

USER GUIDE

Essential Studio

for EJ2 React

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

3D Chart	19
Getting started	19
Dependencies.....	19
Installation and configuration.....	19
Add 3D Chart to the project.....	20
Module injection.....	21
Populate 3D Chart with data.....	22
Add 3D Chart title	24
Enable legend.....	25
Add data label	27
Enable tooltip.....	28
Working with data in React 3D Chart control	30
Local Data.....	30
Remote data.....	32
Binding data using ODataAdaptor	33
Empty points	34
Dimensions in React 3D Chart control	38
Size for container	38
Size for chart	39
Category axis in React 3D Chart control	42
Labels placement	44
Range	45
Indexed category axis.....	47
Numeric axis in React 3D Chart control	48
Range	50
Range padding	51
Label format.....	59
Grouping separator.....	61
Custom label format	62
DateTime axis in React 3D Chart control	64
DateTime axis.....	64
DateTime category axis.....	65
Label format.....	74
Custom label format	76
Logarithmic axis in React 3D Chart control	78

Range	79
Logarithmic base	81
Logarithmic interval	83
Axis labels in React 3D Chart control	85
Smart axis labels.....	85
Edge label placement.....	89
Maximum labels.....	91
Axis customization in React 3D Chart control.....	93
Title	93
Title rotation	94
Tick lines customization	96
Grid lines customization	98
Multiple axis.....	99
Inversed axis.....	101
Opposed position	103
Multiple panes in React 3D Chart control.....	104
Rows.....	104
Columns	108
Chart Types	113
Column Chart in React 3D Chart control.....	113
Stacked column chart in React 3D Chart control	120
100% Stacked column chart in React 3D Chart control	127
Bar Chart in React 3D Chart component.....	132
Stacked bar chart in React Chart component.....	139
100% Stacked bar chart in React 3D Chart control.....	145
Data labels in React 3D Chart control	150
Position	152
Template	154
Text mapping	155
Format.....	156
Margin.....	158
Customization	160
Customizing specific label	161
Legend in React 3D Chart control	163
Position and alignment	163

Legend customization	171
Series selection through legend.....	180
Collapsing legend item	182
Legend title	183
Arrow page navigation	185
Legend item padding	187
Tooltip in React 3D Chart control	189
Default tooltip.....	189
Fixed tooltip	191
Format the tooltip.....	192
Tooltip template	194
Customize the appearance of tooltip	195
Selection in React 3D Chart control	197
Point.....	197
Series.....	199
Cluster	201
Selection type.....	202
Selection during initial loading.....	204
Selection through legend	206
Print and Export in React 3D Chart control.....	208
Print.....	208
Export.....	209
Appearance in React 3D Chart control	211
Custom color palette.....	211
Data point customization.....	213
Point level customization.....	214
Chart area customization.....	216
Animation.....	219
Chart rotation.....	221
Title	222
Accessibility in React 3D Chart control	230
WAI-ARIA.....	230
Keyboard navigation	230
Accordion	231
Getting Started.....	231

Dependencies.....	231
Setup for Local Development.....	231
Adding Syncfusion packages	232
Adding CSS reference.....	232
Initialize the Accordion using Items	232
Initialize the Accordion using HTML elements.....	234
See Also	237
Expand mode in React Accordion component.....	237
Single	237
Multiple.....	239
See Also	241
Accessibility in React Accordion component	241
Keyboard interaction	243
Ensuring accessibility	243
See also	244
Style in React Accordion component	244
Customizing Accordion	244
Customizing the list items.....	244
Customizing Accordion's header.....	244
Customizing Accordion's expand and collapse icons.....	244
Customizing the hover state of Accordion control	245
Customizing selected item of Accordion control	245
How To	245
Set the nested accordion in React Accordion component	245
Load content through post in React Accordion component.....	248
Set custom animation in React Accordion component	251
To keep single pane open always in React Accordion component.....	256
Create wizard using accordion in React Accordion component	259
Load accordion with data source in React Accordion component	272
Load accordion items dynamically in React Accordion component	274
Customize expand collapse actions in React Accordion component.....	280
Render other components in accordion using template in React Accordion component	284
React hooks accordion in React Accordion component	288
Ej1 api migration in React Accordion component	289
Accessibility and Localization.....	289

AjaxSettings.....	290
Animation.....	290
Items	291
Common.....	294
Accumulation Chart	295
Getting Started.....	295
Dependencies.....	296
Installation and configuration.....	296
Add Accumulation Chart to the Project.....	297
Pie dough nut in React Accumulation chart component.....	298
Pie Chart.....	298
Radius customization	299
Pie Center.....	300
Various Radius Pie Chart.....	302
Doughnut Chart.....	303
Start and End angles	304
Color & Text Mapping	305
Customization	306
Hide pie or doughnut border	307
Multi-level pie chart.....	308
See Also	783
Pyramid in React Accumulation chart component.....	783
Mode	785
Size	786
Gap Between the Segments.....	788
Explode.....	789
Customization	791
See Also	793
Funnel in React Accumulation chart component	793
Size	794
Neck Size	796
Gap between the segments.....	798
Explode.....	799
Smart Data Label.....	801
Customization	804

See Also	806
Data label in React Accumulation chart component	806
Positioning.....	807
DataLabel rotation	807
Smart labels.....	808
Format.....	809
Datalabel template	811
Connector Line	812
Text Mapping	813
Customization	814
Text wrap	815
Show percentages in data labels of pie chart	817
Grouping in React Accumulation chart component	819
Broken slice	820
GroupMode.....	822
Customization	823
Empty points in React Accumulation chart component	824
Customization	825
Annotation in React Accumulation chart component	827
Region	828
Co-ordinate Units.....	830
Alignment.....	831
Tooltip in React Accumulation chart component	832
Header.....	833
Format.....	834
Tooltip mapping name	835
Tooltip template	836
Fixed tooltip	838
Customization	839
To customize individual tooltip.....	840
Legend in React Accumulation chart component.....	841
Position and Alignment.....	842
Legend Reverse	843
Legend Shape	846
Legend Size.....	846

Legend Item Size	847
Paging for Legend.....	848
Legend Text Wrap	849
Enable Animation.....	850
Legend Title.....	852
Arrow Page Navigation	853
Legend Item Padding	853
Center label in React Accumulation chart component.....	854
Center label.....	854
Hover text	855
Customization	856
Title and sub title in React Accumulation chart component	857
Title	857
Title customization.....	859
SubTitle	861
SubTitle customization.....	862
Chart print in React Accumulation chart component.....	864
Print.....	864
Export.....	866
Appearance in React Accumulation chart component	868
Custom Color Palette	868
Animation.....	869
Accessibility in React Accumulation chart component.....	873
WAI-ARIA attributes.....	874
Keyboard interaction	875
Ensuring accessibility	875
See also	875
Ej1 api migration in React Accumulation chart component	875
Accumulation Chart	875
Annotation	877
Series.....	879
DataLabel	883
Legend.....	884
Methods.....	886
Events.....	887

How To	892
Percentage tool tip in React Chart component	892
AppBar.....	893
Getting Started.....	893
Dependencies.....	893
Setup your development environment	893
Adding Syncfusion packages	894
Adding Style sheet to the Application.....	895
Add AppBar to the project	895
Run the application	895
Size and color in React AppBar component.....	896
Size	896
Color	898
Accessibility in React AppBar component	902
Keyboard interaction	903
Ensuring accessibility	903
See also	903
Position in React AppBar component	903
Top AppBar	903
Bottom AppBar	905
Sticky AppBar	907
Design in React AppBar component	908
Spacer.....	908
Separator.....	909
Media Query	911
Designing AppBar with Menu	912
Designing AppBar with Buttons	914
Designing AppBar with SideBar.....	915
Style and appearance in React AppBar component	920
CssClass	920
HtmlAttributes	921
AutoComplete.....	922
Getting Started.....	922
Dependencies.....	922
Adding syncfusion packages	923

Adding AutoComplete component	923
Adding CSS reference.....	924
Binding data source	924
Run the application	925
Configure the Popup List.....	927
See Also	928
Data binding in React Auto complete component	928
Binding local data.....	929
Binding remote data	935
See Also	938
Value binding in AutoComplete Component.....	938
Primitive Data Types	938
Object Data Types	939
Templates in React Auto complete component	940
Item template	941
Group template.....	943
Header template	946
Footer template	950
No Records template	952
Action Failure template	954
See Also	957
Virtualization in AutoComplete Component	957
Binding local data.....	957
Binding remote data	959
Grouping	960
Grouping in React Auto complete component	962
Customization	964
See Also	964
Filtering in React Auto complete component.....	964
Change the filter type	964
Filter item count.....	967
Limit the minimum filter character.....	971
Case sensitive filtering	973
Diacritics Filtering.....	975
See Also	977

