules/spellout-cardinal-financial",
        "_type";
        "algorithmic";
    }
    "hebr";
    {
        "_rules";
        "hebrew",
        "_type";
        "algorithmic";
    }
    "hmng";
    {
        "_digits";
        "□□□□□□□□□□",
        "_type";
        "numeric";
    }
    "java";
    {
        "_digits";
        "௦௧௨௩௪௫௬௭௮௯௦௧௨௩௪௫௬௭௮௯௦",
        "_type";
    }

```

```

        "numeric";
    }
    "jpan";
    {
        "_rules";
        "ja/SpelloutRules/spellout-cardinal",
        "_type";
        "algorithmic";
    }
    "jpanfin";
    {
        "_rules";
        "ja/SpelloutRules/spellout-cardinal-financial",
        "_type";
        "algorithmic";
    }
    "kali";
    {
        "_digits";
        "□□□□□□□□",
        "_type";
        "numeric";
    }
    "khamr";
    {
        "_digits";
        "ᱠᱣᱟᱨᱢᱟᱝᱞᱟᱹ",
        "_type";
        "numeric";
    }
    "knda";
    {
        "_digits";
        "೦೧೨೩೪೫೬೭೮೯",
        "_type";
        "numeric";
    }
    "lana";
    {
        "_digits";
        "□□□□□□□□",
        "_type";
        "numeric";
    }
    "lanatham";
    {
        "_digits";
        "□□□□□□□□",
        "_type";
        "numeric";
    }
    "lao";
    {
        "_digits";
        "ᦅᧄᦺᦑᦟᦹᧉ",
        "_type";
    }

```

```

        "numeric";
    }
    "latn";
    {
        "_digits";
        "0123456789",
        "_type";
        "numeric";
    }
    "lepc";
    {
        "_digits";
        "□□□□□□□□□□",
        "_type";
        "numeric";
    }
    "limb";
    {
        "_digits";
        "□□□□□□□□□□",
        "_type";
        "numeric";
    }
    "mathbold";
    {
        "_digits";
        "0123456789",
        "_type";
        "numeric";
    }
    "mathdbl";
    {
        "_digits";
        "0123456789",
        "_type";
        "numeric";
    }
    "mathmono";
    {
        "_digits";
        "0123456789",
        "_type";
        "numeric";
    }
    "mathsanb";
    {
        "_digits";
        "0123456789",
        "_type";
        "numeric";
    }
    "mathsans";
    {
        "_digits";
        "0123456789",
        "_type";

```

```

        "numeric";
    }
    "mlym";
    {
        "_digits";
        "൦൧൨൩൪൫൬൭൮൯",
        "_type";
        "numeric";
    }
    "modi";
    {
        "_digits";
        "□□□□□□□□□□",
        "_type";
        "numeric";
    }
    "mong";
    {
        "_digits";
        "᠐᠑᠒᠓᠐ᠠᠨᠨᠠᠭ",
        "_type";
        "numeric";
    }
    "mroo";
    {
        "_digits";
        "□□□□□□□□□□",
        "_type";
        "numeric";
    }
    "mtei";
    {
        "_digits";
        "ႤႬႬႬႬႬႬႬႬႬႬႬ",
        "_type";
        "numeric";
    }
    "mymr";
    {
        "_digits";
        "၀၁၂၃၄၅၆၇၈",
        "_type";
        "numeric";
    }
    "mymrshan";
    {
        "_digits";
        "0123456788",
        "_type";
        "numeric";
    }
    "mymrtlng";
    {
        "_digits";

```

[illegible]



```

        "_digits";
        "□□□□□□□□",
        "_type";
        "numeric";
    }
    "shrd";
    {
        "_digits";
        "□□□□□□□□",
        "_type";
        "numeric";
    }
    "sind";
    {
        "_digits";
        "□□□□□□□□",
        "_type";
        "numeric";
    }
    "sinh";
    {
        "_digits";
        "අනුමාප්පුරුණි",
        "_type";
        "numeric";
    }
    "sora";
    {
        "_digits";
        "೦೧೨೩೪೫೬೭",
        "_type";
        "numeric";
    }
    "sund";
    {
        "_digits";
        "□□□□□□□□",
        "_type";
        "numeric";
    }
    "takr";
    {
        "_digits";
        "□□□□□□□□",
        "_type";
        "numeric";
    }
    "talv";
    {
        "_digits";
        "တၢ်သုးလၢ",
        "_type";
        "numeric";
    }
    "taml";
    {

```

```

        "_rules";
        "tamil",
        "_type";
        "algorithmic";
    }
    "tamldec";
    {
        "_digits";
        "0௧௨௩௪௫௬௭௮௯",
        "_type";
        "numeric";
    }
    "telu";
    {
        "_digits";
        "౦౧౨౩౪౫౬౭౮౯",
        "_type";
        "numeric";
    }
    "thai";
    {
        "_digits";
        "๐๑๒๓๔๕๖๗๘๙",
        "_type";
        "numeric";
    }
    "tibtb";
    {
        "_digits";
        "༠༡༢༣༤༥༦༧༨༩",
        "_type";
        "numeric";
    }
    "tirh";
    {
        "_digits";
        "୦୧୨୩୪୫୬୭୮୯",
        "_type";
        "numeric";
    }
    "vaih";
    {
        "_digits";
        "୦୧୨୩୪୫୬୭୮୯",
        "_type";
        "numeric";
    }
    "wara";
    {
        "_digits";
        "୦୧୨୩୪୫୬୭୮୯",
        "_type";
        "numeric";
    }
    }
}

```

```
}
```

## NUMBERS.JSON

```
{
  "main": {
    "ar": {
      "identity": {
        "version": {
          "_number": "$Revision: 12879 $",
          "_cldrVersion": "30.0.3"
        },
        "language": "ar"
      },
      "numbers": {
        "defaultNumberingSystem": "arab",
        "otherNumberingSystems": {
          "native": "arab"
        },
        "minimumGroupingDigits": "1",
        "symbols-numberSystem-arab": {
          "decimal": ",",
          "group": ",",
          "list": ":",
          "percentSign": "%",
          "plusSign": "+",
          "minusSign": "-",
          "exponential": "١٠",
          "superscriptingExponent": "×",
          "perMille": "‰",
          "infinity": "∞",
          "nan": "ليس رقم",
          "timeSeparator": ":"
        },
        "symbols-numberSystem-latn": {
          "decimal": ".",
          "group": ",",
          "list": ";",
          "percentSign": "%",
          "plusSign": "+",
          "minusSign": "-",
          "exponential": "E",
          "superscriptingExponent": "×",
          "perMille": "‰",
          "infinity": "∞",
          "nan": "ليس رقمًا",
          "timeSeparator": ":"
        },
        "decimalFormats-numberSystem-arab": {
          "standard": "#,##0.###",
          "long": {
            "decimalFormat": {
              "1000-count-zero": "0 ألف",
              "1000-count-one": "0 ألف",
              "1000-count-two": "0 ألف",
              "1000-count-few": "0 آلاف"
            }
          }
        }
      }
    }
  }
}
```

```
"1000-count-many": "0 ألف",
"1000-count-other": "0 ألف",
"10000-count-zero": "00 ألف",
"10000-count-one": "00 ألف",
"10000-count-two": "00 ألف",
"10000-count-few": "00 ألف",
"10000-count-many": "00 ألف",
"10000-count-other": "00 ألف",
"100000-count-zero": "000 ألف",
"100000-count-one": "000 ألف",
"100000-count-two": "000 ألف",
"100000-count-few": "000 ألف",
"100000-count-many": "000 ألف",
"100000-count-other": "000 ألف",
"1000000-count-zero": "0 مليون",
"1000000-count-one": "0 مليون",
"1000000-count-two": "0 مليون",
"1000000-count-few": "0 ملايين",
"1000000-count-many": "0 مليون",
"1000000-count-other": "0 مليون",
"10000000-count-zero": "00 مليون",
"10000000-count-one": "00 مليون",
"10000000-count-two": "00 مليون",
"10000000-count-few": "00 ملايين",
"10000000-count-many": "00 مليون",
"10000000-count-other": "00 مليون",
"100000000-count-zero": "000 مليون",
"100000000-count-one": "000 مليون",
"100000000-count-two": "000 مليون",
"100000000-count-few": "000 مليون",
"100000000-count-many": "000 مليون",
"100000000-count-other": "000 مليون",
"1000000000-count-zero": "0 مليار",
"1000000000-count-one": "0 مليار",
"1000000000-count-two": "0 مليار",
"1000000000-count-few": "0 مليار",
"1000000000-count-many": "0 مليار",
"1000000000-count-other": "0 مليار",
"10000000000-count-zero": "00 مليار",
"10000000000-count-one": "00 مليار",
"10000000000-count-two": "00 مليار",
"10000000000-count-few": "00 مليار",
"10000000000-count-many": "00 مليار",
"10000000000-count-other": "00 مليار",
"100000000000-count-zero": "000 مليار",
"100000000000-count-one": "000 مليار",
"100000000000-count-two": "000 مليار",
"100000000000-count-few": "000 مليار",
"100000000000-count-many": "000 مليار",
"100000000000-count-other": "000 مليار",
"1000000000000-count-zero": "0 تريليون",
"1000000000000-count-one": "0 تريليون",
"1000000000000-count-two": "0 تريليون",
"1000000000000-count-few": "0 تريليونات",
"1000000000000-count-many": "0 تريليون",
"1000000000000-count-other": "0 تريليون",
"10000000000000-count-zero": "00 تريليون",
```

```

        "10000000000000-count-one": "00 تريليون",
        "10000000000000-count-two": "00 تريليون",
        "10000000000000-count-few": "00 تريليون",
        "10000000000000-count-many": "00 تريليون",
        "10000000000000-count-other": "00 تريليون",
        "10000000000000-count-zero": "000 تريليون",
        "10000000000000-count-one": "000 تريليون",
        "10000000000000-count-two": "000 تريليون",
        "10000000000000-count-few": "000 تريليون",
        "10000000000000-count-many": "000 تريليون",
        "10000000000000-count-other": "000 تريليون"
    }
},
"short": {
    "decimalFormat": {
        "1000-count-zero": "0 ألف",
        "1000-count-one": "0 ألف",
        "1000-count-two": "0 ألف",
        "1000-count-few": "0 آلاف",
        "1000-count-many": "0 ألف",
        "1000-count-other": "0 ألف",
        "10000-count-zero": "00 ألف",
        "10000-count-one": "00 ألف",
        "10000-count-two": "00 ألف",
        "10000-count-few": "00 ألف",
        "10000-count-many": "00 ألف",
        "10000-count-other": "00 ألف",
        "100000-count-zero": "000 ألف",
        "100000-count-one": "000 ألف",
        "100000-count-two": "000 ألف",
        "100000-count-few": "000 ألف",
        "100000-count-many": "000 ألف",
        "100000-count-other": "000 ألف",
        "1000000-count-zero": "0 مليو",
        "1000000-count-one": "0 مليو",
        "1000000-count-two": "0 مليو",
        "1000000-count-few": "0 مليو",
        "1000000-count-many": "0 مليو",
        "1000000-count-other": "0 مليو",
        "10000000-count-zero": "00 مليو",
        "10000000-count-one": "00 مليو",
        "10000000-count-two": "00 مليو",
        "10000000-count-few": "00 مليو",
        "10000000-count-many": "00 مليو",
        "10000000-count-other": "00 مليو",
        "100000000-count-zero": "000 مليو",
        "100000000-count-one": "000 مليو",
        "100000000-count-two": "000 مليو",
        "100000000-count-few": "000 مليو",
        "100000000-count-many": "000 مليو",
        "100000000-count-other": "000 مليو",
        "1000000000-count-zero": "0 مليا",
        "1000000000-count-one": "0 مليا",
        "1000000000-count-two": "0 مليا",
        "1000000000-count-few": "0 مليا",
        "1000000000-count-many": "0 مليا",
        "1000000000-count-other": "0 مليا",
    }
}

```

```

        "10000000000-count-zero": "00 مليا",
        "10000000000-count-one": "00 مليا",
        "10000000000-count-two": "00 مليا",
        "10000000000-count-few": "00 مليا",
        "10000000000-count-many": "00 مليا",
        "10000000000-count-other": "00 مليا",
        "100000000000-count-zero": "000 مليا",
        "100000000000-count-one": "000 مليا",
        "100000000000-count-two": "000 مليا",
        "100000000000-count-few": "000 مليا",
        "100000000000-count-many": "000 مليا",
        "100000000000-count-other": "000 مليا",
        "1000000000000-count-zero": "0 ترليو",
        "1000000000000-count-one": "0 ترليو",
        "1000000000000-count-two": "0 ترليو",
        "1000000000000-count-few": "0 ترليو",
        "1000000000000-count-many": "0 ترليو",
        "1000000000000-count-other": "0 ترليو",
        "10000000000000-count-zero": "00 ترليو",
        "10000000000000-count-one": "00 ترليو",
        "10000000000000-count-two": "00 ترليو",
        "10000000000000-count-few": "00 ترليو",
        "10000000000000-count-many": "00 ترليو",
        "10000000000000-count-other": "00 ترليو",
        "100000000000000-count-zero": "000 ترليو",
        "100000000000000-count-one": "000 ترليو",
        "100000000000000-count-two": "000 ترليو",
        "100000000000000-count-few": "000 ترليو",
        "100000000000000-count-many": "000 ترليو",
        "100000000000000-count-other": "000 ترليو"
    }
}
},
"decimalFormats-numberSystem-latn": {
    "standard": "#,##0.###",
    "long": {
        "decimalFormat": {
            "1000-count-zero": "0 ألف",
            "1000-count-one": "0 ألف",
            "1000-count-two": "0 ألف",
            "1000-count-few": "0 آلاف",
            "1000-count-many": "0 ألف",
            "1000-count-other": "0 ألف",
            "10000-count-zero": "00 ألف",
            "10000-count-one": "00 ألف",
            "10000-count-two": "00 ألف",
            "10000-count-few": "00 ألف",
            "10000-count-many": "00 ألف",
            "10000-count-other": "00 ألف",
            "100000-count-zero": "000 ألف",
            "100000-count-one": "000 ألف",
            "100000-count-two": "000 ألف",
            "100000-count-few": "000 ألف",
            "100000-count-many": "000 ألف",
            "100000-count-other": "000 ألف",
            "1000000-count-zero": "0 مليون",
            "1000000-count-one": "0 مليون",

```

```

        "1000000-count-two": "0 مليون",
        "1000000-count-few": "0 ملايين",
        "1000000-count-many": "0 مليون",
        "1000000-count-other": "0 مليون",
        "10000000-count-zero": "00 مليون",
        "10000000-count-one": "00 مليون",
        "10000000-count-two": "00 مليون",
        "10000000-count-few": "00 ملايين",
        "10000000-count-many": "00 مليون",
        "10000000-count-other": "00 مليون",
        "100000000-count-zero": "000 مليون",
        "100000000-count-one": "000 مليون",
        "100000000-count-two": "000 مليون",
        "100000000-count-few": "000 مليون",
        "100000000-count-many": "000 مليون",
        "100000000-count-other": "000 مليون",
        "1000000000-count-zero": "0 مليار",
        "1000000000-count-one": "0 مليار",
        "1000000000-count-two": "0 مليار",
        "1000000000-count-few": "0 مليار",
        "1000000000-count-many": "0 مليار",
        "1000000000-count-other": "0 مليار",
        "10000000000-count-zero": "00 مليار",
        "10000000000-count-one": "00 مليار",
        "10000000000-count-two": "00 مليار",
        "10000000000-count-few": "00 مليار",
        "10000000000-count-many": "00 مليار",
        "10000000000-count-other": "00 مليار",
        "100000000000-count-zero": "000 مليار",
        "100000000000-count-one": "000 مليار",
        "100000000000-count-two": "000 مليار",
        "100000000000-count-few": "000 مليار",
        "100000000000-count-many": "000 مليار",
        "100000000000-count-other": "000 مليار",
        "1000000000000-count-zero": "0 تريليون",
        "1000000000000-count-one": "0 تريليون",
        "1000000000000-count-two": "0 تريليون",
        "1000000000000-count-few": "0 تريليونات",
        "1000000000000-count-many": "0 تريليون",
        "1000000000000-count-other": "0 تريليون",
        "10000000000000-count-zero": "00 تريليون",
        "10000000000000-count-one": "00 تريليون",
        "10000000000000-count-two": "00 تريليون",
        "10000000000000-count-few": "00 تريليون",
        "10000000000000-count-many": "00 تريليون",
        "10000000000000-count-other": "00 تريليون",
        "100000000000000-count-zero": "000 تريليون",
        "100000000000000-count-one": "000 تريليون",
        "100000000000000-count-two": "000 تريليون",
        "100000000000000-count-few": "000 تريليون",
        "100000000000000-count-many": "000 تريليون",
        "100000000000000-count-other": "000 تريليون"
    },
    },
    "short": {
        "decimalFormat": {
            "1000-count-zero": "0 ألف",

```

```
"1000-count-one": "0 ألف",
"1000-count-two": "0 ألف",
"1000-count-few": "0 آلاف",
"1000-count-many": "0 ألف",
"1000-count-other": "0 ألف",
"10000-count-zero": "00 ألف",
"10000-count-one": "00 ألف",
"10000-count-two": "00 ألف",
"10000-count-few": "00 ألف",
"10000-count-many": "00 ألف",
"10000-count-other": "00 ألف",
"100000-count-zero": "000 ألف",
"100000-count-one": "000 ألف",
"100000-count-two": "000 ألف",
"100000-count-few": "000 ألف",
"100000-count-many": "000 ألف",
"100000-count-other": "000 ألف",
"1000000-count-zero": "0 مليون",
"1000000-count-one": "0 مليون",
"1000000-count-two": "0 مليون",
"1000000-count-few": "0 مليون",
"1000000-count-many": "0 مليون",
"1000000-count-other": "0 مليون",
"10000000-count-zero": "00 مليون",
"10000000-count-one": "00 مليون",
"10000000-count-two": "00 مليون",
"10000000-count-few": "00 مليون",
"10000000-count-many": "00 مليون",
"10000000-count-other": "00 مليون",
"100000000-count-zero": "000 مليون",
"100000000-count-one": "000 مليون",
"100000000-count-two": "000 مليون",
"100000000-count-few": "000 مليون",
"100000000-count-many": "000 مليون",
"100000000-count-other": "000 مليون",
"1000000000-count-zero": "0 مليا",
"1000000000-count-one": "0 مليا",
"1000000000-count-two": "0 مليا",
"1000000000-count-few": "0 مليا",
"1000000000-count-many": "0 مليا",
"1000000000-count-other": "0 مليا",
"10000000000-count-zero": "00 مليا",
"10000000000-count-one": "00 مليا",
"10000000000-count-two": "00 مليا",
"10000000000-count-few": "00 مليا",
"10000000000-count-many": "00 مليا",
"10000000000-count-other": "00 مليا",
"100000000000-count-zero": "000 مليا",
"100000000000-count-one": "000 مليا",
"100000000000-count-two": "000 مليا",
"100000000000-count-few": "000 مليا",
"100000000000-count-many": "000 مليا",
"100000000000-count-other": "000 مليا",
"1000000000000-count-zero": "0 ترليو",
"1000000000000-count-one": "0 ترليو",
"1000000000000-count-two": "0 ترليو",
"1000000000000-count-few": "0 ترليو",
```



```

        "1000000000000-count-many": "0 ترلیو",
        "1000000000000-count-other": "0 ترلیو",
        "1000000000000-count-zero": "00 ترلیو",
        "1000000000000-count-one": "00 ترلیو",
        "1000000000000-count-two": "00 ترلیو",
        "1000000000000-count-few": "00 ترلیو",
        "1000000000000-count-many": "00 ترلیو",
        "1000000000000-count-other": "00 ترلیو",
        "1000000000000-count-zero": "000 ترلیو",
        "1000000000000-count-one": "000 ترلیو",
        "1000000000000-count-two": "000 ترلیو",
        "1000000000000-count-few": "000 ترلیو",
        "1000000000000-count-many": "000 ترلیو",
        "1000000000000-count-other": "000 ترلیو"
    }
}
},
"scientificFormats-numberSystem-arab": {
    "standard": "#E0"
},
"scientificFormats-numberSystem-latn": {
    "standard": "#E0"
},
"percentFormats-numberSystem-arab": {
    "standard": "#,##0 %"
},
"percentFormats-numberSystem-latn": {
    "standard": "#,##0%"
},
"currencyFormats-numberSystem-arab": {
    "currencySpacing": {
        "beforeCurrency": {
            "currencyMatch": "[:^S:]",
            "surroundingMatch": "[:digit:]",
            "insertBetween": " "
        },
        "afterCurrency": {
            "currencyMatch": "[:^S:]",
            "surroundingMatch": "[:digit:]",
            "insertBetween": " "
        }
    },
    "standard": "#,##0.00 ¤",
    "accounting": "#,##0.00 ¤",
    "unitPattern-count-zero": "{0} {1}",
    "unitPattern-count-one": "{0} {1}",
    "unitPattern-count-two": "{0} {1}",
    "unitPattern-count-few": "{0} {1}",
    "unitPattern-count-many": "{0} {1}",
    "unitPattern-count-other": "{0} {1}"
},
"currencyFormats-numberSystem-latn": {
    "currencySpacing": {
        "beforeCurrency": {
            "currencyMatch": "[:^S:]",
            "surroundingMatch": "[:digit:]",
            "insertBetween": " "

```

```

    },
    "afterCurrency": {
      "currencyMatch": "[:^S:]",
      "surroundingMatch": "[:digit:]",
      "insertBetween": " "
    }
  },
  "standard": "¤ #,##0.00",
  "accounting": "¤#,##0.00; (¤#,##0.00)",
  "short": {
    "standard": {
      "1000-count-zero": "¤ ألف 0",
      "1000-count-one": "¤ ألف 1",
      "1000-count-two": "¤ ألف 2",
      "1000-count-few": "¤ ألف 3",
      "1000-count-many": "¤ ألف 4",
      "1000-count-other": "¤ ألف 5",
      "10000-count-zero": "¤ ألف 00",
      "10000-count-one": "¤ ألف 01",
      "10000-count-two": "¤ ألف 02",
      "10000-count-few": "¤ ألف 03",
      "10000-count-many": "¤ ألف 04",
      "10000-count-other": "¤ ألف 05",
      "100000-count-zero": "¤ ألف 000",
      "100000-count-one": "¤ ألف 001",
      "100000-count-two": "¤ ألف 002",
      "100000-count-few": "¤ ألف 003",
      "100000-count-many": "¤ ألف 004",
      "100000-count-other": "¤ ألف 005",
      "1000000-count-zero": "¤ مليون 0",
      "1000000-count-one": "¤ مليون 1",
      "1000000-count-two": "¤ مليون 2",
      "1000000-count-few": "¤ مليون 3",
      "1000000-count-many": "¤ مليون 4",
      "1000000-count-other": "¤ مليون 5",
      "10000000-count-zero": "¤ مليون 00",
      "10000000-count-one": "¤ مليون 01",
      "10000000-count-two": "¤ مليون 02",
      "10000000-count-few": "¤ مليون 03",
      "10000000-count-many": "¤ مليون 04",
      "10000000-count-other": "¤ مليون 05",
      "100000000-count-zero": "¤ مليا 0",
      "100000000-count-one": "¤ مليا 1",
      "100000000-count-two": "¤ مليا 2",
      "100000000-count-few": "¤ مليا 3",
      "100000000-count-many": "¤ مليا 4",
      "100000000-count-other": "¤ مليا 5",
      "1000000000-count-zero": "¤ مليا 00",
      "1000000000-count-one": "¤ مليا 01",
      "1000000000-count-two": "¤ مليا 02",
      "1000000000-count-few": "¤ مليا 03",
      "1000000000-count-many": "¤ مليا 04",
      "1000000000-count-other": "¤ مليا 05"
    }
  }
}

```

```

        "10000000000-count-many": "ملياً 00",
        "10000000000-count-other": "ملياً 00",
        "10000000000-count-zero": "ملياً 000",
        "10000000000-count-one": "ملياً 000",
        "10000000000-count-two": "ملياً 000",
        "10000000000-count-few": "ملياً 000",
        "10000000000-count-many": "ملياً 000",
        "10000000000-count-other": "ملياً 000",
        "10000000000-count-zero": "ترليو 0",
        "10000000000-count-one": "ترليو 0",
        "10000000000-count-two": "ترليو 0",
        "10000000000-count-few": "ترليو 0",
        "10000000000-count-many": "ترليو 0",
        "10000000000-count-other": "ترليو 0",
        "10000000000-count-zero": "ترليو 00",
        "10000000000-count-one": "ترليو 00",
        "10000000000-count-two": "ترليو 00",
        "10000000000-count-few": "ترليو 00",
        "10000000000-count-many": "ترليو 00",
        "10000000000-count-other": "ترليو 00",
        "10000000000-count-zero": "ترليو 000",
        "10000000000-count-one": "ترليو 000",
        "10000000000-count-two": "ترليو 000",
        "10000000000-count-few": "ترليو 000",
        "10000000000-count-many": "ترليو 000",
        "10000000000-count-other": "ترليو 000"
    },
    },
    "unitPattern-count-zero": "{0} {1}",
    "unitPattern-count-one": "{0} {1}",
    "unitPattern-count-two": "{0} {1}",
    "unitPattern-count-few": "{0} {1}",
    "unitPattern-count-many": "{0} {1}",
    "unitPattern-count-other": "{0} {1}"
},
"miscPatterns-numberSystem-arab": {
    "atLeast": "+{0}",
    "range": "{0}-{1}"
},
"miscPatterns-numberSystem-latn": {
    "atLeast": "+{0}",
    "range": "{0}-{1}"
}
}
}
}
}
}

```

## NUMBERS.JSX

```

{
    "main";
    {
        "ar";
        {
            "identity";

```

```

    {
      "version";
      {
        "_number";
        "$Revision: 12879 $",
        "_cldrVersion";
        "30.0.3";
      }
      "language";
      "ar";
    }
    "numbers";
    {
      "defaultNumberingSystem";
      "arab",
      "otherNumberingSystems";
      {
        "native";
        "arab";
      }
      "minimumGroupingDigits";
      "1",
      "symbols-numberSystem-arab";
      {
        "decimal";
        ",",
        "group";
        "list";
        ":",
        "percentSign";
        "%",
        "plusSign";
        "+",
        "minusSign";
        "-",
        "exponential";
        "١٠",
        "superscriptingExponent";
        "x",
        "perMille";
        "%",
        "infinity";
        "∞",
        "nan";
        "ليس رقم",
        "timeSeparator";
        ":";
      }
      "symbols-numberSystem-latn";
      {
        "decimal";
        ".",
        "group";
        ",",
        "list";
        ";";
      }
    }
  }

```

```

        "percentSign";
        "%",
        "plusSign";
        "+",
        "minusSign";
        "-",
        "exponential";
        "E",
        "superscriptingExponent";
        "×",
        "perMille";
        "‰",
        "infinity";
        "∞",
        "nan";
        "ليس رقمًا",
        "timeSeparator";
        ":";
    }
    "decimalFormats-numberSystem-arab";
    {
        "standard";
        "#,##0.###",
        "long";
        {
            "decimalFormat";
            {
                "1000-count-zero";
                "0 ألف",
                "1000-count-one";
                "0 ألف",
                "1000-count-two";
                "0 ألف",
                "1000-count-few";
                "0 آلاف",
                "1000-count-many";
                "0 ألف",
                "1000-count-other";
                "0 ألف",
                "10000-count-zero";
                "00 ألف",
                "10000-count-one";
                "00 ألف",
                "10000-count-two";
                "00 ألف",
                "10000-count-few";
                "00 ألف",
                "10000-count-many";
                "00 ألف",
                "10000-count-other";
                "00 ألف",
                "100000-count-zero";
                "000 ألف",
                "100000-count-one";
                "000 ألف",
                "100000-count-two";
                "000 ألف",
            }
        }
    }

```

```

        "100000-count-few";
    "000 ألف",
        "100000-count-many";
    "000 ألف",
        "100000-count-other";
    "000 ألف",
        "1000000-count-zero";
    "0 مليون",
        "1000000-count-one";
    "0 مليون",
        "1000000-count-two";
    "0 مليون",
        "1000000-count-few";
    "0 ملايين",
        "1000000-count-many";
    "0 مليون",
        "1000000-count-other";
    "0 مليون",
        "10000000-count-zero";
    "00 مليون",
        "10000000-count-one";
    "00 مليون",
        "10000000-count-two";
    "00 مليون",
        "10000000-count-few";
    "00 ملايين",
        "10000000-count-many";
    "00 مليون",
        "10000000-count-other";
    "00 مليون",
        "100000000-count-zero";
    "000 مليون",
        "100000000-count-one";
    "000 مليون",
        "100000000-count-two";
    "000 مليون",
        "100000000-count-few";
    "000 مليون",
        "100000000-count-many";
    "000 مليون",
        "100000000-count-other";
    "000 مليون",
        "1000000000-count-zero";
    "0 مليار",
        "1000000000-count-one";
    "0 مليار",
        "1000000000-count-two";
    "0 مليار",
        "1000000000-count-few";
    "0 مليار",
        "1000000000-count-many";
    "0 مليار",
        "1000000000-count-other";
    "0 مليار",
        "10000000000-count-zero";
    "00 مليار",
        "10000000000-count-one";

```

```

"00 مليار",
"10000000000-count-two";
"00 مليار",
"10000000000-count-few";
"00 مليار",
"10000000000-count-many";
"00 مليار",
"10000000000-count-other";
"00 مليار",
"10000000000-count-zero";
"000 مليار",
"100000000000-count-one";
"000 مليار",
"100000000000-count-two";
"000 مليار",
"100000000000-count-few";
"000 مليار",
"100000000000-count-many";
"000 مليار",
"100000000000-count-other";
"000 مليار",
"100000000000-count-zero";
"0 تريليون",
"1000000000000-count-one";
"0 تريليون",
"1000000000000-count-two";
"0 تريليون",
"1000000000000-count-few";
"0 تريليونات",
"10000000000000-count-many";
"0 تريليون",
"10000000000000-count-other";
"0 تريليون",
"10000000000000-count-zero";
"00 تريليون",
"10000000000000-count-one";
"00 تريليون",
"10000000000000-count-two";
"00 تريليون",
"10000000000000-count-few";
"00 تريليون",
"10000000000000-count-many";
"00 تريليون",
"10000000000000-count-other";
"00 تريليون",
"10000000000000-count-zero";
"000 تريليون",
"100000000000000-count-one";
"000 تريليون",
"100000000000000-count-two";
"000 تريليون",
"100000000000000-count-few";
"000 تريليون",
"100000000000000-count-many";
"000 تريليون",
"100000000000000-count-other";
"000 تريليون";

```

```

    }
  }
  "short";
  {
    "decimalFormat";
    {
      "1000-count-zero";
      "0 ألف",
      "1000-count-one";
      "0 ألف",
      "1000-count-two";
      "0 ألف",
      "1000-count-few";
      "0 آلاف",
      "1000-count-many";
      "0 ألف",
      "1000-count-other";
      "0 ألف",
      "10000-count-zero";
      "00 ألف",
      "10000-count-one";
      "00 ألف",
      "10000-count-two";
      "00 ألف",
      "10000-count-few";
      "00 ألف",
      "10000-count-many";
      "00 ألف",
      "10000-count-other";
      "00 ألف",
      "100000-count-zero";
      "000 ألف",
      "100000-count-one";
      "000 ألف",
      "100000-count-two";
      "000 ألف",
      "100000-count-few";
      "000 ألف",
      "100000-count-many";
      "000 ألف",
      "100000-count-other";
      "000 ألف",
      "1000000-count-zero";
      "0 مليون",
      "1000000-count-one";
      "0 مليون",
      "1000000-count-two";
      "0 مليون",
      "1000000-count-few";
      "0 مليون",
      "1000000-count-many";
      "0 مليون",
      "1000000-count-other";
      "0 مليون",
      "10000000-count-zero";
      "00 مليون",
      "10000000-count-one";
    }
  }
}

```



```
"00 مليو",
    "10000000-count-two";
"00 مليو",
    "10000000-count-few";
"00 مليو",
    "10000000-count-many";
"00 مليو",
    "10000000-count-other";
"00 مليو",
    "100000000-count-zero";
"000 مليو",
    "100000000-count-one";
"000 مليو",
    "100000000-count-two";
"000 مليو",
    "100000000-count-few";
"000 مليو",
    "100000000-count-many";
"000 مليو",
    "100000000-count-other";
"000 مليو",
    "1000000000-count-zero";
"0 مليا",
    "1000000000-count-one";
"0 مليا",
    "1000000000-count-two";
"0 مليا",
    "1000000000-count-few";
"0 مليا",
    "1000000000-count-many";
"0 مليا",
    "1000000000-count-other";
"0 مليا",
    "10000000000-count-zero";
"00 مليا",
    "10000000000-count-one";
"00 مليا",
    "10000000000-count-two";
"00 مليا",
    "10000000000-count-few";
"00 مليا",
    "10000000000-count-many";
"00 مليا",
    "10000000000-count-other";
"00 مليا",
    "100000000000-count-zero";
"000 مليا",
    "100000000000-count-one";
"000 مليا",
    "100000000000-count-two";
"000 مليا",
    "100000000000-count-few";
"000 مليا",
    "100000000000-count-many";
"000 مليا",
    "100000000000-count-other";
"000 مليا",
```

```

        "1000000000000-count-zero";
        "0 ترلیو",
        "1000000000000-count-one";
        "0 ترلیو",
        "1000000000000-count-two";
        "0 ترلیو",
        "1000000000000-count-few";
        "0 ترلیو",
        "1000000000000-count-many";
        "0 ترلیو",
        "1000000000000-count-other";
        "0 ترلیو",
        "1000000000000-count-zero";
        "00 ترلیو",
        "1000000000000-count-one";
        "00 ترلیو",
        "1000000000000-count-two";
        "00 ترلیو",
        "1000000000000-count-few";
        "00 ترلیو",
        "1000000000000-count-many";
        "00 ترلیو",
        "1000000000000-count-other";
        "00 ترلیو",
        "1000000000000-count-zero";
        "000 ترلیو",
        "1000000000000-count-one";
        "000 ترلیو",
        "1000000000000-count-two";
        "000 ترلیو",
        "1000000000000-count-few";
        "000 ترلیو",
        "1000000000000-count-many";
        "000 ترلیو",
        "1000000000000-count-other";
        "000 ترلیو";
    }
}
}
"decimalFormats-numberSystem-latn";
{
    "standard";
    "#,##0.###",
    "long";
    {
        "decimalFormat";
        {
            "1000-count-zero";
            "0 ألف",
            "1000-count-one";
            "0 ألف",
            "1000-count-two";
            "0 ألف",
            "1000-count-few";
            "0 آلاف",
            "1000-count-many";
            "0 ألف",

```

```
"1000-count-other";
"0 ألف",
"10000-count-zero";
"00 ألف",
"10000-count-one";
"00 ألف",
"10000-count-two";
"00 ألف",
"10000-count-few";
"00 ألف",
"10000-count-many";
"00 ألف",
"10000-count-other";
"00 ألف",
"100000-count-zero";
"000 ألف",
"100000-count-one";
"000 ألف",
"100000-count-two";
"000 ألف",
"100000-count-few";
"000 ألف",
"100000-count-many";
"000 ألف",
"100000-count-other";
"000 ألف",
"1000000-count-zero";
"0 مليون",
"1000000-count-one";
"0 مليون",
"1000000-count-two";
"0 مليون",
"1000000-count-few";
"0 ملايين",
"1000000-count-many";
"0 مليون",
"1000000-count-other";
"0 مليون",
"10000000-count-zero";
"00 مليون",
"10000000-count-one";
"00 مليون",
"10000000-count-two";
"00 مليون",
"10000000-count-few";
"00 ملايين",
"10000000-count-many";
"00 مليون",
"10000000-count-other";
"00 مليون",
"100000000-count-zero";
"000 مليون",
"100000000-count-one";
"000 مليون",
"100000000-count-two";
"000 مليون",
"100000000-count-few";
```

```
"000 مليون",
    "1000000000-count-many";
"000 مليون",
    "1000000000-count-other";
"000 مليون",
    "1000000000-count-zero";
"0 مليار",
    "1000000000-count-one";
"0 مليار",
    "1000000000-count-two";
"0 مليار",
    "1000000000-count-few";
"0 مليار",
    "1000000000-count-many";
"0 مليار",
    "1000000000-count-other";
"0 مليار",
    "1000000000-count-zero";
"00 مليار",
    "10000000000-count-one";
"00 مليار",
    "10000000000-count-two";
"00 مليار",
    "10000000000-count-few";
"00 مليار",
    "10000000000-count-many";
"00 مليار",
    "10000000000-count-other";
"00 مليار",
    "10000000000-count-zero";
"000 مليار",
    "100000000000-count-one";
"000 مليار",
    "100000000000-count-two";
"000 مليار",
    "100000000000-count-few";
"000 مليار",
    "100000000000-count-many";
"000 مليار",
    "100000000000-count-other";
"000 مليار",
    "100000000000-count-zero";
"0 تريليون",
    "1000000000000-count-one";
"0 تريليون",
    "1000000000000-count-two";
"0 تريليون",
    "1000000000000-count-few";
"0 تريليونات",
    "10000000000000-count-many";
"0 تريليون",
    "10000000000000-count-other";
"0 تريليون",
    "10000000000000-count-zero";
"00 تريليون",
    "100000000000000-count-one";
"00 تريليون",
```

```

        "1000000000000-count-two";
    "00 تريليون",
    "1000000000000-count-few";
    "00 تريليون",
    "1000000000000-count-many";
    "00 تريليون",
    "1000000000000-count-other";
    "00 تريليون",
    "1000000000000-count-zero";
    "000 تريليون",
    "1000000000000-count-one";
    "000 تريليون",
    "1000000000000-count-two";
    "000 تريليون",
    "1000000000000-count-few";
    "000 تريليون",
    "1000000000000-count-many";
    "000 تريليون",
    "1000000000000-count-other";
    "000 تريليون";
    }
}
"short";
{
    "decimalFormat";
    {
        "1000-count-zero";
        "0 ألف",
        "1000-count-one";
        "0 ألف",
        "1000-count-two";
        "0 ألف",
        "1000-count-few";
        "0 آلاف",
        "1000-count-many";
        "0 ألف",
        "1000-count-other";
        "0 ألف",
        "10000-count-zero";
        "00 ألف",
        "10000-count-one";
        "00 ألف",
        "10000-count-two";
        "00 ألف",
        "10000-count-few";
        "00 ألف",
        "10000-count-many";
        "00 ألف",
        "10000-count-other";
        "00 ألف",
        "100000-count-zero";
        "000 ألف",
        "100000-count-one";
        "000 ألف",
        "100000-count-two";
        "000 ألف",
        "100000-count-few";
    }
}

```

```
"000 ألف",
    "100000-count-many";
"000 ألف",
    "100000-count-other";
"000 ألف",
    "1000000-count-zero";
"0 مليو",
    "1000000-count-one";
"0 مليو",
    "1000000-count-two";
"0 مليو",
    "1000000-count-few";
"0 مليو",
    "1000000-count-many";
"0 مليو",
    "1000000-count-other";
"0 مليو",
    "10000000-count-zero";
"00 مليو",
    "10000000-count-one";
"00 مليو",
    "10000000-count-two";
"00 مليو",
    "10000000-count-few";
"00 مليو",
    "10000000-count-many";
"00 مليو",
    "10000000-count-other";
"00 مليو",
    "100000000-count-zero";
"000 مليو",
    "100000000-count-one";
"000 مليو",
    "100000000-count-two";
"000 مليو",
    "100000000-count-few";
"000 مليو",
    "100000000-count-many";
"000 مليو",
    "100000000-count-other";
"000 مليو",
    "1000000000-count-zero";
"0 مليا",
    "1000000000-count-one";
"0 مليا",
    "1000000000-count-two";
"0 مليا",
    "1000000000-count-few";
"0 مليا",
    "1000000000-count-many";
"0 مليا",
    "1000000000-count-other";
"0 مليا",
    "10000000000-count-zero";
"00 مليا",
    "10000000000-count-one";
"00 مليا",
```

```

        "10000000000-count-two";
    "00 مليا",
    "10000000000-count-few";
    "00 مليا",
    "10000000000-count-many";
    "00 مليا",
    "10000000000-count-other";
    "00 مليا",
    "10000000000-count-zero";
    "000 مليا",
    "10000000000-count-one";
    "000 مليا",
    "10000000000-count-two";
    "000 مليا",
    "10000000000-count-few";
    "000 مليا",
    "10000000000-count-many";
    "000 مليا",
    "10000000000-count-other";
    "000 مليا",
    "10000000000-count-zero";
    "0 ترليو",
    "10000000000-count-one";
    "0 ترليو",
    "10000000000-count-two";
    "0 ترليو",
    "10000000000-count-few";
    "0 ترليو",
    "10000000000-count-many";
    "0 ترليو",
    "10000000000-count-other";
    "0 ترليو",
    "10000000000-count-zero";
    "00 ترليو",
    "10000000000-count-one";
    "00 ترليو",
    "10000000000-count-two";
    "00 ترليو",
    "10000000000-count-few";
    "00 ترليو",
    "10000000000-count-many";
    "00 ترليو",
    "10000000000-count-other";
    "00 ترليو",
    "10000000000-count-zero";
    "000 ترليو",
    "10000000000-count-one";
    "000 ترليو",
    "10000000000-count-two";
    "000 ترليو",
    "10000000000-count-few";
    "000 ترليو",
    "10000000000-count-many";
    "000 ترليو",
    "10000000000-count-other";
    "000 ترليو";
    }

```

```

    }
  }
  "scientificFormats-numberSystem-arab";
  {
    "standard";
    "#E0";
  }
  "scientificFormats-numberSystem-latn";
  {
    "standard";
    "#E0";
  }
  "percentFormats-numberSystem-arab";
  {
    "standard";
    "#,##0 %";
  }
  "percentFormats-numberSystem-latn";
  {
    "standard";
    "#,##0%";
  }
  "currencyFormats-numberSystem-arab";
  {
    "currencySpacing";
    {
      "beforeCurrency";
      {
        "currencyMatch";
        "[:^S:]",
        "surroundingMatch";
        "[:digit:]",
        "insertBetween";
        " ";
      }
      "afterCurrency";
      {
        "currencyMatch";
        "[:^S:]",
        "surroundingMatch";
        "[:digit:]",
        "insertBetween";
        " ";
      }
    }
    "standard";
    "#,##0.00 ¤",
    "accounting";
    "#,##0.00 ¤",
    "unitPattern-count-zero";
    "{0} {1}",
    "unitPattern-count-one";
    "{0} {1}",
    "unitPattern-count-two";
    "{0} {1}",
    "unitPattern-count-few";
    "{0} {1}",

```