Localization in React Auto complete component	977
Loading translations	977
See Also	981
Style in React Auto complete component	981
Customizing the appearance of wrapper element	981
Customizing the dropdown icon's color	981
Customizing the focus color	981
Customizing the outline theme's focus color	981
Customizing the disabled component's text color	982
Customizing the float label element's focusing color	982
Customizing the color of the placeholder text	982
Customizing the text selection color	983
Customizing the background color of focus, hover, and active item's	983
Customizing the appearance of pop-up element	983
Adding mandatory asterisk to placeholder and float label	983
Accessibility in React Auto complete component	985
WAI-ARIA attributes	986
Keyboard Interaction	986
Ensuring accessibility	990
See also	990
How To	990
Autofill in React Auto complete component	990
Icon support in React Auto complete component	991
Custom search in React Auto complete component	992
Ej1 api migration in React Auto complete component	994
DataBinding	994
Filtering	994
Placeholder	995
Popup	995
CSS	997
Grouping	997
Localization	997
Template	997
Sorting	998
Accessibility	998

Selection.....	998
Miscellaneous	999
Common.....	999
Avatar.....	1001
Getting Started.....	1001
Dependencies.....	1001
Installation and configuration.....	1001
Adding Syncfusion packages	1001
Adding CSS Reference	1002
Adding a simple Avatar	1002
Run the application	1002
See Also	1003
Types in React Avatar component	1003
Avatar size	1003
Avatar types	1004
How To	1005
Avatar customization in React Avatar component	1005
Integrate avatar into listview in React Avatar component	1013
Integrate avatar into badge in React Avatar component	1015
Badge	1020
Getting Started.....	1020
Dependencies.....	1020
Installation and configuration.....	1020
Adding Syncfusion packages	1020
Adding CSS Reference	1021
Initialize the Badge Component.....	1021
Run the application	1021
See Also	1023
Types in React Badge component.....	1023
Badge styles	1023
Badge types.....	1027
How To	1039
Badge customization in React Badge component	1039
Integrate badge into listview in React Badge component.....	1045
Dynamic badge content in React Badge component.....	1048

Barcode	1052
Getting Started.....	1052
Dependencies.....	1052
Installation and configuration.....	1052
Adding Barcode Generator control.....	1053
Adding QR Generator control	1054
Adding Datamatrix Generator control	1056
BarcodeGenerator in React Barcode component.....	1057
Code39	1057
Code39 Extended	1058
Code 11	1059
Codabar	1061
Code 32	1062
Code 93	1063
Code 93 Extended	1065
Code 128	1065
Customizing the Barcode color	1067
Customizing the Barcode dimension	1068
Customizing the text	1069
Qrcodegenerator in React Barcode component.....	1070
QR Code	1070
Customizing the Barcode color	1072
Customizing the Barcode dimension	1073
Customizing the text	1074
Datamatrixgenerator in React Barcode component.....	1076
Data Matrix	1076
Customizing the Barcode color	1077
Customizing the Barcode dimension	1078
Customizing the text	1080
Export in React Barcode component	1081
Export.....	1081
Breadcrumb	1082
Getting Started.....	1082
Dependencies.....	1082
Setup your development environment	1083

Adding Syncfusion packages	1083
Adding Style sheet to the Application.....	1084
Add Breadcrumb to the project	1084
Run the application	1084
Add Items to the Breadcrumb Component	1085
Enable or Disable Navigation	1086
Data binding in React Breadcrumb component	1088
Items as tag directive	1088
Items based on current URL.....	1089
Static URL	1089
Customize text when generated items using Url.....	1090
Icons in React Breadcrumb component.....	1091
Loading Icons in Breadcrumb Item	1091
Icon Position.....	1094
Icons only	1096
Show icon only for first item	1096
Navigation in React Breadcrumb component.....	1097
URL	1097
Enable navigation for last Breadcrumb item	1100
Open URL in new page or tab	1101
Template in React Breadcrumb component.....	1102
Item Template.....	1102
Separator Template	1103
Customize Specific Item Template.....	1104
Overflow mode in React Breadcrumb component.....	1106
Collapsed.....	1107
Menu	1109
Wrap.....	1110
Scroll.....	1111
Hidden	1113
None.....	1114
Accessibility in React Breadcrumb component	1114
WAI-ARIA attributes.....	1115
Keyboard interaction	1115
Ensuring accessibility	1115

See also	1116
Bullet Chart	1116
Getting Started.....	1116
Dependencies.....	1116
Installation and configuration.....	1116
Add Bullet Chart to the Project.....	1117
Module Injection.....	1118
Bullet Chart With Data	1119
Add Bullet Chart Title	1120
Ranges.....	1121
Tooltip	1122
Bullet chart dimensions in React Bullet chart component	1123
Size for Container.....	1123
Size for Bullet Chart.....	1124
Axis customization in React Bullet chart component	1128
MajorTickLines and MinorTickLines Customization.....	1128
Tick Placement	1129
Label Format	1130
GroupingSeparator	1132
Custom Label Format.....	1133
Label Placement.....	1134
Opposed Position	1135
Category Label	1137
Category Label Customization	1138
Data binding in React Bullet chart component.....	1139
Ranges in React Bullet chart component.....	1141
Color Customization.....	1141
Value bar in React Bullet chart component.....	1142
Types of actual bar	1143
Actual bar customization	1145
Comparative bar in React Bullet chart component	1147
Types of target bar	1148
Target bar customization	1149
Title in React Bullet chart component	1150
Title	1150

Subtitle	1152
Title and SubTitle Position	1153
Title Customization	1158
SubTitle Customization	1159
Customization in React Bullet chart component	1160
Orientation.....	1160
Right-to-left (RTL).....	1162
Animation.....	1163
Theme	1164
Data label in React Bullet chart component	1165
Data Label Customization	1166
Tool tip in React Bullet chart component	1168
Tooltip Template.....	1170
Customize the Appearance of Tooltip	1170
Tooltip Customization	1170
Accessibility in React Bullet chart component.....	1170
WAI-ARIA attributes.....	1171
Keyboard interaction	1171
Ensuring accessibility	1171
See also	1171
ButtonGroup	1172
Getting Started.....	1172
Dependencies.....	1172
Installation and Configuration	1172
Adding Syncfusion packages	1172
Adding CSS Reference	1172
Adding ButtonGroup component to the Application	1173
Run the application	1173
See Also	1174
Types and styles in React Button group component	1174
ButtonGroup types	1174
ButtonGroup styles	1175
See Also	1176
Selection and nesting in React Button group component	1177
Selection.....	1177

Nesting	1179
See Also	1181
Accessibility in React ButtonGroup component	1181
Keyboard interaction	1182
Ensuring accessibility	1183
See also	1183
Style and appearance in React Button group component.....	1183
How To	1183
Create buttongroup with icons in React Button group component	1183
Create buttongroup with rounded corner in React Button group component	1184
Orientation in React Button group component.....	1185
Disable in React Button group component.....	1186
Enable ripple in React Button group component	1187
Enable rtl in React Button group component	1188
Form submit in React Button group component	1189
Initialize buttongroup using util function in React Button group component	1190
Show buttongroup selected state on initial render in React Button group component	1192
Button	1193
Getting Started.....	1193
Dependencies.....	1193
Installation and Configuration	1194
Adding Syncfusion packages	1194
Adding CSS Reference	1194
Adding Button component to the Application.....	1195
Run the application	1195
See Also	1196
Types and styles in React Button component.....	1196
Button styles	1196
Button types.....	1197
Change Button type	1201
Icons	1202
Button size	1203
See Also	1204
Style and appearance in React Button component	1204
Accessibility in React Button component	1204

WAI-ARIA attributes.....	1205
Keyboard interaction	1206
Ensuring accessibility	1206
See also	1206
How To	1206
Add link to a button in React Button component.....	1206
Create a block button in React Button component	1207
Customize button appearance in React Button component	1208
Customize input and anchor elements in React Button component.....	1208
Repeat button in React Button component.....	1209
Right to left in React Button component.....	1212
Set the disabled state in React Button component	1213
Tooltip for button in React Button component	1213
Ej1 api migration in React Button component.....	1214
Properties.....	1214
Methods.....	1216
Events.....	1216

3D Chart

<!-- markdownlint-disable MD036 -->

Getting started

This section explains you the steps required to create a simple 3D Chart and demonstrate the basic usage of the 3D Chart control.

Dependencies

Below is the list of minimum dependencies required to use the 3D 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
`,`
```

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 3D Chart to the project

Now, you can start adding 3D Chart component in the application.

For getting started, add the 3D Chart component in `src/App.tsx` file using following code.

INDEX.JSX

```
import { Chart3DComponent } from '@syncfusion/ej2-react-charts';
import * as React from 'react';
function App() {
  return (<Chart3DComponent />);
}
export default App;
```

INDEX.TSX

```
import { Chart3DComponent } from '@syncfusion/ej2-react-charts';
import * as React from 'react';
function App() {
  return (<Chart3DComponent />);
}
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 3D Chart.