```

        "unitPattern-count-many";
        "{0} {1}",
        "unitPattern-count-other";
        "{0} {1}";
    }
    "currencyFormats-numberSystem-latn";
    {
        "currencySpacing";
        {
            "beforeCurrency";
            {
                "currencyMatch";
                "[:^S:]",
                "surroundingMatch";
                "[:digit:]",
                "insertBetween";
                " ";
            }
            "afterCurrency";
            {
                "currencyMatch";
                "[:^S:]",
                "surroundingMatch";
                "[:digit:]",
                "insertBetween";
                " ";
            }
        }
    }
    "standard";
    "¤ #,##0.00",
    "accounting";
    "¤#,##0.00; (¤#,##0.00) ",
    "short";
    {
        "standard";
        {
            "1000-count-zero";
            "¤ 0 ألف",
            "1000-count-one";
            "¤ 0 ألف",
            "1000-count-two";
            "¤ 0 ألف",
            "1000-count-few";
            "¤ 0 ألف",
            "1000-count-many";
            "¤ 0 ألف",
            "1000-count-other";
            "¤ 0 ألف",
            "10000-count-zero";
            "¤ 00 ألف",
            "10000-count-one";
            "¤ 00 ألف",
            "10000-count-two";
            "¤ 00 ألف",
            "10000-count-few";
            "¤ 00 ألف",
            "10000-count-many";
        }
    }

```

```
"x 00 ألف",
    "10000-count-other";
"x 00 ألف",
    "100000-count-zero";
"x 000 ألف",
    "100000-count-one";
"x 000 ألف",
    "100000-count-two";
"x 000 ألف",
    "100000-count-few";
"x 000 ألف",
    "100000-count-many";
"x 000 ألف",
    "100000-count-other";
"x 000 ألف",
    "1000000-count-zero";
"x 0 مليون",
    "1000000-count-one";
"x 0 مليون",
    "1000000-count-two";
"x 0 مليون",
    "1000000-count-few";
"x 0 مليون",
    "1000000-count-many";
"x 0 مليون",
    "1000000-count-other";
"x 0 مليون",
    "10000000-count-zero";
"x 00 مليون",
    "10000000-count-one";
"x 00 مليون",
    "10000000-count-two";
"x 00 مليون",
    "10000000-count-few";
"x 00 مليون",
    "10000000-count-many";
"x 00 مليون",
    "10000000-count-other";
"x 00 مليون",
    "100000000-count-zero";
"x 000 مليون",
    "100000000-count-one";
"x 000 مليون",
    "100000000-count-two";
"x 000 مليون",
    "100000000-count-few";
"x 000 مليون",
    "100000000-count-many";
"x 000 مليون",
    "100000000-count-other";
"x 000 مليون",
    "1000000000-count-zero";
"x 0 مليا",
    "1000000000-count-one";
"x 0 مليا",
    "1000000000-count-two";
"x 0 مليا",
```

```
"1000000000-count-few";
"۰ ۰ مليا",
"1000000000-count-many";
"۰ ۰ مليا",
"1000000000-count-other";
"۰ ۰ مليا",
"1000000000-count-zero";
"۰ ۰۰ مليا",
"1000000000-count-one";
"۰ ۰۰ مليا",
"1000000000-count-two";
"۰ ۰۰ مليا",
"1000000000-count-few";
"۰ ۰۰ مليا",
"1000000000-count-many";
"۰ ۰۰ مليا",
"1000000000-count-other";
"۰ ۰۰ مليا",
"1000000000-count-zero";
"۰ ۰۰۰ مليا",
"1000000000-count-one";
"۰ ۰۰۰ مليا",
"1000000000-count-two";
"۰ ۰۰۰ مليا",
"1000000000-count-few";
"۰ ۰۰۰ مليا",
"1000000000-count-many";
"۰ ۰۰۰ مليا",
"1000000000-count-other";
"۰ ۰۰۰ مليا",
"1000000000-count-zero";
"۰ ۰ ترليو",
"1000000000-count-one";
"۰ ۰ ترليو",
"1000000000-count-two";
"۰ ۰ ترليو",
"1000000000-count-few";
"۰ ۰ ترليو",
"1000000000-count-many";
"۰ ۰ ترليو",
"1000000000-count-other";
"۰ ۰ ترليو",
"1000000000-count-zero";
"۰ ۰۰ ترليو",
"1000000000-count-one";
"۰ ۰۰ ترليو",
"1000000000-count-two";
"۰ ۰۰ ترليو",
"1000000000-count-few";
"۰ ۰۰ ترليو",
"1000000000-count-many";
"۰ ۰۰ ترليو",
"1000000000-count-other";
"۰ ۰۰ ترليو",
"1000000000-count-zero";
"۰ ۰۰۰ ترليو",
"1000000000-count-one";
```

```

        "x 000 ترلیو",
        "100000000000000-count-two";
    "x 000 ترلیو",
    "100000000000000-count-few";
    "x 000 ترلیو",
    "100000000000000-count-many";
    "x 000 ترلیو",
    "100000000000000-count-other";
    "x 000 ترلیو";
}

}

unitPattern-count-zero";
"{0} {1}",
    unitPattern-count-one";
"{0} {1}",
    unitPattern-count-two";
"{0} {1}",
    unitPattern-count-few";
"{0} {1}",
    unitPattern-count-many";
"{0} {1}",
    unitPattern-count-other";
"{0} {1}";
}

miscPatterns-numberSystem-arab";
{
    atLeast";
    "+{0}",
    range";
    "{0}-{1}";
}

miscPatterns-numberSystem-latn";
{
    atLeast";
    "+{0}",
    range";
    "{0}-{1}";
}

}

}

```

## TIMEZONENAMES.JSON

```
{
  "main": {
    "ar": {
      "identity": {
        "version": {
          "_number": "$Revision: 12879 $",
          "_cldrVersion": "30.0.3"
        },
        "language": "ar"
      },
      "dates": {
```

```

"timeZoneNames": {
  "hourFormat": "+HH:mm;-HH:mm",
  "gmtFormat": "0{جرينتش",
  "gmtZeroFormat": "جرينتش",
  "regionFormat": "0{توقيت",
  "regionFormat-type-daylight": "الصيفي } 0{توقيت",
  "regionFormat-type-standard": "الرسمي } 0{توقيت",
  "fallbackFormat": "{1} ({0})",
  "zone": {
    "America": {
      "Adak": {
        "exemplarCity": "أداك"
      },
      "Anchorage": {
        "exemplarCity": "أنشوراج"
      },
      "Anguilla": {
        "exemplarCity": "أنغيلا"
      },
      "Antigua": {
        "exemplarCity": "أنتيغوا"
      },
      "Araguaina": {
        "exemplarCity": "أروجوانيا"
      },
      "Argentina": {
        "Rio_Gallegos": {
          "exemplarCity": "ريو جالييوس"
        },
        "San_Juan": {
          "exemplarCity": "سان خوان"
        },
        "Ushuaia": {
          "exemplarCity": "أشوا"
        },
        "La_Rioja": {
          "exemplarCity": "لا ريوجا"
        },
        "San_Luis": {
          "exemplarCity": "سان لويس"
        },
        "Salta": {
          "exemplarCity": "سالطا"
        },
        "Tucuman": {
          "exemplarCity": "تاكمان"
        }
      },
      "Aruba": {
        "exemplarCity": "أروبا"
      },
      "Asuncion": {
        "exemplarCity": "أسونسيون"
      },
      "Bahia": {
        "exemplarCity": "باهيا"
      }
    }
  }
}

```

```
"Bahia_Banderas": {
  "exemplarCity": "باهيا بانديراس"
},
"Barbados": {
  "exemplarCity": "بربادوس"
},
"Belem": {
  "exemplarCity": "بلم"
},
"Belize": {
  "exemplarCity": "بليز"
},
"Blanc-Sablon": {
  "exemplarCity": "بلانك-سابلون"
},
"Boa_Vista": {
  "exemplarCity": "باو فيستا"
},
"Bogota": {
  "exemplarCity": "بوغوتا"
},
"Boise": {
  "exemplarCity": "بويس"
},
"Buenos_Aires": {
  "exemplarCity": "بوينوس آيرس"
},
"Cambridge_Bay": {
  "exemplarCity": "كامبرديج باي"
},
"Campo_Grande": {
  "exemplarCity": "كومبو جراند"
},
"Cancun": {
  "exemplarCity": "كانكون"
},
"Caracas": {
  "exemplarCity": "كاراكاس"
},
"Catamarca": {
  "exemplarCity": "كاتاماركا"
},
"Cayenne": {
  "exemplarCity": "كايين"
},
"Cayman": {
  "exemplarCity": "كيمان"
},
"Chicago": {
  "exemplarCity": "شيكاغو"
},
"Chihuahua": {
  "exemplarCity": "تشيووا"
},
"Coral_Harbour": {
  "exemplarCity": "كورال هاربر"
},
}
```

```
"Cordoba": {
  "exemplarCity": "كوردوبا"
},
"Costa_Rica": {
  "exemplarCity": "كوستاريكا"
},
"Creston": {
  "exemplarCity": "كريستون"
},
"Cuiaba": {
  "exemplarCity": "كيابا"
},
"Curacao": {
  "exemplarCity": "كوراكاو"
},
"Danmarkshavn": {
  "exemplarCity": "دانمرك شافن"
},
"Dawson": {
  "exemplarCity": "داوسان"
},
"Dawson_Creek": {
  "exemplarCity": "داوسن كريك"
},
"Denver": {
  "exemplarCity": "دنفر"
},
"Detroit": {
  "exemplarCity": "ديترويت"
},
"Dominica": {
  "exemplarCity": "دومينيكا"
},
"Edmonton": {
  "exemplarCity": "ايدمونتون"
},
"Eirunepe": {
  "exemplarCity": "ايرونبي"
},
"El_Salvador": {
  "exemplarCity": "السلفادور"
},
"Fort_Nelson": {
  "exemplarCity": "فورت نيلسون"
},
"Fortaleza": {
  "exemplarCity": "فورتاليزا"
},
"Glace_Bay": {
  "exemplarCity": "جلاس باي"
},
"Godthab": {
  "exemplarCity": "غودثاب"
},
"Goose_Bay": {
  "exemplarCity": "جوس باي"
},
}
```

```

"Grand_Turk": {
  "exemplarCity": "غراند ترك"
},
"Grenada": {
  "exemplarCity": "غرينادا"
},
"Guadeloupe": {
  "exemplarCity": "غوادلوب"
},
"Guatemala": {
  "exemplarCity": "غواتيمالا"
},
"Guayaquil": {
  "exemplarCity": "غواياكويل"
},
"Guyana": {
  "exemplarCity": "غيانا"
},
"Halifax": {
  "exemplarCity": "هاليفاكس"
},
"Havana": {
  "exemplarCity": "هافانا"
},
"Hermosillo": {
  "exemplarCity": "هيرموسيلو"
},
"Indiana": {
  "Vincennes": {
    "exemplarCity": "فينسينس"
  },
  "Petersburg": {
    "exemplarCity": "بيتربيرغ"
  },
  "Tell_City": {
    "exemplarCity": "مدينة تل، إنديانا"
  },
  "Knox": {
    "exemplarCity": "كونكس"
  },
  "Winamac": {
    "exemplarCity": "ويناماك"
  },
  "Marengo": {
    "exemplarCity": "مارنجو"
  },
  "Vevay": {
    "exemplarCity": "فيفاي"
  }
},
"Indianapolis": {
  "exemplarCity": "إنديانا بوليس"
},
"Inuvik": {
  "exemplarCity": "اينوفيك"
},
"Iqaluit": {

```



```
    "exemplarCity": "اكويلا"  
  },  
  "Jamaica": {  
    "exemplarCity": "جامايكا"  
  },  
  "Jujuy": {  
    "exemplarCity": "جوجو"  
  },  
  "Juneau": {  
    "exemplarCity": "جونى"  
  },  
  "Kentucky": {  
    "Monticello": {  
      "exemplarCity": "مونتيسيلو"  
    }  
  },  
  "Kralendijk": {  
    "exemplarCity": "كرالنديك"  
  },  
  "La_Paz": {  
    "exemplarCity": "لا باز"  
  },  
  "Lima": {  
    "exemplarCity": "ليما"  
  },  
  "Los_Angeles": {  
    "exemplarCity": "لوس انجلوس"  
  },  
  "Louisville": {  
    "exemplarCity": "لويس فيل"  
  },  
  "Lower_Princes": {  
    "exemplarCity": "حي الأمير السفلي"  
  },  
  "Maceio": {  
    "exemplarCity": "ماشيو"  
  },  
  "Managua": {  
    "exemplarCity": "ماناغوا"  
  },  
  "Manaus": {  
    "exemplarCity": "ماناوس"  
  },  
  "Marigot": {  
    "exemplarCity": "ماريغوت"  
  },  
  "Martinique": {  
    "exemplarCity": "المارتينيك"  
  },  
  "Matamoros": {  
    "exemplarCity": "ماتاموروس"  
  },  
  "Mazatlan": {  
    "exemplarCity": "مازاتلان"  
  },  
  "Mendoza": {  
    "exemplarCity": "ميندوزا"
```

```
},
"Menominee": {
  "exemplarCity": "مينوميني"
},
"Merida": {
  "exemplarCity": "ميريدا"
},
"Metlakatla": {
  "exemplarCity": "ميتلاكاتلا"
},
"Mexico_City": {
  "exemplarCity": "مدينة المكسيك"
},
"Miquelon": {
  "exemplarCity": "ميكلون"
},
"Moncton": {
  "exemplarCity": "وينكتون"
},
"Monterrey": {
  "exemplarCity": "مونتيري"
},
"Montevideo": {
  "exemplarCity": "مونتيفيديو"
},
"Montserrat": {
  "exemplarCity": "مونتسيرات"
},
"Nassau": {
  "exemplarCity": "ناسو"
},
"New_York": {
  "exemplarCity": "نيويورك"
},
"Nipigon": {
  "exemplarCity": "نيبيجون"
},
"Nome": {
  "exemplarCity": "نوم"
},
"Noronha": {
  "exemplarCity": "نوروناه"
},
"North_Dakota": {
  "Beulah": {
    "exemplarCity": "بيولا، داكوتا الشمالية"
  },
  "New_Salem": {
    "exemplarCity": "نيو ساليم"
  },
  "Center": {
    "exemplarCity": "سنتر"
  }
},
"Ojinaga": {
  "exemplarCity": "أوجيناغا"
},
}
```

```
"Panama": {
  "exemplarCity": "بنما"
},
"Pangnirtung": {
  "exemplarCity": "بانجینتینگ"
},
"Paramaribo": {
  "exemplarCity": "باراماریبو"
},
"Phoenix": {
  "exemplarCity": "فینکس"
},
"Port-au-Prince": {
  "exemplarCity": "پورت او برنس"
},
"Port_of_Spain": {
  "exemplarCity": "پورت اوف سبین"
},
"Porto_Velho": {
  "exemplarCity": "پورتو فیلو"
},
"Puerto_Rico": {
  "exemplarCity": "پورتوریکو"
},
"Rainy_River": {
  "exemplarCity": "رانی ریفر"
},
"Rankin_Inlet": {
  "exemplarCity": "رانکن انلت"
},
"Recife": {
  "exemplarCity": "ریسیف"
},
"Regina": {
  "exemplarCity": "ریجینا"
},
"Resolute": {
  "exemplarCity": "ریزولوت"
},
"Rio_Branco": {
  "exemplarCity": "ریوبرانکو"
},
"Santa_Isabel": {
  "exemplarCity": "سانتا ایزابیل"
},
"Santarem": {
  "exemplarCity": "سانتاریم"
},
"Santiago": {
  "exemplarCity": "سانتیآگو"
},
"Santo_Domingo": {
  "exemplarCity": "سانتو دومینگو"
},
"Sao_Paulo": {
  "exemplarCity": "ساو پاولو"
},
}
```

```
"Scoresbysund": {
  "exemplarCity": "سکورسبیسند"
},
"Sitka": {
  "exemplarCity": "سیتکا"
},
"St_Barthelemy": {
  "exemplarCity": "سانت بارتیلیمی"
},
"St_Johns": {
  "exemplarCity": "سانت جونز"
},
"St_Kitts": {
  "exemplarCity": "سانت کیتس"
},
"St_Lucia": {
  "exemplarCity": "سانت لوشیا"
},
"St_Thomas": {
  "exemplarCity": "سانت توماس"
},
"St_Vincent": {
  "exemplarCity": "سانت فنسنت"
},
"Swift_Current": {
  "exemplarCity": "سوفت کارنت"
},
"Tegucigalpa": {
  "exemplarCity": "تیغوسیغالبا"
},
"Thule": {
  "exemplarCity": "ٹیل"
},
"Thunder_Bay": {
  "exemplarCity": "ٹندر بای"
},
"Tijuana": {
  "exemplarCity": "تیخوانا"
},
"Toronto": {
  "exemplarCity": "تورونتو"
},
"Tortola": {
  "exemplarCity": "تورتولا"
},
"Vancouver": {
  "exemplarCity": "فانکوفر"
},
"Whitehorse": {
  "exemplarCity": "وایت هورس"
},
"Winnipeg": {
  "exemplarCity": "وینیپج"
},
"Yakutat": {
  "exemplarCity": "یاکوتات"
},
}
```

```
"Yellowknife": {
  "exemplarCity": "يلونيف"
},
"Atlantic": {
  "Azores": {
    "exemplarCity": "أزورس"
  },
  "Bermuda": {
    "exemplarCity": "برمودا"
  },
  "Canary": {
    "exemplarCity": "كناري"
  },
  "Cape_Verde": {
    "exemplarCity": "الرأس الأخضر"
  },
  "Faeroe": {
    "exemplarCity": "فارو"
  },
  "Madeira": {
    "exemplarCity": "ماديرا"
  },
  "Reykjavik": {
    "exemplarCity": "ريكيافيك"
  },
  "South_Georgia": {
    "exemplarCity": "جورجيا الجنوبية"
  },
  "St_Helena": {
    "exemplarCity": "سانت هيلينا"
  },
  "Stanley": {
    "exemplarCity": "استانلي"
  }
},
"Europe": {
  "Amsterdam": {
    "exemplarCity": "أمستردام"
  },
  "Andorra": {
    "exemplarCity": "أندورا"
  },
  "Astrakhan": {
    "exemplarCity": "أستراخان"
  },
  "Athens": {
    "exemplarCity": "أثينا"
  },
  "Belgrade": {
    "exemplarCity": "بلغراد"
  },
  "Berlin": {
    "exemplarCity": "برلين"
  },
  "Bratislava": {
    "exemplarCity": "براتيسلافا"
  }
```

```
},
"Brussels": {
  "exemplarCity": "بروكسل"
},
"Bucharest": {
  "exemplarCity": "بوخارست"
},
"Budapest": {
  "exemplarCity": "بودابست"
},
"Busingen": {
  "exemplarCity": "بوسنغن"
},
"Chisinau": {
  "exemplarCity": "تشيسيناو"
},
"Copenhagen": {
  "exemplarCity": "كوبنهاغن"
},
"Dublin": {
  "long": {
    "daylight": "توقيت أيرلندا الرسمي"
  },
  "exemplarCity": "دبلن"
},
"Gibraltar": {
  "exemplarCity": "جبل طارق"
},
"Guernsey": {
  "exemplarCity": "غيرنسي"
},
"Helsinki": {
  "exemplarCity": "هلسنكي"
},
"Isle_of_Man": {
  "exemplarCity": "جزيرة مان"
},
"Istanbul": {
  "exemplarCity": "إسطنبول"
},
"Jersey": {
  "exemplarCity": "جيرسي"
},
"Kaliningrad": {
  "exemplarCity": "كالينجراد"
},
"Kiev": {
  "exemplarCity": "كييف"
},
"Kirov": {
  "exemplarCity": "كيروف"
},
"Lisbon": {
  "exemplarCity": "لشبونة"
},
"Ljubljana": {
  "exemplarCity": "ليوبليانا"
```

```
    },  
    "London": {  
      "long": {  
        "daylight": "توقيت بريطانيا الصيفي"  
      },  
      "exemplarCity": "لندن"  
    },  
    "Luxembourg": {  
      "exemplarCity": "لوكسمبورغ"  
    },  
    "Madrid": {  
      "exemplarCity": "مدريد"  
    },  
    "Malta": {  
      "exemplarCity": "مالطة"  
    },  
    "Mariehamn": {  
      "exemplarCity": "ماريهامن"  
    },  
    "Minsk": {  
      "exemplarCity": "مينسك"  
    },  
    "Monaco": {  
      "exemplarCity": "موناكو"  
    },  
    "Moscow": {  
      "exemplarCity": "موسكو"  
    },  
    "Oslo": {  
      "exemplarCity": "أوسلو"  
    },  
    "Paris": {  
      "exemplarCity": "باريس"  
    },  
    "Podgorica": {  
      "exemplarCity": "بودغوريكا"  
    },  
    "Prague": {  
      "exemplarCity": "براغ"  
    },  
    "Riga": {  
      "exemplarCity": "ريغا"  
    },  
    "Rome": {  
      "exemplarCity": "روما"  
    },  
    "Samara": {  
      "exemplarCity": "سمراء"  
    },  
    "San_Marino": {  
      "exemplarCity": "سان مارينو"  
    },  
    "Sarajevo": {  
      "exemplarCity": "سراييفو"  
    },  
    "Simferopol": {  
      "exemplarCity": "سيمفروبول"
```

```
    },
    "Skopje": {
      "exemplarCity": "سكوبي"
    },
    "Sofia": {
      "exemplarCity": "صوفيا"
    },
    "Stockholm": {
      "exemplarCity": "ستوكهولم"
    },
    "Tallinn": {
      "exemplarCity": "تالين"
    },
    "Tirane": {
      "exemplarCity": "تيرانا"
    },
    "Ulyanovsk": {
      "exemplarCity": "أوليانوفسك"
    },
    "Uzhgorod": {
      "exemplarCity": "أوزجروود"
    },
    "Vaduz": {
      "exemplarCity": "فادوز"
    },
    "Vatican": {
      "exemplarCity": "الفاتيكان"
    },
    "Vienna": {
      "exemplarCity": "فيينا"
    },
    "Vilnius": {
      "exemplarCity": "فيلنيوس"
    },
    "Volgograd": {
      "exemplarCity": "فولجوراد"
    },
    "Warsaw": {
      "exemplarCity": "وارسو"
    },
    "Zagreb": {
      "exemplarCity": "زغرب"
    },
    "Zaporozhye": {
      "exemplarCity": "زابوروزي"
    },
    "Zurich": {
      "exemplarCity": "زيورخ"
    }
  },
  "Africa": {
    "Abidjan": {
      "exemplarCity": "أبيدجان"
    },
    "Accra": {
      "exemplarCity": "أكرا"
    }
  },
}
```



```
"Addis_Ababa": {
  "exemplarCity": "أديس أبابا"
},
"Algiers": {
  "exemplarCity": "الجزائر"
},
"Asmera": {
  "exemplarCity": "أسمرّة"
},
"Bamako": {
  "exemplarCity": "باماكو"
},
"Bangui": {
  "exemplarCity": "بانغوي"
},
"Banjul": {
  "exemplarCity": "بانجول"
},
"Bissau": {
  "exemplarCity": "بيساو"
},
"Blantyre": {
  "exemplarCity": "بلانتيير"
},
"Brazzaville": {
  "exemplarCity": "برازافيل"
},
"Bujumbura": {
  "exemplarCity": "بوجومبورا"
},
"Cairo": {
  "exemplarCity": "القاهرة"
},
"Casablanca": {
  "exemplarCity": "الدار البيضاء"
},
"Ceuta": {
  "exemplarCity": "سيتا"
},
"Conakry": {
  "exemplarCity": "كوناكري"
},
"Dakar": {
  "exemplarCity": "داكار"
},
"Dar_es_Salaam": {
  "exemplarCity": "دار السلام"
},
"Djibouti": {
  "exemplarCity": "جيبوتي"
},
"Douala": {
  "exemplarCity": "دوالا"
},
"El_Aaiun": {
  "exemplarCity": "العيون"
},
}
```

```
"Freetown": {
  "exemplarCity": "فري تاون"
},
"Gaborone": {
  "exemplarCity": "غابورون"
},
"Harare": {
  "exemplarCity": "هراري"
},
"Johannesburg": {
  "exemplarCity": "جوهانسبرغ"
},
"Juba": {
  "exemplarCity": "جوبا"
},
"Kampala": {
  "exemplarCity": "كامبالا"
},
"Khartoum": {
  "exemplarCity": "الخرطوم"
},
"Kigali": {
  "exemplarCity": "كيغالي"
},
"Kinshasa": {
  "exemplarCity": "كينشاسا"
},
"Lagos": {
  "exemplarCity": "لاغوس"
},
"Libreville": {
  "exemplarCity": "ليبرفيل"
},
"Lome": {
  "exemplarCity": "لومي"
},
"Luanda": {
  "exemplarCity": "لواندا"
},
"Lubumbashi": {
  "exemplarCity": "لومبباشا"
},
"Lusaka": {
  "exemplarCity": "لوساكا"
},
"Malabo": {
  "exemplarCity": "مالابو"
},
"Maputo": {
  "exemplarCity": "مابوتو"
},
"Maseru": {
  "exemplarCity": "ماسيرو"
},
"Mbabane": {
  "exemplarCity": "مباباني"
},
}
```

```

"Mogadishu": {
  "exemplarCity": "مقديشو"
},
"Monrovia": {
  "exemplarCity": "مونروفيا"
},
"Nairobi": {
  "exemplarCity": "نيروبي"
},
"Ndjamena": {
  "exemplarCity": "نجامينا"
},
"Niamey": {
  "exemplarCity": "نيامي"
},
"Nouakchott": {
  "exemplarCity": "نواكشوط"
},
"Ouagadougou": {
  "exemplarCity": "واغادوغو"
},
"Porto-Novo": {
  "exemplarCity": "بورتو نوفو"
},
"Sao_Tome": {
  "exemplarCity": "ساو تومي"
},
"Tripoli": {
  "exemplarCity": "طرابلس"
},
"Tunis": {
  "exemplarCity": "تونس"
},
"Windhoek": {
  "exemplarCity": "ويندهوك"
}
},
"Asia": {
  "Aden": {
    "exemplarCity": "عدن"
  },
  "Almaty": {
    "exemplarCity": "ألماتي"
  },
  "Amman": {
    "exemplarCity": "عمان"
  },
  "Anadyr": {
    "exemplarCity": "أندير"
  },
  "Aqtan": {
    "exemplarCity": "أكتان"
  },
  "Aqtobe": {
    "exemplarCity": "أكتوب"
  },
  "Ashgabat": {

```

```
    "exemplarCity": "عشق آباد"
  },
  "Baghdad": {
    "exemplarCity": "بغداد"
  },
  "Bahrain": {
    "exemplarCity": "البحرين"
  },
  "Baku": {
    "exemplarCity": "باكو"
  },
  "Bangkok": {
    "exemplarCity": "بانكوك"
  },
  "Barnaul": {
    "exemplarCity": "بارناول"
  },
  "Beirut": {
    "exemplarCity": "بيروت"
  },
  "Bishkek": {
    "exemplarCity": "بشكيك"
  },
  "Brunei": {
    "exemplarCity": "بروناي"
  },
  "Calcutta": {
    "exemplarCity": "كالكتا"
  },
  "Chita": {
    "exemplarCity": "تشيتا"
  },
  "Choibalsan": {
    "exemplarCity": "تشوبالسان"
  },
  "Colombo": {
    "exemplarCity": "كولومبو"
  },
  "Damascus": {
    "exemplarCity": "دمشق"
  },
  "Dhaka": {
    "exemplarCity": "دكا"
  },
  "Dili": {
    "exemplarCity": "ديلي"
  },
  "Dubai": {
    "exemplarCity": "دبي"
  },
  "Dushanbe": {
    "exemplarCity": "دوشانبي"
  },
  "Gaza": {
    "exemplarCity": "غزة"
  },
  "Hebron": {
```

```
    "exemplarCity": "(هيبرون) مدينة الخليل")
  },
  "Hong_Kong": {
    "exemplarCity": "هونغ كونغ"
  },
  "Hovd": {
    "exemplarCity": "هوفد"
  },
  "Irkutsk": {
    "exemplarCity": "ايركيتسك"
  },
  "Jakarta": {
    "exemplarCity": "جاكرتا"
  },
  "Jayapura": {
    "exemplarCity": "جايا بورا"
  },
  "Jerusalem": {
    "exemplarCity": "القدس"
  },
  "Kabul": {
    "exemplarCity": "كابول"
  },
  "Kamchatka": {
    "exemplarCity": "كامتشاتكا"
  },
  "Karachi": {
    "exemplarCity": "كراتشي"
  },
  "Katmandu": {
    "exemplarCity": "كاتماندو"
  },
  "Khandyga": {
    "exemplarCity": "خانديجا"
  },
  "Krasnoyarsk": {
    "exemplarCity": "كراسنويارسك"
  },
  "Kuala_Lumpur": {
    "exemplarCity": "كوالا لامبور"
  },
  "Kuching": {
    "exemplarCity": "كيشينج"
  },
  "Kuwait": {
    "exemplarCity": "الكويت"
  },
  "Macau": {
    "exemplarCity": "ماكاو"
  },
  "Magadan": {
    "exemplarCity": "مجادن"
  },
  "Makassar": {
    "exemplarCity": "ماكسار"
  },
  "Manila": {
```

```
    "exemplarCity": "مانیلا"
  },
  "Muscat": {
    "exemplarCity": "مسقط"
  },
  "Nicosia": {
    "exemplarCity": "نيقوسيا"
  },
  "Novokuznetsk": {
    "exemplarCity": "نوفوكوزنتسك"
  },
  "Novosibirsk": {
    "exemplarCity": "نوفوسبيرسك"
  },
  "Omsk": {
    "exemplarCity": "أومسك"
  },
  "Oral": {
    "exemplarCity": "أورال"
  },
  "Phnom_Penh": {
    "exemplarCity": "بنوم بنه"
  },
  "Pontianak": {
    "exemplarCity": "بونتيانك"
  },
  "Pyongyang": {
    "exemplarCity": "بيونغ يانغ"
  },
  "Qatar": {
    "exemplarCity": "قطر"
  },
  "Qyzylorda": {
    "exemplarCity": "كيزيلوردا"
  },
  "Rangoon": {
    "exemplarCity": "رانغون"
  },
  "Riyadh": {
    "exemplarCity": "الرياض"
  },
  "Saigon": {
    "exemplarCity": "مدينة هو تشي منه"
  },
  "Sakhalin": {
    "exemplarCity": "سكالين"
  },
  "Samarkand": {
    "exemplarCity": "سمرقند"
  },
  "Seoul": {
    "exemplarCity": "سول"
  },
  "Shanghai": {
    "exemplarCity": "شنغهاي"
  },
  "Singapore": {
```

```
    "exemplarCity": "سنغافورة"  
  },  
  "Srednekolymsk": {  
    "exemplarCity": "سريدنكوليمسك"  
  },  
  "Taipei": {  
    "exemplarCity": "تايبيه"  
  },  
  "Tashkent": {  
    "exemplarCity": "تشقند"  
  },  
  "Tbilisi": {  
    "exemplarCity": "تبليسي"  
  },  
  "Tehran": {  
    "exemplarCity": "تهران"  
  },  
  "Thimphu": {  
    "exemplarCity": "تيمفو"  
  },  
  "Tokyo": {  
    "exemplarCity": "طوكيو"  
  },  
  "Tomsk": {  
    "exemplarCity": "تومسك"  
  },  
  "Ulaanbaatar": {  
    "exemplarCity": "آلانباتار"  
  },  
  "Urumqi": {  
    "exemplarCity": "أرومكي"  
  },  
  "Ust-Nera": {  
    "exemplarCity": "أوست نيرا"  
  },  
  "Vientiane": {  
    "exemplarCity": "فيانتيان"  
  },  
  "Vladivostok": {  
    "exemplarCity": "فلاديفوستك"  
  },  
  "Yakutsk": {  
    "exemplarCity": "ياكتسك"  
  },  
  "Yekaterinburg": {  
    "exemplarCity": "يكاترينبيرج"  
  },  
  "Yerevan": {  
    "exemplarCity": "يريفان"  
  },  
  },  
  "Indian": {  
    "Antananarivo": {  
      "exemplarCity": "أنتاناناريفو"  
    },  
    "Chagos": {  
      "exemplarCity": "تشاغوس"  
    }  
  }  
}
```

```

    },
    "Christmas": {
      "exemplarCity": "كريسماس"
    },
    "Cocos": {
      "exemplarCity": "كوكوس"
    },
    "Comoro": {
      "exemplarCity": "جزر القمر"
    },
    "Kerguelen": {
      "exemplarCity": "كيرغويلين"
    },
    "Mahe": {
      "exemplarCity": "ماهي"
    },
    "Maldives": {
      "exemplarCity": "المالديف"
    },
    "Mauritius": {
      "exemplarCity": "موريشيوس"
    },
    "Mayotte": {
      "exemplarCity": "مايوت"
    },
    "Reunion": {
      "exemplarCity": "ريونيون"
    }
  },
  "Australia": {
    "Adelaide": {
      "exemplarCity": "أديليد"
    },
    "Brisbane": {
      "exemplarCity": "برسبان"
    },
    "Broken_Hill": {
      "exemplarCity": "بروكن هيل"
    },
    "Currie": {
      "exemplarCity": "كوري"
    },
    "Darwin": {
      "exemplarCity": "دارون"
    },
    "Eucla": {
      "exemplarCity": "أو كلا"
    },
    "Hobart": {
      "exemplarCity": "هوبارت"
    },
    "Lindeman": {
      "exemplarCity": "ليندمان"
    },
    "Lord_Howe": {
      "exemplarCity": "لورد هاو"
    }
  },

```



```
"Melbourne": {
  "exemplarCity": "ميلبورن"
},
"Perth": {
  "exemplarCity": "برثا"
},
"Sydney": {
  "exemplarCity": "سيدني"
}
},
"Pacific": {
  "Apia": {
    "exemplarCity": "أبيا"
  },
  "Auckland": {
    "exemplarCity": "أوكلاند"
  },
  "Bougainville": {
    "exemplarCity": "بوغانفيل"
  },
  "Chatham": {
    "exemplarCity": "تشاثام"
  },
  "Easter": {
    "exemplarCity": "استر"
  },
  "Efate": {
    "exemplarCity": "إيفات"
  },
  "Enderbury": {
    "exemplarCity": "اندربيرج"
  },
  "Fakaofu": {
    "exemplarCity": "فاكاوفو"
  },
  "Fiji": {
    "exemplarCity": "فيجي"
  },
  "Funafuti": {
    "exemplarCity": "فونافوتي"
  },
  "Galapagos": {
    "exemplarCity": "جلاپاجوس"
  },
  "Gambier": {
    "exemplarCity": "جامبير"
  },
  "Guadalcanal": {
    "exemplarCity": "غواد الكانال"
  },
  "Guam": {
    "exemplarCity": "غوام"
  },
  "Honolulu": {
    "exemplarCity": "هونولولو"
  },
  "Johnston": {
```

```

    "exemplarCity": "جونستون"
  },
  "Kiritimati": {
    "exemplarCity": "كيريتي ماتي"
  },
  "Kosrae": {
    "exemplarCity": "كوسرا"
  },
  "Kwajalein": {
    "exemplarCity": "كواجالين"
  },
  "Majuro": {
    "exemplarCity": "ماجورو"
  },
  "Marquesas": {
    "exemplarCity": "ماركيساس"
  },
  "Midway": {
    "exemplarCity": "ميدواي"
  },
  "Nauru": {
    "exemplarCity": "ناورو"
  },
  "Niue": {
    "exemplarCity": "نيوي"
  },
  "Norfolk": {
    "exemplarCity": "نورفولك"
  },
  "Noumea": {
    "exemplarCity": "نوميا"
  },
  "Pago_Pago": {
    "exemplarCity": "باغو باغو"
  },
  "Palau": {
    "exemplarCity": "بالاو"
  },
  "Pitcairn": {
    "exemplarCity": "بيتكيرن"
  },
  "Ponape": {
    "exemplarCity": "باناب"
  },
  "Port_Moresby": {
    "exemplarCity": "بور مورسبي"
  },
  "Rarotonga": {
    "exemplarCity": "راروتونغا"
  },
  "Saipan": {
    "exemplarCity": "سايبان"
  },
  "Tahiti": {
    "exemplarCity": "تاهيتي"
  },
  "Tarawa": {

```

```
    "exemplarCity": "تاراوا"  
  },  
  "Tongatapu": {  
    "exemplarCity": "تونغاتابو"  
  },  
  "Truk": {  
    "exemplarCity": "ترك"  
  },  
  "Wake": {  
    "exemplarCity": "واك"  
  },  
  "Wallis": {  
    "exemplarCity": "واليس"  
  },  
},  
"Arctic": {  
  "Longyearbyen": {  
    "exemplarCity": "لونجيربين"  
  },  
},  
"Antarctica": {  
  "Casey": {  
    "exemplarCity": "كاساي"  
  },  
  "Davis": {  
    "exemplarCity": "دافيز"  
  },  
  "DumontDUrville": {  
    "exemplarCity": "دي مونت دو روفيل"  
  },  
  "Macquarie": {  
    "exemplarCity": "ماكوارى"  
  },  
  "Mawson": {  
    "exemplarCity": "ماوسون"  
  },  
  "McMurdo": {  
    "exemplarCity": "ماك موردو"  
  },  
  "Palmer": {  
    "exemplarCity": "بالمير"  
  },  
  "Rothera": {  
    "exemplarCity": "روثيرا"  
  },  
  "Syowa": {  
    "exemplarCity": "سايووا"  
  },  
  "Troll": {  
    "exemplarCity": "ترول"  
  },  
  "Vostok": {  
    "exemplarCity": "فوستوك"  
  },  
},  
"Etc": {  
  "GMT": {
```

```
    "exemplarCity": "GMT"
  },
  "GMT1": {
    "exemplarCity": "GMT+1"
  },
  "GMT10": {
    "exemplarCity": "GMT+10"
  },
  "GMT11": {
    "exemplarCity": "GMT+11"
  },
  "GMT12": {
    "exemplarCity": "GMT+12"
  },
  "GMT2": {
    "exemplarCity": "GMT+2"
  },
  "GMT3": {
    "exemplarCity": "GMT+3"
  },
  "GMT4": {
    "exemplarCity": "GMT+4"
  },
  "GMT5": {
    "exemplarCity": "GMT+5"
  },
  "GMT6": {
    "exemplarCity": "GMT+6"
  },
  "GMT7": {
    "exemplarCity": "GMT+7"
  },
  "GMT8": {
    "exemplarCity": "GMT+8"
  },
  "GMT9": {
    "exemplarCity": "GMT+9"
  },
  "GMT-1": {
    "exemplarCity": "GMT-1"
  },
  "GMT-10": {
    "exemplarCity": "GMT-10"
  },
  "GMT-11": {
    "exemplarCity": "GMT-11"
  },
  "GMT-12": {
    "exemplarCity": "GMT-12"
  },
  "GMT-13": {
    "exemplarCity": "GMT-13"
  },
  "GMT-14": {
    "exemplarCity": "GMT-14"
  },
  "GMT-2": {
```

```

        "exemplarCity": "GMT-2"
      },
      "GMT-3": {
        "exemplarCity": "GMT-3"
      },
      "GMT-4": {
        "exemplarCity": "GMT-4"
      },
      "GMT-5": {
        "exemplarCity": "GMT-5"
      },
      "GMT-6": {
        "exemplarCity": "GMT-6"
      },
      "GMT-7": {
        "exemplarCity": "GMT-7"
      },
      "GMT-8": {
        "exemplarCity": "GMT-8"
      },
      "GMT-9": {
        "exemplarCity": "GMT-9"
      },
      "Unknown": {
        "exemplarCity": "مدينة غير معروفة"
      }
    },
    "metazone": {
      "Afghanistan": {
        "long": {
          "standard": "توقيت أفغانستان"
        }
      },
      "Africa_Central": {
        "long": {
          "standard": "توقيت وسط أفريقيا"
        }
      },
      "Africa_Eastern": {
        "long": {
          "standard": "توقيت شرق أفريقيا"
        }
      },
      "Africa_Southern": {
        "long": {
          "standard": "توقيت جنوب أفريقيا"
        }
      },
      "Africa_Western": {
        "long": {
          "generic": "توقيت غرب أفريقيا",
          "standard": "توقيت غرب أفريقيا الرسمي",
          "daylight": "توقيت غرب أفريقيا الصيفي"
        }
      },
      "Alaska": {

```

```

    "long": {
      "generic": "توقيت ألاسكا",
      "standard": "التوقيت الرسمي لألاسكا",
      "daylight": "توقيت ألاسكا الصيفي"
    }
  },
  "Amazon": {
    "long": {
      "generic": "توقيت الأمازون",
      "standard": "توقيت الأمازون الرسمي",
      "daylight": "توقيت الأمازون الصيفي"
    }
  },
  "America_Central": {
    "long": {
      "generic": "التوقيت المركزي لأمريكا الشمالية",
      "standard": "التوقيت الرسمي المركزي لأمريكا الشمالية",
      "daylight": "التوقيت الصيفي المركزي لأمريكا الشمالية"
    }
  },
  "America_Eastern": {
    "long": {
      "generic": "التوقيت الشرقي لأمريكا الشمالية",
      "standard": "التوقيت الرسمي الشرقي لأمريكا الشمالية",
      "daylight": "التوقيت الصيفي الشرقي لأمريكا الشمالية"
    }
  },
  "America_Mountain": {
    "long": {
      "generic": "التوقيت الجبلي لأمريكا الشمالية",
      "standard": "التوقيت الرسمي الجبلي لأمريكا الشمالية",
      "daylight": "التوقيت الصيفي الجبلي لأمريكا الشمالية"
    }
  },
  "America_Pacific": {
    "long": {
      "generic": "توقيت المحيط الهادي",
      "standard": "توقيت المحيط الهادي الرسمي",
      "daylight": "توقيت المحيط الهادي الصيفي"
    }
  },
  "Anadyr": {
    "long": {
      "generic": "توقيت أنادير",
      "standard": "توقيت أنادير الرسمي",
      "daylight": "التوقيت الصيفي لأنادير"
    }
  },
  "Apia": {
    "long": {
      "generic": "توقيت آبيا",
      "standard": "التوقيت الرسمي لآبيا",
      "daylight": "التوقيت الصيفي لآبيا"
    }
  },
  "Arabian": {
    "long": {

```

```

        "generic": "التوقيت العربي",
        "standard": "التوقيت العربي الرسمي",
        "daylight": "التوقيت العربي الصيفي"
    }
},
"Argentina": {
    "long": {
        "generic": "توقيت الأرجنتين",
        "standard": "توقيت الأرجنتين الرسمي",
        "daylight": "توقيت الأرجنتين الصيفي"
    }
},
"Argentina_Western": {
    "long": {
        "generic": "توقيت غرب الأرجنتين",
        "standard": "توقيت غرب الأرجنتين الرسمي",
        "daylight": "توقيت غرب الأرجنتين الصيفي"
    }
},
"Armenia": {
    "long": {
        "generic": "توقيت أرمينيا",
        "standard": "توقيت أرمينيا الرسمي",
        "daylight": "توقيت أرمينيا الصيفي"
    }
},
"Atlantic": {
    "long": {
        "generic": "توقيت الأطلسي",
        "standard": "التوقيت الرسمي الأطلسي",
        "daylight": "التوقيت الصيفي الأطلسي"
    }
},
"Australia_Central": {
    "long": {
        "generic": "توقيت وسط أستراليا",
        "standard": "توقيت وسط أستراليا الرسمي",
        "daylight": "توقيت وسط أستراليا الصيفي"
    }
},
"Australia_CentralWestern": {
    "long": {
        "generic": "توقيت غرب وسط أستراليا",
        "standard": "توقيت غرب وسط أستراليا الرسمي",
        "daylight": "توقيت غرب وسط أستراليا الصيفي"
    }
},
"Australia_Eastern": {
    "long": {
        "generic": "توقيت شرق أستراليا",
        "standard": "توقيت شرق أستراليا الرسمي",
        "daylight": "توقيت شرق أستراليا الصيفي"
    }
},
"Australia_Western": {
    "long": {
        "generic": "توقيت غرب أستراليا",

```

```

        "standard": "توقيت غرب أستراليا الرسمي",
        "daylight": "توقيت غرب أستراليا الصيفي"
    },
    "Azerbaijan": {
        "long": {
            "generic": "توقيت أذربيجان",
            "standard": "توقيت أذربيجان الرسمي",
            "daylight": "توقيت أذربيجان الصيفي"
        }
    },
    "Azores": {
        "long": {
            "generic": "توقيت أزورس",
            "standard": "توقيت أزورس الرسمي",
            "daylight": "توقيت أزورس الصيفي"
        }
    },
    "Bangladesh": {
        "long": {
            "generic": "توقيت بنجلاديش",
            "standard": "توقيت بنجلاديش الرسمي",
            "daylight": "توقيت بنجلاديش الصيفي"
        }
    },
    "Bhutan": {
        "long": {
            "standard": "توقيت بوتان"
        }
    },
    "Bolivia": {
        "long": {
            "standard": "توقيت بوليفيا"
        }
    },
    "Brasilia": {
        "long": {
            "generic": "توقيت برازيليا",
            "standard": "توقيت برازيليا الرسمي",
            "daylight": "توقيت برازيليا الصيفي"
        }
    },
    "Brunei": {
        "long": {
            "standard": "توقيت بروناي"
        }
    },
    "Cape_Verde": {
        "long": {
            "generic": "توقيت الرأس الأخضر",
            "standard": "توقيت الرأس الأخضر الرسمي",
            "daylight": "توقيت الرأس الأخضر الصيفي"
        }
    },
    "Chamorro": {
        "long": {
            "standard": "توقيت تشامورو"
        }
    }

```



```

    }
  },
  "Chatham": {
    "long": {
      "generic": "توقيت تشاتام",
      "standard": "توقيت تشاتام الرسمي",
      "daylight": "توقيت تشاتام الصيفي"
    }
  },
  "Chile": {
    "long": {
      "generic": "توقيت شيلي",
      "standard": "توقيت شيلي الرسمي",
      "daylight": "توقيت شيلي الصيفي"
    }
  },
  "China": {
    "long": {
      "generic": "توقيت الصين",
      "standard": "توقيت الصين الرسمي",
      "daylight": "توقيت الصين الصيفي"
    }
  },
  "Choibalsan": {
    "long": {
      "generic": "توقيت شويبالسان",
      "standard": "توقيت شويبالسان الرسمي",
      "daylight": "التوقيت الصيفي لشويبالسان"
    }
  },
  "Christmas": {
    "long": {
      "standard": "توقيت جزر الكريسماس"
    }
  },
  "Cocos": {
    "long": {
      "standard": "توقيت جزر كوكوس"
    }
  },
  "Colombia": {
    "long": {
      "generic": "توقيت كولومبيا",
      "standard": "توقيت كولومبيا الرسمي",
      "daylight": "توقيت كولومبيا الصيفي"
    }
  },
  "Cook": {
    "long": {
      "generic": "توقيت جزر كوك",
      "standard": "توقيت جزر كوك الرسمي",
      "daylight": "توقيت جزر كوك الصيفي"
    }
  },
  "Cuba": {
    "long": {
      "generic": "توقيت كوبا",

```

```

        "standard": "توقيت كوبا الرسمي",
        "daylight": "توقيت كوبا الصيفي"
    },
    },
    "Davis": {
        "long": {
            "standard": "توقيت دافيز"
        }
    },
    },
    "DumontDUrville": {
        "long": {
            "standard": "توقيت دي مونت دو روفيل"
        }
    },
    },
    "East_Timor": {
        "long": {
            "standard": "توقيت تيمور الشرقية"
        }
    },
    },
    "Easter": {
        "long": {
            "generic": "توقيت جزيرة استر",
            "standard": "توقيت جزيرة استر الرسمي",
            "daylight": "توقيت جزيرة استر الصيفي"
        }
    },
    },
    "Ecuador": {
        "long": {
            "standard": "توقيت الإكوادور"
        }
    },
    },
    "Europe_Central": {
        "long": {
            "generic": "توقيت وسط أوروبا",
            "standard": "توقيت وسط أوروبا الرسمي",
            "daylight": "توقيت وسط أوروبا الصيفي"
        }
    },
    },
    "Europe_Eastern": {
        "long": {
            "generic": "توقيت شرق أوروبا",
            "standard": "توقيت شرق أوروبا الرسمي",
            "daylight": "توقيت شرق أوروبا الصيفي"
        }
    },
    },
    "Europe_Further_Eastern": {
        "long": {
            "standard": "التوقيت الأوروبي (أكثر شرقًا)"
        }
    },
    },
    "Europe_Western": {
        "long": {
            "generic": "توقيت غرب أوروبا",
            "standard": "توقيت غرب أوروبا الرسمي",
            "daylight": "توقيت غرب أوروبا الصيفي"
        }
    },
    },

```

```
"Falkland": {
  "long": {
    "generic": "توقيت جزر فوكلاند",
    "standard": "توقيت جزر فوكلاند الرسمي",
    "daylight": "توقيت جزر فوكلاند الصيفي"
  }
},
"Fiji": {
  "long": {
    "generic": "توقيت فيجي",
    "standard": "توقيت فيجي الرسمي",
    "daylight": "توقيت فيجي الصيفي"
  }
},
"French_Guiana": {
  "long": {
    "standard": "توقيت غايانا الفرنسية"
  }
},
"French_Southern": {
  "long": {
    "standard": "توقيت المقاطعات الفرنسية الجنوبية والأنتارتيكية"
  }
},
"Galapagos": {
  "long": {
    "standard": "توقيت غلاباغوس"
  }
},
"Gambier": {
  "long": {
    "standard": "توقيت جامبير"
  }
},
"Georgia": {
  "long": {
    "generic": "توقيت جورجيا",
    "standard": "توقيت جورجيا الرسمي",
    "daylight": "توقيت جورجيا الصيفي"
  }
},
"Gilbert_Islands": {
  "long": {
    "standard": "توقيت جزر جيلبرت"
  }
},
"GMT": {
  "long": {
    "standard": "توقيت غرينتش"
  }
},
"Greenland_Eastern": {
  "long": {
    "generic": "توقيت شرق غرينلاند",
    "standard": "توقيت شرق غرينلاند الرسمي",
    "daylight": "توقيت شرق غرينلاند الصيفي"
  }
}
```

```

    },
    "Greenland_Western": {
      "long": {
        "generic": "توقيت غرب غرينلاند",
        "standard": "توقيت غرب غرينلاند الرسمي",
        "daylight": "توقيت غرب غرينلاند الصيفي"
      }
    },
    "Guam": {
      "long": {
        "standard": "توقيت غوام"
      }
    },
    "Gulf": {
      "long": {
        "standard": "توقيت الخليج"
      }
    },
    "Guyana": {
      "long": {
        "standard": "توقيت غيانا"
      }
    },
    "Hawaii_Aleutian": {
      "long": {
        "generic": "توقيت هاواي ألوتيان",
        "standard": "توقيت هاواي ألوتيان الرسمي",
        "daylight": "توقيت هاواي ألوتيان الصيفي"
      }
    },
    "Hong_Kong": {
      "long": {
        "generic": "توقيت هونغ كونغ",
        "standard": "توقيت هونغ كونغ الرسمي",
        "daylight": "توقيت هونغ كونغ الصيفي"
      }
    },
    "Hovd": {
      "long": {
        "generic": "توقيت هوفد",
        "standard": "توقيت هوفد الرسمي",
        "daylight": "توقيت هوفد الصيفي"
      }
    },
    "India": {
      "long": {
        "standard": "توقيت الهند"
      }
    },
    "Indian_Ocean": {
      "long": {
        "standard": "توقيت المحيط الهندي"
      }
    },
    "Indochina": {
      "long": {
        "standard": "توقيت الهند الصينية"
      }
    }
  }

```

```

    }
  },
  "Indonesia_Central": {
    "long": {
      "standard": "توقيت وسط إندونيسيا"
    }
  },
  "Indonesia_Eastern": {
    "long": {
      "standard": "توقيت شرق إندونيسيا"
    }
  },
  "Indonesia_Western": {
    "long": {
      "standard": "توقيت غرب إندونيسيا"
    }
  },
  "Iran": {
    "long": {
      "generic": "توقيت إيران",
      "standard": "توقيت إيران الرسمي",
      "daylight": "توقيت إيران الصيفي"
    }
  },
  "Irkutsk": {
    "long": {
      "generic": "توقيت إركوتسك",
      "standard": "توقيت إركوتسك الرسمي",
      "daylight": "توقيت إركوتسك الصيفي"
    }
  },
  "Israel": {
    "long": {
      "generic": "توقيت إسرائيل",
      "standard": "توقيت إسرائيل الرسمي",
      "daylight": "توقيت إسرائيل الصيفي"
    }
  },
  "Japan": {
    "long": {
      "generic": "توقيت اليابان",
      "standard": "توقيت اليابان الرسمي",
      "daylight": "توقيت اليابان الصيفي"
    }
  },
  "Kamchatka": {
    "long": {
      "generic": "توقيت كامشاتكا",
      "standard": "توقيت بيتروبافلوفسك-كامتشاتسكي",
      "daylight": "توقيت بيتروبافلوفسك-كامتشاتسكي الصيفي"
    }
  },
  "Kazakhstan_Eastern": {
    "long": {
      "standard": "توقيت شرق كازاخستان"
    }
  }
},

```

```

    "Kazakhstan_Western": {
      "long": {
        "standard": "توقيت غرب كازاخستان"
      }
    },
    "Korea": {
      "long": {
        "generic": "توقيت كوريا",
        "standard": "توقيت كوريا الرسمي",
        "daylight": "توقيت كوريا الصيفي"
      }
    },
    "Kosrae": {
      "long": {
        "standard": "توقيت كوسرا"
      }
    },
    "Krasnoyarsk": {
      "long": {
        "generic": "توقيت كراسنويارسك",
        "standard": "توقيت كراسنويارسك الرسمي",
        "daylight": "التوقيت الصيفي لكراسنويارسك"
      }
    },
    "Kyrgystan": {
      "long": {
        "standard": "توقيت قرغيزستان"
      }
    },
    "Line_Islands": {
      "long": {
        "standard": "توقيت جزر لاین"
      }
    },
    "Lord_Howe": {
      "long": {
        "generic": "توقيت لورد هاو",
        "standard": "توقيت لورد هاو الرسمي",
        "daylight": "التوقيت الصيفي للورد هاو"
      }
    },
    "Macquarie": {
      "long": {
        "standard": "توقيت ماكوارى"
      }
    },
    "Magadan": {
      "long": {
        "generic": "توقيت ماغادان",
        "standard": "توقيت ماغادان الرسمي",
        "daylight": "توقيت ماغادان الصيفي"
      }
    },
    "Malaysia": {
      "long": {
        "standard": "توقيت ماليزيا"
      }
    }
  }

```

```

    },
    "Maldives": {
      "long": {
        "standard": "توقيت المالديف"
      }
    },
    "Marquesas": {
      "long": {
        "standard": "توقيت ماركيساس"
      }
    },
    "Marshall_Islands": {
      "long": {
        "standard": "توقيت جزر مارشال"
      }
    },
    "Mauritius": {
      "long": {
        "generic": "توقيت موريشيوس",
        "standard": "توقيت موريشيوس الرسمي",
        "daylight": "توقيت موريشيوس الصيفي"
      }
    },
    "Mawson": {
      "long": {
        "standard": "توقيت ماوسون"
      }
    },
    "Mexico_Northwest": {
      "long": {
        "generic": "توقيت شمال غرب المكسيك",
        "standard": "التوقيت الرسمي لشمال غرب المكسيك",
        "daylight": "التوقيت الصيفي لشمال غرب المكسيك"
      }
    },
    "Mexico_Pacific": {
      "long": {
        "generic": "توقيت المحيط الهادي للمكسيك",
        "standard": "توقيت المحيط الهادي الرسمي للمكسيك",
        "daylight": "توقيت المحيط الهادي الصيفي للمكسيك"
      }
    },
    "Mongolia": {
      "long": {
        "generic": "توقيت أولان باتور",
        "standard": "توقيت أولان باتور الرسمي",
        "daylight": "توقيت أولان باتور الصيفي"
      }
    },
    "Moscow": {
      "long": {
        "generic": "توقيت موسكو",
        "standard": "توقيت موسكو الرسمي",
        "daylight": "توقيت موسكو الصيفي"
      }
    },
    "Myanmar": {

```

```

        "long": {
            "standard": "توقيت میانمار"
        }
    },
    "Nauru": {
        "long": {
            "standard": "توقيت ناورو"
        }
    },
    "Nepal": {
        "long": {
            "standard": "توقيت نيبال"
        }
    },
    "New_Caledonia": {
        "long": {
            "generic": "توقيت كاليدونيا الجديدة",
            "standard": "توقيت كاليدونيا الجديدة الرسمي",
            "daylight": "توقيت كاليدونيا الجديدة الصيفي"
        }
    },
    "New_Zealand": {
        "long": {
            "generic": "توقيت نيوزيلندا",
            "standard": "توقيت نيوزيلندا الرسمي",
            "daylight": "توقيت نيوزيلندا الصيفي"
        }
    },
    "Newfoundland": {
        "long": {
            "generic": "توقيت نيوفاوندلاند",
            "standard": "توقيت نيوفاوندلاند الرسمي",
            "daylight": "توقيت نيوفاوندلاند الصيفي"
        }
    },
    "Niue": {
        "long": {
            "standard": "توقيت نيوي"
        }
    },
    "Norfolk": {
        "long": {
            "standard": "توقيت جزيرة نورفولك"
        }
    },
    "Noronha": {
        "long": {
            "generic": "توقيت فيرناندو دي نورونها",
            "standard": "توقيت فرناندو دي نورونها الرسمي",
            "daylight": "توقيت فرناندو دي نورونها الصيفي"
        }
    },
    "North_Mariana": {
        "long": {
            "standard": "توقيت جزر ماريانا الشمالية"
        }
    }
},

```



```

"Novosibirsk": {
  "long": {
    "generic": "توقيت نوفوسيبيرسك",
    "standard": "توقيت نوفوسيبيرسك الرسمي",
    "daylight": "توقيت نوفوسيبيرسك الصيفي"
  }
},
"Omsk": {
  "long": {
    "generic": "توقيت أومسك",
    "standard": "توقيت أومسك الرسمي",
    "daylight": "توقيت أومسك الصيفي"
  }
},
"Pakistan": {
  "long": {
    "generic": "توقيت باكستان",
    "standard": "توقيت باكستان الرسمي",
    "daylight": "توقيت باكستان الصيفي"
  }
},
"Palau": {
  "long": {
    "standard": "توقيت بالاو"
  }
},
"Papua_New_Guinea": {
  "long": {
    "standard": "توقيت بابوا غينيا الجديدة"
  }
},
"Paraguay": {
  "long": {
    "generic": "توقيت باراغواي",
    "standard": "توقيت باراغواي الرسمي",
    "daylight": "توقيت باراغواي الصيفي"
  }
},
"Peru": {
  "long": {
    "generic": "توقيت بيرو",
    "standard": "توقيت بيرو الرسمي",
    "daylight": "توقيت بيرو الصيفي"
  }
},
"Philippines": {
  "long": {
    "generic": "توقيت الفلبين",
    "standard": "توقيت الفلبين الرسمي",
    "daylight": "توقيت الفلبين الصيفي"
  }
},
"Phoenix_Islands": {
  "long": {
    "standard": "توقيت جزر فينكس"
  }
},

```

```

"Pierre_Miquelon": {
  "long": {
    "generic": "توقيت سانت بيير وميكلون",
    "standard": "توقيت سانت بيير وميكلون الرسمي",
    "daylight": "توقيت سانت بيير وميكلون الصيفي"
  }
},
"Pitcairn": {
  "long": {
    "standard": "توقيت بيتكيرن"
  }
},
"Ponape": {
  "long": {
    "standard": "توقيت بونابي"
  }
},
"Pyongyang": {
  "long": {
    "standard": "توقيت بيونغ يانغ"
  }
},
"Reunion": {
  "long": {
    "standard": "توقيت ريونيون"
  }
},
"Rothera": {
  "long": {
    "standard": "توقيت روثيرا"
  }
},
"Sakhalin": {
  "long": {
    "generic": "توقيت ساخالين",
    "standard": "توقيت ساخالين الرسمي",
    "daylight": "توقيت ساخالين الصيفي"
  }
},
"Samara": {
  "long": {
    "generic": "توقيت سامارا",
    "standard": "توقيت سمارة",
    "daylight": "توقيت سمارة الصيفي"
  }
},
"Samoa": {
  "long": {
    "generic": "توقيت ساموا",
    "standard": "توقيت ساموا الرسمي",
    "daylight": "توقيت ساموا الصيفي"
  }
},
"Seychelles": {
  "long": {
    "standard": "توقيت سيشل"
  }
}

```

```

    },
    "Singapore": {
      "long": {
        "standard": "توقيت سنغافورة"
      }
    },
    "Solomon": {
      "long": {
        "standard": "توقيت جزر سليمان"
      }
    },
    "South_Georgia": {
      "long": {
        "standard": "توقيت جنوب جورجيا"
      }
    },
    "Suriname": {
      "long": {
        "standard": "توقيت سورينام"
      }
    },
    "Syowa": {
      "long": {
        "standard": "توقيت سايووا"
      }
    },
    "Tahiti": {
      "long": {
        "standard": "توقيت تاهيتي"
      }
    },
    "Taipei": {
      "long": {
        "generic": "توقيت تايبه",
        "standard": "توقيت تايبه الرسمي",
        "daylight": "توقيت تايبه الصيفي"
      }
    },
    "Tajikistan": {
      "long": {
        "standard": "توقيت طاجكستان"
      }
    },
    "Tokelau": {
      "long": {
        "standard": "توقيت توكيلاو"
      }
    },
    "Tonga": {
      "long": {
        "generic": "توقيت تونغا",
        "standard": "توقيت تونغا الرسمي",
        "daylight": "توقيت تونغا الصيفي"
      }
    },
    "Truk": {
      "long": {

```

```

        "standard": "توقيت شوك"
    },
    },
    "Turkmenistan": {
        "long": {
            "generic": "توقيت تركمانستان",
            "standard": "توقيت تركمانستان الرسمي",
            "daylight": "توقيت تركمانستان الصيفي"
        }
    },
    "Tuvalu": {
        "long": {
            "standard": "توقيت توفالو"
        }
    },
    "Uruguay": {
        "long": {
            "generic": "توقيت أورغواي",
            "standard": "توقيت أورغواي الرسمي",
            "daylight": "توقيت أورغواي الصيفي"
        }
    },
    "Uzbekistan": {
        "long": {
            "generic": "توقيت أوزبكستان",
            "standard": "توقيت أوزبكستان الرسمي",
            "daylight": "توقيت أوزبكستان الصيفي"
        }
    },
    "Vanuatu": {
        "long": {
            "generic": "توقيت فانواتو",
            "standard": "توقيت فانواتو الرسمي",
            "daylight": "توقيت فانواتو الصيفي"
        }
    },
    "Venezuela": {
        "long": {
            "standard": "توقيت فنزويلا"
        }
    },
    "Vladivostok": {
        "long": {
            "generic": "توقيت فلاديفوستوك",
            "standard": "توقيت فلاديفوستوك الرسمي",
            "daylight": "توقيت فلاديفوستوك الصيفي"
        }
    },
    "Volgograd": {
        "long": {
            "generic": "توقيت فولغوغراد",
            "standard": "توقيت فولغوغراد الرسمي",
            "daylight": "توقيت فولغوغراد الصيفي"
        }
    },
    "Vostok": {
        "long": {

```

```

        "standard": "توقيت فوستوك"
      },
    },
    "Wake": {
      "long": {
        "standard": "توقيت جزيرة ويك"
      }
    },
    "Wallis": {
      "long": {
        "standard": "توقيت واليس و فوتونا"
      }
    },
    "Yakutsk": {
      "long": {
        "generic": "توقيت ياكوتسك",
        "standard": "توقيت ياكوتسك الرسمي",
        "daylight": "توقيت ياكوتسك الصيفي"
      }
    },
    "Yekaterinburg": {
      "long": {
        "generic": "توقيت يكاترينبورغ",
        "standard": "توقيت يكاترينبورغ الرسمي",
        "daylight": "توقيت يكاترينبورغ الصيفي"
      }
    }
  }
}

```

### TIMEZONENAMES.JSX

```

{
  "main";
  {
    "ar";
    {
      "identity";
      {
        "version";
        {
          "_number";
          "$Revision: 12879 $",
          "_cldrVersion";
          "30.0.3";
        }
        "language";
        "ar";
      }
      "dates";
      {
        "timeZoneNames";

```

```

    {
      "hourFormat";
      "+HH:mm;-HH:mm",
      "gmtFormat";
    }0{جرينتش",
      "gmtZeroFormat";
      "جرينتش",
      "regionFormat";
    }0{توقيت",
      "regionFormat-type-daylight";
      "الصيفي } 0{توقيت",
      "regionFormat-type-standard";
      "الرسمي } 0{توقيت",
      "fallbackFormat";
    }{1} ({0}) ",
      "zone";
    {
      "America";
      {
        "Adak";
        {
          "exemplarCity";
          "أداك";
        }
        "Anchorage";
        {
          "exemplarCity";
          "أنشوراج";
        }
        "Anguilla";
        {
          "exemplarCity";
          "أنغيلا";
        }
        "Antigua";
        {
          "exemplarCity";
          "أنتيغوا";
        }
        "Araguaina";
        {
          "exemplarCity";
          "أروجوانيا";
        }
        "Argentina";
        {
          "Rio_Gallegos";
          {
            "exemplarCity";
            "ريو جالييوس";
          }
          "San_Juan";
          {
            "exemplarCity";
            "سان خوان";
          }
          "Ushuaia";

```

```
{
  "exemplarCity";
  "أشوا";
}
"La_Rioja";
{
  "exemplarCity";
  "لا ريوجا";
}
"San_Luis";
{
  "exemplarCity";
  "سان لويس";
}
"Salta";
{
  "exemplarCity";
  "سالطا";
}
"Tucuman";
{
  "exemplarCity";
  "تاكمان";
}
}
"Aruba";
{
  "exemplarCity";
  "أروبا";
}
"Asuncion";
{
  "exemplarCity";
  "أسونسيون";
}
"Bahia";
{
  "exemplarCity";
  "باهيا";
}
"Bahia_Banderas";
{
  "exemplarCity";
  "باهيا بانديراس";
}
"Barbados";
{
  "exemplarCity";
  "بربادوس";
}
"Belem";
{
  "exemplarCity";
  "بلم";
}
"Belize";
{
```

```
        "exemplarCity";
        "بلیز";
    }
    "Blanc-Sablon";
    {
        "exemplarCity";
        "بلانک-سابلون";
    }
    "Boa_Vista";
    {
        "exemplarCity";
        "باو فیستا";
    }
    "Bogota";
    {
        "exemplarCity";
        "بوغوتا";
    }
    "Boise";
    {
        "exemplarCity";
        "بویس";
    }
    "Buenos_Aires";
    {
        "exemplarCity";
        "بوينوس آیرس";
    }
    "Cambridge_Bay";
    {
        "exemplarCity";
        "کامبردیج باي";
    }
    "Campo_Grande";
    {
        "exemplarCity";
        "کومبو جراند";
    }
    "Cancun";
    {
        "exemplarCity";
        "کانکون";
    }
    "Caracas";
    {
        "exemplarCity";
        "کاراکاس";
    }
    "Catamarca";
    {
        "exemplarCity";
        "کاتامارکا";
    }
    "Cayenne";
    {
        "exemplarCity";
        "کایین";
    }
```



```
}
"Cayman";
{
  "exemplarCity";
  "كيمان";
}
"Chicago";
{
  "exemplarCity";
  "شيكاغو";
}
"Chihuahua";
{
  "exemplarCity";
  "تشيوواوا";
}
"Coral_Harbour";
{
  "exemplarCity";
  "كورانال هاربر";
}
"Cordoba";
{
  "exemplarCity";
  "كوردوبا";
}
"Costa_Rica";
{
  "exemplarCity";
  "كوستاريكا";
}
"Creston";
{
  "exemplarCity";
  "كريستون";
}
"Cuiaba";
{
  "exemplarCity";
  "كيابا";
}
"Curacao";
{
  "exemplarCity";
  "كوراكاو";
}
"Danmarkshavn";
{
  "exemplarCity";
  "دانمرك شافن";
}
"Dawson";
{
  "exemplarCity";
  "داوسان";
}
"Dawson_Creek";
```

```
{
  "exemplarCity";
  "داوسن كريك";
}
"Denver";
{
  "exemplarCity";
  "دنفر";
}
"Detroit";
{
  "exemplarCity";
  "ديترويت";
}
"Dominica";
{
  "exemplarCity";
  "دومينيكا";
}
"Edmonton";
{
  "exemplarCity";
  "ايدمونتون";
}
"Eirunepe";
{
  "exemplarCity";
  "ايرونبي";
}
"El_Salvador";
{
  "exemplarCity";
  "السلفادور";
}
"Fort_Nelson";
{
  "exemplarCity";
  "فورت نيلسون";
}
"Fortaleza";
{
  "exemplarCity";
  "فورتاليزا";
}
"Glace_Bay";
{
  "exemplarCity";
  "جلاس باي";
}
"Godthab";
{
  "exemplarCity";
  "غودثاب";
}
"Goose_Bay";
{
  "exemplarCity";
```

```
        "جوس باي";
    }
    "Grand_Turk";
    {
        "exemplarCity";
        "گرانند ترك";
    }
    "Grenada";
    {
        "exemplarCity";
        "جرينادا";
    }
    "Guadeloupe";
    {
        "exemplarCity";
        "غوادلوب";
    }
    "Guatemala";
    {
        "exemplarCity";
        "غواتيمالا";
    }
    "Guayaquil";
    {
        "exemplarCity";
        "غواياكويل";
    }
    "Guyana";
    {
        "exemplarCity";
        "غيانا";
    }
    "Halifax";
    {
        "exemplarCity";
        "هاليفاكس";
    }
    "Havana";
    {
        "exemplarCity";
        "هافانا";
    }
    "Hermosillo";
    {
        "exemplarCity";
        "هيرموسيلو";
    }
    "Indiana";
    {
        "Vincennes";
        {
            "exemplarCity";
            "فينسينس";
        }
        "Petersburg";
        {
            "exemplarCity";
```

```
        "بيترسبرغ";
    }
    "Tell_City";
    {
        "exemplarCity";
        "مدينة تل، إنديانا";
    }
    "Knox";
    {
        "exemplarCity";
        "كونكس";
    }
    "Winamac";
    {
        "exemplarCity";
        "ويناماك";
    }
    "Marengo";
    {
        "exemplarCity";
        "مارنجو";
    }
    "Vevay";
    {
        "exemplarCity";
        "فيفاي";
    }
}
"Indianapolis";
{
    "exemplarCity";
    "إنديانا بوليس";
}
"Inuvik";
{
    "exemplarCity";
    "اينوفيك";
}
"Iqaluit";
{
    "exemplarCity";
    "اكيولت";
}
"Jamaica";
{
    "exemplarCity";
    "جامايكا";
}
"Jujuy";
{
    "exemplarCity";
    "جوجو";
}
"Juneau";
{
    "exemplarCity";
    "جونوي";
}
```

```
}
"Kentucky";
{
  "Monticello";
  {
    "exemplarCity";
    "مونتي سيلو";
  }
}
"Kralendijk";
{
  "exemplarCity";
  "كرالنديك";
}
"La_Paz";
{
  "exemplarCity";
  "لا باز";
}
"Lima";
{
  "exemplarCity";
  "ليما";
}
"Los_Angeles";
{
  "exemplarCity";
  "لوس انجلوس";
}
"Louisville";
{
  "exemplarCity";
  "لويس فيل";
}
"Lower_Princes";
{
  "exemplarCity";
  "حي الأمير السفلي";
}
"Maceio";
{
  "exemplarCity";
  "ماشيو";
}
"Managua";
{
  "exemplarCity";
  "ماناغوا";
}
"Manaus";
{
  "exemplarCity";
  "ماناوس";
}
"Marigot";
{
  "exemplarCity";
```

```
        "ماريغوت";
    }
    "Martinique";
    {
        "exemplarCity";
        "المارتينيك";
    }
    "Matamoros";
    {
        "exemplarCity";
        "ماتاموروس";
    }
    "Mazatlan";
    {
        "exemplarCity";
        "مازاتلان";
    }
    "Mendoza";
    {
        "exemplarCity";
        "ميندوزا";
    }
    "Menominee";
    {
        "exemplarCity";
        "مينوميني";
    }
    "Merida";
    {
        "exemplarCity";
        "ميريديا";
    }
    "Metlakatla";
    {
        "exemplarCity";
        "ميتلاكاتلا";
    }
    "Mexico_City";
    {
        "exemplarCity";
        "مدينة المكسيك";
    }
    "Miquelon";
    {
        "exemplarCity";
        "ميكلون";
    }
    "Moncton";
    {
        "exemplarCity";
        "وينكتون";
    }
    "Monterrey";
    {
        "exemplarCity";
        "مونتيري";
    }
}
```

```
"Montevideo";
{
  "exemplarCity";
  "مونتفيدو";
}
"Montserrat";
{
  "exemplarCity";
  "مونتسيرات";
}
"Nassau";
{
  "exemplarCity";
  "ناسو";
}
"New_York";
{
  "exemplarCity";
  "نيويورك";
}
"Nipigon";
{
  "exemplarCity";
  "نيبيجون";
}
"Nome";
{
  "exemplarCity";
  "نوم";
}
"Noronha";
{
  "exemplarCity";
  "نوروناه";
}
"North_Dakota";
{
  "Beulah";
  {
    "exemplarCity";
    "بيولا، داکوتا الشمالية";
  }
  "New_Salem";
  {
    "exemplarCity";
    "نيو ساليم";
  }
  "Center";
  {
    "exemplarCity";
    "سنتر";
  }
}
"Ojinaga";
{
  "exemplarCity";
  "أوجيناجا";
}
```

```
}
"Panama";
{
  "exemplarCity";
  "بنما";
}
"Pangnirtung";
{
  "exemplarCity";
  "بانجینتینگ";
}
"Paramaribo";
{
  "exemplarCity";
  "باراماریبو";
}
"Phoenix";
{
  "exemplarCity";
  "فینکس";
}
"Port-au-Prince";
{
  "exemplarCity";
  "پورت أو برنس";
}
"Port_of_Spain";
{
  "exemplarCity";
  "پورت أوف سبین";
}
"Porto_Velho";
{
  "exemplarCity";
  "پورتو فیلو";
}
"Puerto_Rico";
{
  "exemplarCity";
  "پورتوریکو";
}
"Rainy_River";
{
  "exemplarCity";
  "رانی ریفر";
}
"Rankin_Inlet";
{
  "exemplarCity";
  "رانکن انلت";
}
"Recife";
{
  "exemplarCity";
  "ریسیف";
}
"Regina";
```



```
{
  "exemplarCity";
  "ريجينا";
}
"Resolute";
{
  "exemplarCity";
  "ريزولوت";
}
"Rio_Branco";
{
  "exemplarCity";
  "ريوبرانكو";
}
"Santa_Isabel";
{
  "exemplarCity";
  "سانتا إيزابيل";
}
"Santarem";
{
  "exemplarCity";
  "سانتاريم";
}
"Santiago";
{
  "exemplarCity";
  "سانتياغو";
}
"Santo_Domingo";
{
  "exemplarCity";
  "سانتو دومينغو";
}
"Sao_Paulo";
{
  "exemplarCity";
  "ساو باولو";
}
"Scoresbysund";
{
  "exemplarCity";
  "سكورسبيسند";
}
"Sitka";
{
  "exemplarCity";
  "سيتكا";
}
"St_Barthelemy";
{
  "exemplarCity";
  "سانت بارتيليمي";
}
"St_Johns";
{
  "exemplarCity";
```

```
        "سانت جونس";
    }
    "St_Kitts";
    {
        "exemplarCity";
        "سانت کیتس";
    }
    "St_Lucia";
    {
        "exemplarCity";
        "سانت لوشیا";
    }
    "St_Thomas";
    {
        "exemplarCity";
        "سانت توماس";
    }
    "St_Vincent";
    {
        "exemplarCity";
        "سانت فنسنت";
    }
    "Swift_Current";
    {
        "exemplarCity";
        "سوفت کارنت";
    }
    "Tegucigalpa";
    {
        "exemplarCity";
        "تیغوسیغالبا";
    }
    "Thule";
    {
        "exemplarCity";
        "ٹیل";
    }
    "Thunder_Bay";
    {
        "exemplarCity";
        "ٹنڈر باي";
    }
    "Tijuana";
    {
        "exemplarCity";
        "ٹیخوانا";
    }
    "Toronto";
    {
        "exemplarCity";
        "ٹورونٹو";
    }
    "Tortola";
    {
        "exemplarCity";
        "ٹورتولا";
    }
}
```

```
"Vancouver";
{
  "exemplarCity";
  "فانكوفر";
}
"Whitehorse";
{
  "exemplarCity";
  "وايت هورس";
}
"Winnipeg";
{
  "exemplarCity";
  "وينيبج";
}
"Yakutat";
{
  "exemplarCity";
  "ياكوتات";
}
"Yellowknife";
{
  "exemplarCity";
  "يلونيف";
}
}
"Atlantic";
{
  "Azores";
  {
    "exemplarCity";
    "أزورس";
  }
  "Bermuda";
  {
    "exemplarCity";
    "برمودا";
  }
  "Canary";
  {
    "exemplarCity";
    "كناري";
  }
  "Cape_Verde";
  {
    "exemplarCity";
    "الرأس الأخضر";
  }
  "Faeroe";
  {
    "exemplarCity";
    "فارو";
  }
  "Madeira";
  {
    "exemplarCity";
    "ماديرا";
  }
}
```

```
}
"Reykjavik";
{
  "exemplarCity";
  "ريكيافيك";
}
"South_Georgia";
{
  "exemplarCity";
  "جورجيا الجنوبية";
}
"St_Helena";
{
  "exemplarCity";
  "سانت هيلينا";
}
"Stanley";
{
  "exemplarCity";
  "استانلي";
}
}
"Europe";
{
  "Amsterdam";
  {
    "exemplarCity";
    "أمستردام";
  }
  "Andorra";
  {
    "exemplarCity";
    "أندورا";
  }
  "Astrakhan";
  {
    "exemplarCity";
    "أستراخان";
  }
  "Athens";
  {
    "exemplarCity";
    "أثينا";
  }
  "Belgrade";
  {
    "exemplarCity";
    "بلغراد";
  }
  "Berlin";
  {
    "exemplarCity";
    "برلين";
  }
  "Bratislava";
  {
    "exemplarCity";
```

```
        "براتي سلافا";
    }
    "Brussels";
    {
        "exemplarCity";
        "بروكسل";
    }
    "Bucharest";
    {
        "exemplarCity";
        "بوخارست";
    }
    "Budapest";
    {
        "exemplarCity";
        "بودابست";
    }
    "Busingen";
    {
        "exemplarCity";
        "بوسنغن";
    }
    "Chisinau";
    {
        "exemplarCity";
        "تشييناو";
    }
    "Copenhagen";
    {
        "exemplarCity";
        "كوبنهاغن";
    }
    "Dublin";
    {
        "long";
        {
            "daylight";
            "توقيت أيرلندا الرسمي";
        }
        "exemplarCity";
        "دبلن";
    }
    "Gibraltar";
    {
        "exemplarCity";
        "جبل طارق";
    }
    "Guernsey";
    {
        "exemplarCity";
        "غيرنسي";
    }
    "Helsinki";
    {
        "exemplarCity";
        "هلسنكي";
    }
}
```

```
"Isle_of_Man";
{
  "exemplarCity";
  "جزيرة مان";
}
"Istanbul";
{
  "exemplarCity";
  "إسطنبول";
}
"Jersey";
{
  "exemplarCity";
  "جيرسي";
}
"Kaliningrad";
{
  "exemplarCity";
  "كالينجراد";
}
"Kiev";
{
  "exemplarCity";
  "كييف";
}
"Kirov";
{
  "exemplarCity";
  "كيروف";
}
"Lisbon";
{
  "exemplarCity";
  "لشبونة";
}
"Ljubljana";
{
  "exemplarCity";
  "ليوبليانا";
}
"London";
{
  "long";
  {
    "daylight";
    "توقيت بريطانيا الصيفي";
  }
  "exemplarCity";
  "لندن";
}
"Luxembourg";
{
  "exemplarCity";
  "لوكسمبورغ";
}
"Madrid";
{
```

```
        "exemplarCity";
        "مدريد";
    }
    "Malta";
    {
        "exemplarCity";
        "مالطة";
    }
    "Mariehamn";
    {
        "exemplarCity";
        "ماريهامن";
    }
    "Minsk";
    {
        "exemplarCity";
        "مينسك";
    }
    "Monaco";
    {
        "exemplarCity";
        "موناكو";
    }
    "Moscow";
    {
        "exemplarCity";
        "موسكو";
    }
    "Oslo";
    {
        "exemplarCity";
        "أوسلو";
    }
    "Paris";
    {
        "exemplarCity";
        "باريس";
    }
    "Podgorica";
    {
        "exemplarCity";
        "بودغوريكا";
    }
    "Prague";
    {
        "exemplarCity";
        "براغ";
    }
    "Riga";
    {
        "exemplarCity";
        "ريغا";
    }
    "Rome";
    {
        "exemplarCity";
        "روما";
    }
```

```
}
"Samara";
{
  "exemplarCity";
  "سمراء";
}
"San_Marino";
{
  "exemplarCity";
  "سان مارينو";
}
"Sarajevo";
{
  "exemplarCity";
  "سراييفو";
}
"Simferopol";
{
  "exemplarCity";
  "سيمفروبول";
}
"Skopje";
{
  "exemplarCity";
  "سكوبي";
}
"Sofia";
{
  "exemplarCity";
  "صوفيا";
}
"Stockholm";
{
  "exemplarCity";
  "ستوكهولم";
}
"Tallinn";
{
  "exemplarCity";
  "تالين";
}
"Tirane";
{
  "exemplarCity";
  "تيرانا";
}
"Ulyanovsk";
{
  "exemplarCity";
  "أوليانوفسك";
}
"Uzhgorod";
{
  "exemplarCity";
  "أوزجروود";
}
"Vaduz";
```



```
{
  "exemplarCity";
  "فادوز";
}
"Vatican";
{
  "exemplarCity";
  "الفاتيكان";
}
"Vienna";
{
  "exemplarCity";
  "فيينا";
}
"Vilnius";
{
  "exemplarCity";
  "فيلنيوس";
}
"Volgograd";
{
  "exemplarCity";
  "فولجograd";
}
"Warsaw";
{
  "exemplarCity";
  "وارسو";
}
"Zagreb";
{
  "exemplarCity";
  "زغرب";
}
"Zaporozhye";
{
  "exemplarCity";
  "زابوروزي";
}
"Zurich";
{
  "exemplarCity";
  "زيورخ";
}
}
"Africa";
{
  "Abidjan";
  {
    "exemplarCity";
    "أبيدجان";
  }
  "Accra";
  {
    "exemplarCity";
    "أكرا";
  }
}
```

```
"Addis_Ababa";
{
  "exemplarCity";
  "أديس أبابا";
}
"Algiers";
{
  "exemplarCity";
  "الجزائر";
}
"Asmera";
{
  "exemplarCity";
  "أسمرّة";
}
"Bamako";
{
  "exemplarCity";
  "باماكو";
}
"Bangui";
{
  "exemplarCity";
  "بانغوي";
}
"Banjul";
{
  "exemplarCity";
  "بانجول";
}
"Bissau";
{
  "exemplarCity";
  "بيساو";
}
"Blantyre";
{
  "exemplarCity";
  "بلانتيير";
}
"Brazzaville";
{
  "exemplarCity";
  "برازافيل";
}
"Bujumbura";
{
  "exemplarCity";
  "بوجومبورا";
}
"Cairo";
{
  "exemplarCity";
  "القاهرة";
}
"Casablanca";
{
```

```
        "exemplarCity";
        "الدار البيضاء";
    }
    "Ceuta";
    {
        "exemplarCity";
        "سيتا";
    }
    "Conakry";
    {
        "exemplarCity";
        "كوناكري";
    }
    "Dakar";
    {
        "exemplarCity";
        "داكار";
    }
    "Dar_es_Salaam";
    {
        "exemplarCity";
        "دار السلام";
    }
    "Djibouti";
    {
        "exemplarCity";
        "جيبوتي";
    }
    "Douala";
    {
        "exemplarCity";
        "دوالا";
    }
    "El_Aaiun";
    {
        "exemplarCity";
        "العيون";
    }
    "Freetown";
    {
        "exemplarCity";
        "فري تاون";
    }
    "Gaborone";
    {
        "exemplarCity";
        "غابورون";
    }
    "Harare";
    {
        "exemplarCity";
        "هراري";
    }
    "Johannesburg";
    {
        "exemplarCity";
        "جوهانسبرغ";
    }
}
```

```
}
"Juba";
{
  "exemplarCity";
  "جوبا";
}
"Kampala";
{
  "exemplarCity";
  "كامبالا";
}
"Khartoum";
{
  "exemplarCity";
  "الخرطوم";
}
"Kigali";
{
  "exemplarCity";
  "كيغالي";
}
"Kinshasa";
{
  "exemplarCity";
  "كينشاسا";
}
"Lagos";
{
  "exemplarCity";
  "لاغوس";
}
"Libreville";
{
  "exemplarCity";
  "ليبرفيل";
}
"Lome";
{
  "exemplarCity";
  "لومي";
}
"Luanda";
{
  "exemplarCity";
  "لواندا";
}
"Lubumbashi";
{
  "exemplarCity";
  "لومببasha";
}
"Lusaka";
{
  "exemplarCity";
  "لوساكا";
}
"Malabo";
```