INDEX.JSX

```
import { Chart3DComponent } from '@syncfusion/ej2-react-charts';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
  return <Chart3DComponent id='charts' />;
}
```

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

INDEX.TSX

```
import { Chart3DComponent } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return <Chart3DComponent id='charts' />
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Module injection

3D 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 3D Chart to visualize sales data for a particular year. For this application we are going to use column series, tooltip, data label, category axis and legend feature of the 3D Chart. Please find the relevant feature service name and description as follows. *ColumnSeries3D* - Inject this module in to services to use column series. *Legend3D* - Inject this module in to services to use legend feature. *Tooltip3D* - Inject this module in to services to use tooltip feature. *DataLabel3D* - Inject this module in to services to use datalabel feature. * *Category3D* - Inject this module in to services to use category feature.

These modules should be injected to the `services` section as follows,

INDEX.JSX

```
import { Category3D, Chart3DComponent, DataLabel3D, ColumnSeries3D,
Legend3D, Tooltip3D, Inject } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return <Chart3DComponent id='charts'>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, DataLabel3D,
Category3D]} />
  </Chart3DComponent>;
}
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import { Category3D, Chart3DComponent, DataLabel3D, ColumnSeries3D,
Legend3D, Tooltip3D, Inject } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return <Chart3DComponent id='charts'>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, DataLabel3D,
Category3D]} />
  </Chart3DComponent>;
}
```

```

</Chart3DComponent>;
}
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Populate 3D Chart with data

This section explains how to plot below JSON data to the 3D Chart.

INDEX.JSX

```

export let data = [
  { x: 'Tesla', y: 137429 },
  { x: 'Aion', y: 80308 },
  { x: 'Wuling', y: 76418 },
  { x: 'Changan', y: 52849 },
  { x: 'Geely', y: 47234 },
  { x: 'Nio', y: 31041 },
  { x: 'Neta', y: 22449 },
  { x: 'BMW', y: 18733 }
];

```

INDEX.TSX

```

export let data: any[] = [
  { x: 'Tesla', y: 137429 },
  { x: 'Aion', y: 80308 },
  { x: 'Wuling', y: 76418 },
  { x: 'Changan', y: 52849 },
  { x: 'Geely', y: 47234 },
  { x: 'Nio', y: 31041 },
  { x: 'Neta', y: 22449 },
  { x: 'BMW', y: 18733 }
];

```

Add a series object to the chart by using [series](#) property. Now map the field names **x** and **y** in the JSON data to the [xName](#) and [yName](#) properties of the series, then set the JSON data to [dataSource](#) property.

Since the JSON contains category data, set the [valueType](#) for horizontal axis to **Category**. By default, the axis valueType is **Numeric**.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Category3D, ColumnSeries3D } from
'@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
export let data = [
  { x: 'Tesla', y: 137429 },
  { x: 'Aion', y: 80308 },
  { x: 'Wuling', y: 76418 },
  { x: 'Changan', y: 52849 },
  { x: 'Geely', y: 47234 },
  { x: 'Nio', y: 31041 },
  { x: 'Neta', y: 22449 },
  { x: 'BMW', y: 18733 },
];

```

```

    { x: 'BMW', y: 18733 }
  ];
  function App() {
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
    primaryXAxis={{
      valueType: 'Category',
      labelRotation: -45,
      labelPlacement: 'BetweenTicks'
    }}
    wallColor='transparent'
    primaryYAxis={{
      maximum: 150000, interval: 50000
    }} enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Category3D, Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, ColumnSeries3D, Inject } from '@syncfusion/ej2-
react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
export let data: any[] = [
  { x: 'Tesla', y: 137429 },
  { x: 'Aion', y: 80308 },
  { x: 'Wuling', y: 76418 },
  { x: 'Changan', y: 52849 },
  { x: 'Geely', y: 47234 },
  { x: 'Nio', y: 31041 },
  { x: 'Neta', y: 22449 },
  { x: 'BMW', y: 18733 }
];
function App() {
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
  primaryXAxis={{
    valueType: 'Category',
    labelRotation: -45,
    labelPlacement: 'BetweenTicks'
  }}
  wallColor='transparent'
  primaryYAxis={{
    maximum: 150000, interval: 50000
  }} enableRotation={true} rotation={7} tilt={10} depth={100}>
  <Inject services={[ColumnSeries3D, Category3D]} />
  <Chart3DSeriesCollectionDirective >

```

```

        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
</Chart3DComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Add 3D Chart title

You can add a title using [title](#) property to the 3D Chart to provide quick information to the user about the data plotted in the 3D Chart.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Category3D, ColumnSeries3D } from
'@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
export let data = [
    { x: 'Tesla', y: 137429 },
    { x: 'Aion', y: 80308 },
    { x: 'Wuling', y: 76418 },
    { x: 'Changan', y: 52849 },
    { x: 'Geely', y: 47234 },
    { x: 'Nio', y: 31041 },
    { x: 'Neta', y: 22449 },
    { x: 'BMW', y: 18733 }
];
function App() {
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
        valueType: 'Category',
        labelRotation: -45,
        labelPlacement: 'BetweenTicks'
    }}
    wallColor='transparent'
    primaryYAxis={{
        maximum: 150000, interval: 50000
    }} enableRotation={true} rotation={7} tilt={10} depth={100}
title='Top Selling Electric Cars in China'>
    <Inject services={[ColumnSeries3D, Category3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX


```

import { Category3D, Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, ColumnSeries3D, Inject } from '@syncfusion/ej2-
react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
export let data: any[] = [
  { x: 'Tesla', y: 137429 },
  { x: 'Aion', y: 80308 },
  { x: 'Wuling', y: 76418 },
  { x: 'Changan', y: 52849 },
  { x: 'Geely', y: 47234 },
  { x: 'Nio', y: 31041 },
  { x: 'Neta', y: 22449 },
  { x: 'BMW', y: 18733 }
];
function App() {
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
  valueType: 'Category',
  labelRotation: -45,
  labelPlacement: 'BetweenTicks'
}}
  wallColor='transparent'
  primaryYAxis={{
    maximum: 150000, interval: 50000
  }} enableRotation={true} rotation={7} tilt={10} depth={100} title='Top
Selling Electric Cars in China'>
    <Inject services={[ColumnSeries3D, Category3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>
  };
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Enable legend

You can use legend for the 3D Chart by setting the [visible](#) property to true in [legendSettings](#) object and by injecting the [Legend3D](#) module into the [services](#).

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, Category3D, ColumnSeries3D,
Highlight3D } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
export let data = [
  { x: 'Tesla', y: 137429 },
  { x: 'Aion', y: 80308 },
  { x: 'Wuling', y: 76418 },
  { x: 'Changan', y: 52849 },
  { x: 'Geely', y: 47234 },
  { x: 'Nio', y: 31041 },

```

```

    { x: 'Neta', y: 22449 },
    { x: 'BMW', y: 18733 }
  ];
function App() {
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
  primaryXAxis={{
    valueType: 'Category',
    labelRotation: -45,
    labelPlacement: 'BetweenTicks'
  }}
  wallColor='transparent'
  primaryYAxis={{
    maximum: 150000, interval: 50000
  }} enableRotation={true} rotation={7} legendSettings={{
enableHighlight: true, visible: true }} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Category3D,
Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x' name='Sales'
yName='y' type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Category3D, Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Highlight3D, ColumnSeries3D, Legend3D, Inject } from
'@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
export let data: any[] = [
  { x: 'Tesla', y: 137429 },
  { x: 'Aion', y: 80308 },
  { x: 'Wuling', y: 76418 },
  { x: 'Changan', y: 52849 },
  { x: 'Geely', y: 47234 },
  { x: 'Nio', y: 31041 },
  { x: 'Neta', y: 22449 },
  { x: 'BMW', y: 18733 }
];
function App() {
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
  primaryXAxis={{
    valueType: 'Category',
    labelRotation: -45,
    labelPlacement: 'BetweenTicks'
  }}
  wallColor='transparent'
  primaryYAxis={{
    maximum: 150000, interval: 50000
  }} enableRotation={true} rotation={7} legendSettings={{
enableHighlight: true, visible: true }} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Category3D,
Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x' name='Sales'
yName='y' type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

```

    }} legendSettings={{ enableHighlight: true, visible: true }}
enableRotation={true} rotation={7} tilt={10} depth={100}>
<Inject services={[ColumnSeries3D, Legend3D, Category3D, Highlight3D]}
/>
<Chart3DSeriesCollectionDirective >
  <Chart3DSeriesDirective dataSource={data} xName='x' name='Sales'
yName='y' type='Column'>
    </Chart3DSeriesDirective>
  </Chart3DSeriesCollectionDirective>
</Chart3DComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Add data label

You can add data labels to improve the readability of the 3D Chart. This can be achieved by setting the [visible](#) property to true in the [dataLabel](#) object and by injecting [DataLabel3D](#) module into the [services](#).