```
{
  "exemplarCity";
  "مالابو";
}
"Maputo";
{
  "exemplarCity";
  "مابوتو";
}
"Maseru";
{
  "exemplarCity";
  "ماسيرو";
}
"Mbabane";
{
  "exemplarCity";
  "مباباني";
}
"Mogadishu";
{
  "exemplarCity";
  "مقديشيو";
}
"Monrovia";
{
  "exemplarCity";
  "مونروفيا";
}
"Nairobi";
{
  "exemplarCity";
  "نيروبي";
}
"Ndjamena";
{
  "exemplarCity";
  "نجامينا";
}
"Niamey";
{
  "exemplarCity";
  "نيامي";
}
"Nouakchott";
{
  "exemplarCity";
  "نواكشوط";
}
"Ouagadougou";
{
  "exemplarCity";
  "واغادوغو";
}
"Porto-Novo";
{
  "exemplarCity";
```

```

        "بورتو نوفو";
    }
    "Sao_Tome";
    {
        "exemplarCity";
        "ساو تومي";
    }
    "Tripoli";
    {
        "exemplarCity";
        "طرابلس";
    }
    "Tunis";
    {
        "exemplarCity";
        "تونس";
    }
    "Windhoek";
    {
        "exemplarCity";
        "ويندهوك";
    }
}
"Asia";
{
    "Aden";
    {
        "exemplarCity";
        "عدن";
    }
    "Almaty";
    {
        "exemplarCity";
        "ألماتي";
    }
    "Amman";
    {
        "exemplarCity";
        "عمان";
    }
    "Anadyr";
    {
        "exemplarCity";
        "أندير";
    }
    "Aqtau";
    {
        "exemplarCity";
        "أكتاو";
    }
    "Aqtobe";
    {
        "exemplarCity";
        "أكتوب";
    }
    "Ashgabat";
    {

```

```
        "exemplarCity";
        "عشق آباد";
    }
    "Baghdad";
    {
        "exemplarCity";
        "بغداد";
    }
    "Bahrain";
    {
        "exemplarCity";
        "البحرين";
    }
    "Baku";
    {
        "exemplarCity";
        "باكو";
    }
    "Bangkok";
    {
        "exemplarCity";
        "بانكوك";
    }
    "Barnaul";
    {
        "exemplarCity";
        "بارناول";
    }
    "Beirut";
    {
        "exemplarCity";
        "بيروت";
    }
    "Bishkek";
    {
        "exemplarCity";
        "بشكيك";
    }
    "Brunei";
    {
        "exemplarCity";
        "بروناي";
    }
    "Calcutta";
    {
        "exemplarCity";
        "كالكتا";
    }
    "Chita";
    {
        "exemplarCity";
        "تشيتا";
    }
    "Choibalsan";
    {
        "exemplarCity";
        "تشوبالسان";
    }
```

```
}
"Colombo";
{
  "exemplarCity";
  "كولومبو";
}
"Damascus";
{
  "exemplarCity";
  "دمشق";
}
"Dhaka";
{
  "exemplarCity";
  "دكا";
}
"Dili";
{
  "exemplarCity";
  "ديلي";
}
"Dubai";
{
  "exemplarCity";
  "دبي";
}
"Dushanbe";
{
  "exemplarCity";
  "دوشانبي";
}
"Gaza";
{
  "exemplarCity";
  "غزة";
}
"Hebron";
{
  "exemplarCity";
  "هيبرون (مدينة الخليل)";
}
"Hong_Kong";
{
  "exemplarCity";
  "هونغ كونغ";
}
"Hovd";
{
  "exemplarCity";
  "هوفد";
}
"Irkutsk";
{
  "exemplarCity";
  "ايركيتسك";
}
"Jakarta";
```



```
{
  "exemplarCity";
  "جاكرتا";
}
"Jayapura";
{
  "exemplarCity";
  "جاياپورا";
}
"Jerusalem";
{
  "exemplarCity";
  "القدس";
}
"Kabul";
{
  "exemplarCity";
  "كابل";
}
"Kamchatka";
{
  "exemplarCity";
  "كامتشاتكا";
}
"Karachi";
{
  "exemplarCity";
  "كراتشي";
}
"Katmandu";
{
  "exemplarCity";
  "كاتماندو";
}
"Khandyga";
{
  "exemplarCity";
  "خانديجا";
}
"Krasnoyarsk";
{
  "exemplarCity";
  "كراسنويارسك";
}
"Kuala_Lumpur";
{
  "exemplarCity";
  "كوالا لامبور";
}
"Kuching";
{
  "exemplarCity";
  "كيشينج";
}
"Kuwait";
{
  "exemplarCity";
```

```
        "الكويت";
    }
    "Macau";
    {
        "exemplarCity";
        "ماكاو";
    }
    "Magadan";
    {
        "exemplarCity";
        "مجادن";
    }
    "Makassar";
    {
        "exemplarCity";
        "ماكassar";
    }
    "Manila";
    {
        "exemplarCity";
        "مانيل";
    }
    "Muscat";
    {
        "exemplarCity";
        "مسقط";
    }
    "Nicosia";
    {
        "exemplarCity";
        "نيقوسيا";
    }
    "Novokuznetsk";
    {
        "exemplarCity";
        "نوفوكوزنتسك";
    }
    "Novosibirsk";
    {
        "exemplarCity";
        "نوفوسبيرسك";
    }
    "Omsk";
    {
        "exemplarCity";
        "أومسك";
    }
    "Oral";
    {
        "exemplarCity";
        "أورال";
    }
    "Phnom_Penh";
    {
        "exemplarCity";
        "بنوم بنه";
    }
}
```

```
"Pontianak";
{
  "exemplarCity";
  "بونتيانك";
}
"Pyongyang";
{
  "exemplarCity";
  "بيونغ يانغ";
}
"Qatar";
{
  "exemplarCity";
  "قطر";
}
"Qyzylorda";
{
  "exemplarCity";
  "كيزيلوردا";
}
"Rangoon";
{
  "exemplarCity";
  "رانغون";
}
"Riyadh";
{
  "exemplarCity";
  "الرياض";
}
"Saigon";
{
  "exemplarCity";
  "مدينة هو تشي منه";
}
"Sakhalin";
{
  "exemplarCity";
  "سكالين";
}
"Samarkand";
{
  "exemplarCity";
  "سمرقند";
}
"Seoul";
{
  "exemplarCity";
  "سول";
}
"Shanghai";
{
  "exemplarCity";
  "شنغهاي";
}
"Singapore";
{
```

```
        "exemplarCity";
        "سنغافورة";
    }
    "Srednekolymsk";
    {
        "exemplarCity";
        "سريدنكوليمسك";
    }
    "Taipei";
    {
        "exemplarCity";
        "تايبه";
    }
    "Tashkent";
    {
        "exemplarCity";
        "طشقند";
    }
    "Tbilisi";
    {
        "exemplarCity";
        "تبليسي";
    }
    "Tehran";
    {
        "exemplarCity";
        "طهران";
    }
    "Thimphu";
    {
        "exemplarCity";
        "تيمفو";
    }
    "Tokyo";
    {
        "exemplarCity";
        "طوكيو";
    }
    "Tomsk";
    {
        "exemplarCity";
        "تومسك";
    }
    "Ulaanbaatar";
    {
        "exemplarCity";
        "آلانباتار";
    }
    "Urumqi";
    {
        "exemplarCity";
        "أرومكي";
    }
    "Ust-Nera";
    {
        "exemplarCity";
        "أوست نيرا";
    }
```

```
}
  "Vientiane";
  {
    "exemplarCity";
    "فيانتيان";
  }
  "Vladivostok";
  {
    "exemplarCity";
    "فلاديفوستك";
  }
  "Yakutsk";
  {
    "exemplarCity";
    "ياكتسك";
  }
  "Yekaterinburg";
  {
    "exemplarCity";
    "يكاترينبيرج";
  }
  "Yerevan";
  {
    "exemplarCity";
    "يريفان";
  }
}
"Indian";
{
  "Antananarivo";
  {
    "exemplarCity";
    "أنتاناناريفو";
  }
  "Chagos";
  {
    "exemplarCity";
    "تشاغوس";
  }
  "Christmas";
  {
    "exemplarCity";
    "كريسماس";
  }
  "Cocos";
  {
    "exemplarCity";
    "كوكوس";
  }
  "Comoro";
  {
    "exemplarCity";
    "جزر القمر";
  }
  "Kerguelen";
  {
    "exemplarCity";
```

```

        "كيرغويلين";
    }
    "Mahe";
    {
        "exemplarCity";
        "ماهي";
    }
    "Maldives";
    {
        "exemplarCity";
        "المالديف";
    }
    "Mauritius";
    {
        "exemplarCity";
        "موريشيوس";
    }
    "Mayotte";
    {
        "exemplarCity";
        "مايوت";
    }
    "Reunion";
    {
        "exemplarCity";
        "ريونيون";
    }
}
"Australia";
{
    "Adelaide";
    {
        "exemplarCity";
        "أديليد";
    }
    "Brisbane";
    {
        "exemplarCity";
        "برسبان";
    }
    "Broken_Hill";
    {
        "exemplarCity";
        "بروكن هيل";
    }
    "Currie";
    {
        "exemplarCity";
        "كوري";
    }
    "Darwin";
    {
        "exemplarCity";
        "دارون";
    }
    "Eucla";
    {

```

```
        "exemplarCity";
        "أوكلا";
    }
    "Hobart";
    {
        "exemplarCity";
        "هوبارت";
    }
    "Lindeman";
    {
        "exemplarCity";
        "ليندمان";
    }
    "Lord_Howe";
    {
        "exemplarCity";
        "لورد هاو";
    }
    "Melbourne";
    {
        "exemplarCity";
        "ميلبورن";
    }
    "Perth";
    {
        "exemplarCity";
        "برثا";
    }
    "Sydney";
    {
        "exemplarCity";
        "سيدني";
    }
}
"Pacific";
{
    "Apia";
    {
        "exemplarCity";
        "أبيا";
    }
    "Auckland";
    {
        "exemplarCity";
        "أوكلاند";
    }
    "Bougainville";
    {
        "exemplarCity";
        "بوغانفيل";
    }
    "Chatham";
    {
        "exemplarCity";
        "تشاثام";
    }
    "Easter";
```

```
{
  "exemplarCity";
  "استر";
}
"Efate";
{
  "exemplarCity";
  "إيفات";
}
"Enderbury";
{
  "exemplarCity";
  "اندربيرج";
}
"Fakaofu";
{
  "exemplarCity";
  "فاكاوفو";
}
"Fiji";
{
  "exemplarCity";
  "فيجي";
}
"Funafuti";
{
  "exemplarCity";
  "فونافوتي";
}
"Galapagos";
{
  "exemplarCity";
  "جلاپاجوس";
}
"Gambier";
{
  "exemplarCity";
  "جامبير";
}
"Guadalcanal";
{
  "exemplarCity";
  "غواد الكانال";
}
"Guam";
{
  "exemplarCity";
  "غوام";
}
"Honolulu";
{
  "exemplarCity";
  "هونولولو";
}
"Johnston";
{
  "exemplarCity";
```



```
        "جونستون";
    }
    "Kiritimati";
    {
        "exemplarCity";
        "كيريتي ماتي";
    }
    "Kosrae";
    {
        "exemplarCity";
        "كوسرا";
    }
    "Kwajalein";
    {
        "exemplarCity";
        "كواجالين";
    }
    "Majuro";
    {
        "exemplarCity";
        "ماجورو";
    }
    "Marquesas";
    {
        "exemplarCity";
        "ماركيساس";
    }
    "Midway";
    {
        "exemplarCity";
        "ميدواي";
    }
    "Nauru";
    {
        "exemplarCity";
        "ناورو";
    }
    "Niue";
    {
        "exemplarCity";
        "نيوي";
    }
    "Norfolk";
    {
        "exemplarCity";
        "نورفولك";
    }
    "Noumea";
    {
        "exemplarCity";
        "نوميا";
    }
    "Pago_Pago";
    {
        "exemplarCity";
        "باغو باغو";
    }
}
```

```
"Palau";
{
  "exemplarCity";
  "بالاو";
}
"Pitcairn";
{
  "exemplarCity";
  "بيتكيرن";
}
"Ponape";
{
  "exemplarCity";
  "باناب";
}
"Port_Moresby";
{
  "exemplarCity";
  "بور مورسبي";
}
"Rarotonga";
{
  "exemplarCity";
  "راروتونغا";
}
"Saipan";
{
  "exemplarCity";
  "سايبان";
}
"Tahiti";
{
  "exemplarCity";
  "تاهيتي";
}
"Tarawa";
{
  "exemplarCity";
  "تاراوا";
}
"Tongatapu";
{
  "exemplarCity";
  "تونغاتابو";
}
"Truk";
{
  "exemplarCity";
  "ترك";
}
"Wake";
{
  "exemplarCity";
  "واك";
}
"Wallis";
{
```

```
        "exemplarCity";
        "واليس";
    }
}
"Arctic";
{
    "Longyearbyen";
    {
        "exemplarCity";
        "لونجيربين";
    }
}
"Antarctica";
{
    "Casey";
    {
        "exemplarCity";
        "كاساي";
    }
    "Davis";
    {
        "exemplarCity";
        "دافيز";
    }
    "DumontDUrville";
    {
        "exemplarCity";
        "دي مونت دو روفيل";
    }
    "Macquarie";
    {
        "exemplarCity";
        "ماكوارى";
    }
    "Mawson";
    {
        "exemplarCity";
        "ماوسون";
    }
    "McMurdo";
    {
        "exemplarCity";
        "ماك موردو";
    }
    "Palmer";
    {
        "exemplarCity";
        "بالمير";
    }
    "Rothera";
    {
        "exemplarCity";
        "روثيرا";
    }
    "Syowa";
    {
        "exemplarCity";
```

```
        "ساپووا";
    }
    "Troll";
    {
        "exemplarCity";
        "ترول";
    }
    "Vostok";
    {
        "exemplarCity";
        "فوستوك";
    }
}
"Etc";
{
    "GMT";
    {
        "exemplarCity";
        "GMT";
    }
    "GMT1";
    {
        "exemplarCity";
        "GMT+1";
    }
    "GMT10";
    {
        "exemplarCity";
        "GMT+10";
    }
    "GMT11";
    {
        "exemplarCity";
        "GMT+11";
    }
    "GMT12";
    {
        "exemplarCity";
        "GMT+12";
    }
    "GMT2";
    {
        "exemplarCity";
        "GMT+2";
    }
    "GMT3";
    {
        "exemplarCity";
        "GMT+3";
    }
    "GMT4";
    {
        "exemplarCity";
        "GMT+4";
    }
    "GMT5";
    {
```

```
        "exemplarCity";
        "GMT+5";
    }
    "GMT6";
    {
        "exemplarCity";
        "GMT+6";
    }
    "GMT7";
    {
        "exemplarCity";
        "GMT+7";
    }
    "GMT8";
    {
        "exemplarCity";
        "GMT+8";
    }
    "GMT9";
    {
        "exemplarCity";
        "GMT+9";
    }
    "GMT-1";
    {
        "exemplarCity";
        "GMT-1";
    }
    "GMT-10";
    {
        "exemplarCity";
        "GMT-10";
    }
    "GMT-11";
    {
        "exemplarCity";
        "GMT-11";
    }
    "GMT-12";
    {
        "exemplarCity";
        "GMT-12";
    }
    "GMT-13";
    {
        "exemplarCity";
        "GMT-13";
    }
    "GMT-14";
    {
        "exemplarCity";
        "GMT-14";
    }
    "GMT-2";
    {
        "exemplarCity";
        "GMT-2";
    }
```

```

    }
    "GMT-3";
    {
      "exemplarCity";
      "GMT-3";
    }
    "GMT-4";
    {
      "exemplarCity";
      "GMT-4";
    }
    "GMT-5";
    {
      "exemplarCity";
      "GMT-5";
    }
    "GMT-6";
    {
      "exemplarCity";
      "GMT-6";
    }
    "GMT-7";
    {
      "exemplarCity";
      "GMT-7";
    }
    "GMT-8";
    {
      "exemplarCity";
      "GMT-8";
    }
    "GMT-9";
    {
      "exemplarCity";
      "GMT-9";
    }
    "Unknown";
    {
      "exemplarCity";
      "مدينة غير معروفة";
    }
  }
}
"metazone";
{
  "Afghanistan";
  {
    "long";
    {
      "standard";
      "توقيت أفغانستان";
    }
  }
}
"Africa_Central";
{
  "long";
  {

```

```

        "standard";
        "توقيت وسط أفريقيا";
    }
}
"Africa_Eastern";
{
    "long";
    {
        "standard";
        "توقيت شرق أفريقيا";
    }
}
"Africa_Southern";
{
    "long";
    {
        "standard";
        "توقيت جنوب أفريقيا";
    }
}
"Africa_Western";
{
    "long";
    {
        "generic";
        "توقيت غرب أفريقيا",
        "standard";
        "توقيت غرب أفريقيا الرسمي",
        "daylight";
        "توقيت غرب أفريقيا الصيفي";
    }
}
"Alaska";
{
    "long";
    {
        "generic";
        "توقيت ألاسكا",
        "standard";
        "التوقيت الرسمي لألاسكا",
        "daylight";
        "توقيت ألاسكا الصيفي";
    }
}
"Amazon";
{
    "long";
    {
        "generic";
        "توقيت الأمازون",
        "standard";
        "توقيت الأمازون الرسمي",
        "daylight";
        "توقيت الأمازون الصيفي";
    }
}
"America_Central";

```

```

    {
      "long";
      {
        "generic";
        "التوقيت المركزي لأمريكا الشمالية",
        "standard";
        "التوقيت الرسمي المركزي لأمريكا الشمالية",
        "daylight";
        "التوقيت الصيفي المركزي لأمريكا الشمالية";
      }
    }
    "America_Eastern";
    {
      "long";
      {
        "generic";
        "التوقيت الشرقي لأمريكا الشمالية",
        "standard";
        "التوقيت الرسمي الشرقي لأمريكا الشمالية",
        "daylight";
        "التوقيت الصيفي الشرقي لأمريكا الشمالية";
      }
    }
    "America_Mountain";
    {
      "long";
      {
        "generic";
        "التوقيت الجبلي لأمريكا الشمالية",
        "standard";
        "التوقيت الجبلي الرسمي لأمريكا الشمالية",
        "daylight";
        "التوقيت الجبلي الصيفي لأمريكا الشمالية";
      }
    }
    "America_Pacific";
    {
      "long";
      {
        "generic";
        "توقيت المحيط الهادي",
        "standard";
        "توقيت المحيط الهادي الرسمي",
        "daylight";
        "توقيت المحيط الهادي الصيفي";
      }
    }
    "Anadyr";
    {
      "long";
      {
        "generic";
        "توقيت أنادير",
        "standard";
        "توقيت أنادير الرسمي",
        "daylight";
        "التوقيت الصيفي لأنادير";
      }
    }
  }

```



```
    }  
  }  
  "Apia";  
  {  
    "long";  
    {  
      "generic";  
      "توقيت آبيا",  
      "standard";  
      "التوقيت الرسمي لآبيا",  
      "daylight";  
      "التوقيت الصيفي لآبيا";  
    }  
  }  
  "Arabian";  
  {  
    "long";  
    {  
      "generic";  
      "التوقيت العربي",  
      "standard";  
      "التوقيت العربي الرسمي",  
      "daylight";  
      "التوقيت العربي الصيفي";  
    }  
  }  
  "Argentina";  
  {  
    "long";  
    {  
      "generic";  
      "توقيت الأرجنتين",  
      "standard";  
      "توقيت الأرجنتين الرسمي",  
      "daylight";  
      "توقيت الأرجنتين الصيفي";  
    }  
  }  
  "Argentina_Western";  
  {  
    "long";  
    {  
      "generic";  
      "توقيت غرب الأرجنتين",  
      "standard";  
      "توقيت غرب الأرجنتين الرسمي",  
      "daylight";  
      "توقيت غرب الأرجنتين الصيفي";  
    }  
  }  
  "Armenia";  
  {  
    "long";  
    {  
      "generic";  
      "توقيت أرمينيا",  
      "standard";
```

```

        "توقيت أرمينيا الرسمي",
        "daylight";
        "توقيت أرمينيا الصيفي";
    }
}
"Atlantic";
{
    "long";
    {
        "generic";
        "توقيت الأطلسي",
        "standard";
        "التوقيت الرسمي الأطلسي",
        "daylight";
        "التوقيت الصيفي الأطلسي";
    }
}
"Australia_Central";
{
    "long";
    {
        "generic";
        "توقيت وسط أستراليا",
        "standard";
        "توقيت وسط أستراليا الرسمي",
        "daylight";
        "توقيت وسط أستراليا الصيفي";
    }
}
"Australia_CentralWestern";
{
    "long";
    {
        "generic";
        "توقيت غرب وسط أستراليا",
        "standard";
        "توقيت غرب وسط أستراليا الرسمي",
        "daylight";
        "توقيت غرب وسط أستراليا الصيفي";
    }
}
"Australia_Eastern";
{
    "long";
    {
        "generic";
        "توقيت شرق أستراليا",
        "standard";
        "توقيت شرق أستراليا الرسمي",
        "daylight";
        "توقيت شرق أستراليا الصيفي";
    }
}
"Australia_Western";
{
    "long";
    {

```

```
        "generic";
        "توقيت غرب أستراليا",
        "standard";
        "توقيت غرب أستراليا الرسمي",
        "daylight";
        "توقيت غرب أستراليا الصيفي";
    }
}
"Azerbaijan";
{
    "long";
    {
        "generic";
        "توقيت أذربيجان",
        "standard";
        "توقيت أذربيجان الرسمي",
        "daylight";
        "توقيت أذربيجان الصيفي";
    }
}
"Azores";
{
    "long";
    {
        "generic";
        "توقيت أزورس",
        "standard";
        "توقيت أزورس الرسمي",
        "daylight";
        "توقيت أزورس الصيفي";
    }
}
"Bangladesh";
{
    "long";
    {
        "generic";
        "توقيت بنجلاديش",
        "standard";
        "توقيت بنجلاديش الرسمي",
        "daylight";
        "توقيت بنجلاديش الصيفي";
    }
}
"Bhutan";
{
    "long";
    {
        "standard";
        "توقيت بوتان";
    }
}
"Bolivia";
{
    "long";
    {
        "standard";
```

```
        "توقيت بوليفيا";
    }
}
"Brasilia";
{
    "long";
    {
        "generic";
        "توقيت برازيليا",
        "standard";
        "توقيت برازيليا الرسمي",
        "daylight";
        "توقيت برازيليا الصيفي";
    }
}
"Brunei";
{
    "long";
    {
        "standard";
        "توقيت بروناي";
    }
}
"Cape_Verde";
{
    "long";
    {
        "generic";
        "توقيت الرأس الأخضر",
        "standard";
        "توقيت الرأس الأخضر الرسمي",
        "daylight";
        "توقيت الرأس الأخضر الصيفي";
    }
}
"Chamorro";
{
    "long";
    {
        "standard";
        "توقيت تشامورو";
    }
}
"Chatham";
{
    "long";
    {
        "generic";
        "توقيت تشاثام",
        "standard";
        "توقيت تشاثام الرسمي",
        "daylight";
        "توقيت تشاثام الصيفي";
    }
}
"Chile";
{
```

```
        "long";
        {
            "generic";
            "توقيت شيلي",
            "standard";
            "توقيت شيلي الرسمي",
            "daylight";
            "توقيت شيلي الصيفي";
        }
    }
    "China";
    {
        "long";
        {
            "generic";
            "توقيت الصين",
            "standard";
            "توقيت الصين الرسمي",
            "daylight";
            "توقيت الصين الصيفي";
        }
    }
    "Choibalsan";
    {
        "long";
        {
            "generic";
            "توقيت شويبالسان",
            "standard";
            "توقيت شويبالسان الرسمي",
            "daylight";
            "التوقيت الصيفي لشويبالسان";
        }
    }
    "Christmas";
    {
        "long";
        {
            "standard";
            "توقيت جزر الكريسماس";
        }
    }
    "Cocos";
    {
        "long";
        {
            "standard";
            "توقيت جزر كوكوس";
        }
    }
    "Colombia";
    {
        "long";
        {
            "generic";
            "توقيت كولومبيا",
            "standard";
```

```

        "توقيت كولومبيا الرسمي",
        "daylight";
        "توقيت كولومبيا الصيفي";
    }
}
"Cook";
{
    "long";
    {
        "generic";
        "توقيت جزر كوك",
        "standard";
        "توقيت جزر كوك الرسمي",
        "daylight";
        "توقيت جزر كوك الصيفي";
    }
}
"Cuba";
{
    "long";
    {
        "generic";
        "توقيت كوبا",
        "standard";
        "توقيت كوبا الرسمي",
        "daylight";
        "توقيت كوبا الصيفي";
    }
}
"Davis";
{
    "long";
    {
        "standard";
        "توقيت دافيز";
    }
}
"DumontDUrville";
{
    "long";
    {
        "standard";
        "توقيت دي مونت دو روفيل";
    }
}
"East_Timor";
{
    "long";
    {
        "standard";
        "توقيت تيمور الشرقية";
    }
}
"Easter";
{
    "long";
    {

```

```

        "generic";
        "توقيت جزيرة استر",
        "standard";
        "توقيت جزيرة استر الرسمي",
        "daylight";
        "توقيت جزيرة استر الصيفي";
    }
}
"Ecuador";
{
    "long";
    {
        "standard";
        "توقيت الإكوادور";
    }
}
"Europe_Central";
{
    "long";
    {
        "generic";
        "توقيت وسط أوروبا",
        "standard";
        "توقيت وسط أوروبا الرسمي",
        "daylight";
        "توقيت وسط أوروبا الصيفي";
    }
}
"Europe_Eastern";
{
    "long";
    {
        "generic";
        "توقيت شرق أوروبا",
        "standard";
        "توقيت شرق أوروبا الرسمي",
        "daylight";
        "توقيت شرق أوروبا الصيفي";
    }
}
"Europe_Further_Eastern";
{
    "long";
    {
        "standard";
        "التوقيت الأوروبي (أكثر شرقًا)";
    }
}
"Europe_Western";
{
    "long";
    {
        "generic";
        "توقيت غرب أوروبا",
        "standard";
        "توقيت غرب أوروبا الرسمي",
        "daylight";
    }
}

```

```

        "توقيت غرب أوروبا الصيفي";
    }
}
"Falkland";
{
    "long";
    {
        "generic";
        "توقيت جزر فوكلاند",
        "standard";
        "توقيت جزر فوكلاند الرسمي",
        "daylight";
        "توقيت جزر فوكلاند الصيفي";
    }
}
"Fiji";
{
    "long";
    {
        "generic";
        "توقيت فيجي",
        "standard";
        "توقيت فيجي الرسمي",
        "daylight";
        "توقيت فيجي الصيفي";
    }
}
"French_Guiana";
{
    "long";
    {
        "standard";
        "توقيت غايانا الفرنسية";
    }
}
"French_Southern";
{
    "long";
    {
        "standard";
        "توقيت المقاطعات الفرنسية الجنوبية";
    }
}
"Galapagos";
{
    "long";
    {
        "standard";
        "توقيت غلاباغوس";
    }
}
"Gambier";
{
    "long";
    {
        "standard";
    }
}

```



```

        "توقيت جامبير";
    }
}
"Georgia";
{
    "long";
    {
        "generic";
        "توقيت جورجيا",
        "standard";
        "توقيت جورجيا الرسمي",
        "daylight";
        "توقيت جورجيا الصيفي";
    }
}
"Gilbert_Islands";
{
    "long";
    {
        "standard";
        "توقيت جزر جيلبرت";
    }
}
"GMT";
{
    "long";
    {
        "standard";
        "توقيت غرينتش";
    }
}
"Greenland_Eastern";
{
    "long";
    {
        "generic";
        "توقيت شرق غرينلاند",
        "standard";
        "توقيت شرق غرينلاند الرسمي",
        "daylight";
        "توقيت شرق غرينلاند الصيفي";
    }
}
"Greenland_Western";
{
    "long";
    {
        "generic";
        "توقيت غرب غرينلاند",
        "standard";
        "توقيت غرب غرينلاند الرسمي",
        "daylight";
        "توقيت غرب غرينلاند الصيفي";
    }
}
"Guam";
{

```

```
        "long";
        {
            "standard";
            "توقيت غوام";
        }
    }
    "Gulf";
    {
        "long";
        {
            "standard";
            "توقيت الخليج";
        }
    }
    "Guyana";
    {
        "long";
        {
            "standard";
            "توقيت غيانا";
        }
    }
    "Hawaii_Aleutian";
    {
        "long";
        {
            "generic";
            "توقيت هاواي ألوتيان",
            "standard";
            "توقيت هاواي ألوتيان الرسمي",
            "daylight";
            "توقيت هاواي ألوتيان الصيفي";
        }
    }
    "Hong_Kong";
    {
        "long";
        {
            "generic";
            "توقيت هونغ كونغ",
            "standard";
            "توقيت هونغ كونغ الرسمي",
            "daylight";
            "توقيت هونغ كونغ الصيفي";
        }
    }
    "Hovd";
    {
        "long";
        {
            "generic";
            "توقيت هوفد",
            "standard";
            "توقيت هوفد الرسمي",
            "daylight";
            "توقيت هوفد الصيفي";
        }
    }
}
```

```
}
"India";
{
  "long";
  {
    "standard";
    "توقيت الهند";
  }
}
"Indian_Ocean";
{
  "long";
  {
    "standard";
    "توقيت المحيط الهندي";
  }
}
"Indochina";
{
  "long";
  {
    "standard";
    "توقيت الهند الصينية";
  }
}
"Indonesia_Central";
{
  "long";
  {
    "standard";
    "توقيت وسط إندونيسيا";
  }
}
"Indonesia_Eastern";
{
  "long";
  {
    "standard";
    "توقيت شرق إندونيسيا";
  }
}
"Indonesia_Western";
{
  "long";
  {
    "standard";
    "توقيت غرب إندونيسيا";
  }
}
"Iran";
{
  "long";
  {
    "generic";
    "توقيت إيران",
    "standard";
    "توقيت إيران الرسمي",
  }
}
```

```
        "daylight";
        "توقيت إيران الصيفي";
    }
}
"Irkutsk";
{
    "long";
    {
        "generic";
        "توقيت إركوتسك",
        "standard";
        "توقيت إركوتسك الرسمي",
        "daylight";
        "توقيت إركوتسك الصيفي";
    }
}
"Israel";
{
    "long";
    {
        "generic";
        "توقيت إسرائيل",
        "standard";
        "توقيت إسرائيل الرسمي",
        "daylight";
        "توقيت إسرائيل الصيفي";
    }
}
"Japan";
{
    "long";
    {
        "generic";
        "توقيت اليابان",
        "standard";
        "توقيت اليابان الرسمي",
        "daylight";
        "توقيت اليابان الصيفي";
    }
}
"Kamchatka";
{
    "long";
    {
        "generic";
        "توقيت كامشاتكا",
        "standard";
        "توقيت بيتروبافلوفسك-كامتشاتسكي",
        "daylight";
        "توقيت بيتروبافلوفسك-كامتشاتسكي الصيفي";
    }
}
"Kazakhstan_Eastern";
{
    "long";
    {
        "standard";
```

```

        "توقيت شرق كازاخستان";
    }
}
"Kazakhstan_Western";
{
    "long";
    {
        "standard";
        "توقيت غرب كازاخستان";
    }
}
"Korea";
{
    "long";
    {
        "generic";
        "توقيت كوريا",
        "standard";
        "توقيت كوريا الرسمي",
        "daylight";
        "توقيت كوريا الصيفي";
    }
}
"Kosrae";
{
    "long";
    {
        "standard";
        "توقيت كوسرا";
    }
}
"Krasnoyarsk";
{
    "long";
    {
        "generic";
        "توقيت كراسنويارسك",
        "standard";
        "توقيت كراسنويارسك الرسمي",
        "daylight";
        "التوقيت الصيفي لكراسنويارسك";
    }
}
"Kyrgystan";
{
    "long";
    {
        "standard";
        "توقيت قرغيزستان";
    }
}
"Line_Islands";
{
    "long";
    {
        "standard";
        "توقيت جزر لاین";
    }
}

```

```
    }  
  }  
  "Lord_Howe";  
  {  
    "long";  
    {  
      "generic";  
      "توقيت لورد هاو",  
      "standard";  
      "توقيت لورد هاو الرسمي",  
      "daylight";  
      "التوقيت الصيفي للورد هاو";  
    }  
  }  
  "Macquarie";  
  {  
    "long";  
    {  
      "standard";  
      "توقيت ماكوارى";  
    }  
  }  
  "Magadan";  
  {  
    "long";  
    {  
      "generic";  
      "توقيت ماغادان",  
      "standard";  
      "توقيت ماغادان الرسمي",  
      "daylight";  
      "توقيت ماغادان الصيفي";  
    }  
  }  
  "Malaysia";  
  {  
    "long";  
    {  
      "standard";  
      "توقيت ماليزيا";  
    }  
  }  
  "Maldives";  
  {  
    "long";  
    {  
      "standard";  
      "توقيت المالديف";  
    }  
  }  
  "Marquesas";  
  {  
    "long";  
    {  
      "standard";  
      "توقيت ماركيساس";  
    }  
  }
```

```

    }
    "Marshall_Islands";
    {
        "long";
        {
            "standard";
            "توقيت جزر مارشال";
        }
    }
    "Mauritius";
    {
        "long";
        {
            "generic";
            "توقيت موريشيوس",
            "standard";
            "توقيت موريشيوس الرسمي",
            "daylight";
            "توقيت موريشيوس الصيفي";
        }
    }
    "Mawson";
    {
        "long";
        {
            "standard";
            "توقيت ماوسون";
        }
    }
    "Mexico_Northwest";
    {
        "long";
        {
            "generic";
            "توقيت شمال غرب المكسيك",
            "standard";
            "التوقيت الرسمي لشمال غرب المكسيك",
            "daylight";
            "التوقيت الصيفي لشمال غرب المكسيك";
        }
    }
    "Mexico_Pacific";
    {
        "long";
        {
            "generic";
            "توقيت المحيط الهادي للمكسيك",
            "standard";
            "توقيت المحيط الهادي الرسمي للمكسيك",
            "daylight";
            "توقيت المحيط الهادي الصيفي للمكسيك";
        }
    }
    "Mongolia";
    {
        "long";
        {

```

```

        "generic";
        "توقيت أولان باتور",
        "standard";
        "توقيت أولان باتور الرسمي",
        "daylight";
        "توقيت أولان باتور الصيفي";
    }
}
"Moscow";
{
    "long";
    {
        "generic";
        "توقيت موسكو",
        "standard";
        "توقيت موسكو الرسمي",
        "daylight";
        "توقيت موسكو الصيفي";
    }
}
"Myanmar";
{
    "long";
    {
        "standard";
        "توقيت ميانمار";
    }
}
"Nauru";
{
    "long";
    {
        "standard";
        "توقيت ناورو";
    }
}
"Nepal";
{
    "long";
    {
        "standard";
        "توقيت نيبال";
    }
}
"New_Caledonia";
{
    "long";
    {
        "generic";
        "توقيت كاليدونيا الجديدة",
        "standard";
        "توقيت كاليدونيا الجديدة الرسمي",
        "daylight";
        "توقيت كاليدونيا الجديدة الصيفي";
    }
}
"New_Zealand";

```



```

    {
      "long";
      {
        "generic";
        "توقيت نيوزيلندا",
        "standard";
        "توقيت نيوزيلندا الرسمي",
        "daylight";
        "توقيت نيوزيلندا الصيفي";
      }
    }
    "Newfoundland";
    {
      "long";
      {
        "generic";
        "توقيت نيوفاوندلاند",
        "standard";
        "توقيت نيوفاوندلاند الرسمي",
        "daylight";
        "توقيت نيوفاوندلاند الصيفي";
      }
    }
    "Niue";
    {
      "long";
      {
        "standard";
        "توقيت نيوي";
      }
    }
    "Norfolk";
    {
      "long";
      {
        "standard";
        "توقيت جزيرة نورفولك";
      }
    }
    "Noronha";
    {
      "long";
      {
        "generic";
        "توقيت فيرناندو دي نورونها",
        "standard";
        "توقيت فرناندو دي نورونها الرسمي",
        "daylight";
        "توقيت فرناندو دي نورونها الصيفي";
      }
    }
    "North_Mariana";
    {
      "long";
      {
        "standard";
        "توقيت جزر ماريانا الشمالية";
      }
    }
  }

```

```

    }
  }
  "Novosibirsk";
  {
    "long";
    {
      "generic";
      "توقيت نوفوسيبيرسك",
      "standard";
      "توقيت نوفوسيبيرسك الرسمي",
      "daylight";
      "توقيت نوفوسيبيرسك الصيفي";
    }
  }
  "Omsk";
  {
    "long";
    {
      "generic";
      "توقيت أومسك",
      "standard";
      "توقيت أومسك الرسمي",
      "daylight";
      "توقيت أومسك الصيفي";
    }
  }
  "Pakistan";
  {
    "long";
    {
      "generic";
      "توقيت باكستان",
      "standard";
      "توقيت باكستان الرسمي",
      "daylight";
      "توقيت باكستان الصيفي";
    }
  }
  "Palau";
  {
    "long";
    {
      "standard";
      "توقيت بالاو";
    }
  }
  "Papua_New_Guinea";
  {
    "long";
    {
      "standard";
      "توقيت بابوا غينيا الجديدة";
    }
  }
  "Paraguay";
  {
    "long";

```

```

        {
            "generic";
            "توقيت باراغواي",
            "standard";
            "توقيت باراغواي الرسمي",
            "daylight";
            "توقيت باراغواي الصيفي";
        }
    }
    "Peru";
    {
        "long";
        {
            "generic";
            "توقيت بيرو",
            "standard";
            "توقيت بيرو الرسمي",
            "daylight";
            "توقيت بيرو الصيفي";
        }
    }
    "Philippines";
    {
        "long";
        {
            "generic";
            "توقيت الفلبين",
            "standard";
            "توقيت الفلبين الرسمي",
            "daylight";
            "توقيت الفلبين الصيفي";
        }
    }
    "Phoenix_Islands";
    {
        "long";
        {
            "standard";
            "توقيت جزر فينكس";
        }
    }
    "Pierre_Miquelon";
    {
        "long";
        {
            "generic";
            "توقيت سانت بيير وميكلون",
            "standard";
            "توقيت سانت بيير وميكلون الرسمي",
            "daylight";
            "توقيت سانت بيير وميكلون الصيفي";
        }
    }
    "Pitcairn";
    {
        "long";
        {

```

```

        "standard";
        "توقيت بيتكيرن";
    }
}
"Ponape";
{
    "long";
    {
        "standard";
        "توقيت بونا بي";
    }
}
"Pyongyang";
{
    "long";
    {
        "standard";
        "توقيت بيونغ يانغ";
    }
}
"Reunion";
{
    "long";
    {
        "standard";
        "توقيت ريونيون";
    }
}
"Rothera";
{
    "long";
    {
        "standard";
        "توقيت روثيرا";
    }
}
"Sakhalin";
{
    "long";
    {
        "generic";
        "توقيت ساخالين",
        "standard";
        "توقيت ساخالين الرسمي",
        "daylight";
        "توقيت ساخالين الصيفي";
    }
}
"Samara";
{
    "long";
    {
        "generic";
        "توقيت سامارا",
        "standard";
        "توقيت سامارا",
        "daylight";
    }
}

```

```
        "توقيت سمارا الصيفي";
    }
}
"Samoa";
{
    "long";
    {
        "generic";
        "توقيت ساموا",
        "standard";
        "توقيت ساموا الرسمي",
        "daylight";
        "توقيت ساموا الصيفي";
    }
}
"Seychelles";
{
    "long";
    {
        "standard";
        "توقيت سيشل";
    }
}
"Singapore";
{
    "long";
    {
        "standard";
        "توقيت سنغافورة";
    }
}
"Solomon";
{
    "long";
    {
        "standard";
        "توقيت جزر سليمان";
    }
}
"South_Georgia";
{
    "long";
    {
        "standard";
        "توقيت جنوب جورجيا";
    }
}
"Suriname";
{
    "long";
    {
        "standard";
        "توقيت سورينام";
    }
}
"Syowa";
{
```

```
        "long";
        {
            "standard";
            "توقيت سايووا";
        }
    }
    "Tahiti";
    {
        "long";
        {
            "standard";
            "توقيت تاهيتي";
        }
    }
    "Taipei";
    {
        "long";
        {
            "generic";
            "توقيت تايبه",
            "standard";
            "توقيت تايبه الرسمي",
            "daylight";
            "توقيت تايبه الصيفي";
        }
    }
    "Tajikistan";
    {
        "long";
        {
            "standard";
            "توقيت طاجكستان";
        }
    }
    "Tokelau";
    {
        "long";
        {
            "standard";
            "توقيت توكيلاو";
        }
    }
    "Tonga";
    {
        "long";
        {
            "generic";
            "توقيت تونغا",
            "standard";
            "توقيت تونغا الرسمي",
            "daylight";
            "توقيت تونغا الصيفي";
        }
    }
    "Truk";
    {
        "long";
```

```

        {
            "standard";
            "توقيت شوك";
        }
    }
    "Turkmenistan";
    {
        "long";
        {
            "generic";
            "توقيت تركمانستان",
            "standard";
            "توقيت تركمانستان الرسمي",
            "daylight";
            "توقيت تركمانستان الصيفي";
        }
    }
    "Tuvalu";
    {
        "long";
        {
            "standard";
            "توقيت توفالو";
        }
    }
    "Uruguay";
    {
        "long";
        {
            "generic";
            "توقيت أورغواي",
            "standard";
            "توقيت أورغواي الرسمي",
            "daylight";
            "توقيت أورغواي الصيفي";
        }
    }
    "Uzbekistan";
    {
        "long";
        {
            "generic";
            "توقيت أوزبكستان",
            "standard";
            "توقيت أوزبكستان الرسمي",
            "daylight";
            "توقيت أوزبكستان الصيفي";
        }
    }
    "Vanuatu";
    {
        "long";
        {
            "generic";
            "توقيت فانواتو",
            "standard";
            "توقيت فانواتو الرسمي",

```

```

        "daylight";
        "توقيت فانواتو الصيفي";
    }
}
"Venezuela";
{
    "long";
    {
        "standard";
        "توقيت فنزويلا";
    }
}
"Vladivostok";
{
    "long";
    {
        "generic";
        "توقيت فلاديفوستوك",
        "standard";
        "توقيت فلاديفوستوك الرسمي",
        "daylight";
        "توقيت فلاديفوستوك الصيفي";
    }
}
"Volgograd";
{
    "long";
    {
        "generic";
        "توقيت فولغوغراد",
        "standard";
        "توقيت فولغوغراد الرسمي",
        "daylight";
        "توقيت فولغوغراد الصيفي";
    }
}
"Vostok";
{
    "long";
    {
        "standard";
        "توقيت فوستوك";
    }
}
"Wake";
{
    "long";
    {
        "standard";
        "توقيت جزيرة ويك";
    }
}
"Wallis";
{
    "long";
    {
        "standard";

```



```

        "توقيت واليس و فوتونا";
    }
}
"Yakutsk";
{
    "long";
    {
        "generic";
        "توقيت ياكوتسك",
        "standard";
        "توقيت ياكوتسك الرسمي",
        "daylight";
        "توقيت ياكوتسك الصيفي";
    }
}
"Yekaterinburg";
{
    "long";
    {
        "generic";
        "توقيت يكاترينبورغ",
        "standard";
        "توقيت يكاترينبورغ الرسمي",
        "daylight";
        "توقيت يكاترينبورغ الصيفي";
    }
}
}
}
}
}
}
}
}

```

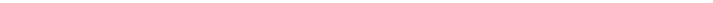
## Accessibility in React Timepicker component

The TimePicker component followed the accessibility guidelines and standards, including [ADA](#), [Section 508](#), [WCAG 2.2](#) standards, and [WCAG roles](#) that are commonly used to evaluate accessibility.

The accessibility compliance for the TimePicker component is outlined below.

| Accessibility Criteria | Compatibility |

| - - | - - |

| [WCAG 2.2 Support](#) |  alt="Yes" |

| [Section 508 Support](#) | 

| [Screen Reader Support](#) |  |

| [Right-To-Left Support](#) |  |

| [Color Contrast](#) |  |

| [Mobile Device Support](#) |  |

| [Keyboard Navigation Support](#) |  |

| [Accessibility Checker Validation](#) |  |

| [Axe-core Accessibility Validation](#) |  |

<style>

.post .post-content img {

display: inline-block;

margin: 0.5em 0;

}

</style>

<div> - All features of the component meet the requirement.</div>

<div> - Some features of the component do not meet the requirement.</div>

<div> - The component does not meet the requirement.</div>

### WAI-ARIA attributes

The web accessibility makes web applications and its content more accessible to people with disabled without any barriers. Especially it tracks the dynamic value changes and DOM changes.

TimePicker component has covered the [WAI-ARIA](#) specifications with the following list of WAI-ARIA attributes `aria-haspopup`, `aria-selected`, `aria-disabled`, `aria-activedescendant`,

`aria-expanded`, `aria-owns`, and `aria-autocomplete`.

Here, the `combobox` as a role for input element and `listbox` as a role for popup element in the TimePicker.

- **Aria-haspopup** : Provides the information about whether this element display a pop-up window or not.
- **Aria-selected** : Indicates the current selected value of the TimePicker component.
- **Aria-disabled** : Indicates disabled state of the TimePicker component.
- **Aria-expanded** : Indicates expanded state of the popup.
- **Aria-autocomplete** : Indicates whether user input completion suggestions are provided or not.
- **Aria-owns** : Attribute that creates a parent/child relationship between the two DOM element in accessibility layer

- **Aria-activedescendent** : Attribute that helps in managing the current active child of the TimePicker component.
- **Role** : Attributes that gives assistive technologies information for handling each element in a widget.

### Keyboard Interaction

Keyboard accessibility is one of the most important aspects of web accessibility. It will be more useful to all the computer users, as they often allow to interact keyboard more than a mouse.

Among people with disabilities like blind or who have motor disabilities are also can make frequent use of keyboard shortcuts.

The TimePicker component has built-in keyboard accessibility support by following the [WAI-ARIA practices](#).

It supports the following list of shortcut keys to interact with the TimePicker component.

| Press | To do this |

| --- | --- |

| Upper Arrow | Navigate and select the previous item. |

| Down Arrow | Navigate and select the next item. |

| Left Arrow | Move the cursor towards arrow key pressed direction. |

| Right Arrow | Move the cursor towards arrow key pressed direction. |

| Home | Navigate and select the first item. |

| End | Navigate and select the last item. |

| Enter | Select the currently focused item and close the popup. |

| Alt + Upper Arrow | Close the popup. |

| Alt + Down Arrow | Open the popup. |

| Esc | Close the popup. |

The following sample use the `alt+t` keys to focus the TimePicker component.

### INDEX.JSX

```
{% raw %}
// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
export default class App extends React.Component {
  timepickerInstance;
  componentDidMount() {
    const proxy = this;
    document.onkeyup = (e) => {
```

```

        if (e.altKey && e.keyCode === 84 /* t */) {
            // press alt+t to focus the control.
            proxy.timepickerInstance.focus();
        }
    };
}
render() {
    return <TimePickerComponent id="timepicker" placeholder="Select a
Time" ref={ (scope) => { this.timepickerInstance = scope; } } />;
}
;
ReactDOM.render(<App />, document.getElementById('timer'));
{% endraw %}

```

### INDEX.TSX

```

{% raw %}
// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePicker, TimePickerComponent } from '@syncfusion/ej2-react-
calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
export default class App extends React.Component<{}, {}> {
    private timepickerInstance: TimePicker;
    public componentDidMount() {
        const proxy = this;
        document.onkeyup = (e) => {
            if (e.altKey && e.keyCode === 84 /* t */) {
                // press alt+t to focus the control.
                proxy.timepickerInstance.focus();
            }
        };
    }
    public render() {
        return <TimePickerComponent id="timepicker" placeholder="Select a
Time" ref={ (scope) => { (this.timepickerInstance as TimePicker | null) =
scope; } } />
    }
};
ReactDOM.render(<App />, document.getElementById('timer'));
{% endraw %}

```

### INDEX.JSX

```

{% raw %}
// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";

```

```
// enable ripple effect
enableRipple(true);
function App() {
  let timepickerInstance;
  React.useEffect(() => {
    const proxy = this;
    document.onkeyup = (e) => {
      if (e.altKey && e.keyCode === 84 /* t */) {
        // press alt+t to focus the control.
        proxy.timepickerInstance.focus();
      }
    };
  }, []);
  return <TimePickerComponent id="timepicker" placeholder="Select a Time"
  ref={(scope) => { timepickerInstance = scope; }}/>;
}
;
ReactDOM.render(<App />, document.getElementById('timer'));
{% endraw %}
```

## INDEX.TSX

```
{% raw %}
// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePicker, TimePickerComponent } from '@syncfusion/ej2-react-
calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
function App() {
  let timepickerInstance: TimePicker;
  React.useEffect(() => {
    const proxy = this;
    document.onkeyup = (e) => {
      if (e.altKey && e.keyCode === 84 /* t */) {
        // press alt+t to focus the control.
        proxy.timepickerInstance.focus();
      }
    };
  }, []);
  return <TimePickerComponent id="timepicker" placeholder="Select a Time"
  ref={(scope) => { (timepickerInstance as TimePicker | null) = scope; }}/>;
};
ReactDOM.render(<App />, document.getElementById('timer'));
{% endraw %}
```

## Ensuring accessibility

The TimePicker component's accessibility levels are ensured through an [accessibility-checker](#) and [axe-core](#) software tools during automated testing.

The accessibility compliance of the TimePicker component is shown in the following sample. Open the [sample](#) in a new window to evaluate the accessibility of the TimePicker component with accessibility tools.

See also

- [Accessibility in Syncfusion React components](#)

### Style appearance in React Timepicker component

The following content provides the exact CSS structure that can be used to modify the control's appearance based on the user preference.

#### Customizing the appearance of TimePicker wrapper element

Use the following CSS to customize the appearance of wrapper element.

```
`css
/ To specify height and font size /
.e-input-group input.e-input, .e-input-group.e-control-wrapper input.e-input, .e-input-group textarea.e-input, .e-input-group.e-control-wrapper textarea.e-input {
font-size: 20px;
height: 40px;
}
```

#### Customizing the TimePicker icon element

Use the following CSS to customize the TimePicker icon element

```
`css
/ To specify background color and font size /
.e-time-wrapper .e-time-icon.e-icons, *.e-control-wrapper.e-time-wrapper .e-time-icon.e-icons {
font-size: 20px;
background-color: beige;
}
```

#### Customizing the TimePicker popup

Use the following CSS to customize the TimePicker popup

```
`css
/ To specify height /
.e-timepicker.e-popup {
height: 100px;
}
```

`

### Customizing the TimePicker popup content

Use the following CSS to customize the TimePicker popup content

```
`css
/ To specify height /
.e-timepicker.e-popup .e-list-parent.e-ul li.e-list-item {
background-color: beige;
font-size: 20px;
}
```

`

### Full screen mode support in mobiles and tablets

The TimePicker component's full-screen mode feature enables users to view the component popup element in full-screen mode on mobile devices with improved visibility and a better user experience. It is important to mention that this feature is exclusively available for mobile and tablet devices in both landscape and portrait orientations. To activate the full screen mode within the TimePicker component, simply set the [fullScreenMode](#) API value to `true`. This action will extend the popup element to occupy the entire screen on mobile devices.

```
`typescript
// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';

// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";

// enable ripple effect
enableRipple(true);

function App(){
const mobileMode:boolean = true;
return <TimePickerComponent id="timepicker" fullScreenMode={this.mobileMode} />
};

ReactDOM.render(<App />, document.getElementById('timer'));
```

`

**Default Sample**

12:00 AM







## How To

### Css customization in React Timepicker component

TimePicker allows you to customize the textbox and popup list appearance to suit for your application by using [cssClass](#) property.

The below sample demonstrates customization of the text appearance in a textbox, popup button, and popup list along with hover and active state by using `e-custom-style` class. Following is the list of available classes used to customize the entire TimePicker component.

Class Name	Description
---	---
e-time-wrapper	Applied to TimePicker wrapper element.
e-timepicker	Applied to input element and TimePicker popup element.
e-time-wrapper.e-timepicker	Applied to input element only.
e-input-group-icon.e-time-icon	Applied to popup button.
e-float-text	Applied to floating label text element.
e-popup	Applied to popup element.
e-timepicker.e-popup	Applied to TimePicker popup element only.
e-list-parent	Applied to popup UL element.
e-timepicker.e-list-parent	Applied to TimePicker popup UL element only.
e-list-item	Applied to LI elements.
e-hover	Applied to LI element hovering time.
e-active	Applied to active LI element.

### INDEX.JSX

```
// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
export default class App extends React.Component {
  render() {
    return <TimePickerComponent id="timepicker" cssClass="e-custom-style"
placeholder="Select a Time"/>;
  }
}
;
ReactDOM.render(<App />, document.getElementById('timer'));
```

### INDEX.TSX

```
// import the ripple effect
```

```
import { enableRipple } from '@syncfusion/ej2-base';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
export default class App extends React.Component<{}, {}> {
    public render() {
        return <TimePickerComponent id="timepicker" cssClass="e-custom-style"
placeholder="Select a Time" />
    }
};
ReactDOM.render(<App />, document.getElementById('timer'));
```

### Client side validation using form validator in React Timepicker component

To achieve the client side validation in a TimePicker component by using [Essential JavaScript 2 FormValidator](#). It provides an option to customize the feedback error messages to the corresponding fields to take action to resolve the issue.

In the following example, the required field validation is implemented by mapping the name attribute value to the rules property. It validates the TimePicker component and displays the validation message when the textbox value is empty, during form post back or focus out.

### INDEX.JSX

```
// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the form validator related functions
import { FormValidator } from '@syncfusion/ej2-inputs';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
export default class App extends React.Component {
    componentDidMount() {
        const options = {
            customPlacement: (inputElement, errorElement) => {
                const parentElement = inputElement.parentElement;
                parentElement.parentElement.appendChild(errorElement);
            },
            rules: {
                // must specify the name attribute value in rules section
                // with required validation
                'timepicker': { required: true }
            }
        };
        const formObject = new FormValidator('#form-element', options);
        formObject.addRules('timepicker', { required: true });
    }
    render() {
        return (<form id='form-element'>
            <TimePickerComponent id="timepicker" placeholder="Select a
Time"/>
        </form>);
    }
}
```

```

        </form>;
    }
}
;
ReactDOM.render(<App />, document.getElementById('timer'));

```

## INDEX.TSX

```

// import the ripple effect
import { enableRipple } from '@syncfusion/ej2-base';
// import the form validator related functions
import { FormValidator, FormValidatorModel } from '@syncfusion/ej2-inputs';
// import the timepicker
import { TimePickerComponent } from '@syncfusion/ej2-react-calendars';
import * as React from "react";
import * as ReactDOM from "react-dom";
// enable ripple effect
enableRipple(true);
export default class App extends React.Component<{}, {}> {
    public componentDidMount(): void {
        const options: FormValidatorModel = {
            customPlacement: (inputElement: HTMLElement, errorElement:
HTMLElement) => {
                const parentElement: HTMLElement = inputElement.parentElement
as HTMLElement;
                (parentElement.parentElement as
HTMLElement).appendChild(errorElement);
            },
            rules: {
                // must specify the name attribute value in rules section
                // with required validation
                'timepicker': { required: true }
            }
        };
        const formObject: FormValidator = new FormValidator('#form-element',
options);
        formObject.addRules('timepicker', {required: true});
    }
    public render() {
        return (
            <form id='form-element'>
                <TimePickerComponent id="timepicker" placeholder="Select a
Time" />
            </form>)
        );
    }
};
ReactDOM.render(<App />, document.getElementById('timer'));

```

## Ej1 api migration in React Timepicker component

This article describes the API migration process of TimePicker component from Essential JS 1 to Essential JS 2.

### Time Selection

{% raw %}

&lt;!-- markdownlint-disable MD033 --&gt;

Behavior	API in Essential JS 1	API in Essential JS 2
Setting the value	Property: value	Property: Value

### Time Format

&lt;!-- markdownlint-disable MD033 --&gt;

Behavior	API in Essential JS 1	API in Essential JS 2
Display time format	Property: timeFormat	Property: format

### Time Range

&lt;!-- markdownlint-disable MD033 --&gt;

Behavior	API in Essential JS 1	API in Essential JS 2
Minimum time	Property: minTime	Property: min
Maximum time	Property: maxTime	Property: Max
Set current time	Method: setCurrentTime() function onCreate() { var timeObj = \$("#timepicker").data("ejTimePicker"); timeObj.setCurrentTime();}	Can be achieved by private timeValue: Date = new Date();

### Disabled Time Ranges

&lt;!-- markdownlint-disable MD033 --&gt;

Behavior	API in Essential JS 1	API in Essential JS 2
Disable time ranges	Property: disableTimeRanges var disableTime = [{ startTime: '3:00 AM', endTime: '6:00 AM' }, { startTime: '1:00 PM', endTime: '3:00 PM' }, { startTime: '8:00 PM', endTime: '10:00 PM' }];	Can be achieved by itemRenderHandler(args) { if (args.value.getHours() === 4) { args.isDisabled = true; }}

### Customization

&lt;!-- markdownlint-disable MD033 --&gt;

Behavior	API in Essential JS 1	API in Essential JS 2
CSS Class	Property: cssClass	Property: cssClass
Popup list customization	Not Applicable	Event: itemRenderitemRanderHandler() {/ code block */ }
Show/Hide the popup button	Property: showPopupButton	Can be achieved byonFocus(args) { this.show();}.e-control- wrapper .e-input-group-icon.e- time-icon { display: none;}
Enable/Disable the rounded corner	Property: showRoundedCorner	Can be achieved by.e-control- wrapper.e-custom-style.e-time- wrapper.e-input-group { border-radius: 4px;}
Enable/Disable the animation	Property: enableAnimation	Not Applicable
Interval	Property: interval	Property: step
FocusIn event	Event: focusInfunction onFocus() { / code block */ }	Event: focusonFocus() {/ code block */ }
FocusOut event	Event: focusOutfunction onFocusout() { }	Event: bluronBlur(args) { / code block */ }
FocusIn method	Not Applicable	Method: focusIn()create(args) { this.focusIn();}
FocusOut method	Not Applicable	Method: focusOutcreate(args) { this.focusOut();}
Prevent popup close	Not Applicable	Event: closeonClose(args) { args.cancel = true;}
Prevent popup open	Not Applicable	Event: openonOpen(args) { args.cancel = true;}

### Accessibility

<!-- markdownlint-disable MD033 -->

Behavior	API in Essential JS 1	API in Essential JS 2
Enable/Disable the RTL	Property: enableRTL	Property: enableRtl

### Persistence

<!-- markdownlint-disable MD033 -->

Behavior	API in Essential JS 1	API in Essential JS 2
Enable/Disable the persistence	Property: enablePersistence	Property: enablePersistence

### Validation

<!-- markdownlint-disable MD033 -->

Behavior	API in Essential JS 1	API in Essential JS 2
Validation rules	Property: validationRules var validationRules = {required: {true}};\$\$.validator.setDefault({ errorClass: 'e-validation-error', errorPlacement: function (error, element) { \$(error).insertAfter(element.closest(".e-widget")); } });	Can be achieved by var options = { rules: { 'timepicker': { required: true } } };var formObject = new ej.inputs.FormValidator('#form-element', options);
Validation message	Property: validationMessages var validationRules = {required: {true}};var validationMessage = {required: "Required value"};\$\$.validator.setDefault({ errorClass: 'e-validation-error', errorPlacement: function (error, element) { \$(error).insertAfter(element.closest(".e-widget")); } });	A Can be achieved by var options = { rules: { 'timepicker': { required: true } }, customPlacement: (inputElement, errorElement) => { inputElement.parentElement.parentElement.appendChild(errorElement); } };var formObject = new ej.inputs.FormValidator('#form-element', options);

### Common

<!-- markdownlint-disable MD033 -->

Behavior	API in Essential JS 1	API in Essential JS 2
Width	Property: width	Property: width
Readonly	Property: readOnly	Property: Readonly
Show/Hide the clear button	Not Applicable	Property: showClearButton
Height	Property: Height	Can be achieved by.e-control-wrapper.e-custom-style.e-time-wrapper.e-input-group { height: 35px;}
Html Attributes	Property: HtmlAttributes var htmlAttributes = {required:"required"}	Not Applicable
Watermark text	Property: watermarkText	Property: placeholder
Disable/Enable	Property: enabled	Property: enabled
Disable the TimePicker	Method: disable() function onCreate(args) { var timeObject = \$("#time").data("ejTimePicker"); timeObject.disable();}	Property: enabled
Enable the TimePicker	Method: enable() function onCreate(args) { var timeObject = \$("#time").data("ejTimePicker"); timeObject.enable();}	Not Applicable
Enable/Disable the textBox editing	Not Applicable	Property: allowEdit
zIndex	Not Applicable	Property: zIndex
Specify the placeholder text behavior	Not Applicable	Property: floatLabelType

### Globalization

<!-- markdownlint-disable MD033 -->

Behavior	API in Essential JS 1	API in Essential JS 2
Locale	Property: locale	Property: locale

### Strict Mode

<!-- markdownlint-disable MD033 -->

Behavior	API in Essential JS 1	API in Essential JS 2
Strict mode	Property: enableStrictMode	Property: strictMode

### Open and Close

<!-- markdownlint-disable MD033 -->

Behavior	API in Essential JS 1	API in Essential JS 2
Close	Event: closefunction onClose() {/ code block */ }	Event: closeonClose() {/ code block */ }
Open	Event: openfunction onOpen(args) {/ code block */ }	Event: openonOpen(args) {/ code block */ }
Hide	Method: hide()function onCreate(args) { var timeObject = \$("#time").data("ejTimePicker"); timeObject.show(); timeObject.hide();}	Method: hide()create() { this.show(); this.hide();}
Show	Method: show()function onCreate(args) { var timeObject = \$("#time").data("ejTimePicker"); timeObject.show();}	Method: show()onCreate() { this.show();}

{% endraw %}

## Toast

### Getting Started

This section briefly explains you the steps required to create a simple Toast and demonstrate the basic usage of the Toast component.

### Dependencies

The following list of dependencies are required to use the Toast component in your application.

```
`javascript
```

```
|-- @syncfusion/ej2-react-notifications
```

```
|-- @syncfusion/ej2-react-base
```

```
|-- @syncfusion/ej2-notifications
```

```
|-- @syncfusion/ej2-base
```

```
|-- @syncfusion/ej2-buttons
```

```
|-- @syncfusion/ej2-react-buttons
```



```
|-- @syncfusion/ej2-popups  
,
```

### Installation and configuration

You can use [Create-react-app](#) to setup the applications. To install `create-react-app` run the following command.

```
`bash  
npm install -g create-react-app  
,
```

Start a new project using create-react-app command as follows

```
<div class='tsx'>  
,
```

```
create-react-app quickstart --scripts-version=react-scripts-ts  
cd quickstart  
,
```

```
</div>  
<div class='jsx'>  
,
```

```
create-react-app quickstart  
cd quickstart  
,
```

```
</div>
```

Install the below required dependency package in order to use the `Toast` component in your application.

```
`bash  
npm install @syncfusion/ej2-react-notifications --save  
,
```

The above package installs [Toast dependencies](#) which are required to render the Toast component in React environment.

- Toast CSS files are available in the `ej2-react-notifications` package folder.

Import the Toast component's required CSS references as follows in `src/App.css`.

```
`css  
@import '../node_modules/@syncfusion/ej2-base/styles/material.css';  
@import '../node_modules/@syncfusion/ej2-react-buttons/styles/material.css';
```

```
@import '../node_modules/@syncfusion/ej2-popups/styles/material.css';
@import '../node_modules/@syncfusion/ej2-react-notifications/styles/material.css';
`
```

### Initialize the Toast with message

The Toast message can be rendered by defining an `title` or `content`.

- Import the Toast component to your `src/App.tsx` file using following code.

#### [Class-component]

```
`ts
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
class App extends React.Component<{}, {}> {
  public toastInstance:ToastComponent;
  public toastCreated(): void {
    this.toastInstance.show();
  }
  public render() {
    return (
      <ToastComponent ref={toast => this.toastInstance = toast!} title="Sample Toast Title" content="Sample
      Toast Content" created={this.toastCreated = this.toastCreated.bind(this)} />
    );
  }
};
ReactDOM.render(<App />, document.getElementById('element'));
```

#### [Functional-component]

```
`ts
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
function App() {
  let toastInstance:ToastComponent;
  function toastCreated(): void {
    toastInstance.show();
  }
}
```

```

return (
  <ToastComponent ref={toast => toastInstance = toast!} title="Sample Toast Title" content="Sample
  Toast Content" created={toastCreated.bind(this)} />
);
};

ReactDOM.render(<App />, document.getElementById('element'));
`

```

- Run the application in the browser using the following command.

```

npm start
`

```

Output will be as follows:

[Class-component]

#### APP.JSX

```

import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
class App extends React.Component {
  toastInstance;
  toastCreated() {
    this.toastInstance.show();
  }
  render() {
    return (<ToastComponent ref={toast => this.toastInstance = toast}
    title="Matt sent you a friend request" content="Hey, wanna dress up as
    wizards and ride our hoverboards?" created={this.toastCreated =
    this.toastCreated.bind(this)} />);
  }
}
;
export default App;

```

#### APP.TSX

```

import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import * as ReactDOM from 'react-dom';
class App extends React.Component<{}, {}> {
  public toastInstance: ToastComponent;
  public toastCreated(): void {
    this.toastInstance.show();
  }
  public render() {
    return (
      <ToastComponent ref={toast => this.toastInstance = toast!} title="Matt
      sent you a friend request" content="Hey, wanna dress up as wizards and ride

```

```

    our hoverboards?" created={this.toastCreated = this.toastCreated.bind(this)}
  />
    );
  }
};
export default App;

```

## [Functional-component]

### APP.JSX

```

import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
function App() {
  let toastInstance;
  function toastCreated() {
    toastInstance.show();
  }
  return (<ToastComponent ref={toast => toastInstance = toast} title="Matt
sent you a friend request" content="Hey, wanna dress up as wizards and ride
our hoverboards?" created={toastCreated.bind(this)} />);
}
;
export default App;

```

### APP.TSX

```

import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import * as ReactDOM from 'react-dom';
function App() {
  let toastInstance:ToastComponent;
  function toastCreated(): void {
    toastInstance.show();
  }
  return (
    <ToastComponent ref={toast => toastInstance = toast!} title="Matt sent
you a friend request" content="Hey, wanna dress up as wizards and ride our
hoverboards?" created={toastCreated.bind(this)} />
  );
};
export default App;

```

## Initialize the Toast with target

By default toast can be rendered in document body, we can change the target position for toast rendering using `target` property.

In the above sample code, `#element` is the `id` of the HTML element in a page to which the Toast is initialized.

## [Class-component]

### APP.JSX

```

import { ToastComponent } from '@syncfusion/ej2-react-notifications';

```

```
import * as React from "react";
class App extends React.Component {
  toastInstance;
  toastCreated() {
    this.toastInstance.show();
  }
  render() {
    return (
      <div>
        <div id='#toast_target' />
        <ToastComponent id='toast_target' ref={toast => this.toastInstance =
toast} title="Sample Toast Title" content="Sample Toast Content"
created={this.toastCreated = this.toastCreated.bind(this)} />
      </div>
    );
  }
}
;
export default App;
```

**APP.TSX**

```
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
class App extends React.Component<{}, {}> {
  public toastInstance: ToastComponent;
  public toastCreated(): void {
    this.toastInstance.show();
  }
  public render() {
    return (
      <div>
        <div id='#toast_target' />
        <ToastComponent id='toast_target' ref={toast => this.toastInstance =
toast!} title="Sample Toast Title" content="Sample Toast Content"
created={this.toastCreated = this.toastCreated.bind(this)} />
      </div>
    );
  }
};
export default App;
```

**[Functional-component]****APP.JSX**

```
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
function App() {
  let toastInstance;
  function toastCreated() {
    toastInstance.show();
  }
  return (
    <div>
      <div id='#toast_target' />
      <ToastComponent id='toast_target' ref={toast => toastInstance = toast}
title="Sample Toast Title" content="Sample Toast Content"
created={toastCreated.bind(this)} />
    </div>
  );
}
```

```
    </div>);  
  }  
;  
export default App;
```

### APP.TSX

```
import { ToastComponent } from '@syncfusion/ej2-react-notifications';  
import * as React from "react";  
function App() {  
  let toastInstance: ToastComponent;  
  function toastCreated(): void {  
    toastInstance.show();  
  }  
  return (  
    <div>  
      <div id='#toast_target' />  
      <ToastComponent id='toast_target' ref={toast => toastInstance = toast!}  
title="Sample Toast Title" content="Sample Toast Content"  
created={toastCreated.bind(this)} />  
    </div>  
  );  
};  
export default App;
```

### See Also

- [Render different types of toast](#)

**Note:** You can refer to our [React Toast](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Toast Example](#) that shows you how to render the Toast in React.

### Getting Started

This section briefly explains you the steps required to create a simple Toast and demonstrate the basic usage of the Toast component.

### Dependencies

The following list of dependencies are required to use the Toast component in your application.

```
`javascript  
|-- @syncfusion/ej2-react-notifications  
|-- @syncfusion/ej2-react-base  
|-- @syncfusion/ej2-notifications  
|-- @syncfusion/ej2-base  
|-- @syncfusion/ej2-buttons  
|-- @syncfusion/ej2-react-buttons  
|-- @syncfusion/ej2-popups
```

,

### Installation and configuration

You can use [Create-react-app](#) to setup the applications. To install `create-react-app` run the following command.

```
`bash
```

```
npm install -g create-react-app
```

,

Start a new project using create-react-app command as follows

```
<div class='tsx'>
```

,

```
create-react-app quickstart --scripts-version=react-scripts-ts
```

```
cd quickstart
```

,

```
</div>
```

```
<div class='jsx'>
```

,

```
create-react-app quickstart
```

```
cd quickstart
```

,

```
</div>
```

Install the below required dependency package in order to use the `Toast` component in your application.

```
`bash
```

```
npm install @syncfusion/ej2-react-notifications --save
```

,

The above package installs [Toast dependencies](#) which are required to render the Toast component in React environment.

- Toast CSS files are available in the `ej2-react-notifications` package folder.

Import the Toast component's required CSS references as follows in `src/App.css`.

```
`css
```

```
@import '../node_modules/@syncfusion/ej2-base/styles/material.css';
```

```
@import '../node_modules/@syncfusion/ej2-react-buttons/styles/material.css';
```

```
@import '../node_modules/@syncfusion/ej2-react-popups/styles/material.css';
```

```
@import '../node_modules/@syncfusion/ej2-react-notifications/styles/material.css';  
`
```

### Initialize the Toast with message

The Toast message can be rendered by defining an `title` or `content`.

- Import the Toast component to your `src/App.tsx` file using following code.

```
`ts  
import { ToastComponent } from '@syncfusion/ej2-react-notifications';  
import * as React from "react";  
import { useRef } from 'react';  
function App() {  
  const toastInstance = useRef<ToastComponent>(null);  
  function toastCreated() {  
    toastInstance.current.show();  
  }  
  return (  
    <ToastComponent  
      ref={toastInstance}  
      title="Sample Toast Title"  
      content="Sample Toast Content"  
      created={toastCreated}  
    />  
  );  
};  
export default App;  
ReactDOM.render(<App />, document.getElementById('element'));  
`
```

- Run the application in the browser using the following command.

```
`  
npm start  
`
```

Output will be as follows:

**APP.JSX**



```
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import { useRef } from 'react';
function App() {
    const toastInstance = useRef(null);
    function toastCreated() {
        toastInstance.current.show();
    }
    return (<div>
        <ToastComponent ref={toastInstance} title="Matt sent you a friend
request" content="Hey, wanna dress up as wizards and ride our hoverboards?"
created={toastCreated}/>
    </div>);
}
export default App;
```

### APP.TSX

```
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import * as ReactDOM from 'react-dom';
import { useRef } from 'react';
function App() {
    const toastInstance = useRef<ToastComponent>(null);
    function toastCreated() {
        toastInstance.current.show();
    }
    return (
        <div>
            <ToastComponent
                ref={toastInstance}
                title="Matt sent you a friend request"
                content="Hey, wanna dress up as wizards and ride our hoverboards?"
                created={toastCreated}
            />
        </div>
    );
}
export default App;
```

### Initialize the Toast with target

By default toast can be rendered in document body, we can change the target position for toast rendering using `target` property.

In the above sample code, `#element` is the `id` of the HTML element in a page to which the Toast is initialized.

### APP.JSX

```
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import { useRef } from 'react';
function App() {
    const toastInstance = useRef(null);
    function toastCreated() {
```

```

        toastInstance.current.show();
    }
    return (<div id='#toast_target' />
        ,
        <ToastComponent id="toast_target" ref={toastInstance}
        title="Sample Toast Title" content="Sample Toast Content"
        created={toastCreated} />);
    div >
    ;
    ;
}
export default App;

```

### APP.TSX

```

import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
function App() {
    let toastInstance;
    function toastCreated() {
        toastInstance.show();
    }
    return (<div>
        <div id='#toast_target' />
        <ToastComponent id='toast_target' ref={toast => toastInstance = toast}
        title="Sample Toast Title" content="Sample Toast Content"
        created={toastCreated.bind(this)} />
        </div>);
    }
    ;
export default App;

```

See Also

- [Render different types of toast](#)

### Config in React Toast component

This section explains on customizing the Toast appearance using built-in APIs.

#### Title and content template

Toast can be created with the notification message. The message contains [title](#) and [content](#) of the Toasts. Title and contents are adaptable in any resolution.

Title or Content property can be given as HTML element/element ID as a string that can be displayed as a Toast.

[Class-component]

### APP.JSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from 'react';
class App extends React.Component {
    toastInstance;

```

```

position = { X: 'Center' };
toastCreated() {
    this.toastInstance.show();
}
toastShow() {
    this.toastInstance.show();
}
render() {
    return (<div>
        <ButtonComponent cssClass="e-primary" onClick={this.toastShow =
this.toastShow.bind(this)}> Show Toast </ButtonComponent>
        <ToastComponent ref={toast => this.toastInstance = toast} title="Matt
sent you a friend request" content="Hey, wanna dress up as wizards and ride
our hoverboards?" position={this.position} created={this.toastCreated =
this.toastCreated.bind(this)} />
    </div>);
}
}
export default App;

```

### APP.TSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
class App extends React.Component<{}, {}> {
    public toastInstance: ToastComponent;
    public position = { X: 'Center' };
    public toastCreated(): void {
        this.toastInstance.show();
    }
    public toastShow() {
        this.toastInstance.show();
    }
    public render() {
        return (
            <div>
                <ButtonComponent cssClass="e-primary" onClick={this.toastShow =
this.toastShow.bind(this)}> Show Toast </ButtonComponent>
                <ToastComponent ref={toast => this.toastInstance = toast!}
title="Matt sent you a friend request" content="Hey, wanna dress up as
wizards and ride our hoverboards?" position={this.position}
created={this.toastCreated = this.toastCreated.bind(this)} />
            </div>
        );
    }
}
export default App;

```

### [Functional-component]

#### APP.JSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';

```

```
import * as React from 'react';
function App() {
  let toastInstance;
  let position = { X: 'Center' };
  function toastCreated() {
    toastInstance.show();
  }
  function toastShow() {
    toastInstance.show();
  }
  return (<div>
    <ButtonComponent cssClass="e-primary" onClick={toastShow.bind(this)}>
Show Toast </ButtonComponent>
    <ToastComponent ref={toast => toastInstance = toast} title="Matt sent
you a friend request" content="Hey, wanna dress up as wizards and ride our
hoverboards?" position={position} created={toastCreated.bind(this)} />
    </div>);
}
export default App;
```

### APP.TSX

```
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
  let toastInstance: ToastComponent;
  let position = { X: 'Center' };

  function toastCreated(): void {
    toastInstance.show();
  }

  function toastShow() {
    toastInstance.show();
  }

  return (
    <div>
      <ButtonComponent cssClass="e-primary" onClick={toastShow.bind(this)}>
Show Toast </ButtonComponent>
      <ToastComponent ref={toast => toastInstance = toast!} title="Matt
sent you a friend request" content="Hey, wanna dress up as wizards and ride
our hoverboards?" position={position} created={toastCreated.bind(this)} />
    </div>
  );
}
export default App;
```

### Specifying custom target

By default toast can be rendered in the document body, we can change the target position for toast rendering using [target](#) property. Based on the target [position](#) will update.

### Close Button

By default [showCloseButton](#) will not be enabled. We can enable it by setting true value. Before expiring toast we can use to close or destroy toasts manually.

### Progress bar

By default [showProgressBar](#) will not be enabled. If we enabled it can visually indicate how long time to get toast expires. Based on the `timeOut` property Progress bar will appear.

### Progress bar direction

By default, the [progressDirection](#) is set to "Rtl" and it will appear from right to left direction. You can change the progressDirection to "Ltr" to make it appear from left to right direction.

### Newest on top

In default, newly created toasts will append next with existing toast. We can change the Sequence like inserting before the toast, by enabling the [newestOnTop](#).

Here below sample demonstrates the combination of `target`, `showCloseButton`, `showProgressBar` and `newestOnTop` properties in toast.

[Class-component]

### APP.JSX

```
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from 'react';
class App extends React.Component {
  toastInstance;
  position = { X: 'Center' };
  toastCreated() {
    this.toastShow();
  }
  toastShow() {
    setTimeout(() => {
      this.toastInstance.show();
    }, 600);
  }
  render() {
    return (
      <div>
        <ButtonComponent cssClass="e-primary" onClick={this.toastShow =
this.toastShow.bind(this)}> Show Toast </ButtonComponent>
        <ToastComponent ref={toast => this.toastInstance = toast} title="File
Downloading" content="<div class='progress'><span style='width:
80%'></span></div>" position={this.position} showCloseButton='true'
target="#toast_target" newestOnTop='true' showProgressBar='true'
progressDirection='Ltr' created={this.toastCreated =
this.toastCreated.bind(this)} />
      </div>
    );
  }
}
export default App;
```

### APP.TSX

```
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
```

```

import * as React from 'react';
import * as ReactDOM from 'react-dom';
class App extends React.Component<{}, {}> {
  public toastInstance: ToastComponent;
  public position = { X: 'Center' };
  public toastCreated(): void {
    this.toastShow();
  }
  public toastShow() {
    setTimeout(
      () => {
        this.toastInstance.show();
      }, 600);
  }
  public render() {
    return (
      <div>
        <ButtonComponent cssClass="e-primary" onClick={this.toastShow =
this.toastShow.bind(this)}> Show Toast </ButtonComponent>
        <ToastComponent ref={toast => this.toastInstance = toast!}
title="File Downloading" content="<div class='progress'><span style='width:
80%'></span></div>" position={this.position} showCloseButton='true'
target="#toast_target" newestOnTop='true' showProgressBar='true'
progressDirection='Ltr' created={this.toastCreated =
this.toastCreated.bind(this)} />
      </div>
    );
  }
}
export default App;

```

### [Functional-component]

#### APP.JSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from 'react';
function App() {
  let toastInstance;
  let position = { X: 'Center' };
  function toastCreated() {
    toastShow();
  }
  function toastShow() {
    setTimeout(() => {
      toastInstance.show();
    }, 600);
  }
  return (
    <div>
      <ButtonComponent cssClass="e-primary"
onClick={toastShow.bind(this)}> Show Toast </ButtonComponent>
      <ToastComponent ref={toast => toastInstance = toast} title="File
Downloading" content="<div class='progress'><span style='width:
80%'></span></div>" position={position} showCloseButton={true}

```

```
target="#toast_target" newestOnTop={true} showProgressBar={true}
progressDirection='Ltr' created={toastCreated.bind(this)} />
</div>;
}
export default App;
```

### APP.TSX

```
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
  let toastInstance: ToastComponent;
  let position = { X: 'Center' };

  function toastCreated(): void {
    toastShow();
  }

  function toastShow() {
    setTimeout(
      () => {
        toastInstance.show();
      }, 600);
  }

  return (
    <div>
      <ButtonComponent cssClass="e-primary"
onClick={toastShow.bind(this)}> Show Toast </ButtonComponent>
      <ToastComponent ref={toast => toastInstance = toast!} title="File
Downloading" content="<div class='progress'><span style='width:
80%'></span></div>" position={position} showCloseButton={true}
target="#toast_target" newestOnTop={true} showProgressBar={true}
progressDirection='Ltr' created={toastCreated.bind(this)} />
      </div>
    );
  }
}
export default App;
```

### Width and height

we can set toast dimensions through [width](#) and [height](#) property. The height and width can be applied for specific set of toasts. So you can create different toasts with custom dimension.

In default toast can be rendered with '300px' width with 'auto' height

In mobile device toast default width gets '100%' width of the page.

When we sets toast width as '100%' toast will occupies full width and displayed top or bottom based on position [Y](#) property.

Both width and height property allows setting pixels/numbers/percentage. The number value is considered as pixels.

**[Class-component]****APP.JSX**

```

import { ButtonComponent, CheckBoxComponent, RadioButtonComponent } from
 '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from 'react';
class App extends React.Component {
  toastInstance;
  checkboxObj;
  constructor(props) {
    super(props);
    this.state = {
      content: 'Hey, wanna dress up as wizards and ride our
hoverboards?',
      position: { X: 'Center', Y: "Bottom" },
      title: 'Matt sent you a friend request',
      width: 300
    };
  }
  toastCreated() {
    this.toastShow(600);
  }
  toastShow(timeOutDelay) {
    setTimeout(() => {
      this.toastInstance.show();
    }, timeOutDelay);
  }
  checkboxChange(e) {
    if (e.event.target.checked) {
      this.setState({ position: { Y: "Top" } });
      this.toastInstance.hide('All');
      this.toastShow(700);
    }
  }
  checkboxChange1(e) {
    if (e.event.target.checked) {
      this.setState({ position: { Y: "Bottom" } });
      this.toastInstance.hide('All');
      this.toastShow(700);
    }
  }
  onChange(e) {
    if (e.checked) {
      this.toastInstance.hide('All');
      this.setState({
        content: "<div class='e-custom'>Take a look at our next
generation <b>Javascript</b> <a
href='https://ej2.syncfusion.com/home/index.html' target='_blank'>LEARN
MORE</a></div>",
        width: '100%',
        title: ''
      });
      this.toastShow(500);
    }
    else {

```



```

        this.toastInstance.hide('All');
        this.setState({
            content: 'Hey, wanna dress up as wizards and ride our
hoverboards?',
            width: 300,
            title: 'Matt sent you a friend request'
        });
        this.toastShow(500);
    }
}
render() {
    return (<div id="container">
        <div id="toast_full_width" className='row'>
            <table>
                <tbody>
                    <tr>
                        <td>
                            <div>
                                <RadioButtonComponent checked={true} label='Top'
name="toast" change={this.checkboxChange = this.checkboxChange.bind(this)} />
                            </div>
                        </td>
                        <td>
                            <div>
                                <RadioButtonComponent checked={true} label='Bottom'
name="toast" change={this.checkboxChange1 =
this.checkboxChange1.bind(this)} />
                            </div>
                        </td>
                    </tr>
                    <tr>
                        <td>
                            <div>
                                <CheckBoxComponent label='100% Width' ref={scope =>
this.checkboxObj = scope} change={this.onChange = this.onChange.bind(this)} />
                            </div>
                        </td>
                    </tr>
                </tbody>
            </table>
            <div>
                <ButtonComponent cssClass="e-primary" onClick={this.toastShow =
this.toastShow.bind(this, 500)}> Show Toast </ButtonComponent>
                <ToastComponent ref={toast => this.toastInstance = toast}
width={this.state.width} title={this.state.title}
content={this.state.content} position={this.state.position}
created={this.toastCreated = this.toastCreated.bind(this)} />
            </div>
        </div>
    );
}
}
export default App;

```

**APP.TSX**

```

import { ButtonComponent, ChangeEventArgs, CheckBoxComponent,
RadioButtonComponent } from '@syncfusion/ej2-react-buttons';

```

```

import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
class App extends React.Component<{}>, {}> {
  public toastInstance: ToastComponent;
  public checkboxObj: CheckBoxComponent;
  constructor(props: any) {
    super(props);
    this.state = {
      content: 'Hey, wanna dress up as wizards and ride our hoverboards?',
      position: { X: 'Center', Y: "Bottom" },
      title: 'Matt sent you a friend request',
      width: 300
    }
  }
  public toastCreated(): void {
    this.toastShow(600);
  }
  public toastShow(timeOutDelay: number) {
    setTimeout(
      () => {
        this.toastInstance.show();
      }, timeOutDelay);
  }
  public checkboxChange(e: any): void {
    if (e.event.target.checked) {
      this.setState({ position: { Y: "Top" } });
      this.toastInstance.hide('All');
      this.toastShow(700);
    }
  }
  public checkboxChange1(e: any): void {
    if (e.event.target.checked) {
      this.setState({ position: { Y: "Bottom" } });
      this.toastInstance.hide('All');
      this.toastShow(700);
    }
  }
  public onChange(e: ChangeEventArgs): void {
    if (e.checked) {
      this.toastInstance.hide('All');
      this.setState({
        content: "<div class='e-custom'>Take a look at our next generation
<b>Javascript</b> <a href='https://ej2.syncfusion.com/home/index.html'
target='_blank'>LEARN MORE</a></div>",
        width: '100%',
        title: ''
      });
      this.toastShow(500);
    } else {
      this.toastInstance.hide('All');
      this.setState({
        content: 'Hey, wanna dress up as wizards and ride our hoverboards?',
        width: 300,
        title: 'Matt sent you a friend request'
      });
      this.toastShow(500);
    }
  }
}

```