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
export let data = [
  { x: 'Tesla', y: 137429 },
  { x: 'Aion', y: 80308 },
  { x: 'Wuling', y: 76418 },
  { x: 'Changan', y: 52849 },
  { x: 'Geely', y: 47234 },
  { x: 'Nio', y: 31041 },
  { x: 'Neta', y: 22449 },
  { x: 'BMW', y: 18733 }
];
function App() {
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
  valueType: 'Category',
  labelRotation: -45,
  labelPlacement: 'BetweenTicks'
}}
wallColor='transparent'
primaryYAxis={{
  maximum: 150000, interval: 50000
}} enableRotation={true} rotation={7} tilt={10} depth={100}
legendSettings={{ enableHighlight: true, visible: true }}>
  <Inject services={[ColumnSeries3D, Legend3D, Category3D,
DataLabel3D, Highlight3D]} />
  <Chart3DSeriesCollectionDirective >
    <Chart3DSeriesDirective dataSource={data} dataLabel={{ visible:
true }} xName='x' name='Sales' yName='y' type='Column'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
}

```

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

INDEX.TSX

```
import { Category3D, Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, DataLabel3D, Highlight3D, ColumnSeries3D, Legend3D,
Inject } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
export let data: any[] = [
  { x: 'Tesla', y: 137429 },
  { x: 'Aion', y: 80308 },
  { x: 'Wuling', y: 76418 },
  { x: 'Changan', y: 52849 },
  { x: 'Geely', y: 47234 },
  { x: 'Nio', y: 31041 },
  { x: 'Neta', y: 22449 },
  { x: 'BMW', y: 18733 }
];
function App() {
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
  valueType: 'Category',
  labelRotation: -45,
  labelPlacement: 'BetweenTicks'
}}
wallColor='transparent'
primaryYAxis={{
  maximum: 150000, interval: 50000
}} enableRotation={true} rotation={7} tilt={10} depth={100}
legendSettings={{ enableHighlight: true, visible: true }}>
  <Inject services={[ColumnSeries3D, Legend3D, DataLabel3D, Category3D,
Highlight3D]} />
  <Chart3DSeriesCollectionDirective >
    <Chart3DSeriesDirective dataLabel={{ visible: true }}
dataSource={data} xName='x' name='Sales' yName='y' type='Column'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Enable tooltip

The tooltip is useful when you cannot display information by using the data labels due to space constraints. You can enable tooltip by setting the [enable](#) property as true in [tooltip](#) object and by injecting [Tooltip3D](#) module into the [services](#).

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
```

```

Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
export let data = [
  { x: 'Tesla', y: 137429 },
  { x: 'Aion', y: 80308 },
  { x: 'Wuling', y: 76418 },
  { x: 'Changan', y: 52849 },
  { x: 'Geely', y: 47234 },
  { x: 'Nio', y: 31041 },
  { x: 'Neta', y: 22449 },
  { x: 'BMW', y: 18733 }
];
function App() {
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
    primaryXAxis={{
      valueType: 'Category',
      labelRotation: -45,
      labelPlacement: 'BetweenTicks'
    }}
    wallColor='transparent'
    primaryYAxis={{
      maximum: 150000, interval: 50000
    }} enableRotation={true} rotation={7} tilt={10} depth={100}
    tooltip={{ enable: true }} legendSettings={{ enableHighlight: true, visible: true }}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, Category3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x' name='Sales'
        yName='y' type='Column'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Category3D, Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, DataLabel3D, Highlight3D, ColumnSeries3D, Legend3D,
Tooltip3D, Inject } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
export let data: any[] = [
  { x: 'Tesla', y: 137429 },
  { x: 'Aion', y: 80308 },
  { x: 'Wuling', y: 76418 },
  { x: 'Changan', y: 52849 },
  { x: 'Geely', y: 47234 },
  { x: 'Nio', y: 31041 },
  { x: 'Neta', y: 22449 },
  { x: 'BMW', y: 18733 }
]

```

```

];
function App() {
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
  primaryXAxis={{
    valueType: 'Category',
    labelRotation: -45,
    labelPlacement: 'BetweenTicks'
  }}
  wallColor='transparent'
  primaryYAxis={{
    maximum: 150000, interval: 50000
  }} enableRotation={true} rotation={7} tilt={10} depth={100} tooltip={{
    enable: true }} legendSettings={{ enableHighlight: true, visible: true }}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, DataLabel3D,
    Category3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x' name='Sales '
      yName='y' type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>
  };
  export default App;
  ReactDOM.render(<App />, document.getElementById("charts"));
}

```

You can refer to our [React 3D Charts](#) feature tour page for its groundbreaking feature representations. You can also explore our [React 3D Charts example](#) that shows various 3D Chart types and how to represent time-dependent data, showing trends in data at equal intervals.

<!-- markdownlint-disable MD036 -->

Working with data in React 3D Chart control

Local Data

A simple JSON data can be bound to the 3D chart using [dataSource](#) property in series. Now map the fields in JSON to [xName](#) and [yName](#) properties.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
export let data = [
  { month: 'Jan', sales: 35 }, { month: 'Feb', sales: 28 },
  { month: 'Mar', sales: 34 }, { month: 'Apr', sales: 32 },
  { month: 'May', sales: 40 }, { month: 'Jun', sales: 32 },
  { month: 'Jul', sales: 35 }, { month: 'Aug', sales: 55 },
  { month: 'Sep', sales: 38 }, { month: 'Oct', sales: 30 },
  { month: 'Nov', sales: 25 }, { month: 'Dec', sales: 32 }
];

function App() {

```

```

    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
    primaryXAxis={{
        valueType: 'Category',
        labelRotation: -45,
        labelPlacement: 'BetweenTicks'
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, Category3D,
    DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='month'
    name='Sales' yName='sales' type='Column'>
            <Chart3DSeriesDirective>
                </Chart3DSeriesDirective>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
export let data = [
    { month: 'Jan', sales: 35 }, { month: 'Feb', sales: 28 },
    { month: 'Mar', sales: 34 }, { month: 'Apr', sales: 32 },
    { month: 'May', sales: 40 }, { month: 'Jun', sales: 32 },
    { month: 'Jul', sales: 35 }, { month: 'Aug', sales: 55 },
    { month: 'Sep', sales: 38 }, { month: 'Oct', sales: 30 },
    { month: 'Nov', sales: 25 }, { month: 'Dec', sales: 32 }
];
function App() {
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
    primaryXAxis={{
        valueType: 'Category',
        labelRotation: -45,
        labelPlacement: 'BetweenTicks'
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, Category3D,
    DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='month' name='Sales'
    yName='sales' type='Column'>
            <Chart3DSeriesDirective>
                </Chart3DSeriesDirective>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;

```

```
ReactDOM.render(<App />, document.getElementById("charts"));
```

Remote data

The remote data can be bound to the 3D chart using the [DataManager](#). The [DataManager](#) requires minimal information like web service URL, adaptor and cross domain to interact with service endpoint properly. Assign the instance of the [DataManager](#) to the [dataSource](#) property in series and map the fields of data to [xName](#) and [yName](#) properties. You can also use the [query](#) property of the series to filter the data.

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import { DataManager, Query } from '@syncfusion/ej2-data';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const dataManager = new DataManager({
        url: 'https://services.syncfusion.com/js/production/api/orders',
    });
    const query = new Query().take(5).where('Estimate', 'lessThan', 3,
false);
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
    valueType: 'Category',
    labelRotation: -45,
    labelPlacement: 'BetweenTicks'
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
<Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, Category3D,
DataLabel3D, Highlight3D]} />
<Chart3DSeriesCollectionDirective >
    <Chart3DSeriesDirective dataSource={dataManager}
xName='CustomerID' yName='Freight' type='Column' query={query}>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import { DataManager, Query } from '@syncfusion/ej2-data';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
```



```

const dataManager = new DataManager({
  url: 'https://services.syncfusion.com/js/production/api/orders',
});
const query = new Query().take(5).where('Estimate', 'lessThan', 3, false);
return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
  valueType: 'Category',
  labelRotation: -45,
  labelPlacement: 'BetweenTicks'
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
  <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, Category3D,
DataLabel3D, Highlight3D]} />
  <Chart3DSeriesCollectionDirective >
    <Chart3DSeriesDirective dataSource={dataManager} xName='CustomerID'
yName='Freight' type='Column' query={query}>
    </Chart3DSeriesDirective>
  </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Binding data using ODataAdaptor

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.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Category3D, ColumnSeries3D } from
'@syncfusion/ej2-react-charts';
import { DataManager, Query, ODataAdaptor } from '@syncfusion/ej2-data';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const dataManager = new DataManager({
    url: 'https://services.syncfusion.com/js/production/api/orders',
    adaptor: new ODataAdaptor()
  });
  const query = new Query();
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
  valueType: 'Category',
  labelRotation: -45,
  labelPlacement: 'BetweenTicks'
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, Category3D,
DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={dataManager}
xName='CustomerID' yName='Freight' type='Column' query={query}>

```

```

        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import { DataManager, Query, ODataAdaptor } from '@syncfusion/ej2-data';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const dataManager = new DataManager({
    url: 'https://services.syncfusion.com/js/production/api/orders',
    adaptor: new ODataAdaptor()
  });
  const query = new Query();
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
  valueType: 'Category',
  labelRotation: -45,
  labelPlacement: 'BetweenTicks'
}}
  wallColor='transparent'
  enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, Category3D,
DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={dataManager} xName='CustomerID'
yName='Freight' type='Column' query={query}>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Empty points

The data points that uses the `null` or `undefined` as value are considered as empty points. The empty data points are ignored and is not plotted in the chart. When the data is provided by using the points property, by using [emptyPointSettings](#) property in series, the empty can be customized. The default [mode](#) of the empty point is **Gap**.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-charts';
import * as React from "react";

```