```

    }
  }
  public render() {
    return (
      <div id="container">
        <div id="toast_full_width" className='row'>
          <table>
            <tr>
              <td>
                <div>
                  <RadioButtonComponent checked={true} label='Top'
name="toast" change={this.checkboxChange = this.checkboxChange.bind(this)} />
                </div>
              </td>
              <td>
                <div>
                  <RadioButtonComponent checked={true} label='Bottom'
name="toast" change={this.checkboxChange1 = this.checkboxChange1.bind(this)} />
                </div>
              </td>
            </tr>
            <tr>
              <td>
                <div>
                  <CheckBoxComponent label='100% Width' ref={scope =>
this.checkboxObj = scope!} change={this.onChange = this.onChange.bind(this)} />
                </div>
              </td>
            </tr>
          </table>
        </div>
        <ButtonComponent cssClass="e-primary" onClick={this.toastShow =
this.toastShow.bind(this, 500)}> Show Toast </ButtonComponent>
        <ToastComponent ref={toast => this.toastInstance = toast!}
width={this.state.width} title={this.state.title}
content={this.state.content} position={this.state.position}
created={this.toastCreated = this.toastCreated.bind(this)} />
      </div>
    );
  }
}
export default App;

```

### [Functional-component]

#### APP.JSX

```

import { ButtonComponent, CheckBoxComponent, RadioButtonComponent } from
'@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from 'react';
function App() {
  let toastInstance;
  let checkboxObj;

```

```

    const [status, setStatus] = React.useState({ content: 'Hey, wanna dress
up as wizards and ride our hoverboards?', title: 'Matt sent you a friend
request', width: '600', position: { X: 'Center', Y: "Bottom" } });
    function toastCreated() {
        toastShow(600);
    }
    function toastShow(timeOutDelay) {
        setTimeout(() => {
            toastInstance.show();
        }, timeOutDelay);
    }
    function checkboxChange(e) {
        if (e.event.target.checked) {
            setStatus({
                content: '',
                width: '',
                title: '',
                position: { Y: "Top", X: '' }
            });
            toastInstance.hide('All');
            toastShow(700);
        }
    }
    function checkboxChangel(e) {
        if (e.event.target.checked) {
            setStatus({
                content: '',
                width: '',
                title: '',
                position: { Y: "Bottom", X: '' }
            });
            toastInstance.hide('All');
            toastShow(700);
        }
    }
    function onChange(e) {
        if (e.checked) {
            toastInstance.hide('All');
            setStatus({
                content: "<div class='e-custom'>Take a look at our next
generation <b>Javascript</b> <a
href='https://ej2.syncfusion.com/home/index.html' target='_blank'>LEARN
MORE</a></div>",
                width: '100%',
                title: '',
                position: { X: '', Y: '' }
            });
            toastShow(500);
        }
        else {
            toastInstance.hide('All');
            setStatus({
                content: 'Hey, wanna dress up as wizards and ride our
hoverboards?',
                width: '300',
                title: 'Matt sent you a friend request',
                position: { X: '', Y: '' }
            });
        }
    }

```

```

        });
        toastShow(500);
    }
}
return (<div id="container">
    <div id="toast_full_width" className='row'>
        <table>
            <tr>
                <td>
                    <div>
                        <RadioButtonComponent checked={true} label='Top'
name="toast" change={checkboxChange.bind(this)} />
                    </div>
                </td>
                <td>
                    <div>
                        <RadioButtonComponent checked={true} label='Bottom'
name="toast" change={checkboxChange1.bind(this)} />
                    </div>
                </td>
            </tr>
            <tr>
                <td>
                    <div>
                        <CheckBoxComponent label='100% Width' ref={scope =>
checkboxObj = scope} change={onChange.bind(this)} />
                    </div>
                </td>
            </tr>
        </table>
        <div>
            <ButtonComponent cssClass="e-primary" onClick={toastShow.bind(this,
500)}> Show Toast </ButtonComponent>
            <ToastComponent ref={toast => toastInstance = toast}
width={status.width} title={status.title} content={status.content}
position={status.position} created={toastCreated.bind(this)} />
        </div>);
    }
export default App;

```

### APP.TSX

```

import { ButtonComponent, ChangeEventArgs, CheckBoxComponent,
RadioButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    let toastInstance: ToastComponent;
    let checkboxObj: CheckBoxComponent;
    const [status, setStatus] = React.useState({ content: 'Hey, wanna dress up
as wizards and ride our hoverboards?', title: 'Matt sent you a friend
request', width: '600', position: { X: 'Center', Y: "Bottom" } });
    function toastCreated(): void {
        toastShow(600);
    }
}

```

```

function toastShow(timeOutDelay: number) {
  setTimeout(
    () => {
      toastInstance.show();
    }, timeOutDelay);
}
function checkboxChange(e: any): void {
  if (e.event.target.checked) {
    setStatus({
      content: '',
      width: '',
      title: '',
      position: { Y: "Top", X: '' }
    });
    toastInstance.hide('All');
    toastShow(700);
  }
}
function checkboxChange1(e: any): void {
  if (e.event.target.checked) {
    setStatus({
      content: '',
      width: '',
      title: '',
      position: { Y: "Bottom", X: '' }
    });
    toastInstance.hide('All');
    toastShow(700);
  }
}
function onChange(e: ChangeEventArgs): void {
  if (e.checked) {
    toastInstance.hide('All');
    setStatus({
      content: "<div class='e-custom'>Take a look at our next generation
<b>Javascript</b> <a href='https://ej2.syncfusion.com/home/index.html'
target='_blank'>LEARN MORE</a></div>",
      width: '100%',
      title: '',
      position: {X: '',Y: ''}
    });
    toastShow(500);
  } else {
    toastInstance.hide('All');
    setStatus({
      content: 'Hey, wanna dress up as wizards and ride our hoverboards?',
      width: '300',
      title: 'Matt sent you a friend request',
      position: {X: '',Y: ''}
    });
    toastShow(500);
  }
}
return (
  <div id="container">
    <div id="toast_full_width" className='row'>
      <table>

```

```

        <tr>
          <td>
            <div>
              <RadioButtonComponent checked={true} label='Top'
name="toast" change={ checkboxChange.bind(this)} />
            </div>
          </td>
          <td>
            <div>
              <RadioButtonComponent checked={true} label='Bottom'
name="toast" change={checkboxChange1.bind(this)} />
            </div>
          </td>
        </tr>
        <tr>
          <td>
            <div>
              <CheckBoxComponent label='100% Width' ref={scope =>
checkboxObj = scope!} change={onChange.bind(this)} />
            </div>
          </td>
        </tr>
      </table>
    </div>
    <ButtonComponent cssClass="e-primary" onClick={toastShow.bind(this,
500)}> Show Toast </ButtonComponent>
    <ToastComponent ref={toast => toastInstance = toast!}
width={status.width} title={status.title} content={status.content}
position={status.position} created={ toastCreated.bind(this)} />
  </div>
);
}
export default App;

```

See Also

[Prevent duplicate toasts](#)

[Customize the progress bar](#)

## Position in React Toast component

Toast position can be updated based on predefined positions or user customizable positions. Predefined position combinations are updated in [X](#) and [Y](#) position properties.

### Predefined

#### X Positions

- Left
- Center
- Right

#### Y Positions

- Top

- Bottom

In the case of multiple Toast display, new Toast position will not update on dynamic change of property values, until the old Toast messages removed.

Toast occupies full width when we set width as '100%', so X positions won't affect changes when '100%' width.

#### Custom

Custom X and Y Position can be given as pixels/numbers/percentage. The number value is considered as pixels. based value top and left value updated in the toast.

#### [Class-component]

#### APP.JSX

```
import { ButtonComponent, RadioButtonComponent } from '@syncfusion/ej2-react-buttons';
import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
class App extends React.Component {
  toastInstance;
  customFlag;
  checkboxObj;
  dropDownInstance;
  radioInstance1;
  radioInstance;
  radioInstance2;
  radioInstance3;
  timeoutDelay = 600;
  dropdownDB = ['Top Left', 'Top Right', 'Top Center', 'Bottom Left', 'Bottom Right', 'Bottom Center'];
  constructor(props) {
    super(props);
    this.state = {
      position: { X: 'Center', Y: "Bottom" },
      target: document.body,
      width: 300
    };
  }
  toastCreated() {
    this.toastShow(600);
  }
  toastShow(timeoutDelay) {
    setTimeout(() => {
      this.toastInstance.show();
    }, timeoutDelay);
  }
  valueChange(e) {
    this.toastInstance.hide('All');
    this.setToastPosValue(e.value);
    this.toastShow(1000);
  }
  radioChange(e) {
    if (this.radioInstance.element.checked) {
```



```

        this.toastInstance.hide('All');
        document.getElementById('dropdownChoose').style.display = 'table-
cell';

        document.getElementById('customChoose').style.display = 'none';
        this.setToastPosValue(this.dropDownInstance.value.toString());
        this.customFlag = true;
        this.toastShow(1000);
    }
}
radioChange3(e) {
    if (this.radioInstance3.element.checked) {
        this.toastInstance.hide('All');
        this.setState({ target: document.body });
        this.toastShow(1000);
    }
}
radioChange2(e) {
    if (this.radioInstance2.element.checked) {
        this.toastInstance.hide('All');
        this.setState({ target: '#toast_pos_target' });
        this.toastShow(1000);
    }
}
radioChange1(e) {
    if (this.radioInstance1.element.checked) {
        this.toastInstance.hide('All');
        document.getElementById('dropdownChoose').style.display = 'none';
        document.getElementById('customChoose').style.display = 'table-
cell';

        this.setCustomPosValue();
        this.customFlag = true;
        this.toastShow(1000);
    }
}
setCustomPosValue() {
    this.setState({
        position: {
            X: parseInt(document.getElementById('xPos').value, 10),
            Y: parseInt(document.getElementById('yPos').value, 10)
        }
    });
}
setToastPosValue(value) {
    value = value.toLowerCase().replace(' ', '');
    switch (value) {
        case 'topleft':
            this.setState({ position: { X: 'Left', Y: 'Top' } });
            break;
        case 'topright':
            this.setState({ position: { X: 'Right', Y: 'Top' } });
            break;
        case 'topcenter':
            this.setState({ position: { X: 'Center', Y: 'Top' } });
            break;
        case 'topfullwidth':
            this.setState({ position: { X: 'Center', Y: 'Top' }, width:
'100%' });

```

```

        break;
      case 'bottomleft':
        this.setState({ position: { X: 'Left', Y: 'Bottom' } });
        break;
      case 'bottomright':
        this.setState({ position: { X: 'Right', Y: 'Bottom' } });
        break;
      case 'bottomcenter':
        this.setState({ position: { X: 'Center', Y: 'Bottom' } });
        break;
      case 'bottomfullwidth':
        this.setState({ position: { X: 'Center', Y: 'Bottom' },
width: '100%' });
        break;
    }
  }
  render() {
    return (<div id="toast_pos_target">
      <div id="toast_full_Position" className='row'>
        <table>
          <tbody>
            <tr>
              <td>
                <div>
                  <RadioButtonComponent ref={custom => this.radioInstance =
custom} checked={true} label='Position' name="position"
change={this.radioChange = this.radioChange.bind(this)} />
                </div>
              </td>
              <td>
                <div>
                  <RadioButtonComponent ref={custom => this.radioInstance1
= custom} label='Custom' name="position" change={this.radioChange1 =
this.radioChange1.bind(this)} />
                </div>
              </td>
            </tr>
            <tr>
              <td id="dropdownChoose" colSpan={2}>
                <div>
                  <DropDownListComponent ref={drop => this.dropDownInstance
= drop} id="ddlelement" dataSource={this.dropdownDB} placeholder="Select a
Position" change={this.valueChange = this.valueChange.bind(this)} index='4' />
                </div>
              </td>
            </tr>
            <tr>
              <td id='customChoose'>
                <div className='e-row'>
                  <div className='e-float-input'>
                    <input type='text' className='e-input' id='xPos'
defaultValue={'50'} required={true} />
                    <span className='e-float-line' />
                    <label className="e-float-text">X Position</label>
                  </div>
                </div>
                <div className='e-row'>

```

```

        <div className='e-float-input'>
            <input type='text' className='e-input' id='yPos'
defaultValue={'50'} required={true}/>
            <span className='e-float-line' />
            <label className="e-float-text">Y Position</label>
        </div>
    </div>
</td>
</tr>
<tr>
<td>
        <div>
            <RadioButtonComponent ref={custom => this.radioInstance2
= custom} checked={true} label='Target' name="toast"
change={this.radioChange2 = this.radioChange2.bind(this)} />
        </div>
    </td>
<td>
        <div>
            <RadioButtonComponent ref={custom => this.radioInstance3
= custom} checked={true} label='Global' name="toast"
change={this.radioChange3 = this.radioChange3.bind(this)} />
        </div>
    </td>
</tr>
</tbody>
</table>
</div>
    <ButtonComponent cssClass="e-primary" onClick={this.toastShow =
this.toastShow.bind(this, 500)}> Show Toast </ButtonComponent>
    <ToastComponent ref={toast => this.toastInstance = toast} title="Matt
sent you a friend request" content="Hey, wanna dress up as wizards and ride
our hoverboards?" target={this.state.target} width={this.state.width}
position={this.state.position} created={this.toastCreated =
this.toastCreated.bind(this)} />
    </div>);
    }
}
;
export default App;

```

### APP.TSX

```

import { ButtonComponent, CheckBoxComponent, RadioButtonComponent } from
'@syncfusion/ej2-react-buttons';
import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
class App extends React.Component<{}, {position: any, target: any, width:
any}> {
    public toastInstance: ToastComponent;
    public customFlag: boolean;
    public checkboxObj: CheckBoxComponent;
    public dropDownInstance: DropDownListComponent;
    public radioInstance1: RadioButtonComponent;
    public radioInstance: RadioButtonComponent;

```

```

public radioInstance2: RadioButtonComponent;
public radioInstance3: RadioButtonComponent;
public timeOutDelay: number = 600;
public dropdownDB = ['Top Left', 'Top Right', 'Top Center', 'Bottom Left',
'Bottom Right', 'Bottom Center'];
constructor(props: any) {
  super(props);
  this.state = {
    position: { X: 'Center', Y: "Bottom" },
    target: document.body,
    width: 300
  }
}
public toastCreated(): void {
  this.toastShow(600);
}
public toastShow(timeOutDelay: number) {
  setTimeout(
    () => {
      this.toastInstance.show();
    }, timeOutDelay);
}
public valueChange(e: any): void {
  this.toastInstance.hide('All');
  this.setToastPosValue(e.value);
  this.toastShow(1000);
}
public radioChange(e: any): void {
  if (this.radioInstance.element.checked) {
    this.toastInstance.hide('All');
    (document.getElementById('dropdownChoose') as
HTMLElement).style.display = 'table-cell';
    (document.getElementById('customChoose') as HTMLElement).style.display
= 'none';
    this.setToastPosValue(this.dropDownInstance.value.toString());
    this.customFlag = true;
    this.toastShow(1000);
  }
}
public radioChange3(e: any): void {
  if (this.radioInstance3.element.checked) {
    this.toastInstance.hide('All');
    this.setState({ target: document.body });
    this.toastShow(1000);
  }
}
public radioChange2(e: any): void {
  if (this.radioInstance2.element.checked) {
    this.toastInstance.hide('All');
    this.setState({ target: '#toast_pos_target' });
    this.toastShow(1000);
  }
}
public radioChange1(e: any): void {
  if (this.radioInstance1.element.checked) {
    this.toastInstance.hide('All');
  }
}

```

```

        (document.getElementById('dropdownChoose') as
HTMLInputElement).style.display = 'none';
        (document.getElementById('customChoose') as HTMLInputElement).style.display
= 'table-cell';
        this.setCustomPosValue();
        this.customFlag = true;
        this.toastShow(1000);
    }
}
public setCustomPosValue(): void {
    this.setState({
        position: {
            X: parseInt((document.getElementById('xPos') as
HTMLInputElement).value, 10),
            Y: parseInt((document.getElementById('yPos') as
HTMLInputElement).value, 10)
        }
    });
}
public setToastPosValue(value: string): void {
    value = value.toLowerCase().replace(' ', '');
    switch (value) {
        case 'topleft':
            this.setState({ position: { X: 'Left', Y: 'Top' } });
            break;
        case 'topright':
            this.setState({ position: { X: 'Right', Y: 'Top' } });
            break;
        case 'topcenter':
            this.setState({ position: { X: 'Center', Y: 'Top' } });
            break;
        case 'topfullwidth':
            this.setState({ position: { X: 'Center', Y: 'Top' }, width: '100%'
    });
            break;
        case 'bottomleft':
            this.setState({ position: { X: 'Left', Y: 'Bottom' } });
            break;
        case 'bottomright':
            this.setState({ position: { X: 'Right', Y: 'Bottom' } });
            break;
        case 'bottomcenter':
            this.setState({ position: { X: 'Center', Y: 'Bottom' } });
            break;
        case 'bottomfullwidth':
            this.setState({ position: { X: 'Center', Y: 'Bottom' }, width: '100%'
    });
            break;
    }
}
public render() {
    return (
        <div id="toast_pos_target">
            <div id="toast_full_Position" className='row'>
                <table>
                    <tbody>
                        <tr>

```

```

        <td>
            <div>
                <RadioButtonComponent ref={custom => this.radioInstance =
custom!} checked={true} label='Position' name="position"
change={this.radioChange = this.radioChange.bind(this)} />
            </div>
        </td>
        <td>
            <div>
                <RadioButtonComponent ref={custom => this.radioInstance1
= custom!} label='Custom' name="position" change={this.radioChange1 =
this.radioChange1.bind(this)} />
            </div>
        </td>
    </tr>
    <tr>
        <td id="dropdownChoose" colSpan={2}>
            <div>
                <DropDownListComponent ref={drop => this.dropDownInstance
= drop!} id="ddlelement" dataSource={this.dropdownDB} placeholder="Select a
Position" change={this.valueChange = this.valueChange.bind(this)} index='4'
/>
            </div>
        </td>
    </tr>
    <tr>
        <td id='customChoose'>
            <div className='e-row'>
                <div className='e-float-input'>
                    <input type='text' className='e-input' id='xPos'
defaultValue={'50'} required={true} />
                    <span className='e-float-line' />
                    <label className="e-float-text">X Position</label>
                </div>
            </div>
            <div className='e-row'>
                <div className='e-float-input'>
                    <input type='text' className='e-input' id='yPos'
defaultValue={'50'} required={true} />
                    <span className='e-float-line' />
                    <label className="e-float-text">Y Position</label>
                </div>
            </div>
        </td>
    </tr>
    <tr>
        <td>
            <div>
                <RadioButtonComponent ref={custom => this.radioInstance2
= custom!} checked={true} label='Target' name="toast"
change={this.radioChange2 = this.radioChange2.bind(this)} />
            </div>
        </td>
    </tr>
    <td>
        <div>

```

```

        <RadioButtonComponent ref={custom => this.radioInstance3
= custom!} checked={true} label='Global' name="toast"
change={this.radioChange3 = this.radioChange3.bind(this)} />
      </div>
    </td>
  </tr>
</tbody>
</table>
</div>
  <ButtonComponent cssClass="e-primary" onClick={this.toastShow =
this.toastShow.bind(this, 500)}> Show Toast </ButtonComponent>
  <ToastComponent ref={toast => this.toastInstance = toast!}
title="Matt sent you a friend request" content="Hey, wanna dress up as
wizards and ride our hoverboards?" target={this.state.target}
width={this.state.width} position={this.state.position}
created={this.toastCreated = this.toastCreated.bind(this)} />
</div>
);
}
};
export default App;

```

### [Functional-component]

#### APP.JSX

```

import { ButtonComponent, RadioButtonComponent } from '@syncfusion/ej2-react-
buttons';
import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
function App(position, target, width) {
  let toastInstance;
  let customFlag;
  let checkboxObj;
  let dropDownInstance;
  let radioInstance1;
  let radioInstance;
  let radioInstance2;
  let radioInstance3;
  let timeOutDelay = 600;
  let dropdownDB = ['Top Left', 'Top Right', 'Top Center', 'Bottom Left',
'Bottom Right', 'Bottom Center'];
  const [state1, setStatus1] = React.useState({ position: { X: 'Center', Y:
"Bottom" } });
  const [state2, setStatus2] = React.useState({ target: document.body });
  const [state3, setStatus3] = React.useState({ width: '300' });
  function toastCreated() {
    toastShow(600);
  }
  function toastShow(timeOutDelay) {
    setTimeout(() => {
      toastInstance.show();
    }, timeOutDelay);
  }
  function valueChange(e) {

```

```

        toastInstance.hide('All');
        setToastPosValue(e.value);
        toastShow(1000);
    }
    function radioChange(e) {
        if (radioInstance.element.checked) {
            toastInstance.hide('All');
            document.getElementById('dropdownChoose').style.display = 'table-
cell';

            document.getElementById('customChoose').style.display = 'none';
            setToastPosValue(dropDownInstance.value.toString());
            customFlag = true;
            toastShow(1000);
        }
    }
    function radioChange3(e) {
        if (radioInstance3.element.checked) {
            toastInstance.hide('All');
            setStatus2({ target: document.body });
            toastShow(1000);
        }
    }
    function radioChange2(e) {
        if (radioInstance2.element.checked) {
            toastInstance.hide('All');
            setStatus2({ target: '#toast_pos_target' });
            toastShow(1000);
        }
    }
    function radioChange1(e) {
        if (radioInstance1.element.checked) {
            toastInstance.hide('All');
            document.getElementById('dropdownChoose').style.display = 'none';
            document.getElementById('customChoose').style.display = 'table-
cell';

            setCustomPosValue();
            customFlag = true;
            toastShow(1000);
        }
    }
    function setCustomPosValue() {
        setStatus1({
            position: {
                X: parseInt(document.getElementById('xPos').value),
                Y: parseInt(document.getElementById('yPos').value)
            }
        });
    }
    function setToastPosValue(value) {
        value = value.toLowerCase().replace(' ', '');
        switch (value) {
            case 'topleft':
                setStatus1({ position: { X: 'Left', Y: 'Top' } });
                break;
            case 'topright':
                setStatus1({ position: { X: 'Right', Y: 'Top' } });
                break;
        }
    }

```



```

        case 'topcenter':
            setStatus1({ position: { X: 'Center', Y: 'Top' } });
            break;
        case 'topfullwidth':
            setStatus1({ position: { X: 'Center', Y: 'Top' } });
            setStatus3({ width: '100%' });
            break;
        case 'bottomleft':
            setStatus1({ position: { X: 'Left', Y: 'Bottom' } });
            break;
        case 'bottomright':
            setStatus1({ position: { X: 'Right', Y: 'Bottom' } });
            break;
        case 'bottomcenter':
            setStatus1({ position: { X: 'Center', Y: 'Bottom' } });
            break;
        case 'bottomfullwidth':
            setStatus1({ position: { X: 'Center', Y: 'Bottom' } });
            setStatus3({ width: '100%' });
            break;
    }
}
return (<div id="toast_pos_target">
    <div id="toast_full_Position" className='row'>
        <table>
            <tbody>
                <tr>
                    <td>
                        <div>
                            <RadioButtonComponent ref={custom => radioInstance =
custom} checked={true} label='Position' name="position"
change={radioChange.bind(this)} />
                        </div>
                    </td>
                    <td>
                        <div>
                            <RadioButtonComponent ref={custom => radioInstance1 =
custom} label='Custom' name="position" change={radioChange1.bind(this)} />
                        </div>
                    </td>
                </tr>
                <tr>
                    <td id="dropdownChoose" colspan={2}>
                        <div>
                            <DropDownListComponent ref={drop => dropDownInstance =
drop} id="ddlelement" dataSource={dropdownDB} placeholder="Select a Position"
change={valueChange.bind(this)} index={4} />
                        </div>
                    </td>
                </tr>
                <tr>
                    <td id='customChoose'>
                        <div className='e-row'>
                            <div className='e-float-input'>
                                <input type='text' className='e-input' id='xPos'
defaultValue={'50'} required={true} />
                                <span className='e-float-line' />

```

```

        <label className="e-float-text">X Position</label>
      </div>
    </div>
    <div className='e-row'>
      <div className='e-float-input'>
        <input type='text' className='e-input' id='yPos'
defaultValue={'50'} required={true}/>
        <span className='e-float-line' />
        <label className="e-float-text">Y Position</label>
      </div>
    </div>
  </td>
</tr>
<tr>
  <td>
    <div>
      <RadioButtonComponent ref={custom => radioInstance2 =
custom} checked={true} label='Target' name="toast"
change={radioChange2.bind(this)} />
    </div>
  </td>
  <td>
    <div>
      <RadioButtonComponent ref={custom => radioInstance3 =
custom} checked={true} label='Global' name="toast"
change={radioChange3.bind(this)} />
    </div>
  </td>
</tr>
</tbody>
</table>
</div>
<ButtonComponent cssClass="e-primary" onClick={toastShow.bind(this,
500)}> Show Toast </ButtonComponent>
<ToastComponent ref={toast => toastInstance = toast} title="Matt sent
you a friend request" content="Hey, wanna dress up as wizards and ride our
hoverboards?" target={state2.target} width={state3.width}
position={state1.position} created={toastCreated.bind(this)} />
</div>;
}
export default App;

```

## APP.TSX

```

{% raw %}
import { ButtonComponent, CheckBoxComponent, RadioButtonComponent } from
'@syncfusion/ej2-react-buttons';
import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
function App(position: any, target: any, width: any){
  let toastInstance: ToastComponent;
  let customFlag: boolean;
  let checkboxObj: CheckBoxComponent;
  let dropDownInstance: DropDownListComponent;
  let radioInstancel: RadioButtonComponent;

```

```

let radioInstance: RadioButtonComponent;
let radioInstance2: RadioButtonComponent;
let radioInstance3: RadioButtonComponent;
let timeOutDelay: number = 600;
let dropdownDB = ['Top Left', 'Top Right', 'Top Center', 'Bottom Left',
'Bottom Right', 'Bottom Center'];
const [state1, setStatus1] = React.useState({position: { X: 'Center', Y:
"Bottom" }});
const [state2, setStatus2] = React.useState({target: document.body});
const [state3, setStatus3] = React.useState({width: '300'});
function toastCreated(): void {
    toastShow(600);
}
function toastShow(timeOutDelay: number) {
    setTimeout(
        () => {
            toastInstance.show();
        }, timeOutDelay);
}
function valueChange(e: any): void {
    toastInstance.hide('All');
    setToastPosValue(e.value);
    toastShow(1000);
}
function radioChange(e: any): void {
    if (radioInstance.element.checked) {
        toastInstance.hide('All');
        (document.getElementById('dropdownChoose') as
HTMLElement).style.display = 'table-cell';
        (document.getElementById('customChoose') as HTMLElement).style.display
= 'none';
        setToastPosValue(dropDownInstance.value.toString());
        customFlag = true;
        toastShow(1000);
    }
}
function radioChange3(e: any): void {
    if (radioInstance3.element.checked) {
        toastInstance.hide('All');
        setStatus2({ target: document.body });
        toastShow(1000);
    }
}
function radioChange2(e: any): void {
    if (radioInstance2.element.checked) {
        toastInstance.hide('All');
        setStatus2({ target: '#toast_pos_target' });
        toastShow(1000);
    }
}
function radioChangel(e: any): void {
    if (radioInstance1.element.checked) {
        toastInstance.hide('All');
        (document.getElementById('dropdownChoose') as
HTMLElement).style.display = 'none';
        (document.getElementById('customChoose') as HTMLElement).style.display
= 'table-cell';

```

```

        setCustomPosValue();
        customFlag = true;
        toastShow(1000);
    }
}
function setCustomPosValue(): void {
    setStatus1({
        position: {
            X: parseInt((document.getElementById('xPos') as
HTMLInputElement).value),
            Y: parseInt((document.getElementById('yPos') as
HTMLInputElement).value)
        }
    });
}
function setToastPosValue(value: string): void {
    value = value.toLowerCase().replace(' ', '');
    switch (value) {
        case 'topleft':
            setStatus1({ position: { X: 'Left', Y: 'Top' } });
            break;
        case 'topright':
            setStatus1({ position: { X: 'Right', Y: 'Top' } });
            break;
        case 'topcenter':
            setStatus1({ position: { X: 'Center', Y: 'Top' } });
            break;
        case 'topfullwidth':
            setStatus1({ position: { X: 'Center', Y: 'Top' } });
            setStatus3({width: '100%' });
            break;
        case 'bottomleft':
            setStatus1({ position: { X: 'Left', Y: 'Bottom' } });
            break;
        case 'bottomright':
            setStatus1({ position: { X: 'Right', Y: 'Bottom' } });
            break;
        case 'bottomcenter':
            setStatus1({ position: { X: 'Center', Y: 'Bottom' } });
            break;
        case 'bottomfullwidth':
            setStatus1({ position: { X: 'Center', Y: 'Bottom' } });
            setStatus3({width: '100%' });
            break;
    }
}
return (
    <div id="toast_pos_target">
        <div id="toast_full_Position" className='row'>
            <table>
                <tbody>
                    <tr>
                        <td>
                            <div>
                                <RadioButtonComponent ref={custom => radioInstance =
custom!} checked={true} label='Position' name="position"
change={radioChange.bind(this)} />

```

```

        </div>
      </td>
      <td>
        <div>
          <RadioButtonComponent ref={custom => radioInstance1 =
custom!} label='Custom' name="position" change={radioChange1.bind(this)} />
        </div>
      </td>
    </tr>
    <tr>
      <td id="dropdownChoose" colSpan={2}>
        <div>
          <DropDownListComponent ref={drop => dropDownInstance =
drop!} id="ddlelement" dataSource={dropdownDB} placeholder="Select a
Position" change={valueChange.bind(this)} index={4} />
        </div>
      </td>
    </tr>
    <tr>
      <td id='customChoose'>
        <div className='e-row'>
          <div className='e-float-input'>
            <input type='text' className='e-input' id='xPos'
defaultValue={'50'} required={true} />
            <span className='e-float-line' />
            <label className="e-float-text">X Position</label>
          </div>
        </div>
        <div className='e-row'>
          <div className='e-float-input'>
            <input type='text' className='e-input' id='yPos'
defaultValue={'50'} required={true} />
            <span className='e-float-line' />
            <label className="e-float-text">Y Position</label>
          </div>
        </div>
      </td>
    </tr>
    <tr>
      <td>
        <div>
          <RadioButtonComponent ref={custom => radioInstance2 =
custom!} checked={true} label='Target' name="toast"
change={radioChange2.bind(this)} />
        </div>
      </td>
      <td>
        <div>
          <RadioButtonComponent ref={custom => radioInstance3 =
custom!} checked={true} label='Global' name="toast"
change={radioChange3.bind(this)} />
        </div>
      </td>
    </tr>
  </tbody>
</table>
</div>

```

```

        <ButtonComponent cssClass="e-primary" onClick={toastShow.bind(this,
500)}> Show Toast </ButtonComponent>
        <ToastComponent ref={toast => toastInstance = toast!} title="Matt
sent you a friend request" content="Hey, wanna dress up as wizards and ride
our hoverboards?" target={state2.target} width={state3.width}
position={state1.position} created={toastCreated.bind(this)} />
    </div>
    );
}
export default App;
{% enddraw %}

```

## See Also

- [Render toast with different positions](#)

## Action buttons in React Toast component

You can include action Buttons into toast by adding [buttons](#) property. You can bind collections of Essential JS 2 Button Model to `model` property inside buttons property, You can also include click event callback function, for each button.

[Class-component]

### APP.JSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
class App extends React.Component {
    toastInstance;
    position = { X: "Right", Y: "Bottom" };
    buttons = [
        { model: { content: "Ignore" } },
        { model: { content: "reply" } }
    ];
    toastShow() {
        this.toastInstance.show();
    }
    toastCreated() {
        this.toastInstance.show();
    }
    contentTemplate() {
        return <p><img src='./laura.png'/></p>;
    }
    render() {
        return (<div>
            <ButtonComponent cssClass="e-primary" onClick={this.toastShow =
this.toastShow.bind(this)}> Show Toast </ButtonComponent>
            <ToastComponent ref={toast => this.toastInstance = toast}
title="Anjolie Stokes" content={this.contentTemplate}
position={this.position} width='230' height='250' buttons={this.buttons}
created={this.toastCreated = this.toastCreated.bind(this)} />
            </div>);
    }
}

```

```
;
export default App;
```

**APP.TSX**

```
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
class App extends React.Component<{}, {}> {
  public toastInstance: ToastComponent;
  public position = { X: "Right", Y: "Bottom" };
  private buttons: object[] = [
    { model: { content: "Ignore" } },
    { model: { content: "reply" } }
  ];
  public toastShow() {
    this.toastInstance.show();
  }
  public toastCreated(): void {
    this.toastInstance.show();
  }
  public contentTemplate() {
    return <p><img src='./laura.png'/></p>;
  }
  public render() {
    return (
      <div>
        <ButtonComponent cssClass="e-primary" onClick={this.toastShow =
this.toastShow.bind(this)}> Show Toast </ButtonComponent>
        <ToastComponent ref={toast => this.toastInstance = toast!}
title="Anjolie Stokes" content={this.contentTemplate}
position={this.position} width='230' height='250' buttons={this.buttons}
created={this.toastCreated = this.toastCreated.bind(this)} />
      </div>
    );
  }
};
export default App;
```

**[Functional-component]****APP.JSX**

```
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
function App() {
  let toastInstance;
  let position = { X: "Right", Y: "Bottom" };
  let buttons = [
    { model: { content: "Ignore" } },
    { model: { content: "reply" } }
  ];
  function toastShow() {
    toastInstance.show();
  }
}
```

```

function toastCreated() {
    toastInstance.show();
}
function contentTemplate() {
    return <p><img src='./laura.png' /></p>;
}
return (<div>
    <ButtonComponent cssClass="e-primary"
onClick={toastShow.bind(this)}> Show Toast </ButtonComponent>
    <ToastComponent ref={toast => toastInstance = toast} title="Anjolie
Stokes" content={contentTemplate} position={position} width='230'
height='250' buttons={buttons} created={toastCreated.bind(this)} />
    </div>);
}
export default App;

```

**APP.TSX**

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
function App() {
    let toastInstance: ToastComponent;
    let position = { X: "Right", Y: "Bottom" };
    let buttons: object[] = [
        { model: { content: "Ignore" } },
        { model: { content: "reply" } }
    ];

    function toastShow() {
        toastInstance.show();
    }

    function toastCreated() {
        toastInstance.show();
    }

    function contentTemplate() {
        return <p><img src='./laura.png' /></p>;
    }

    return (
        <div>
            <ButtonComponent cssClass="e-primary"
onClick={toastShow.bind(this)}> Show Toast </ButtonComponent>
            <ToastComponent ref={toast => toastInstance = toast!}
title="Anjolie Stokes" content={contentTemplate} position={position}
width='230' height='250' buttons={buttons} created={toastCreated.bind(this)}
/>
            </div>
        );
    }
export default App;

```



See Also

- [Configuring options](#)

### Timeout in React Toast component

The [timeOut](#) property helps to change the default value of toast expire time. By default, the set timeOut value will be 5000 milliseconds. React Toast will live till the timeOut reaches without user interaction, the timeOut value will be taken in milliseconds. Once the toast reached a given timeOut it will expire automatically.

### Visual representation of timeOut

The toast component's [expire time](#) can be visually shown using a [Progress Bar](#). To show progress bar you must set showProgressbar api as true.

### How long the toast displayed in page

The [extendedTimeOut](#) property is used to decide how long the toast to be displayed after the user hovers on it.

You can destroy toast at any time by clicking on the close button. The close button can be enabled by [showCloseButton](#) property as true.

[Class-component]

### APP.JSX

```
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
class App extends React.Component {
  toastInstance;
  position = { X: "Right", Y: "Bottom" };
  buttons = [
    { model: { content: "Ignore" } },
    { model: { content: "reply" } }
  ];
  toastCreated() {
    this.toastShow();
  }
  toastShow() {
    const value = parseInt(document.getElementById('timeOut').value, 0);
    this.toastInstance.show({ timeOut: value });
  }
  contentTemplate() {
    return <p><img src='./laura.png'/></p>;
  }
  render() {
    return (
      <div>
        <div className='e-float-input'>
          <input type='text' className='e-input' id='timeOut'
            defaultValue={'0'} required={true}/>
          <span className='e-float-line'/>
          <label className='e-float-text'>TimeOut</label>
        </div>
        <ButtonComponent cssClass="e-primary" onClick={this.toastShow =
          this.toastShow.bind(this)}> Show Toast </ButtonComponent>
      </div>
    );
  }
}
```

```

        <ToastComponent id="elementToastTime" ref={toast =>
this.toastInstance = toast} title="Anjolie Stokes"
content={this.contentTemplate} position={this.position} width={230}
height={250} buttons={this.buttons} showProgressBar={true}
created={this.toastCreated = this.toastCreated.bind(this)} />
      </div>;
    }
  }
;
export default App;

```

## APP.TSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
class App extends React.Component<{}, {}> {
  public toastInstance: ToastComponent;
  public position = { X: "Right", Y: "Bottom" };
  private buttons: object[] = [
    { model: { content: "Ignore" } },
    { model: { content: "reply" } }
  ];
  public toastCreated(): void {
    this.toastShow();
  }
  public toastShow() {
    const value: number = parseInt((document.getElementById('timeOut') as
HTMLInputElement).value, 0);
    this.toastInstance.show({ timeOut: value });
  }
  public contentTemplate() {
    return <p><img src='./laura.png' /></p>;
  }
  public render() {
    return (
      <div>
        <div className='e-float-input'>
          <input type='text' className='e-input' id='timeOut'
defaultValue={'0'} required={true} />
          <span className='e-float-line' />
          <label className="e-float-text">TimeOut</label>
        </div>
        <ButtonComponent cssClass="e-primary" onClick={this.toastShow =
this.toastShow.bind(this)}> Show Toast </ButtonComponent>
        <ToastComponent id="elementToastTime" ref={toast =>
this.toastInstance = toast!} title="Anjolie Stokes"
content={this.contentTemplate} position={this.position} width={230}
height={250} buttons={this.buttons} showProgressBar={true}
created={this.toastCreated = this.toastCreated.bind(this)} />
      </div>
    );
  }
};
export default App;

```

**[Functional-component]****APP.JSX**

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
function App() {
    let toastInstance;
    let position = { X: "Right", Y: "Bottom" };
    let buttons = [
        { model: { content: "Ignore" } },
        { model: { content: "reply" } }
    ];
    function toastCreated() {
        toastShow();
    }
    function toastShow() {
        const value = parseInt(document.getElementById('timeOut').value, 0);
        toastInstance.show({ timeOut: value });
    }
    function contentTemplate() {
        return <p><img src='./laura.png'/></p>;
    }
    return (<div>
        <div className='e-float-input'>
            <input type='text' className='e-input' id='timeOut'
defaultValue={'0'} required={true}/>
            <span className='e-float-line' />
            <label className="e-float-text">TimeOut</label>
        </div>
        <ButtonComponent cssClass="e-primary" onClick={toastShow.bind(this)}>
Show Toast </ButtonComponent>
        <ToastComponent id="elementToastTime" ref={toast => toastInstance =
toast} title="Anjolie Stokes" content={contentTemplate} position={position}
width={230} height={250} buttons={buttons} showProgressBar={true}
created={toastCreated.bind(this)} />
        </div>);
}
export default App;

```

**APP.TSX**

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
function App() {
    let toastInstance: ToastComponent;
    let position = { X: "Right", Y: "Bottom" };
    let buttons: object[] = [
        { model: { content: "Ignore" } },
        { model: { content: "reply" } }
    ];
    function toastCreated(): void {
        toastShow();
    }
    function toastShow() {

```

```

const value: number = parseInt((document.getElementById('timeOut') as
HTMLInputElement).value, 0);
toastInstance.show({ timeOut: value });
}
function contentTemplate() {
  return <p><img src='./laura.png' /></p>;
}
return (
  <div>
    <div className='e-float-input'>
      <input type='text' className='e-input' id='timeOut'
defaultValue='0' required={true} />
      <span className='e-float-line' />
      <label className="e-float-text">TimeOut</label>
    </div>
    <ButtonComponent cssClass="e-primary" onClick={toastShow.bind(this)}>
Show Toast </ButtonComponent>
    <ToastComponent id="elementToastTime" ref={toast => toastInstance =
toast!} title="Anjolie Stokes" content={contentTemplate} position={position}
width={230} height={250} buttons={buttons} showProgressBar={true}
created={toastCreated.bind(this)} />
  </div>
);
}
export default App;

```

### Static toast

You can prevent auto-hiding of React toast by set timeOut value of timeOut property as zero (0).

### [Class-component]

#### APP.JSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
class App extends React.Component {
  toastInstance;
  position = { X: "Right", Y: "Bottom" };
  buttons = [
    { model: { content: "Ignore" } },
    { model: { content: "reply" } }
  ];
  toastCreated() {
    this.toastShow();
  }
  toastShow() {
    const value = parseInt(document.getElementById('timeOut').value, 0);
    this.toastInstance.show({ timeOut: value });
  }
  contentTemplate() {
    return <p><img src='./laura.png' /></p>;
  }
  render() {
    return <div>
      <div className='e-float-input'>

```

```

        <input type='text' className='e-input' id='timeOut'
        defaultValue={'0'} required={true}/>
        <span className='e-float-line' />
        <label className="e-float-text">TimeOut</label>
    </div>
    <ButtonComponent cssClass="e-primary" onClick={this.toastShow =
    this.toastShow.bind(this)}> Show Toast </ButtonComponent>
    <ToastComponent id="elementToastTime" ref={toast =>
    this.toastInstance = toast} title="Anjolie Stokes"
    content={this.contentTemplate} position={this.position} width={230}
    height={250} buttons={this.buttons} showProgressBar={true}
    created={this.toastCreated = this.toastCreated.bind(this)} />
    </div>;
    }
}
;
export default App;

```

### APP.TSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
class App extends React.Component<{}, {}> {
    public toastInstance: ToastComponent;
    public position = { X: "Right", Y: "Bottom" };
    private buttons: object[] = [
        { model: { content: "Ignore" } },
        { model: { content: "reply" } }
    ];
    public toastCreated(): void {
        this.toastShow();
    }
    public toastShow() {
        const value: number = parseInt((document.getElementById('timeOut') as
        HTMLInputElement).value, 0);
        this.toastInstance.show({ timeOut: value });
    }
    public contentTemplate() {
        return <p><img src='./laura.png' /></p>;
    }
    public render() {
        return (
            <div>
                <div className='e-float-input'>
                    <input type='text' className='e-input' id='timeOut'
                    defaultValue={'0'} required={true} />
                    <span className='e-float-line' />
                    <label className="e-float-text">TimeOut</label>
                </div>
                <ButtonComponent cssClass="e-primary" onClick={this.toastShow =
                this.toastShow.bind(this)}> Show Toast </ButtonComponent>
                <ToastComponent id="elementToastTime" ref={toast =>
                this.toastInstance = toast!} title="Anjolie Stokes"
                content={this.contentTemplate} position={this.position} width={230}

```

```

height={250} buttons={this.buttons} showProgressBar={true}
created={this.toastCreated = this.toastCreated.bind(this)} />
    </div>
  );
}
};
export default App;

```

## [Functional-component]

### APP.JSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
function App() {
  let toastInstance;
  let position = { X: "Right", Y: "Bottom" };
  let buttons = [
    { model: { content: "Ignore" } },
    { model: { content: "reply" } }
  ];
  function toastCreated() {
    toastShow();
  }
  function toastShow() {
    const value = parseInt(document.getElementById('timeOut').value, 0);
    toastInstance.show({ timeOut: value });
  }
  function contentTemplate() {
    return <p><img src='./laura.png' /></p>;
  }
  return <div>
    <div className='e-float-input'>
      <input type='text' className='e-input' id='timeOut'
defaultValue='0' required={true}/>
      <span className='e-float-line' />
      <label className="e-float-text">TimeOut</label>
    </div>
    <ButtonComponent cssClass="e-primary" onClick={toastShow.bind(this)}>
Show Toast </ButtonComponent>
    <ToastComponent id="elementToastTime" ref={toast => toastInstance =
toast} title="Anjolie Stokes" content={contentTemplate} position={position}
width={230} height={250} buttons={buttons} showProgressBar={true}
created={toastCreated.bind(this)} />
    </div>;
}
export default App;

```

### APP.TSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
function App() {
  let toastInstance: ToastComponent;

```

```

let position = { X: "Right", Y: "Bottom" };
let buttons: object[] = [
  { model: { content: "Ignore" } },
  { model: { content: "reply" } }
];
function toastCreated(): void {
  toastShow();
}
function toastShow() {
  const value: number = parseInt((document.getElementById('timeOut') as
HTMLInputElement).value, 0);
  toastInstance.show({ timeOut: value });
}
function contentTemplate() {
  return <p><img src='./laura.png'/></p>;
}
return (
  <div>
    <div className='e-float-input'>
      <input type='text' className='e-input' id='timeOut'
defaultValue={'0'} required={true} />
      <span className='e-float-line' />
      <label className="e-float-text">TimeOut</label>
    </div>
    <ButtonComponent cssClass="e-primary" onClick={toastShow.bind(this)}>
Show Toast </ButtonComponent>
    <ToastComponent id="elementToastTime" ref={toast => toastInstance =
toast!} title="Anjolie Stokes" content={contentTemplate} position={position}
width={230} height={250} buttons={buttons} showProgressBar={true}
created={toastCreated.bind(this)} />
  </div>
);
}
export default App;

```

See Also

- [Hide the toast on click](#)

## Template in React Toast component

Template property can be given as the HTML element that is either a string or a query selector.