```

import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { month: 'Jan', sales: 35 }, { month: 'Feb', sales: null },
    { month: 'Mar', sales: 34 }, { month: 'Apr', sales: 32 },
    { month: 'May', sales: 40 }, { month: 'Jun', sales: undefined },
    { month: 'Jul', sales: 35 }, { month: 'Aug', sales: 55 },
    { month: 'Sep', sales: 38 }, { month: 'Oct', sales: 30 },
    { month: 'Nov', sales: 25 }, { month: 'Dec', sales: 32 }
  ];
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
    primaryXAxis={{
      valueType: 'Category',
      labelRotation: -45,
      labelPlacement: 'BetweenTicks'
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, Category3D,
    DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='month'
      yName='sales' type='Column' emptyPointSettings={{
        mode: "Gap"
      }}>
    </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Category3D, Chart3DComponent, Chart3DSeriesCollectionDirective,
Tooltip3D, Chart3DSeriesDirective, DataLabel3D, Highlight3D, ColumnSeries3D,
Legend3D, Inject } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { month: 'Jan', sales: 35 }, { month: 'Feb', sales: null },
    { month: 'Mar', sales: 34 }, { month: 'Apr', sales: 32 },
    { month: 'May', sales: 40 }, { month: 'Jun', sales: undefined },
    { month: 'Jul', sales: 35 }, { month: 'Aug', sales: 55 },
    { month: 'Sep', sales: 38 }, { month: 'Oct', sales: 30 },
    { month: 'Nov', sales: 25 }, { month: 'Dec', sales: 32 }
  ];
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
    primaryXAxis={{
      valueType: 'Category',
      labelRotation: -45,
      labelPlacement: 'BetweenTicks'
    }}
    wallColor='transparent'

```

```

    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, Category3D,
DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='month' yName='sales'
type='Column' emptyPointSettings={{
            mode: "Gap"
        }}>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Customizing empty point

The specific color for empty point can be set by the [fill](#) property in [emptyPointSettings](#).

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
export let data = [
    { x: 'Tesla', y: 137429 },
    { x: 'Aion', y: 80308 },
    { x: 'Wuling', y: 76418 },
    { x: 'Changan', y: 52849 },
    { x: 'Geely', y: 47234 },
    { x: 'Nio', y: 31041 },
    { x: 'Neta', y: 22449 },
    { x: 'BMW', y: 18733 }
];
function App() {
    const data = [
        { month: 'Jan', sales: 35 }, { month: 'Feb', sales: null },
        { month: 'Mar', sales: 34 }, { month: 'Apr', sales: 32 },
        { month: 'May', sales: 40 }, { month: 'Jun', sales: undefined },
        { month: 'Jul', sales: 35 }, { month: 'Aug', sales: 55 },
        { month: 'Sep', sales: 38 }, { month: 'Oct', sales: 30 },
        { month: 'Nov', sales: 25 }, { month: 'Dec', sales: 32 }
    ];
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
        valueType: 'Category',
        labelRotation: -45,
        labelPlacement: 'BetweenTicks'
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, Category3D,
DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >

```

```

        <Chart3DSeriesDirective dataSource={data} xName='month'
yName='sales' type='Column' emptyPointSettings={{
            mode: 'Average',
            fill: 'green'
        }}>
    </Chart3DSeriesDirective>
</Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Category3D, Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, DataLabel3D, Highlight3D, ColumnSeries3D, Legend3D,
Tooltip3D, Inject } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
export let data: any[] = [
    { x: 'Tesla', y: 137429 },
    { x: 'Aion', y: 80308 },
    { x: 'Wuling', y: 76418 },
    { x: 'Changan', y: 52849 },
    { x: 'Geely', y: 47234 },
    { x: 'Nio', y: 31041 },
    { x: 'Neta', y: 22449 },
    { x: 'BMW', y: 18733 }
];
function App() {
    const data = [
        { month: 'Jan', sales: 35 }, { month: 'Feb', sales: null },
        { month: 'Mar', sales: 34 }, { month: 'Apr', sales: 32 },
        { month: 'May', sales: 40 }, { month: 'Jun', sales: undefined },
        { month: 'Jul', sales: 35 }, { month: 'Aug', sales: 55 },
        { month: 'Sep', sales: 38 }, { month: 'Oct', sales: 30 },
        { month: 'Nov', sales: 25 }, { month: 'Dec', sales: 32 }
    ];
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
    valueType: 'Category',
    labelRotation: -45,
    labelPlacement: 'BetweenTicks'
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, Category3D,
DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='month'
yName='sales' type='Column' emptyPointSettings={{
            mode: 'Average',
            fill: 'green'
        }}>
    </Chart3DSeriesDirective>

```

```

    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Dimensions in React 3D Chart control

Size for container

The 3D chart can be rendered to its container size and it can be set via inline or CSS as demonstrated below.

`javascript`

```

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

```

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { month: 'Jan', sales: 35 }, { month: 'Feb', sales: 28 },
    { month: 'Mar', sales: 34 }, { month: 'Apr', sales: 32 },
    { month: 'May', sales: 40 }, { month: 'Jun', sales: 32 },
    { month: 'Jul', sales: 35 }, { month: 'Aug', sales: 55 },
    { month: 'Sep', sales: 38 }, { month: 'Oct', sales: 30 },
    { month: 'Nov', sales: 25 }, { month: 'Dec', sales: 32 }
  ];
  return <Chart3DComponent id='charts' width="650px" height="350px"
style={{ textAlign: "center" }} primaryXAxis={{
  valueType: 'Category',
  labelRotation: -45,
  labelPlacement: 'BetweenTicks'
}}
  wallColor='transparent'
  enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D,
Category3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='month'
yName='sales' type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;

```

```
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { month: 'Jan', sales: 35 }, { month: 'Feb', sales: 28 },
    { month: 'Mar', sales: 34 }, { month: 'Apr', sales: 32 },
    { month: 'May', sales: 40 }, { month: 'Jun', sales: 32 },
    { month: 'Jul', sales: 35 }, { month: 'Aug', sales: 55 },
    { month: 'Sep', sales: 38 }, { month: 'Oct', sales: 30 },
    { month: 'Nov', sales: 25 }, { month: 'Dec', sales: 32 }
  ];
  return <Chart3DComponent id='charts' width="650px" height="350px" style={{
textAlign: "center" }} primaryXAxis={{
  valueType: 'Category',
  labelRotation: -45,
  labelPlacement: 'BetweenTicks'
}}
  wallColor='transparent'
  enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, Category3D,
DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='month'
yName='sales' type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

```
<!-- markdownlint-disable MD036 -->
```

Size for chart

```
<!-- markdownlint-disable MD036 -->
```

The size of the 3D chart can be set directly through [width](#) and [height](#) properties.

In Pixel

The size of the 3D chart can be set in pixel as demonstrated below.

INDEX.JSX

```
import { Category3D, Chart3DComponent, Chart3DSeriesCollectionDirective,
DataLabel3D, Highlight3D, Legend3D, Tooltip3D, Chart3DSeriesDirective,
ColumnSeries3D, Inject } from '@syncfusion/ej2-react-charts';
import * as ReactDOM from "react-dom";
function App() {
```

```

const data = [
  { month: 'Jan', sales: 35 }, { month: 'Feb', sales: 28 },
  { month: 'Mar', sales: 34 }, { month: 'Apr', sales: 32 },
  { month: 'May', sales: 40 }, { month: 'Jun', sales: 32 },
  { month: 'Jul', sales: 35 }, { month: 'Aug', sales: 55 },
  { month: 'Sep', sales: 38 }, { month: 'Oct', sales: 30 },
  { month: 'Nov', sales: 25 }, { month: 'Dec', sales: 32 }
];

return <Chart3DComponent id='charts' width="650" height="350" style={{
  textAlign: "center" }} primaryXAxis={{
    valueType: 'Category',
    labelRotation: -45,
    labelPlacement: 'BetweenTicks'
  }}
  wallColor='transparent'
  enableRotation={true} rotation={7} tilt={10} depth={100}>
  <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D,
    Category3D, DataLabel3D, Highlight3D]} />
  <Chart3DSeriesCollectionDirective >
    <Chart3DSeriesDirective dataSource={data} xName='month'
    yName='sales' type='Column'>
    </Chart3DSeriesDirective>
  </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Category3D, Chart3DComponent, Chart3DSeriesCollectionDirective,
  DataLabel3D, Highlight3D, Legend3D, Tooltip3D, Chart3DSeriesDirective,
  ColumnSeries3D, Inject } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { month: 'Jan', sales: 35 }, { month: 'Feb', sales: 28 },
    { month: 'Mar', sales: 34 }, { month: 'Apr', sales: 32 },
    { month: 'May', sales: 40 }, { month: 'Jun', sales: 32 },
    { month: 'Jul', sales: 35 }, { month: 'Aug', sales: 55 },
    { month: 'Sep', sales: 38 }, { month: 'Oct', sales: 30 },
    { month: 'Nov', sales: 25 }, { month: 'Dec', sales: 32 }
  ];
  return <Chart3DComponent id='charts' width="650" height="350" style={{
    textAlign: "center" }} primaryXAxis={{
      valueType: 'Category',
      labelRotation: -45,
      labelPlacement: 'BetweenTicks'
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, Category3D,
      DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >

```



```

        <Chart3DSeriesDirective dataSource={data} xName='month'
yName='sales' type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

In Percentage

By setting the value in percentage, 3D chart gets its dimension with respect to its container. For example, when the height is **50%**, chart renders to half of the container height.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { month: 'Jan', sales: 35 }, { month: 'Feb', sales: 28 },
        { month: 'Mar', sales: 34 }, { month: 'Apr', sales: 32 },
        { month: 'May', sales: 40 }, { month: 'Jun', sales: 32 },
        { month: 'Jul', sales: 35 }, { month: 'Aug', sales: 55 },
        { month: 'Sep', sales: 38 }, { month: 'Oct', sales: 30 },
        { month: 'Nov', sales: 25 }, { month: 'Dec', sales: 32 }
    ];
    return <Chart3DComponent id='charts' width="80%" height="90%" style={{
textAlign: "center" }} primaryXAxis={{
        valueType: 'Category',
        labelRotation: -45,
        labelPlacement: 'BetweenTicks'
    }}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D,
Category3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='month'
yName='sales' type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,

```

```

Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { month: 'Jan', sales: 35 }, { month: 'Feb', sales: 28 },
    { month: 'Mar', sales: 34 }, { month: 'Apr', sales: 32 },
    { month: 'May', sales: 40 }, { month: 'Jun', sales: 32 },
    { month: 'Jul', sales: 35 }, { month: 'Aug', sales: 55 },
    { month: 'Sep', sales: 38 }, { month: 'Oct', sales: 30 },
    { month: 'Nov', sales: 25 }, { month: 'Dec', sales: 32 }
  ];
  return <Chart3DComponent id='charts' width="80%" height="90%" style={{
    textAlign: "center" }} primaryXAxis={{
    valueType: 'Category',
    labelRotation: -45,
    labelPlacement: 'BetweenTicks'
  }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, Category3D,
    DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='month'
yName='sales' type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Note: When you do not specify the size, it takes 450px as the height and window size as its width.

Category axis in React 3D Chart control

The category axis is the horizontal axis of a 3D chart that shows text values rather than numerical values. Compared to the vertical axis, this axis has fewer labels. The following sample shows to render the 3D chart using category axis.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },

```

```

        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
    primaryXAxis={{
        valueType: 'Category'
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, Category3D,
    DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
    yName='gold' type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
    primaryXAxis={{
        valueType: 'Category'
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, Category3D,
    DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
    yName='gold' type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};

```

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

Note: To use category axis, we need to inject **Category3D** module into the services and set the [valueType](#) of axis to **Category**.

Labels placement

By default, category axis labels are placed between ticks in an axis. It can also be placed on ticks using the [labelPlacement](#) property.

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
  valueType: 'Category',
  // label placement as on ticks
  labelPlacement: 'OnTicks',
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
<Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, Category3D,
DataLabel3D, Highlight3D]} />
<Chart3DSeriesCollectionDirective >
  <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
    </Chart3DSeriesDirective>
  </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
```

```

Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
    primaryXAxis={{
      valueType: 'Category',
      // label placement as on ticks
      labelPlacement: 'OnTicks',
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, Category3D,
    DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
      yName='gold' type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Range

The range of the category axis can be customized using [minimum](#), [maximum](#) and [interval](#) properties of the axis.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },

```

```

    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
  primaryXAxis={{
    valueType: 'Category',
    interval: 2, minimum: 1, maximum: 5
  }}
  wallColor='transparent'
  enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, Category3D,
    DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];
  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: 'Category',
    interval: 2, minimum: 1, maximum: 5
  }}
  wallColor='transparent'
  enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, Category3D,
    DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};

```

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

Indexed category axis

The category axis also can be rendered based on the index values of the data source. This can be achieved by defining the [isIndexed](#) property to **true** in the axis.

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
    valueType: 'Category',
    isIndexed: true
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
<Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, Category3D,
DataLabel3D, Highlight3D]} />
<Chart3DSeriesCollectionDirective >
    <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
    </Chart3DSeriesDirective>
    <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' type='Column'>
    </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
```

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
    primaryXAxis={{
      valueType: 'Category',
      isIndexed: true
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, Category3D,
    DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' type='Column'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

<!-- markdownlint-disable MD036 -->

Numeric axis in React 3D Chart control

The numeric axis can be used to represent the numeric values of data in 3D chart. By default, the **valueType** of an axis is **Double**.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: 1, y: 7 }, { x: 2, y: 1 }, { x: 3, y: 1 }, { x: 4, y: 14 }, {
x: 5, y: 1 }, { x: 6, y: 10 },
    { x: 7, y: 8 }, { x: 8, y: 6 }, { x: 9, y: 10 }, { x: 10, y: 10 }, {
x: 11, y: 16 }, { x: 12, y: 6 },

```



```

        { x: 13, y: 14 }, { x: 14, y: 7 }, { x: 15, y: 5 }, { x: 16, y: 2 },
        { x: 17, y: 14 }, { x: 18, y: 7 },
        { x: 19, y: 7 }, { x: 20, y: 10 }
    ];
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
    primaryXAxis={{
        valueType: 'Double',
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, DataLabel3D,
    Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
    type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 1, y: 7 }, { x: 2, y: 1 }, { x: 3, y: 1 }, { x: 4, y: 14 }, {
    x: 5, y: 1 }, { x: 6, y: 10 },
        { x: 7, y: 8 }, { x: 8, y: 6 }, { x: 9, y: 10 }, { x: 10, y: 10 }, {
    x: 11, y: 16 }, { x: 12, y: 6 },
        { x: 13, y: 14 }, { x: 14, y: 7 }, { x: 15, y: 5 }, { x: 16, y: 2 },
        { x: 17, y: 14 }, { x: 18, y: 7 },
        { x: 19, y: 7 }, { x: 20, y: 10 }
    ];
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
    primaryXAxis={{
        valueType: 'Double',
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, DataLabel3D,
    Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
    type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};

```

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

Range

The range of an axis, will be calculated automatically based on the provided data, and it can also be customized by using the [minimum](#), [maximum](#) and [interval](#) properties of the axis.

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 1, y: 7 }, { x: 2, y: 1 }, { x: 3, y: 1 }, { x: 4, y: 14 }, {
x: 5, y: 1 }, { x: 6, y: 10 },
        { x: 7, y: 8 }, { x: 8, y: 6 }, { x: 9, y: 10 }, { x: 10, y: 10 }, {
x: 11, y: 16 }, { x: 12, y: 6 },
        { x: 13, y: 14 }, { x: 14, y: 7 }, { x: 15, y: 5 }, { x: 16, y: 2 },
{ x: 17, y: 14 }, { x: 18, y: 7 },
        { x: 19, y: 7 }, { x: 20, y: 10 }
    ];
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
        valueType: 'Double',
        minimum: 1,
        maximum: 20,
        interval: 5
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, DataLabel3D,
Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
```

```
function App() {
    const data = [
        { x: 1, y: 7 }, { x: 2, y: 1 }, { x: 3, y: 1 }, { x: 4, y: 14 }, {
x: 5, y: 1 }, { x: 6, y: 10 },
        { x: 7, y: 8 }, { x: 8, y: 6 }, { x: 9, y: 10 }, { x: 10, y: 10 }, {
x: 11, y: 16 }, { x: 12, y: 6 },
        { x: 13, y: 14 }, { x: 14, y: 7 }, { x: 15, y: 5 }, { x: 16, y: 2 },
{ x: 17, y: 14 }, { x: 18, y: 7 },
        { x: 19, y: 7 }, { x: 20, y: 10 }
    ];
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
        valueType: 'Double',
        minimum: 1,
        maximum: 20,
        interval: 5
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, DataLabel3D,
Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Range padding

Padding can be applied to the minimum and maximum extremes of an axis range by using the [rangePadding](#) property. Numeric axis supports the following types of padding.

- None
- Round
- Additional
- Normal
- Auto

Numeric - None

When the [rangePadding](#) is set to **None**, minimum and maximum of the axis is based on the data.

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
```

```

    const data = [
      { x: 1, y: 7 }, { x: 2, y: 1 }, { x: 3, y: 1 }, { x: 4, y: 14 }, {
x: 5, y: 1 }, { x: 6, y: 10 },
      { x: 7, y: 8 }, { x: 8, y: 6 }, { x: 9, y: 10 }, { x: 10, y: 10 }, {
x: 11, y: 16 }, { x: 12, y: 6 },
      { x: 13, y: 14 }, { x: 14, y: 7 }, { x: 15, y: 5 }, { x: 16, y: 2 },
{ x: 17, y: 14 }, { x: 18, y: 7 },
      { x: 19, y: 7 }, { x: 20, y: 10 }
    ];
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
      valueType: 'Double'
    }}
    primaryYAxis={{
      //RangePadding as none in Y Axis
      rangePadding: 'None'
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, DataLabel3D,
Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: 1, y: 7 }, { x: 2, y: 1 }, { x: 3, y: 1 }, { x: 4, y: 14 }, {
x: 5, y: 1 }, { x: 6, y: 10 },
    { x: 7, y: 8 }, { x: 8, y: 6 }, { x: 9, y: 10 }, { x: 10, y: 10 }, {
x: 11, y: 16 }, { x: 12, y: 6 },
    { x: 13, y: 14 }, { x: 14, y: 7 }, { x: 15, y: 5 }, { x: 16, y: 2 },
{ x: 17, y: 14 }, { x: 18, y: 7 },
    { x: 19, y: 7 }, { x: 20, y: 10 }
  ];
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
    valueType: 'Double'
  }}
  primaryYAxis={{
    //RangePadding as none in Y Axis

```

```

        rangePadding: 'None'
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, DataLabel3D,
Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Numeric - Round

When the [rangePadding](#) is set to **Round**, minimum and maximum will be rounded to the nearest possible value, which is divisible by interval. For example, when the [minimum](#) is 3.5 and the [interval](#) is 1, then the minimum will be rounded to 3.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 1, y: 7 }, { x: 2, y: 1 }, { x: 3, y: 1 }, { x: 4, y: 14 }, {
x: 5, y: 1 }, { x: 6, y: 10 },
        { x: 7, y: 8 }, { x: 8, y: 6 }, { x: 9, y: 10 }, { x: 10, y: 10 }, {
x: 11, y: 16 }, { x: 12, y: 6 },
        { x: 13, y: 14 }, { x: 14, y: 7 }, { x: 15, y: 5 }, { x: 16, y: 2 },
{ x: 17, y: 14 }, { x: 18, y: 7 },
        { x: 19, y: 7 }, { x: 20, y: 10 }
    ];
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
    valueType: 'Double'
}}
primaryYAxis={{
    rangePadding: 'Round'
}}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, DataLabel3D,
Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>

```

```

    </Chart3DComponent>;
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: 1, y: 7 }, { x: 2, y: 1 }, { x: 3, y: 1 }, { x: 4, y: 14 }, {
x: 5, y: 1 }, { x: 6, y: 10 },
    { x: 7, y: 8 }, { x: 8, y: 6 }, { x: 9, y: 10 }, { x: 10, y: 10 }, {
x: 11, y: 16 }, { x: 12, y: 6 },
    { x: 13, y: 14 }, { x: 14, y: 7 }, { x: 15, y: 5 }, { x: 16, y: 2 },
{ x: 17, y: 14 }, { x: 18, y: 7 },
    { x: 19, y: 7 }, { x: 20, y: 10 }
  ];
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
  valueType: 'Double'
}}
primaryYAxis={{
  rangePadding: 'Round'
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, DataLabel3D,
Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Numeric - Additional

When the [rangePadding](#) is set to **Additional**, interval of an axis will be added to the minimum and maximum of the axis.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';

```

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 1, y: 7 }, { x: 2, y: 1 }, { x: 3, y: 1 }, { x: 4, y: 14 }, {
x: 5, y: 1 }, { x: 6, y: 10 },
        { x: 7, y: 8 }, { x: 8, y: 6 }, { x: 9, y: 10 }, { x: 10, y: 10 }, {
x: 11, y: 16 }, { x: 12, y: 6 },
        { x: 13, y: 14 }, { x: 14, y: 7 }, { x: 15, y: 5 }, { x: 16, y: 2 },
{ x: 17, y: 14 }, { x: 18, y: 7 },
        { x: 19, y: 7 }, { x: 20, y: 10 }
    ];
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
        valueType: 'Double'
    }}
primaryYAxis={{
        rangePadding: 'Additional'
    }}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, DataLabel3D,
Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
                </Chart3DSeriesDirective>
            </Chart3DSeriesCollectionDirective>
        </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 1, y: 7 }, { x: 2, y: 1 }, { x: 3, y: 1 }, { x: 4, y: 14 }, {
x: 5, y: 1 }, { x: 6, y: 10 },
        { x: 7, y: 8 }, { x: 8, y: 6 }, { x: 9, y: 10 }, { x: 10, y: 10 }, {
x: 11, y: 16 }, { x: 12, y: 6 },
        { x: 13, y: 14 }, { x: 14, y: 7 }, { x: 15, y: 5 }, { x: 16, y: 2 },
{ x: 17, y: 14 }, { x: 18, y: 7 },
        { x: 19, y: 7 }, { x: 20, y: 10 }
    ];
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
        valueType: 'Double'
    }}

```

```

    primaryYAxis={{
      rangePadding: 'Additional'
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, DataLabel3D,
Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
  };
  export default App;
  ReactDOM.render(<App />, document.getElementById("charts"));

```

Numeric - Normal

When the [rangePadding](#) is set to **Normal**, padding is applied to the axis based on default range calculation.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: 1, y: 7 }, { x: 2, y: 1 }, { x: 3, y: 1 }, { x: 4, y: 14 }, {
x: 5, y: 1 }, { x: 6, y: 10 },
    { x: 7, y: 8 }, { x: 8, y: 6 }, { x: 9, y: 10 }, { x: 10, y: 10 }, {
x: 11, y: 16 }, { x: 12, y: 6 },
    { x: 13, y: 14 }, { x: 14, y: 7 }, { x: 15, y: 5 }, { x: 16, y: 2 },
{ x: 17, y: 14 }, { x: 18, y: 7 },
    { x: 19, y: 7 }, { x: 20, y: 10 }
  ];
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
  valueType: 'Double'
}}
primaryYAxis={{
  rangePadding: 'Normal'
}}
  wallColor='transparent'
  enableRotation={true} rotation={7} tilt={10} depth={100}>
  <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, DataLabel3D,
Highlight3D]} />
  <Chart3DSeriesCollectionDirective >
    <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>

```



```

    </Chart3DComponent>;
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: 1, y: 7 }, { x: 2, y: 1 }, { x: 3, y: 1 }, { x: 4, y: 14 }, {
x: 5, y: 1 }, { x: 6, y: 10 },
    { x: 7, y: 8 }, { x: 8, y: 6 }, { x: 9, y: 10 }, { x: 10, y: 10 }, {
x: 11, y: 16 }, { x: 12, y: 6 },
    { x: 13, y: 14 }, { x: 14, y: 7 }, { x: 15, y: 5 }, { x: 16, y: 2 },
    { x: 17, y: 14 }, { x: 18, y: 7 },
    { x: 19, y: 7 }, { x: 20, y: 10 }
  ];
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
  valueType: 'Double'
}}
primaryYAxis={{
  rangePadding: 'Normal'
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
<Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, DataLabel3D,
Highlight3D]} />
<Chart3DSeriesCollectionDirective >
  <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
    </Chart3DSeriesDirective>
  </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Numeric - Auto

When the [rangePadding](#) is set to **Auto**, horizontal numeric axis takes **None** as padding calculation, while the vertical numeric axis takes **Normal** as padding calculation.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';

```

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 1, y: 7 }, { x: 2, y: 1 }, { x: 3, y: 1 }, { x: 4, y: 14 }, {
x: 5, y: 1 }, { x: 6, y: 10 },
        { x: 7, y: 8 }, { x: 8, y: 6 }, { x: 9, y: 10 }, { x: 10, y: 10 }, {
x: 11, y: 16 }, { x: 12, y: 6 },
        { x: 13, y: 14 }, { x: 14, y: 7 }, { x: 15, y: 5 }, { x: 16, y: 2 },
{ x: 17, y: 14 }, { x: 18, y: 7 },
        { x: 19, y: 7 }, { x: 20, y: 10 }
    ];
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
    valueType: 'Double',
    // Set the rangePadding as auto in X Axis
    rangePadding: 'Auto'
}}
primaryYAxis={{
    rangePadding: 'Auto'
}}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, DataLabel3D,
Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 1, y: 7 }, { x: 2, y: 1 }, { x: 3, y: 1 }, { x: 4, y: 14 }, {
x: 5, y: 1 }, { x: 6, y: 10 },
        { x: 7, y: 8 }, { x: 8, y: 6 }, { x: 9, y: 10 }, { x: 10, y: 10 }, {
x: 11, y: 16 }, { x: 12, y: 6 },
        { x: 13, y: 14 }, { x: 14, y: 7 }, { x: 15, y: 5 }, { x: 16, y: 2 },
{ x: 17, y: 14 }, { x: 18, y: 7 },
        { x: 19, y: 7 }, { x: 20, y: 10 }
    ];
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{

```

```

        valueType: 'Double',
        // Set the rangePadding as auto in X Axis
        rangePadding: 'Auto'
    }}
    primaryYAxis={{
        rangePadding: 'Auto'
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, DataLabel3D,
Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Label format

Numeric label format

Numeric labels can be formatted by using the [labelFormat](#) property. Also, it supports all globalize format.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 1, y: 7 }, { x: 2, y: 1 }, { x: 3, y: 1 }, { x: 4, y: 14 }, {
x: 5, y: 1 }, { x: 6, y: 10 },
        { x: 7, y: 8 }, { x: 8, y: 6 }, { x: 9, y: 10 }, { x: 10, y: 10 }, {
x: 11, y: 16 }, { x: 12, y: 6 },
        { x: 13, y: 14 }, { x: 14, y: 7 }, { x: 15, y: 5 }, { x: 16, y: 2 },
{ x: 17, y: 14 }, { x: 18, y: 7 },
        { x: 19, y: 7 }, { x: 20, y: 10 }
    ];
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
        valueType: 'Double'
    }}
    primaryYAxis={{
        //Label format as currency
        labelFormat: 'c'
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>

```