The HTML element tag can be given as a string for the template property.

`ts

template: "<div>Toast Content</div>"

,

The template property also allows getting template content through query selector. Here, element 'ID' attribute is specified in the template.

,

template: "#Template"

## [Class-component]

**APP.JSX**

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
// import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
class App extends React.Component {
    toastInstance;
    position = { X: 'Right', Y: 'Bottom' };
    template = document.getElementById("template_toast_ele").innerHTML;
    toastCreated() {
        this.toastInstance.show();
    }
    toastShow() {
        this.toastInstance.show();
    }
    render() {
        return (<div>
            <ButtonComponent cssClass="e-primary" id='template_toast'
onClick={this.toastShow = this.toastShow.bind(this)}> Show Toast
</ButtonComponent>
            <ToastComponent id='template_toast' ref={toast => this.toastInstance
= toast} template={this.template} position={this.position}
extendedTimeout={0} timeOut={120000} created={this.toastCreated =
this.toastCreated.bind(this)}>/>
        </div>);
    }
}
;
export default App;

```

**APP.TSX**

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
// import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
class App extends React.Component<{}, {position: any, target: any, width:
any}> {
    public toastInstance: ToastComponent;
    public position = { X: 'Right', Y: 'Bottom' };
    public template = (document.getElementById("template_toast_ele") as
HTMLElement).innerHTML;
    public toastCreated(): void {
        this.toastInstance.show();
    }
    public toastShow() {
        this.toastInstance.show();
    }
    public render() {
        return (
            <div>

```



```

    <ButtonComponent cssClass="e-primary" id='template_toast'
onClick={this.toastShow = this.toastShow.bind(this)}> Show Toast
</ButtonComponent>
    <ToastComponent id='template_toast' ref={toast => this.toastInstance
= toast!} template={this.template} position={this.position}
extendedTimeout={0} timeout={120000} created={this.toastCreated =
this.toastCreated.bind(this)} />
  </div>
);
}
};
export default App;

```

### [Functional-component]

#### APP.JSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
// import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
function App() {
    let toastInstance;
    let position;
    let target;
    let width;
    let position = { X: 'Right', Y: 'Bottom' };
    let template = document.getElementById("template_toast_ele").innerHTML;
    function toastCreated() {
        toastInstance.show();
    }
    function toastShow() {
        toastInstance.show();
    }
    return (<div>
        <ButtonComponent cssClass="e-primary" id='template_toast'
onClick={toastShow = toastShow.bind(this)}> Show Toast </ButtonComponent>
        <ToastComponent id='template_toast' ref={toast => toastInstance =
toast} template={template} position={position} extendedTimeout={0}
timeout={120000} created={toastCreated = toastCreated.bind(this)} />
    </div>);
}
export default App;

```

#### APP.TSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
// import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
function App() {
    let toastInstance: ToastComponent;
    let position: any;
    let target: any;
    let width: any;
    let position = { X: 'Right', Y: 'Bottom' };

```

```

let template = (document.getElementById("template_toast_ele") as
HTMLElement).innerHTML;
function toastCreated(): void {
    toastInstance.show();
}
function toastShow() {
    toastInstance.show();
}
return (
    <div>
        <ButtonComponent cssClass="e-primary" id='template_toast'
onClick={toastShow = toastShow.bind(this)}> Show Toast </ButtonComponent>
        <ToastComponent id='template_toast' ref={toast => toastInstance =
toast!} template={template} position={position} extendedTimeout={0}
timeOut={120000} created={toastCreated = toastCreated.bind(this)} />
    </div>
);
}
export default App;

```

See Also

- [Add template dynamically](#)

### Style in React Toast component

The following content provides the exact CSS structure that can be used to modify the control's appearance based on the user preference.

#### Customizing the toast title

Use the following CSS to customize the default toast's content properties like font-family, font-size and color.

```

`css
/ To change color, font family and font size /
.e-toast-container .e-toast .e-toast-message .e-toast-title {
color: red;
font-size: 18px;
font-weight: bold;
}
`

```

#### Customizing the toast content

Use the following CSS to customize the default toast's content properties like font-family, font-size and color.

```

`css
/ To change color, font family and font size /
.e-toast-container .e-toast .e-toast-message .e-toast-content {

```

```
color: aqua;
font-size: 13px;
font-weight: normal;
}
`
```

### Customizing the toast icon

Use the following CSS to customize the default toast icon color.

```
`css
/ To change icon color /
.e-toast-container .e-toast .e-toast-icon {
color: yellow;
}
`
```

### Customizing the toast background

Use the following CSS to customize the default toast's background color.

```
`css
/ To change background color /
.e-toast-container .e-toast {
background-color: navy;
}
`
```

### Animation in React Toast component

[React Toasts](#) support custom animations for both shows and hide actions from the provided animation option of [Animation](#) library.

Default animation is given as [FadeIn](#) for showing the toast and [FadeOut](#) for hiding the toast.

The sample demonstrates some types of animation that suits Toast. You can check all the animation effects [here](#).

[Class-component]

#### APP.JSX

```
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from 'react';
class App extends React.Component {
  toastInstance;
  timeoutDelay = 600;
  position = { X: 'Right', Y: 'Bottom' };
  animation;
```

```

    style = { paddingTop: '20px' };
    dropDownShowInstance;
    dropDownHideInstance;
    AnimationDB = ['FadeIn', 'Fadeout', 'FadeZoomIn', 'FadeZoomOut',
'FlipLeftDownIn', 'FlipLeftDownOut', 'FlipLeftUpIn', 'FlipLeftUpOut',
'FlipRightDownIn', 'FlipRightDownOut', 'SlideBottomIn', 'SlideBottomOut',
'ZoomIn', 'ZoomOut'];
    toastCreated() {
        this.toastShow();
    }
    toastShow() {
        setTimeout(() => {
            this.toastInstance.show();
        }, this.timeOutDelay);
    }
    btnClick() {
        this.toastShow();
    }
    valueChange() {
        this.toastInstance.animation.show = { effect:
this.dropDownShowInstance.value };
    }
    valueChangeHide() {
        this.toastInstance.animation.hide = { effect:
this.dropDownHideInstance.value };
    }
    render() {
        return (<div>
            <ButtonComponent className="e-primary" onClick={this.btnClick =
this.btnClick.bind(this)}> Show Toast</ButtonComponent>
            <div className='e-row' style={this.style}>
                <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
                    <label>Show Animation</label>
                </div>
                <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
                    <DropDownListComponent ref={drop => this.dropDownShowInstance =
drop} dataSource={this.AnimationDB} placeholder="Select a ProgressBar Color"
change={this.valueChange = this.valueChange.bind(this)} index={0}/>
                </div>
            </div>
            <div className='e-row' style={this.style}>
                <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
                    <label>Hide Animation</label>
                </div>
                <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
                    <DropDownListComponent ref={drop => this.dropDownHideInstance =
drop} dataSource={this.AnimationDB} placeholder="Select a ProgressBar Color"
change={this.valueChangeHide = this.valueChangeHide.bind(this)} index={0}/>
                </div>
            </div>
            <ToastComponent ref={toast => this.toastInstance = toast} title='Matt
sent you a friend request' content='Hey, wanna dress up as wizards and ride
our hoverboards?' position={this.position} animation={this.animation}
showProgressBar='true' created={this.toastCreated =
this.toastCreated.bind(this)} />
        </div>);
    }
}

```

```

}
;
export default App;

```

### APP.TSX

```

import { Effect } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
class App extends React.Component<{}, {}> {
    public toastInstance: ToastComponent;
    public timeOutDelay: number = 600;
    public position = { X: 'Right', Y: 'Bottom' };
    public animation: { show: { effect: 'SlideRightIn' }, hide: { effect:
'SlideLeftOut' } };
    public style = { padding: '20px' };
    public dropDownShowInstance: DropDownListComponent;
    public dropDownHideInstance: DropDownListComponent;
    public AnimationDB = ['FadeIn', 'Fadeout', 'FadeZoomIn', 'FadeZoomOut',
'FlipLeftDownIn', 'FlipLeftDownOut', 'FlipLeftUpIn', 'FlipLeftUpOut',
'FlipRightDownIn', 'FlipRightDownOut', 'SlideBottomIn', 'SlideBottomOut',
'ZoomIn', 'ZoomOut'];
    public toastCreated(): void {
        this.toastShow();
    }
    public toastShow() {
        setTimeout(
            () => {
                this.toastInstance.show();
            }, this.timeOutDelay);
    }
    public btnClick(): void {
        this.toastShow()
    }
    public valueChange(): void {
        this.toastInstance.animation.show = { effect:
this.dropDownShowInstance.value as Effect };
    }
    public valueChangeHide(): void {
        this.toastInstance.animation.hide = { effect:
this.dropDownHideInstance.value as Effect };
    }
    public render() {
        return (
            <div>
                <ButtonComponent cssClass="e-primary" onClick={this.btnClick =
this.btnClick.bind(this)}> Show Toast</ButtonComponent>
                <div className="e-row" style={this.style}>
                    <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
                        <label>Show Animation</label>
                    </div>
                    <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">

```

```

        <DropDownListComponent ref={drop => this.dropDownShowInstance =
drop!} dataSource={this.AnimationDB} placeholder="Select a ProgressBar Color"
change={this.valueChange = this.valueChange.bind(this)} index={0} />
      </div>
    </div>
    <div className='e-row' style={this.style}>
      <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
        <label>Hide Animation</label>
      </div>
      <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
        <DropDownListComponent ref={drop => this.dropDownHideInstance =
drop!} dataSource={this.AnimationDB} placeholder="Select a ProgressBar Color"
change={this.valueChangeHide = this.valueChangeHide.bind(this)} index={0} />
      </div>
    </div>
    <ToastComponent ref={toast => this.toastInstance = toast!}
title='Matt sent you a friend request' content='Hey, wanna dress up as
wizards and ride our hoverboards?' position={this.position}
animation={this.animation} showProgressBar='true' created={this.toastCreated
= this.toastCreated.bind(this)} />
  </div>
);
}
};
export default App;

```

## [Functional-component]

### APP.JSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
function App() {
  let toastInstance;
  let timeoutDelay = 600;
  let position = { X: 'Right', Y: 'Bottom' };
  let animation;
  let style = { paddingTop: '20px' };
  let dropDownShowInstance;
  let dropDownHideInstance;
  let AnimationDB = ['FadeIn', 'Fadeout', 'FadeZoomIn', 'FadeZoomOut',
'FlipLeftDownIn', 'FlipLeftDownOut', 'FlipLeftUpIn', 'FlipLeftUpOut',
'FlipRightDownIn', 'FlipRightDownOut', 'SlideBottomIn', 'SlideBottomOut',
'ZoomIn', 'ZoomOut'];
  function toastCreated() {
    toastShow();
  }
  function toastShow() {
    setTimeout(() => {
      toastInstance.show();
    }, timeoutDelay);
  }
  function btnClick() {
    toastShow();
  }
}

```

```

    }
    function valueChange() {
        toastInstance.animation.show = { effect: dropDownHideInstance.value };
    };
    }
    function valueChangeHide() {
        toastInstance.animation.hide = { effect: dropDownHideInstance.value };
    };
    }
    return (<div>
        <ButtonComponent className="e-primary" onClick={btnClick.bind(this)}>
Show Toast</ButtonComponent>
        <div className='e-row' style={style}>
            <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
                <label>Show Animation</label>
            </div>
            <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
                <DropDownListComponent ref={drop => dropDownShowInstance = drop}
dataSource={AnimationDB} placeholder="Select a ProgressBar Color"
change={valueChange.bind(this)} index={0}/>
            </div>
            </div>
            <div className='e-row' style={style}>
                <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
                    <label>Hide Animation</label>
                </div>
                <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
                    <DropDownListComponent ref={drop => dropDownHideInstance = drop}
dataSource={AnimationDB} placeholder="Select a ProgressBar Color"
change={valueChangeHide.bind(this)} index={0}/>
                </div>
                </div>
                <ToastComponent ref={toast => toastInstance = toast} title='Matt sent
you a friend request' content='Hey, wanna dress up as wizards and ride our
hoverboards?' position={position} showProgressBar={true}
created={toastCreated.bind(this)} animation={animation}/>
            </div>);
    }
    ;
export default App;

```

**APP.TSX**

```

import { Effect } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
function App() {
    let toastInstance: ToastComponent;
    let timeoutDelay: number = 600;
    let position = { X: 'Right', Y: 'Bottom' };
    let animation: { show: { effect: 'SlideRightIn' }, hide: { effect:
'SlideLeftOut' } };
    let style = { paddingTop: '20px' };
    let dropDownShowInstance: DropDownListComponent;

```

```

    let dropDownHideInstance: DropDownListComponent;
    let AnimationDB = ['FadeIn', 'Fadeout', 'FadeZoomIn', 'FadeZoomOut',
'FlipLeftDownIn', 'FlipLeftDownOut', 'FlipLeftUpIn', 'FlipLeftUpOut',
'FlipRightDownIn', 'FlipRightDownOut', 'SlideBottomIn', 'SlideBottomOut',
'ZoomIn', 'ZoomOut'];

    function toastCreated(): void {
        toastShow();
    }

    function toastShow() {
        setTimeout(
            () => {
                toastInstance.show();
            }, timeOutDelay);
    }

    function btnClick(): void {
        toastShow()
    }

    function valueChange(): void {
        toastInstance.animation.show = { effect: dropDownHideInstance.value as
any}
    }

    function valueChangeHide(): void {
        toastInstance.animation.hide = { effect: dropDownHideInstance.value as
any };
    }

    return (
        <div>
            <ButtonComponent cssClass="e-primary" onClick={btnClick.bind(this)}>
Show Toast</ButtonComponent>
            <div className='e-row' style={style}>
                <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
                    <label>Show Animation</label>
                </div>
                <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
                    <DropDownListComponent ref={drop => dropDownShowInstance = drop!}
dataSource={AnimationDB} placeholder="Select a ProgressBar Color"
change={valueChange.bind(this)} index={0} />
                </div>
            </div>
            <div className='e-row' style={style}>
                <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
                    <label>Hide Animation</label>
                </div>
                <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
                    <DropDownListComponent ref={drop => dropDownHideInstance = drop!}
dataSource={AnimationDB} placeholder="Select a ProgressBar Color"
change={valueChangeHide.bind(this)} index={0} />
                </div>
            </div>
            <ToastComponent ref={toast => toastInstance = toast!} title='Matt
sent you a friend request' content='Hey, wanna dress up as wizards and ride

```



```

our hoverboards?' position={position} showProgressBar={true}
created={toastCreated.bind(this)} animation={animation} />
    </div>
    );
};
export default App;

```

### Toast services in React Toast component

The Toast component provides a built-in utility function to render the toast with minimal code. The utility function will render the toast without the need of rendering the container element in the DOM where the toast is appended. So that, the toast can now be rendered on the go. The following are the option to render the toast using the utility function.

#### Show Toast with predefined types

The Toast component support 4 types of predefined toast with essential colors for various situations which can be shown using the `ToastUtility.show` by just defining the type of the toast without defining any class names. The following options are used as an argument on calling the utility function for predefined types:

Options	Description
-----	-----
<a href="#">content</a>	Specifies the content that can be displayed on the Toast.
type	Specifies the type of the predefined Toasts. The 4 types of predefined toasts are <b>Information</b> , <b>Success</b> , <b>Error</b> , <b>Warning</b>
<a href="#">timeOut</a>	Specifies the Toast display time duration on the page in milliseconds. Once the time expires, Toast message will be removed. Setting 0 as a time out value displays the Toast on the page until the user closes it manually.

[Class-component]

#### APP.JSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastUtility } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
const toastObj;
class App extends React.Component {
    infoShow() {
        toastObj = ToastUtility.show('Please read the comments carefully',
        'Information', 20000);
    }
    successShow() {
        toastObj = ToastUtility.show('Your message has been sent
        successfully', 'Success', 20000);
    }
    warningShow() {
        toastObj = ToastUtility.show('There was a problem with your network
        connection', 'Warning', 20000);
    }
    dangerShow() {
        toastObj = ToastUtility.show('A problem has been occurred while
        submitting the data', 'Error', 20000);
    }
}

```

```

    }
    hideToast() {
        toastObj.hide('All');
    }
    render() {
        return (<div>
            <div>
                <ButtonComponent cssClass="e-btn e-control e-info"
onClick={this.infoShow = this.infoShow.bind(this)}> Info Message
            </ButtonComponent>
                <ButtonComponent cssClass="e-btn e-control e-success"
onClick={this.successShow = this.successShow.bind(this)}> Success Message
            </ButtonComponent>
                <ButtonComponent cssClass="e-btn e-control e-warning"
onClick={this.warningShow = this.warningShow.bind(this)}> Warning Message
            </ButtonComponent>
                <ButtonComponent cssClass="e-btn e-control e-danger"
onClick={this.dangerShow = this.dangerShow.bind(this)}> Danger Message
            </ButtonComponent>
            </div>
            <br />
            <div className="toastUtility">
                <ButtonComponent cssClass="e-btn e-control" onClick={this.hideToast
= this.hideToast.bind(this)}> Hide All </ButtonComponent>
            </div>
        </div>);
    }
}
;
export default App;

```

**APP.TSX**

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent, ToastUtility } from '@syncfusion/ej2-react-
notifications';
import * as React from "react";
const toastObj;
class App extends React.Component<{}, {}> {
    public infoShow() {
        toastObj = ToastUtility.show('Please read the comments carefully',
'Information', 20000);
    }
    public successShow() {
        toastObj = ToastUtility.show('Your message has been sent successfully',
'Success', 20000);
    }
    public warningShow() {
        toastObj = ToastUtility.show('There was a problem with your network
connection', 'Warning', 20000);
    }
    public dangerShow() {
        toastObj = ToastUtility.show('A problem has been occurred while
submitting the data', 'Error', 20000);
    }
    public hideToast() {

```

```

    toastObj.hide('All');
  }
  public render() {
    return (
      <div>
        <div>
          <ButtonComponent cssClass="e-btn e-control e-info"
            onClick={this.infoShow = this.infoShow.bind(this)}> Info Message
          </ButtonComponent>
          <ButtonComponent cssClass="e-btn e-control e-success"
            onClick={this.successShow = this.successShow.bind(this)}> Success Message
          </ButtonComponent>
          <ButtonComponent cssClass="e-btn e-control e-warning"
            onClick={this.warningShow = this.warningShow.bind(this)}> Warning Message
          </ButtonComponent>
          <ButtonComponent cssClass="e-btn e-control e-danger"
            onClick={this.dangerShow = this.dangerShow.bind(this)}> Danger Message
          </ButtonComponent>
        </div>
        <br/>
        <div className="toastUtility">
          <ButtonComponent cssClass="e-btn e-control" onClick={this.hideToast =
            this.hideToast.bind(this)}> Hide All </ButtonComponent>
        </div>
      </div>
    );
  }
};
export default App;

```

### [Functional-component]

#### APP.JSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastUtility } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
let toastObj;
function App() {
  function infoShow() {
    toastObj = ToastUtility.show('Please read the comments carefully',
    'Information', 20000);
  }
  function successShow() {
    toastObj = ToastUtility.show('Your message has been sent
    successfully', 'Success', 20000);
  }
  function warningShow() {
    toastObj = ToastUtility.show('There was a problem with your network
    connection', 'Warning', 20000);
  }
  function dangerShow() {
    toastObj = ToastUtility.show('A problem has been occurred while
    submitting the data', 'Error', 20000);
  }
  function hideToast() {

```

```

        toastObj.hide('All');
    }
    return (<div>
        <div>
            <ButtonComponent cssClass="e-btn e-control e-info"
onClick={infoShow.bind(this)}> Info Message </ButtonComponent>
            <ButtonComponent cssClass="e-btn e-control e-success"
onClick={successShow.bind(this)}> Success Message </ButtonComponent>
            <ButtonComponent cssClass="e-btn e-control e-warning"
onClick={warningShow.bind(this)}> Warning Message </ButtonComponent>
            <ButtonComponent cssClass="e-btn e-control e-danger"
onClick={dangerShow.bind(this)}> Danger Message </ButtonComponent>
        </div>
        <br />
        <div className="toastUtility">
            <ButtonComponent cssClass="e-btn e-control"
onClick={hideToast.bind(this)}> Hide All </ButtonComponent>
        </div>
    </div>);
}
export default App;

```

### APP.TSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent, ToastUtility } from '@syncfusion/ej2-react-
notifications';
import * as React from "react";
let toastObj;
function App() {
    function infoShow() {
        toastObj = ToastUtility.show('Please read the comments carefully',
'Information', 20000);
    }
    function successShow() {
        toastObj = ToastUtility.show('Your message has been sent successfully',
'Success', 20000);
    }
    function warningShow() {
        toastObj = ToastUtility.show('There was a problem with your network
connection', 'Warning', 20000);
    }
    function dangerShow() {
        toastObj = ToastUtility.show('A problem has been occurred while
submitting the data', 'Error', 20000);
    }
    function hideToast() {
        toastObj.hide('All');
    }
    return (
        <div>
            <div>
                <ButtonComponent cssClass="e-btn e-control e-info"
onClick={infoShow.bind(this)}> Info Message </ButtonComponent>
                <ButtonComponent cssClass="e-btn e-control e-success"
onClick={successShow.bind(this)}> Success Message </ButtonComponent>

```

```

        <ButtonComponent cssClass="e-btn e-control e-warning"
onOnClick={warningShow.bind(this)}> Warning Message </ButtonComponent>
        <ButtonComponent cssClass="e-btn e-control e-danger"
onOnClick={dangerShow.bind(this)}> Danger Message </ButtonComponent>
    </div>
    <br/>
    <div className="toastUtility">
        <ButtonComponent cssClass="e-btn e-control"
onOnClick={hideToast.bind(this)}> Hide All </ButtonComponent>
    </div>
</div>
);
}
export default App;

```

### Show Toast with ToastModel

The utility function can be called using the [ToastModel](#) as argument to show the toast where all the properties in the `ToastModel` like any events, position, close icon, action buttons, etc. can be used in the `ToastUtility.show`.

[Class-component]

### APP.JSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastUtility } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
const toastObj;
class App extends React.Component {
    toastShow() {
        toastObj = ToastUtility.show({
            title: 'Toast Title',
            content: 'Toast shown using utility function with ToastModel',
            timeout: 20000,
            position: { X: 'Right', Y: 'Bottom' },
            showCloseButton: true,
            click: this.toastClick.bind(this),
            buttons: [{
                model: { content: 'Close' }, click:
this.toastClose.bind(this)
            }]
        });
    }
    toastClick() {
        console.log('Toast click event triggered');
    }
    toastClose() {
        toastObj.hide('All');
    }
    render() {
        return (<div>
            <div className="toastUtility">
                <ButtonComponent cssClass="e-btn e-control" onOnClick={this.hideToast
= this.toastShow.bind(this)}> Show Toast </ButtonComponent>
            </div>
        </div>);
    }
}

```

```

    }
  }
;
export default App;

```

### APP.TSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent, ToastUtility } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
const toastObj;
class App extends React.Component<{}>, {}> {
  public toastShow() {
    toastObj = ToastUtility.show({
      title: 'Toast Title',
      content: 'Toast shown using utility function with ToastModel',
      timeout: 20000,
      position: { X: 'Right', Y: 'Bottom' },
      showCloseButton: true,
      click: this.toastClick.bind(this),
      buttons: [{
        model: { content: 'Close' }, click: this.toastClose.bind(this)
      }]
    });
  }
  public toastClick() {
    console.log('Toast click event triggered');
  }
  public toastClose() {
    toastObj.hide('All');
  }
  public render() {
    return (
      <div>
        <div className="toastUtility">
          <ButtonComponent cssClass="e-btn e-control" onClick={this.hideToast} = this.toastShow.bind(this)>> Show Toast </ButtonComponent>
        </div>
      </div>
    );
  }
};
export default App;

```

### [Functional-component]

### APP.JSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastUtility } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
let toastObj;
function App() {
  function toastShow() {
    toastObj = ToastUtility.show({

```

```

        title: 'Toast Title',
        content: 'Toast shown using utility function with ToastModel',
        timeout: 20000,
        position: { X: 'Right', Y: 'Bottom' },
        showCloseButton: true,
        click: toastClick.bind(this),
        buttons: [{
            model: { content: 'Close' }, click: toastClose.bind(this)
        }]
    });
}
function toastClick() {
    console.log('Toast click event triggered');
}
function toastClose() {
    toastObj.hide('All');
}
return (<div>
    <div className="toastUtility">
        <ButtonComponent cssClass="e-btn e-control"
onClick={toastShow.bind(this)}> Show Toast </ButtonComponent>
    </div>
</div>);
}
export default App;

```

**APP.TSX**

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent, ToastUtility } from '@syncfusion/ej2-react-
notifications';
import * as React from "react";
let toastObj;
function App() {
    function toastShow() {
        toastObj = ToastUtility.show({
            title: 'Toast Title',
            content: 'Toast shown using utility function with ToastModel',
            timeout: 20000,
            position: { X: 'Right', Y: 'Bottom' },
            showCloseButton: true,
            click: toastClick.bind(this),
            buttons: [{
                model: { content: 'Close' }, click: toastClose.bind(this)
            }]
        });
    }
    function toastClick() {
        console.log('Toast click event triggered');
    }
    function toastClose() {
        toastObj.hide('All');
    }
    return (
        <div>
            <div className="toastUtility">

```

```

    <ButtonComponent cssClass="e-btn e-control"
    onClick={toastShow.bind(this)}> Show Toast </ButtonComponent>
    </div>
  </div>
);
}
export default App;

```

## Accessibility in React Toast component

The Toast component followed the accessibility guidelines and standards, including [ADA, Section 508](#), [WCAG 2.2](#) standards, and [WCAG roles](#) that are commonly used to evaluate accessibility.

The accessibility compliance for the Toast component is outlined below.

| Accessibility Criteria | Compatibility |

| -- | -- |

| [WCAG 2.2 Support](#) |  |

| [Section 508 Support](#) |  |

| [Screen Reader Support](#) |  |

| [Right-To-Left Support](#) |  |

| [Color Contrast](#) |  |

| [Mobile Device Support](#) |  |

| [Keyboard Navigation Support](#) |  |

| [Accessibility Checker Validation](#) |  |

| [Axe-core Accessibility Validation](#) |  |

<style>

.post .post-content img {

display: inline-block;

margin: 0.5em 0;

}

</style>

<div> - All features of the component meet the requirement.</div>





```
    );  
  }  
};  
export default App;
```

### [Functional-component]

#### APP.JSX

```
import { ToastComponent } from '@syncfusion/ej2-react-notifications';  
import * as React from "react";  
function App() {  
  let toastInstance;  
  function toastCreated() {  
    toastInstance.show();  
  }  
  return (<ToastComponent ref={toast => toastInstance = toast}  
title="Sample Toast Title" content="Sample Toast Content"  
created={toastCreated.bind(this)} />);  
}  
;  
export default App;
```

#### APP.TSX

```
import { ToastComponent } from '@syncfusion/ej2-react-notifications';  
import * as React from "react";  
function App() {  
  let toastInstance:ToastComponent;  
  function toastCreated() {  
    toastInstance.show();  
  }  
  return (  
    <ToastComponent ref={ toast => toastInstance = toast!} title="Sample  
Toast Title" content="Sample Toast Content" created={toastCreated.bind(this)}  
/>  
  );  
};  
export default App;
```

### Ensuring accessibility

The Toast component's accessibility levels are ensured through an [accessibility-checker](#) and [axe-core](#) software tools during automated testing.

The accessibility compliance of the Toast component is shown in the following sample. Open the [sample](#) in a new window to evaluate the accessibility of the Toast component with accessibility tools.

See also

- [Accessibility in Syncfusion React components](#)

## How To

### Prevent duplicate toast display in React Toast component

You can prevent identical same Toast displaying in a screen by event function. You can terminate the Toast displaying process by setting cancel event property in [beforeOpen](#) Event.

Here below sample demonstrates preventing duplicate title Toast element displaying, with custom code blocks.

#### [Class-component]

#### APP.JSX

```
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import './App.css';
class App extends React.Component {
    toastInstance;
    toasts = [
        { title: 'Warning !', content: 'There was a problem with your network connection.', isOpen: false },
        { title: 'Success !', content: 'Your message has been sent successfully.', isOpen: false },
        { title: 'Error !', content: 'A problem has been occurred while submitting your data.', isOpen: false }
    ];
    toastFlag = 0;
    timeOutDelay = 600;
    position = { X: 'Center' };
    toastCreated() {
        this.toastInstance.show(this.toasts[this.toastFlag]);
        ++this.toastFlag;
    }
    onClose(e) {
        for (let i = 0; i < this.toasts.length; i++) {
            if (this.toasts[i].title === e.options.title) {
                this.toasts[i].isOpen = false;
            }
        }
    }
    toastShow() {
        setTimeout(() => {
            this.toastInstance.show(this.toasts[this.toastFlag]);
            ++this.toastFlag;
            if (this.toastFlag === (this.toasts.length)) {
                this.toastFlag = 0;
            }
        }, this.timeOutDelay);
    }
    btnClick() {
        this.toastShow();
    }
    onBeforeOpen(e) {
        if (this.preventDuplicate(e)) {
            e.cancel = true;
        }
    }
}
```

```

    }
    preventDuplicate(e) {
        for (let i = 0; i < this.toasts.length; i++) {
            if (this.toasts[i].title === e.options.title &&
!this.toasts[i].isOpen) {
                this.toasts[i].isOpen = true;
                return false;
            }
        }
        return true;
    }
    render() {
        return (<div>
            <ButtonComponent cssClass="e-primary" onClick={this.btnClick =
this.btnClick.bind(this)}> Show Toast </ButtonComponent>
            <ToastComponent ref={toast => this.toastInstance = toast}
position={this.position} beforeOpen={this.onBeforeOpen =
this.onBeforeOpen.bind(this)} created={this.toastCreated =
this.toastCreated.bind(this)} close={this.onClose =
this.onClose.bind(this)}>/>
        </div>);
    }
}
;
export default App;

```

### APP.TSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastBeforeOpenArgs, ToastComponent } from '@syncfusion/ej2-react-
notifications';
import * as React from "react";
import './App.css';
class App extends React.Component<{}, {}> {
    public toastInstance: ToastComponent;
    public toasts = [
        { title: 'Warning !', content: 'There was a problem with your network
connection.', isOpen: false },
        { title: 'Success !', content: 'Your message has been sent
successfully.', isOpen: false },
        { title: 'Error !', content: 'A problem has been occurred while
submitting your data.', isOpen: false }
    ];
    public toastFlag: number = 0;
    public timeOutDelay: number = 600;
    public position = { X: 'Center' };
    public toastCreated(): void {
        this.toastInstance.show(this.toasts[this.toastFlag]);
        ++this.toastFlag;
    }
    public onClose(e: any): void {
        for (let i: number = 0; i < this.toasts.length; i++) {
            if (this.toasts[i].title === e.options.title) {
                this.toasts[i].isOpen = false;
            }
        }
    }
}

```

```

    }
    public toastShow() {
      setTimeout(
        () => {
          this.toastInstance.show(this.toasts[this.toastFlag]);
          ++this.toastFlag;
          if (this.toastFlag === (this.toasts.length)) {
            this.toastFlag = 0;
          }
        }, this.timeOutDelay);
    }
    public btnClick(): void {
      this.toastShow()
    }
    public onBeforeOpen(e: ToastBeforeOpenArgs): void {
      if (this.preventDuplicate(e)) {
        e.cancel = true;
      }
    }
    public preventDuplicate(e: ToastBeforeOpenArgs): boolean {
      for (let i: number = 0; i < this.toasts.length; i++) {
        if (this.toasts[i].title === e.options.title && !this.toasts[i].isOpen)
        {
          this.toasts[i].isOpen = true;
          return false;
        }
      }
      return true;
    }
    public render() {
      return (
        <div>
          <ButtonComponent cssClass="e-primary" onClick={this.btnClick =
this.btnClick.bind(this)}> Show Toast </ButtonComponent>
          <ToastComponent ref={toast => this.toastInstance = toast!}
position={this.position} beforeOpen={this.onBeforeOpen =
this.onBeforeOpen.bind(this)} created={this.toastCreated =
this.toastCreated.bind(this)} close={this.onClose = this.onClose.bind(this)}
/>
        </div>
      );
    }
  };
export default App;

```

### [Functional-component]

#### APP.JSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import './App.css';
function App() {
  let toastInstance;
  let toasts = [

```

```

    { title: 'Warning !', content: 'There was a problem with your network
connection.', isOpen: false },
    { title: 'Success !', content: 'Your message has been sent
successfully.', isOpen: false },
    { title: 'Error !', content: 'A problem has been occurred while
submitting your data.', isOpen: false }
  ];
  let toastFlag = 0;
  let timeOutDelay = 600;
  let position = { X: 'Center' };
  function toastCreated() {
    toastInstance.show(toasts[toastFlag]);
    ++toastFlag;
  }
  function onClose(e) {
    for (let i = 0; i < toasts.length; i++) {
      if (toasts[i].title === e.options.title) {
        toasts[i].isOpen = false;
      }
    }
  }
  function toastShow() {
    setTimeout(() => {
      toastInstance.show(toasts[toastFlag]);
      ++toastFlag;
      if (toastFlag === (toasts.length)) {
        toastFlag = 0;
      }
    }, timeOutDelay);
  }
  function btnClick() {
    toastShow();
  }
  function onBeforeOpen(e) {
    if (preventDuplicate(e)) {
      e.cancel = true;
    }
  }
  function preventDuplicate(e) {
    for (let i = 0; i < toasts.length; i++) {
      if (toasts[i].title === e.options.title && !toasts[i].isOpen) {
        toasts[i].isOpen = true;
        return false;
      }
    }
    return true;
  }
  return (<div>
    <ButtonComponent cssClass="e-primary" onClick={btnClick.bind(this)}>
Show Toast </ButtonComponent>
    <ToastComponent ref={toast => toastInstance = toast}
position={position} beforeOpen={onBeforeOpen.bind(this)}
created={toastCreated.bind(this)} close={onClose.bind(this)} />
    </div>);
}
;
export default App;

```

**APP.TSX**

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastBeforeOpenArgs, ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import './App.css';
function App() {
    let toastInstance: ToastComponent;
    let toasts = [
        { title: 'Warning !', content: 'There was a problem with your network connection.', isOpen: false },
        { title: 'Success !', content: 'Your message has been sent successfully.', isOpen: false },
        { title: 'Error !', content: 'A problem has been occurred while submitting your data.', isOpen: false }
    ];
    let toastFlag: number = 0;
    let timeOutDelay: number = 600;
    let position = { X: 'Center' };
    function toastCreated(): void {
        toastInstance.show(toasts[toastFlag]);
        ++toastFlag;
    }
    function onClose(e: any): void {
        for (let i: number = 0; i < toasts.length; i++) {
            if (toasts[i].title === e.options.title) {
                toasts[i].isOpen = false;
            }
        }
    }
    function toastShow() {
        setTimeout(
            () => {
                toastInstance.show(toasts[toastFlag]);
                ++toastFlag;
                if (toastFlag === (toasts.length)) {
                    toastFlag = 0;
                }
            }, timeOutDelay);
    }
    function btnClick(): void {
        toastShow()
    }
    function onBeforeOpen(e: ToastBeforeOpenArgs): void {
        if (preventDuplicate(e)) {
            e.cancel = true;
        }
    }
    function preventDuplicate(e: ToastBeforeOpenArgs): boolean {
        for (let i: number = 0; i < toasts.length; i++) {
            if (toasts[i].title === e.options.title && !toasts[i].isOpen) {
                toasts[i].isOpen = true;
                return false;
            }
        }
    }
}

```

```

    }
    return true;
  }
  return (
    <div>
      <ButtonComponent cssClass="e-primary" onClick={btnClick.bind(this)}>
Show Toast </ButtonComponent>
      <ToastComponent ref={toast => toastInstance = toast!}
position={position} beforeOpen={onBeforeOpen.bind(this)}
created={toastCreated.bind(this)} close={onClose.bind(this)} />
    </div>
  );
};
export default App;

```

### Restrict the maximum toast to show in React Toast component

You can restrict the maximum toast count by event callback function. You can terminate the toast displaying process by setting boolean as true to cancel, the argument which is obtained from [beforeOpen](#) Event.

You can restrict by your own count with custom code blocks. The below sample demonstrates that, how to restrict toast displaying up to 3.

#### [Class-component]

##### APP.JSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import './App.css';
class App extends React.Component {
  toastInstance;
  toasts = [
    { title: 'Warning !', content: 'There was a problem with your network connection.' },
    { title: 'Success !', content: 'Your message has been sent successfully.' },
    { title: 'Error !', content: 'A problem has been occurred while submitting your data.' },
  ];
  toastFlag = 0;
  maxCount = 3;
  timeoutDelay = 600;
  position = { X: 'Right', Y: 'Bottom' };
  toastCreated() {
    this.toastInstance.show(this.toasts[this.toastFlag]);
    ++this.toastFlag;
  }
  toastShow() {
    setTimeout(() => {
      this.toastInstance.show(this.toasts[this.toastFlag]);
      ++this.toastFlag;
      if (this.toastFlag === (this.toasts.length)) {
        this.toastFlag = 0;
      }
    });
  }
}

```



```

    }, this.timeOutDelay);
  }
  btnClick() {
    this.toastShow();
  }
  onBeforeOpen(e) {
    if (this.maxCount === this.toastInstance.element.childElementCount) {
      e.cancel = true;
    }
    else {
      e.cancel = false;
    }
  }
  render() {
    return (<div>
      <ButtonComponent cssClass="e-primary" onClick={this.btnClick =
this.btnClick.bind(this)}> Show Toast </ButtonComponent>
      <ToastComponent ref={toast => this.toastInstance = toast}
position={this.position} beforeOpen={this.onBeforeOpen =
this.onBeforeOpen.bind(this)} created={this.toastCreated =
this.toastCreated.bind(this)}>/>
    </div>);
  }
}
;
export default App;

```

### APP.TSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastBeforeOpenArgs, ToastComponent } from '@syncfusion/ej2-react-
notifications';
import * as React from "react";
import './App.css';
class App extends React.Component<{}, {}> {
  public toastInstance: ToastComponent;
  public toasts = [
    { title: 'Warning !', content: 'There was a problem with your network
connection.' },
    { title: 'Success !', content: 'Your message has been sent successfully.'
},
    { title: 'Error !', content: 'A problem has been occurred while
submitting your data.' },
  ];
  public toastFlag: number = 0;
  public maxCount: number = 3;
  public timeOutDelay: number = 600;
  public position = { X: 'Right', Y: 'Bottom' };
  public toastCreated(): void {
    this.toastInstance.show(this.toasts[this.toastFlag]);
    ++this.toastFlag;
  }
  public toastShow() {
    setTimeout(
      () => {
        this.toastInstance.show(this.toasts[this.toastFlag]);
        ++this.toastFlag;
      }
    );
  }
}

```

```

        if (this.toastFlag === (this.toasts.length)) {
            this.toastFlag = 0;
        }
    }, this.timeOutDelay);
}
public btnClick(): void {
    this.toastShow()
}
public onBeforeOpen(e: ToastBeforeOpenArgs): void {
    if (this.maxCount === this.toastInstance.element.childElementCount) {
        e.cancel = true;
    } else {
        e.cancel = false;
    }
}
public render() {
    return (
        <div>
            <ButtonComponent cssClass="e-primary" onClick={this.btnClick =
this.btnClick.bind(this)}> Show Toast </ButtonComponent>
            <ToastComponent ref={toast => this.toastInstance = toast!}
position={this.position}
beforeOpen={this.onBeforeOpen = this.onBeforeOpen.bind(this)}
created={this.toastCreated = this.toastCreated.bind(this)} />
        </div>
    );
}
};
export default App;

```

### [Functional-component]

#### APP.JSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import './App.css';
function App() {
    let toastInstance;
    let toasts = [
        { title: 'Warning !', content: 'There was a problem with your network
connection.' },
        { title: 'Success !', content: 'Your message has been sent
successfully.' },
        { title: 'Error !', content: 'A problem has been occurred while
submitting your data.' },
    ];
    let toastFlag = 0;
    let maxCount = 3;
    let timeOutDelay = 600;
    let position = { X: 'Right', Y: 'Bottom' };
    function toastCreated() {
        toastInstance.show(toasts[toastFlag]);
        ++toastFlag;
    }
}

```

```

function toastShow() {
  setTimeout(() => {
    toastInstance.show(toasts[toastFlag]);
    ++toastFlag;
    if (toastFlag === (toasts.length)) {
      toastFlag = 0;
    }
  }, timeOutDelay);
}
function btnClick() {
  toastShow();
}
function onBeforeOpen(e) {
  if (maxCount === toastInstance.element.childElementCount) {
    e.cancel = true;
  }
  else {
    e.cancel = false;
  }
}
return (<div>
  <ButtonComponent cssClass="e-primary" onClick={btnClick.bind(this)}>
Show Toast </ButtonComponent>
  <ToastComponent ref={toast => toastInstance = toast}
position={position} beforeOpen={onBeforeOpen.bind(this)}
created={toastCreated.bind(this)}>/>
</div>);
}
export default App;

```

### APP.TSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastBeforeOpenArgs, ToastComponent } from '@syncfusion/ej2-react-
notifications';
import * as React from "react";
import './App.css';
function App() {
  let toastInstance: ToastComponent;
  let toasts = [
    { title: 'Warning !', content: 'There was a problem with your network
connection.' },
    { title: 'Success !', content: 'Your message has been sent successfully.'
},
    { title: 'Error !', content: 'A problem has been occurred while
submitting your data.' },
];
let toastFlag: number = 0;
let maxCount: number = 3;
let timeOutDelay: number = 600;
let position = { X: 'Right', Y: 'Bottom' };
function toastCreated(): void {
  toastInstance.show(toasts[toastFlag]);
  ++toastFlag;
}
function toastShow() {
  setTimeout(

```

```

        () => {
            toastInstance.show(toasts[toastFlag]);
            ++toastFlag;
            if (toastFlag === (toasts.length)) {
                toastFlag = 0;
            }
        }, timeOutDelay);
    }
    function btnClick(): void {
        toastShow()
    }
    function onBeforeOpen(e: ToastBeforeOpenArgs): void {
        if (maxCount === toastInstance.element.childElementCount) {
            e.cancel = true;
        } else {
            e.cancel = false;
        }
    }
    return (
        <div>
            <ButtonComponent cssClass="e-primary" onClick={btnClick.bind(this)}>
Show Toast </ButtonComponent>
            <ToastComponent ref={toast => toastInstance = toast!}
position={position}
beforeOpen={onBeforeOpen.bind(this)} created={toastCreated.bind(this)}
/>
        </div>
    );
}
export default App;

```

### Customize progress bar theme and sizing in React Toast component

In default, the Progress bar will appear based on the theme stylings and dimensions. You can customize progress bar stylings through custom CSS or Event functions.

Here below sample demonstrates customize the progress bar Stylings using [beforeOpen](#) Event.

[Class-component]

#### APP.JSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import './App.css';
class App extends React.Component {
    toastInstance;
    timeOutDelay = 600;
    position = { X: 'Right', Y: 'Bottom' };
    style = { padding: '20px' };
    dropDownInstance;
    progressBarColor = ['Red', 'Cyan', 'Blue', 'Yellow', 'Pink'];
    toastCreated() {
        this.toastInstance.show();
    }
}

```

```

toastShow() {
  setTimeout(() => {
    this.toastInstance.show();
  }, this.timeOutDelay);
}
btnClick() {
  this.toastShow();
}
onBeforeOpen(e) {
  const progress = e.element.querySelector('.e-toast-progress');
  progress.style.height =
(document.getElementById('progressHeight').value + 'px').toString();
  progress.style.backgroundColor =
this.dropDownInstance.value.toString();
}
valueChange() {
  const progressElements =
this.toastInstance.element.querySelectorAll('.e-toast-progress');
  progressElements.forEach((ele) => {
    ele.style.backgroundColor =
this.dropDownInstance.value.toString();
  });
}
render() {
  return (<div>
    <ButtonComponent cssClass="e-primary" onClick={this.btnClick =
this.btnClick.bind(this)}> Show Toast </ButtonComponent>
    <div className="e-row" style={this.style}>
      <div className="e-float-input">
        <input type="text" className="e-input" id="progressHeight"
defaultValue={'4'} required={true}/>
        <span className="e-float-line"/>
        <label className="e-float-text">ProgressBar Height</label>
      </div>
    </div>
    <div className="e-row" style={this.style}>
      <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
        <label>Progress Bar</label>
      </div>
      <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
        <DropDownListComponent ref={drop => this.dropDownInstance = drop}
dataSource={this.progressBarColor} placeholder="Select a ProgressBar Color"
change={this.valueChange = this.valueChange.bind(this)} index={0}/>
      </div>
    </div>
    <ToastComponent ref={toast => this.toastInstance = toast}
showProgressBar={true} title="Matt sent you a friend request" content="Hey,
wanna dress up as wizards and ride our hoverboards?" position={this.position}
beforeOpen={this.onBeforeOpen = this.onBeforeOpen.bind(this)}
created={this.toastCreated = this.toastCreated.bind(this)}/>
  </div>);
}
}
;
export default App;

```

**APP.TSX**

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { ToastBeforeOpenArgs, ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import './App.css';

class App extends React.Component<{}, {}> {
    public toastInstance: ToastComponent;
    public timeoutDelay: number = 600;
    public position = { X: 'Right', Y: 'Bottom' };
    public style = { padding: '20px' };
    public dropDownInstance: DropDownListComponent;
    public progressBarColor = ['Red', 'Cyan', 'Blue', 'Yellow', 'Pink'];
    public toastCreated(): void {
        this.toastInstance.show();
    }
    public toastShow() {
        setTimeout(
            () => {
                this.toastInstance.show();
            }, this.timeoutDelay);
    }
    public btnClick(): void {
        this.toastShow()
    }
    public onBeforeOpen(e: ToastBeforeOpenArgs): void {
        const progress: HTMLElement = e.element.querySelector('.e-toast-progress') as HTMLElement;
        progress.style.height = ((document.getElementById('progressHeight') as HTMLInputElement).value + 'px').toString();
        progress.style.backgroundColor = this.dropDownInstance.value.toString();
    }
    public valueChange(): void {
        const progressElements: NodeList =
this.toastInstance.element.querySelectorAll('.e-toast-progress');
        progressElements.forEach((ele: HTMLElement) => {
            ele.style.backgroundColor = this.dropDownInstance.value.toString();
        })
    }
    public render() {
        return (
            <div>
                <ButtonComponent cssClass="e-primary" onClick={this.btnClick =
this.btnClick.bind(this)}> Show Toast </ButtonComponent>
                <div className="e-row" style={this.style}>
                    <div className="e-float-input">
                        <input type="text" className="e-input" id="progressHeight"
defaultValue={'4'} required={true} />
                        <span className="e-float-line" />
                        <label className="e-float-text">ProgressBar Height</label>
                    </div>
                </div>
                <div className="e-row" style={this.style}>
                    <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
                        <label>Progress Bar</label>
                    </div>
                </div>
            </div>
        );
    }
}

```

```

        </div>
        <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
            <DropDownListComponent ref={drop => this.dropDownInstance =
drop!} dataSource={this.progressBarColor} placeholder="Select a ProgressBar
Color" change={this.valueChange = this.valueChange.bind(this)} index={0} />
        </div>
    </div>
    <ToastComponent ref={toast => this.toastInstance = toast!}
showProgressBar={true} title='Matt sent you a friend request' content='Hey,
wanna dress up as wizards and ride our hoverboards?' position={this.position}
beforeOpen={this.onBeforeOpen = this.onBeforeOpen.bind(this)}
created={this.toastCreated = this.toastCreated.bind(this)} />
    </div>
    );
}
};
export default App;

```

## [Functional-component]

### APP.JSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import './App.css';
function App() {
    let toastInstance;
    let timeoutDelay = 600;
    let position = { X: 'Right', Y: 'Bottom' };
    let style = { paddingTop: '20px' };
    let dropDownInstance;
    let progressBarColor = ['Red', 'Cyan', 'Blue', 'Yellow', 'Pink'];
    function toastCreated() {
        toastInstance.show();
    }
    function toastShow() {
        setTimeout(() => {
            toastInstance.show();
        }, timeoutDelay);
    }
    function btnClick() {
        toastShow();
    }
    function onBeforeOpen(e) {
        const progress = e.element.querySelector('.e-toast-progress');
        progress.style.height =
(document.getElementById('progressHeight').value + 'px').toString();
        progress.style.backgroundColor = dropDownInstance.value.toString();
    }
    function valueChange() {
        const progressElements = toastInstance.element.querySelectorAll('.e-
toast-progress');
        progressElements.forEach((ele) => {
            ele.style.backgroundColor = dropDownInstance.value.toString();

```

```

    });
  }
  return (<div>
    <ButtonComponent className="e-primary" onClick={btnClick.bind(this)}>
Show Toast </ButtonComponent>
    <div className='e-row' style={style}>
      <div className='e-float-input'>
        <input type='text' className='e-input' id='progressHeight'
defaultValue={'4'} required={true}/>
        <span className='e-float-line' />
        <label className="e-float-text">ProgressBar Height</label>
      </div>
    </div>
    <div className='e-row' style={style}>
      <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
        <label>Progress Bar</label>
      </div>
      <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
        <DropDownListComponent ref={drop => dropDownInstance = drop}
dataSource={progressBarColor} placeholder="Select a ProgressBar Color"
change={valueChange.bind(this)} index={0}/>
      </div>
    </div>
    <ToastComponent ref={toast => toastInstance = toast}
showProgressBar={true} title='Matt sent you a friend request' content='Hey,
wanna dress up as wizards and ride our hoverboards?' position={position}
beforeOpen={onBeforeOpen.bind(this)} created={toastCreated.bind(this)} />
  </div>);
}
export default App;

```

### APP.TSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { ToastBeforeOpenArgs, ToastComponent } from '@syncfusion/ej2-react-
notifications';
import * as React from "react";
import './App.css';
function App() {
  let toastInstance: ToastComponent;
  let timeOutDelay: number = 600;
  let position = { X: 'Right', Y: 'Bottom' };
  let style = { paddingTop: '20px' };
  let dropDownInstance: DropDownListComponent;
  let progressBarColor = ['Red', 'Cyan', 'Blue', 'Yellow', 'Pink'];
  function toastCreated(): void {
    toastInstance.show();
  }
  function toastShow() {
    setTimeout(
      () => {
        toastInstance.show();
      }, timeOutDelay);
  }
  function btnClick(): void {

```



```

    toastShow()
  }
  function onBeforeOpen(e: ToastBeforeOpenArgs): void {
    const progress: HTMLElement = e.element.querySelector('.e-toast-progress') as HTMLElement;
    progress.style.height = ((document.getElementById('progressHeight') as HTMLInputElement).value + 'px').toString();
    progress.style.backgroundColor = dropDownInstance.value.toString();
  }
  function valueChange(): void {
    const progressElements: NodeList =
toastInstance.element.querySelectorAll('.e-toast-progress');
    progressElements.forEach((ele: HTMLElement) => {
      ele.style.backgroundColor = dropDownInstance.value.toString();
    })
  }
  return (
    <div>
      <ButtonComponent cssClass="e-primary" onClick={btnClick.bind(this)}>
Show Toast </ButtonComponent>
      <div className='e-row' style={style}>
        <div className='e-float-input'>
          <input type='text' className='e-input' id='progressHeight'
defaultValue={'4'} required={true} />
          <span className='e-float-line' />
          <label className="e-float-text">ProgressBar Height</label>
        </div>
      </div>
      <div className='e-row' style={style}>
        <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
          <label>Progress Bar</label>
        </div>
        <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
          <DropDownListComponent ref={drop => dropDownInstance = drop!}
dataSource={progressBarColor} placeholder="Select a ProgressBar Color"
change={valueChange.bind(this)} index={0} />
        </div>
      </div>
      <ToastComponent ref={toast => toastInstance = toast!}
showProgressBar={true} title='Matt sent you a friend request' content='Hey,
wanna dress up as wizards and ride our hoverboards?' position={position}
beforeOpen={onBeforeOpen.bind(this)} created={toastCreated.bind(this)} />
    </div>
  );
}
export default App;

```

### Play an audio before open the toast in React Toast component

Here below sample demonstrates to playing audio background while opening Toast. Here we have included audio play codes into beforeOpen event Function.

If you want to stop the audio after displaying Toast use [open](#) event in Toast. please check the Toast Events [api's](#) for further customization.

[Class-component]

**APP.JSX**

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import './App.css';
class App extends React.Component {
    toastInstance;
    timeOutDelay = 600;
    position = { X: 'Right', Y: 'Bottom' };
    toastCreated() {
        this.toastInstance.show();
    }
    toastShow() {
        const audio = new
Audio('https://drive.google.com/uc?export=download&id=1M95VOptolcQ4FQHzNBaLf0
WFQglrtWi7');
        audio.play();
        setTimeout(() => {
            this.toastInstance.show();
        }, this.timeOutDelay);
    }
    btnClick() {
        this.toastShow();
    }
    render() {
        return (<div>
            <ButtonComponent cssClass="e-primary" onClick={this.btnClick =
this.btnClick.bind(this)}> Show Toast </ButtonComponent>
            <ToastComponent ref={toast => this.toastInstance = toast} title='Matt
sent you a friend request' content='Hey, wanna dress up as wizards and ride
our hoverboards?' position={this.position} created={this.toastCreated =
this.toastCreated.bind(this)} />
        </div>);
    }
}
;
export default App;

```

**APP.TSX**

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import './App.css';
class App extends React.Component<{}, {}> {
    public toastInstance: ToastComponent;
    public timeOutDelay: number = 600;
    public position = { X: 'Right', Y: 'Bottom' };
    public toastCreated(): void {
        this.toastInstance.show();
    }
    public toastShow() {
        const audio: HTMLAudioElement = new
Audio('https://drive.google.com/uc?export=download&id=1M95VOptolcQ4FQHzNBaLf0
WFQglrtWi7');

```

```

        audio.play();
        setTimeout(
            () => {
                this.toastInstance.show();
            }, this.timeOutDelay);
    }
    public btnClick(): void {
        this.toastShow()
    }
    public render() {
        return (
            <div>
                <ButtonComponent cssClass="e-primary" onClick={this.btnClick =
this.btnClick.bind(this)}> Show Toast </ButtonComponent>
                <ToastComponent ref={toast => this.toastInstance = toast!}
title='Matt sent you a friend request' content='Hey, wanna dress up as
wizards and ride our hoverboards?' position={this.position}
created={this.toastCreated = this.toastCreated.bind(this)} />
            </div>
        );
    }
};
export default App;

```

## [Functional-component]

### APP.JSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import './App.css';
function App() {
    let toastInstance;
    let timeOutDelay = 600;
    let position = { X: 'Right', Y: 'Bottom' };
    function toastCreated() {
        toastInstance.show();
    }
    function toastShow() {
        const audio = new
Audio('https://drive.google.com/uc?export=download&id=1M95VOpto1cQ4FQHzNBaLf0
WFQglrtWi7');
        audio.play();
        setTimeout(() => {
            toastInstance.show();
        }, timeOutDelay);
    }
    function btnClick() {
        toastShow();
    }
    return (<div>
        <ButtonComponent cssClass="e-primary" onClick={btnClick.bind(this)}>
Show Toast </ButtonComponent>

```

```

    <ToastComponent ref={toast => toastInstance = toast} title='Matt sent
you a friend request' content='Hey, wanna dress up as wizards and ride our
hoverboards?' position={position} created={toastCreated.bind(this)} />
  </div>;
}
export default App;

```

## APP.TSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import './App.css';
function App() {
  let toastInstance: ToastComponent;
  let timeoutDelay: number = 600;
  let position = { X: 'Right', Y: 'Bottom' };
  function toastCreated(): void {
    toastInstance.show();
  }
  function toastShow() {
    const audio: HTMLAudioElement = new
Audio('https://drive.google.com/uc?export=download&id=1M95VOpto1cQ4FQHZNbALf0
WFQglrtWi7');
    audio.play();
    setTimeout(
      () => {
        toastInstance.show();
      }, timeoutDelay);
  }
  function btnClick(): void {
    toastShow()
  }
  return (
    <div>
      <ButtonComponent cssClass="e-primary" onClick={btnClick.bind(this)}>
Show Toast </ButtonComponent>
      <ToastComponent ref={toast => toastInstance = toast!} title='Matt sent
you a friend request' content='Hey, wanna dress up as wizards and ride our
hoverboards?' position={position} created={toastCreated.bind(this)} />
    </div>
  );
}
export default App;

```

### Show different types of toast in React Toast component

The Essential JS 2 Toast has the following predefined styles that can be defined using the [cssClass](#) property for achieving different types of Toast.

Class	Description
e-toast-success	Used to represent a positive Toast.
e-toast-info	Used to represent an informative Toast.

| e-toast-warning | Used to represent a Toast with caution. |

| e-toast-danger | Used to represent a negative Toast. |

[Class-component]

### APP.JSX

```
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import './App.css';
class App extends React.Component {
  toastInstance;
  toasts = [
    { title: 'Warning !', content: 'There was a problem with your network connection.', cssClass: 'e-toast-warning' },
    { title: 'Success !', content: 'Your message has been sent successfully.', cssClass: 'e-toast-success' },
    { title: 'Error !', content: 'A problem has been occurred while submitting your data.', cssClass: 'e-toast-danger' },
    { title: 'Information !', content: 'Please read the comments carefully.', cssClass: 'e-toast-info' }
  ];
  timeoutDelay = 600;
  toastFlag = 0;
  position = { X: 'Right', Y: 'Bottom' };
  toastCreated() {
    this.toastInstance.show(this.toasts[this.toastFlag]);
    ++this.toastFlag;
  }
  toastShow() {
    setTimeout(() => {
      this.toastInstance.show(this.toasts[this.toastFlag]);
      ++this.toastFlag;
      if (this.toastFlag === (this.toasts.length)) {
        this.toastFlag = 0;
      }
    }, this.timeoutDelay);
  }
  btnClick() {
    this.toastShow();
  }
  render() {
    return (
      <div>
        <ButtonComponent cssClass="e-primary" onClick={this.btnClick = this.btnClick.bind(this)}> Show Toast </ButtonComponent>
        <ToastComponent ref={toast => this.toastInstance = toast} position={this.position} created={this.toastCreated = this.toastCreated.bind(this)} />
      </div>
    );
  }
}
;
export default App;
```

### APP.TSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import './App.css';
class App extends React.Component<{}, {}> {
    public toastInstance: ToastComponent;
    public toasts = [
        { title: 'Warning !', content: 'There was a problem with your network connection.', cssClass: 'e-toast-warning' },
        { title: 'Success !', content: 'Your message has been sent successfully.', cssClass: 'e-toast-success' },
        { title: 'Error !', content: 'A problem has been occurred while submitting your data.', cssClass: 'e-toast-danger' },
        { title: 'Information !', content: 'Please read the comments carefully.', cssClass: 'e-toast-info' }
    ];
    public timeOutDelay: number = 600;
    public toastFlag: number = 0;
    public position = { X: 'Right', Y: 'Bottom' };
    public toastCreated(): void {
        this.toastInstance.show(this.toasts[this.toastFlag]);
        ++this.toastFlag;
    }
    public toastShow() {
        setTimeout(
            () => {
                this.toastInstance.show(this.toasts[this.toastFlag]);
                ++this.toastFlag;
                if (this.toastFlag === (this.toasts.length)) {
                    this.toastFlag = 0;
                }
            }, this.timeOutDelay);
    }
    public btnClick(): void {
        this.toastShow()
    }
    public render() {
        return (
            <div>
                <ButtonComponent cssClass="e-primary" onClick={this.btnClick = this.btnClick.bind(this)}> Show Toast </ButtonComponent>
                <ToastComponent ref={toast => this.toastInstance = toast!} position={this.position} created={this.toastCreated = this.toastCreated.bind(this)} />
            </div>
        );
    }
};
export default App;

```

### [Functional-component]

#### APP.JSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';

```

```

import * as React from "react";
import './App.css';
function App() {
  let toastInstance;
  let toasts = [
    { title: 'Warning !', content: 'There was a problem with your network connection.', cssClass: 'e-toast-warning' },
    { title: 'Success !', content: 'Your message has been sent successfully.', cssClass: 'e-toast-success' },
    { title: 'Error !', content: 'A problem has been occurred while submitting your data.', cssClass: 'e-toast-danger' },
    { title: 'Information !', content: 'Please read the comments carefully.', cssClass: 'e-toast-info' }
  ];
  let timeOutDelay = 600;
  let toastFlag = 0;
  let position = { X: 'Right', Y: 'Bottom' };
  function toastCreated() {
    toastInstance.show(toasts[toastFlag]);
    ++toastFlag;
  }
  function toastShow() {
    setTimeout(() => {
      toastInstance.show(toasts[toastFlag]);
      ++toastFlag;
      if (toastFlag === (toasts.length)) {
        toastFlag = 0;
      }
    }, timeOutDelay);
  }
  function btnClick() {
    toastShow();
  }
  return (<div>
    <ButtonComponent cssClass="e-primary" onClick={btnClick.bind(this)}>
    Show Toast </ButtonComponent>
    <ToastComponent ref={toast => toastInstance = toast}
    position={position} created={toastCreated.bind(this)}>/>
  </div>);
}
;
export default App;

```

### APP.TSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import './App.css';
function App(){
  let toastInstance: ToastComponent;
  let toasts = [
    { title: 'Warning !', content: 'There was a problem with your network connection.', cssClass: 'e-toast-warning' },
    { title: 'Success !', content: 'Your message has been sent successfully.', cssClass: 'e-toast-success' },

```

```

    { title: 'Error !', content: 'A problem has been occurred while
submitting your data.', cssClass: 'e-toast-danger' },
    { title: 'Information !', content: 'Please read the comments carefully.',
cssClass: 'e-toast-info' }
  ];
  let timeOutDelay: number = 600;
  let toastFlag: number = 0;
  let position = { X: 'Right', Y: 'Bottom' };
  function toastCreated(): void {
    toastInstance.show(toasts[toastFlag]);
    ++toastFlag;
  }
  function toastShow() {
    setTimeout(
      () => {
        toastInstance.show(toasts[toastFlag]);
        ++toastFlag;
        if (toastFlag === (toasts.length)) {
          toastFlag = 0;
        }
      }, timeOutDelay);
  }
  function btnClick(): void {
    toastShow()
  }
  return (
    <div>
      <ButtonComponent cssClass="e-primary" onClick={btnClick.bind(this)}>
Show Toast </ButtonComponent>
      <ToastComponent ref={toast => toastInstance = toast!}
position={position} created={toastCreated.bind(this)} />
    </div>
  );
};
export default App;

```

### Show multiple toasts in various positions in React Toast component

In default Toast position only updates once visible Toasts get destroyed. If You needs to display multiple Toasts with different position means needs to initiate another Toast for achieving this.

Here below sample demonstrates to add multiple Toasts adding in the different position.

#### [Class-component]

#### APP.JSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import './App.css';
class App extends React.Component {
  toastInstance;
  toastLeftInstance;
  timeOutDelay = 600;
  position = { X: 'Right', Y: 'Bottom' };
  leftPosition = { X: 'Right', };

```



```

toastCreated() {
    this.toastShow(this.toastInstance);
    this.toastShow(this.toastLeftInstance);
}
toastShow(toastObj) {
    setTimeout(() => {
        toastObj.show();
    }, this.timeOutDelay);
}
btnClick() {
    this.toastShow(this.toastInstance);
}
btnToastClick() {
    this.toastShow(this.toastLeftInstance);
}
onClick(e) {
    e.clickToClose = true;
}
render() {
    return (<div id='multiToast'>
        <div><ButtonComponent cssClass="e-primary" onClick={this.btnClick =
this.btnClick.bind(this)}> Show Bottom Position Toast</ButtonComponent></div>
        <div><ButtonComponent cssClass="e-primary"
onClick={this.btnToastClick = this.btnToastClick.bind(this)}> Show Top
Position Toast</ButtonComponent></div>
        <ToastComponent click={this.onClick = this.onClick.bind(this)}
ref={toast => this.toastInstance = toast} title='Warning !' content='There
was a problem with your network connection.' position={this.position}
created={this.toastCreated = this.toastCreated.bind(this)}>/>
        <ToastComponent click={this.onClick = this.onClick.bind(this)}
ref={toast => this.toastLeftInstance = toast} title='Warning !'
content='There was a problem with your network connection.'
position={this.leftPosition} created={this.toastCreated =
this.toastCreated.bind(this)}>/>
    </div>);
}
}
;
export default App;

```

### APP.TSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastClickEventArgs, ToastComponent } from '@syncfusion/ej2-react-
notifications';
import * as React from "react";
import './App.css';
class App extends React.Component<{}, {}> {
    public toastInstance: ToastComponent;
    public toastLeftInstance: ToastComponent;
    public timeOutDelay: number = 600;
    public position = { X: 'Right', Y: 'Bottom' };
    public leftPosition = { X: 'Right', };
    public toastCreated(): void {
        this.toastShow(this.toastInstance);
        this.toastShow(this.toastLeftInstance);
    }
}

```

```

    }
    public toastShow(toastObj: ToastComponent) {
      setTimeout(
        () => {
          toastObj.show();
        }, this.timeOutDelay);
    }
    public btnClick(): void {
      this.toastShow(this.toastInstance);
    }
    public btnToastClick(): void {
      this.toastShow(this.toastLeftInstance);
    }
    public onClick(e: ToastClickEventArgs): void {
      e.clickToClose = true;
    }
    public render() {
      return (
        <div id='multiToast'>
          <div><ButtonComponent cssClass="e-primary" onClick={this.btnClick =
this.btnClick.bind(this)}> Show Bottom Position Toast</ButtonComponent></div>
          <div><ButtonComponent cssClass="e-primary"
onClick={this.btnToastClick = this.btnToastClick.bind(this)}> Show Top
Position Toast</ButtonComponent></div>
          <ToastComponent click={this.onClick = this.onClick.bind(this)}
ref={toast => this.toastInstance = toast!}
title='Warning !' content='There was a problem with your network
connection.' position={this.position} created={this.toastCreated =
this.toastCreated.bind(this)} />
          <ToastComponent click={this.onClick = this.onClick.bind(this)}
ref={toast => this.toastLeftInstance = toast!}
title='Warning !' content='There was a problem with your network
connection.' position={this.leftPosition} created={this.toastCreated =
this.toastCreated.bind(this)} />
        </div>
      );
    }
  };
export default App;

```

## [Functional-component]

### APP.JSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import './App.css';
function App() {
  let toastInstance;
  let toastLeftInstance;
  let timeOutDelay = 600;
  let position = { X: 'Right', Y: 'Bottom' };
  let leftPosition = { X: 'Right', };
  function toastCreated() {
    toastShow(toastInstance);
  }

```

```

        toastShow(toastLeftInstance);
    }
    function toastShow(toastObj) {
        setTimeout(() => {
            toastObj.show();
        }, timeOutDelay);
    }
    function btnClick() {
        toastShow(toastInstance);
    }
    function btnToastClick() {
        toastShow(toastLeftInstance);
    }
    function onClick(e) {
        e.clickToClose = true;
    }
    return (<div id='multiToast'>
        <div><ButtonComponent cssClass="e-primary"
onClick={btnClick.bind(this)}> Show Bottom Position
Toast</ButtonComponent></div>
        <div><ButtonComponent cssClass="e-primary"
onClick={btnToastClick.bind(this)}> Show Top Position
Toast</ButtonComponent></div>
        <ToastComponent click={onClick.bind(this)} ref={toast => toastInstance
= toast} title='Warning !' content='There was a problem with your network
connection.' position={position} created={toastCreated.bind(this)}>/>
        <ToastComponent click={onClick.bind(this)} ref={toast =>
toastLeftInstance = toast} title='Warning !' content='There was a problem
with your network connection.' position={leftPosition}
created={toastCreated.bind(this)}>/>
    </div>);
}
export default App;

```

### APP.TSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastClickEventArgs, ToastComponent } from '@syncfusion/ej2-react-
notifications';
import * as React from "react";
import './App.css';
function App() {
    let toastInstance: ToastComponent;
    let toastLeftInstance: ToastComponent;
    let timeOutDelay: number = 600;
    let position = { X: 'Right', Y: 'Bottom' };
    let leftPosition = { X: 'Right', };
    function toastCreated(): void {
        toastShow(toastInstance);
        toastShow(toastLeftInstance);
    }
    function toastShow(toastObj: ToastComponent) {
        setTimeout(
            () => {
                toastObj.show();
            }, timeOutDelay);
    }
}

```

```

    }
    function btnClick(): void {
        toastShow(toastInstance);
    }
    function btnToastClick(): void {
        toastShow(toastLeftInstance);
    }
    function onClick(e: ToastClickEventArgs): void {
        e.clickToClose = true;
    }
    return (
        <div id='multiToast'>
            <div><ButtonComponent cssClass="e-primary"
onClick={btnClick.bind(this)}> Show Bottom Position
Toast</ButtonComponent></div>
            <div><ButtonComponent cssClass="e-primary" onClick={
btnToastClick.bind(this)}> Show Top Position Toast</ButtonComponent></div>
            <ToastComponent click={onClick.bind(this)} ref={toast => toastInstance
= toast!}
                title='Warning !' content='There was a problem with your network
connection.' position={position} created={toastCreated.bind(this)} />
            <ToastComponent click={onClick.bind(this)} ref={toast =>
toastLeftInstance = toast!}
                title='Warning !' content='There was a problem with your network
connection.' position={leftPosition} created={toastCreated.bind(this)} />
            </div>
        );
    }
}
export default App;

```

### Close the toast with click tap in React Toast component

By default, Toast gets expired based on timeout value. You can customize Toast hide with click/tap action by setting boolean value to [clickToClose](#) in click callback function with [static Toast](#).

### [Class-component]

#### APP.JSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import './App.css';
class App extends React.Component {
    toastInstance;
    timeoutDelay = 600;
    position = { X: 'Right', Y: 'Bottom' };
    toastCreated() {
        this.toastShow(this.toastInstance);
    }
    toastShow(toastObj) {
        setTimeout(() => {
            toastObj.show();
        }, this.timeoutDelay);
    }
    btnClick() {
        this.toastShow(this.toastInstance);
    }
}

```

```

    }
    onClick(e) {
        e.clickToClose = true;
    }
    render() {
        return (<div>
            <div><ButtonComponent cssClass="e-primary" onClick={this.btnClick =
this.btnClick.bind(this)}> Show Bottom Position Toast</ButtonComponent></div>
            <ToastComponent click={this.onClick = this.onClick.bind(this)}
ref={toast => this.toastInstance = toast} title='Warning !' content='There
was a problem with your network connection.' position={this.position}
created={this.toastCreated = this.toastCreated.bind(this)} />
            </div>);
    }
}
;
export default App;

```

### APP.TSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastClickEventArgs, ToastComponent } from '@syncfusion/ej2-react-
notifications';
import * as React from "react";
import './App.css';
class App extends React.Component<{}, {}> {
    public toastInstance: ToastComponent;
    public timeoutDelay: number = 600;
    public position = { X: 'Right', Y: 'Bottom' };
    public toastCreated(): void {
        this.toastShow(this.toastInstance);
    }
    public toastShow(toastObj: ToastComponent) {
        setTimeout(
            () => {
                toastObj.show();
            }, this.timeoutDelay);
    }
    public btnClick(): void {
        this.toastShow(this.toastInstance);
    }
    public onClick(e: ToastClickEventArgs): void {
        e.clickToClose = true;
    }
    public render() {
        return (
            <div>
                <div><ButtonComponent cssClass="e-primary" onClick={this.btnClick =
this.btnClick.bind(this)}> Show Bottom Position Toast</ButtonComponent></div>
                <ToastComponent click={this.onClick = this.onClick.bind(this)}
ref={toast => this.toastInstance = toast!} title='Warning !' content='There
was a problem with your network connection.'
                position={this.position} created={this.toastCreated =
this.toastCreated.bind(this)} />
            </div>
        );
    }
}

```

```

    }
  };
  export default App;

```

## [Functional-component]

### APP.JSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import './App.css';
function App() {
    let toastInstance;
    let timeOutDelay = 600;
    let position = { X: 'Right', Y: 'Bottom' };
    function toastCreated() {
        toastShow(toastInstance);
    }
    function toastShow(toastObj) {
        setTimeout(() => {
            toastObj.show();
        }, timeOutDelay);
    }
    function btnClick() {
        toastShow(toastInstance);
    }
    function onClick(e) {
        e.clickToClose = true;
    }
    return (<div>
        <div><ButtonComponent cssClass="e-primary"
        onClick={btnClick.bind(this)}> Show Bottom Position
        Toast</ButtonComponent></div>
        <ToastComponent click={onClick.bind(this)} ref={toast => toastInstance
        = toast} title='Warning !' content='There was a problem with your network
        connection.' position={position} created={toastCreated.bind(this)}/>
        </div>);
    }
  export default App;

```

### APP.TSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastClickEventArgs, ToastComponent } from '@syncfusion/ej2-react-
notifications';
import * as React from "react";
import './App.css';
function App() {
    let toastInstance: ToastComponent;
    let timeOutDelay: number = 600;
    let position = { X: 'Right', Y: 'Bottom' };
    function toastCreated(): void {
        toastShow(toastInstance);
    }
    function toastShow(toastObj: ToastComponent) {

```

```

        setTimeout(
            () => {
                toastObj.show();
            }, timeOutDelay);
    }
    function btnClick(): void {
        toastShow(toastInstance);
    }
    function onClick(e: ToastClickEventArgs): void {
        e.clickToClose = true;
    }
    return (
        <div>
            <div><ButtonComponent cssClass="e-primary"
onClick={btnClick.bind(this)}> Show Bottom Position
Toast</ButtonComponent></div>
            <ToastComponent click={onClick.bind(this)} ref={toast => toastInstance
= toast!} title='Warning !' content='There was a problem with your network
connection.'
                position={position} created={toastCreated.bind(this)} />
            </div>
        );
    }
}
export default App;

```

### Add dynamic template in React Toast component

Toast provides the support to change its templates dynamically, So that you can update templates to multiple Toasts. We can change Toast properties while calling [show](#) method.

#### [Class-component]

#### APP.JSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import './App.css';
class App extends React.Component {
    toastInstance;
    toasts = [
        { template: '2 Mail has received' },
        { template: 'User Guest Logged in' },
        { template: 'Logging in as Guest' },
        { template: 'Ticket has reserved' },
        { template: '#templateToast' }
    ];
    toastFlag = 0;
    maxCount = 3;
    timeOutDelay = 600;
    position = { X: 'Right', Y: 'Bottom' };
    toastCreated() {
        this.toastInstance.show(this.toasts[this.toastFlag]);
        ++this.toastFlag;
    }
    toastShow() {
        setTimeout(() => {

```

```

        this.toastInstance.show(this.toasts[this.toastFlag]);
        ++this.toastFlag;
        if (this.toastFlag === (this.toasts.length)) {
            this.toastFlag = 0;
        }
    }, this.timeOutDelay);
}
btnClick() {
    this.toastShow();
}
onClick(e) {
    e.clickToClose = true;
}
render() {
    return (<div>
        <div><ButtonComponent cssClass="e-primary" onClick={this.btnClick =
this.btnClick.bind(this)}> Show Bottom Position Toast</ButtonComponent></div>
        <ToastComponent click={this.onClick = this.onClick.bind(this)}
ref={toast => this.toastInstance = toast} position={this.position}
created={this.toastCreated = this.toastCreated.bind(this)} />
        </div>);
}
}
;
export default App;

```

### APP.TSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastClickEventArgs, ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import './App.css';
class App extends React.Component<{}, {}> {
    public toastInstance: ToastComponent;
    public toasts = [
        { template: '2 Mail has received' },
        { template: 'User Guest Logged in' },
        { template: 'Logging in as Guest' },
        { template: 'Ticket has reserved' },
        { template: '#templateToast' }
    ];
    public toastFlag: number = 0;
    public maxCount: number = 3;
    public timeOutDelay: number = 600;
    public position = { X: 'Right', Y: 'Bottom' };
    public toastCreated(): void {
        this.toastInstance.show(this.toasts[this.toastFlag]);
        ++this.toastFlag;
    }
    public toastShow() {
        setTimeout(
            () => {
                this.toastInstance.show(this.toasts[this.toastFlag]);
                ++this.toastFlag;
                if (this.toastFlag === (this.toasts.length)) {

```



```

        this.toastFlag = 0;
    }
    }, this.timeOutDelay);
}
public btnClick(): void {
    this.toastShow();
}
public onClick(e: ToastClickEventArgs): void {
    e.clickToClose = true;
}
public render() {
    return (
        <div>
            <div><ButtonComponent cssClass="e-primary" onClick={this.btnClick =
this.btnClick.bind(this)}> Show Bottom Position Toast</ButtonComponent></div>
            <ToastComponent click={this.onClick = this.onClick.bind(this)}
ref={toast => this.toastInstance = toast!} position={this.position}
created={this.toastCreated = this.toastCreated.bind(this)} />
        </div>
    );
}
};
export default App;

```

## [Functional-component]

### APP.JSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import './App.css';
function App() {
    let toastInstance;
    let toasts = [
        { template: '2 Mail has received' },
        { template: 'User Guest Logged in' },
        { template: 'Logging in as Guest' },
        { template: 'Ticket has reserved ' },
        { template: '#templateToast' }
    ];
    let toastFlag = 0;
    let maxCount = 3;
    let timeOutDelay = 600;
    let position = { X: 'Right', Y: 'Bottom' };
    function toastCreated() {
        toastInstance.show(toasts[toastFlag]);
        ++toastFlag;
    }
    function toastShow() {
        setTimeout(() => {
            toastInstance.show(toasts[toastFlag]);
            ++toastFlag;
            if (toastFlag === (toasts.length)) {
                toastFlag = 0;
            }
        });
    }
}

```

```

        }, timeOutDelay);
    }
    function btnClick() {
        toastShow();
    }
    function onClick(e) {
        e.clickToClose = true;
    }
    return (<div>
        <div><ButtonComponent cssClass="e-primary"
onClick={btnClick.bind(this)}> Show Bottom Position
Toast</ButtonComponent></div>
        <ToastComponent click={onClick.bind(this)} ref={toast => toastInstance
= toast} position={position} created={toastCreated.bind(this)}/>
        </div>);
    }
    export default App;

```

### APP.TSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastClickEventArgs, ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from "react";
import './App.css';
function App() {
    let toastInstance: ToastComponent;
    let toasts = [
        { template: '2 Mail has received' },
        { template: 'User Guest Logged in' },
        { template: 'Logging in as Guest' },
        { template: 'Ticket has reserved ' },
        { template: '#templateToast' }
    ];
    let toastFlag: number = 0;
    let maxCount: number = 3;
    let timeOutDelay: number = 600;
    let position = { X: 'Right', Y: 'Bottom' };
    function toastCreated(): void {
        toastInstance.show(toasts[toastFlag]);
        ++toastFlag;
    }
    function toastShow() {
        setTimeout(
            () => {
                toastInstance.show(toasts[toastFlag]);
                ++toastFlag;
                if (toastFlag === (toasts.length)) {
                    toastFlag = 0;
                }
            }, timeOutDelay);
    }
    function btnClick(): void {
        toastShow();
    }
    function onClick(e: ToastClickEventArgs): void {

```

```

    e.clickToClose = true;
  }
  return (
    <div>
      <div><ButtonComponent cssClass="e-primary"
onClick={btnClick.bind(this)}> Show Bottom Position
Toast</ButtonComponent></div>
      <ToastComponent click={onClick.bind(this)} ref={toast => toastInstance
= toast!} position={position} created={toastCreated.bind(this)} />
    </div>
  );
}
export default App;

```

### Prevent toast close with mobile swipe in React Toast component

You can prevent the toast close with mobile swipe action by setting [beforeClose](#) argument cancel value to true while argument type as a swipe. The following code shows how to prevent toast close with mobile swipe.

The following sample demonstrates preventing toast close with mobile swipe element displaying with custom code blocks.

#### [Class-component]

#### APP.JSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from 'react';
class App extends React.Component {
  constructor() {
    super(...arguments);
    this.position = { X: 'Center' };
  }
  onBeforeClose(e) {
    if (e.type === "swipe") {
      e.cancel = true;
    }
  }
  toastCreated() {
    this.toastInstance.show();
  }
  toastShow() {
    this.toastInstance.show();
  }
  render() {
    return (<div>
      <ButtonComponent cssClass="e-primary" onClick={this.toastShow =
this.toastShow.bind(this)}> Show Toast </ButtonComponent>
      <ToastComponent ref={toast => this.toastInstance = toast} title="Matt
sent you a friend request" content="Hey, wanna dress up as wizards and ride
our hoverboards?" position={this.position} beforeClose={this.onBeforeClose =
this.onBeforeClose.bind(this)} created={this.toastCreated =
this.toastCreated.bind(this)} />
    </div>);
  }
}

```

```
}
export default App;
```

### APP.TSX

```
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from 'react';
class App extends React.Component {
  constructor() {
    super(...arguments);
    this.position = { X: 'Center' };
  }
  onBeforeClose(e: any):void {
    if (e.type === "swipe") {
      e.cancel = true;
    }
  }
  toastCreated() {
    this.toastInstance.show();
  }
  toastShow() {
    this.toastInstance.show();
  }
  render() {
    return (<div>
      <ButtonComponent cssClass="e-primary" onClick={this.toastShow =
this.toastShow.bind(this)}> Show Toast </ButtonComponent>
      <ToastComponent ref={toast => this.toastInstance = toast} title="Matt
sent you a friend request" content="Hey, wanna dress up as wizards and ride
our hoverboards?" position={this.position}
beforeClose={this.onBeforeClose=this.onBeforeClose.bind(this)}
created={this.toastCreated = this.toastCreated.bind(this)} />
    </div>);
  }
}
export default App;
```

### [Functional-component]

### APP.JSX

```
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from 'react';
function App() {
  let position = { X: 'Center' };
  let toastInstance;
  function onBeforeClose(e) {
    if (e.type === "swipe") {
      e.cancel = true;
    }
  }
  function toastCreated() {
    toastInstance.show();
  }
}
```

```

function toastShow() {
    toastInstance.show();
}
return (<div>
    <ButtonComponent cssClass="e-primary" onClick={toastShow.bind(this)}>
Show Toast </ButtonComponent>
    <ToastComponent ref={toast => toastInstance = toast} title="Matt sent
you a friend request" content="Hey, wanna dress up as wizards and ride our
hoverboards?" position={position} beforeClose={onBeforeClose.bind(this)}
created={toastCreated.bind(this)} />
    </div>);
}
export default App;

```

### APP.TSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { ToastComponent } from '@syncfusion/ej2-react-notifications';
import * as React from 'react';
function App() {
    let position = { X: 'Center' };
    let toastInstance: ToastComponent;
    function onBeforeClose(e: any): void {
        if (e.type === "swipe") {
            e.cancel = true;
        }
    }
    function toastCreated() {
        toastInstance.show();
    }
    function toastShow() {
        toastInstance.show();
    }
    return (<div>
        <ButtonComponent cssClass="e-primary" onClick={toastShow.bind(this)}>
Show Toast </ButtonComponent>
        <ToastComponent ref={toast => toastInstance = toast} title="Matt sent
you a friend request" content="Hey, wanna dress up as wizards and ride our
hoverboards?" position={position} beforeClose={onBeforeClose.bind(this)}
created={toastCreated.bind(this)} />
        </div>);
    }
export default App;

```