```

    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, DataLabel3D,
Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 1, y: 7 }, { x: 2, y: 1 }, { x: 3, y: 1 }, { x: 4, y: 14 }, {
x: 5, y: 1 }, { x: 6, y: 10 },
        { x: 7, y: 8 }, { x: 8, y: 6 }, { x: 9, y: 10 }, { x: 10, y: 10 }, {
x: 11, y: 16 }, { x: 12, y: 6 },
        { x: 13, y: 14 }, { x: 14, y: 7 }, { x: 15, y: 5 }, { x: 16, y: 2 },
{ x: 17, y: 14 }, { x: 18, y: 7 },
        { x: 19, y: 7 }, { x: 20, y: 10 }
    ];
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
        valueType: 'Double'
    }}
    primaryYAxis={{
        //Label format as currency
        labelFormat: 'c'
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, DataLabel3D,
Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
                </Chart3DSeriesDirective>
            </Chart3DSeriesCollectionDirective>
        </Chart3DComponent>;
    };
    export default App;
    ReactDOM.render(<App />, document.getElementById("charts"));
}

```

The following table describes the result of applying some commonly used label formats on numeric values.

<!-- markdownlint-disable MD033 -->

Label Value	Label Format: Property value	Result	Description
1000	n1	1000.0	The Number is rounded to 1 decimal place.
1000	n2	1000.00	The Number is rounded to 2 decimal place.
1000	n3	1000.000	The Number is rounded to 3 decimal place.
0.01	p1	1.0%	The Number is converted to percentage with 1 decimal place.
0.01	p2	1.00%	The Number is converted to percentage with 2 decimal place.
0.01	p3	1.000%	The Number is converted to percentage with 3 decimal place.
1000	c1	\$1000.0	The Currency symbol is appended to number and number is rounded to 1 decimal place.
1000	c2	\$1000.00	The Currency symbol is appended to number and number is rounded to 2 decimal place.

Grouping separator

To separate the y-axis labels to groups of thousands, set the [useGroupingSeparator](#) property to **true** in the 3D chart.

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 10, y: 7000 }, { x: 20, y: 1000 }, { x: 30, y: 12000 }, { x:
40, y: 14000 }, { x: 50, y: 11000 }, { x: 60, y: 5000 },
        { x: 70, y: 7300 }, { x: 80, y: 9000 }, { x: 90, y: 12000 }, { x:
100, y: 14000 }, { x: 110, y: 11000 }, { x: 120, y: 5000 }
    ];
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
        useGroupingSeparator={true}
        primaryXAxis={{
            valueType: 'Double'
        }}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, DataLabel3D,
Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
```

```

        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 10, y: 7000 }, { x: 20, y: 1000 }, { x: 30, y: 12000 }, { x:
40, y: 14000 }, { x: 50, y: 11000 }, { x: 60, y: 5000 },
        { x: 70, y: 7300 }, { x: 80, y: 9000 }, { x: 90, y: 12000 }, { x:
100, y: 14000 }, { x: 110, y: 11000 }, { x: 120, y: 5000 }
    ];
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
useGroupingSeparator={true}
primaryXAxis={{
    valueType: 'Double'
}}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, DataLabel3D,
Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Custom label format

The axis supports custom label format using placeholder like {value}°C, in which the value represent the axis label e.g 20°C.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";

```

```

import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 1, y: 7 }, { x: 2, y: 1 }, { x: 3, y: 1 }, { x: 4, y: 14 }, {
x: 5, y: 1 }, { x: 6, y: 10 },
        { x: 7, y: 8 }, { x: 8, y: 6 }, { x: 9, y: 10 }, { x: 10, y: 10 }, {
x: 11, y: 16 }, { x: 12, y: 6 },
        { x: 13, y: 14 }, { x: 14, y: 7 }, { x: 15, y: 5 }, { x: 16, y: 2 },
{ x: 17, y: 14 }, { x: 18, y: 7 },
        { x: 19, y: 7 }, { x: 20, y: 10 }
    ];
    const yAxis = { labelFormat: '${value}K' };
    return <Chart3DComponent id='charts'
primaryXAxis={{
    valueType: 'Double'
}}
primaryYAxis={yAxis}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, DataLabel3D,
Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
                </Chart3DSeriesDirective>
            </Chart3DSeriesCollectionDirective>
        </Chart3DComponent>;
    }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 1, y: 7 }, { x: 2, y: 1 }, { x: 3, y: 1 }, { x: 4, y: 14 }, {
x: 5, y: 1 }, { x: 6, y: 10 },
        { x: 7, y: 8 }, { x: 8, y: 6 }, { x: 9, y: 10 }, { x: 10, y: 10 }, {
x: 11, y: 16 }, { x: 12, y: 6 },
        { x: 13, y: 14 }, { x: 14, y: 7 }, { x: 15, y: 5 }, { x: 16, y: 2 },
{ x: 17, y: 14 }, { x: 18, y: 7 },
        { x: 19, y: 7 }, { x: 20, y: 10 }
    ];
    const yAxis = { labelFormat: '${value}K' };
    return <Chart3DComponent id='charts'
primaryXAxis={{
    valueType: 'Double'
}}
primaryYAxis={yAxis}

```

```

        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Legend3D, Tooltip3D, DataLabel3D,
Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
                </Chart3DSeriesDirective>
            </Chart3DSeriesCollectionDirective>
        </Chart3DComponent>;
    };
    export default App;
    ReactDOM.render(<App />, document.getElementById("charts"));

```

DateTime axis in React 3D Chart control

DateTime axis

DateTime axis uses date time scale and displays the date time values as axis labels in the specified format.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, DateTime3D, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: new Date(2000, 6, 11), y: 10 },
        { x: new Date(2002, 3, 7), y: 30 },
        { x: new Date(2004, 3, 6), y: 15 },
        { x: new Date(2006, 3, 30), y: 65 },
        { x: new Date(2008, 3, 8), y: 90 },
        { x: new Date(2010, 3, 8), y: 85 }
    ];
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
        // Date time scale in primary X Axis
        valueType: 'DateTime',
    }}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, DateTime3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
                </Chart3DSeriesDirective>
            </Chart3DSeriesCollectionDirective>
        </Chart3DComponent>;
    }
    ;
    export default App;
    ReactDOM.render(<App />, document.getElementById("charts"));

```


INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, DateTime3D, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: new Date(2000, 6, 11), y: 10 },
        { x: new Date(2002, 3, 7), y: 30 },
        { x: new Date(2004, 3, 6), y: 15 },
        { x: new Date(2006, 3, 30), y: 65 },
        { x: new Date(2008, 3, 8), y: 90 },
        { x: new Date(2010, 3, 8), y: 85 }
    ];
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
    primaryXAxis={{
        // Date time scale in primary X Axis
        valueType: 'DateTime',
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, DateTime3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Note: To use datetime axis, we need to inject the **DateTime3D** module into the services and set the [valueType](#) of axis to **DateTime**.

DateTime category axis

DateTime category axis is used to display the date time values with non-linear intervals. For example, the business days alone have been depicted in a week here.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, DateTimeCategory3D, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [

```

```

    { x: new Date(2017, 11, 20), y: 21 }, { x: new Date(2017, 11, 21),
y: 24 },
    { x: new Date(2017, 11, 22), y: 24 }, { x: new Date(2017, 11, 26),
y: 70 },
    { x: new Date(2017, 11, 27), y: 75 }, { x: new Date(2018, 0, 2), y:
82 },
    { x: new Date(2018, 0, 3), y: 53 }, { x: new Date(2018, 0, 4), y: 54
},
    { x: new Date(2018, 0, 5), y: 53 }, { x: new Date(2018, 0, 8), y: 45
}
];

return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
    valueType: 'DateTimeCategory',
    skeleton: 'Ed',
}}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, DateTimeCategory3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, DateTimeCategory3D, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: new Date(2017, 11, 20), y: 21 }, { x: new Date(2017, 11, 21),
y: 24 },
        { x: new Date(2017, 11, 22), y: 24 }, { x: new Date(2017, 11, 26),
y: 70 },
        { x: new Date(2017, 11, 27), y: 75 }, { x: new Date(2018, 0, 2), y:
82 },
        { x: new Date(2018, 0, 3), y: 53 }, { x: new Date(2018, 0, 4), y: 54
},
        { x: new Date(2018, 0, 5), y: 53 }, { x: new Date(2018, 0, 8), y: 45
}
    ];

    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{

```

```

        valueType: 'DateTimeCategory',
        skeleton: 'Ed',
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, DateTimeCategory3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Note: To use datetime category axis, we need to inject **DateTimeCategory3D** module into the services and set the **valueType** of axis to **DateTimeCategory**.

Range

Range of an axis will be calculated automatically based on the provided data. You can also customize the range of an axis using [minimum](#), [maximum](#) and [interval](#) properties.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, DateTime3D, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: new Date(2000, 6, 11), y: 10 }, { x: new Date(2002, 3, 7), y:
30 },
        { x: new Date(2004, 3, 6), y: 15 }, { x: new Date(2006, 3, 30), y:
65 },
        { x: new Date(2008, 3, 8), y: 90 }, { x: new Date(2010, 3, 8), y: 85
    }
    ];
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
        valueType: 'DateTime',
        minimum: new Date(2000, 6, 1),
        maximum: new Date(2010, 6, 1), interval: 1
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, DateTime3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>

```

```

    </Chart3DComponent>;
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, DateTime3D, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: new Date(2000, 6, 11), y: 10 }, { x: new Date(2002, 3, 7), y:
30 },
    { x: new Date(2004, 3, 6), y: 15 }, { x: new Date(2006, 3, 30), y:
65 },
    { x: new Date(2008, 3, 8), y: 90 }, { x: new Date(2010, 3, 8), y: 85
  }
];
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
  valueType: 'DateTime',
  minimum: new Date(2000, 6, 1),
  maximum: new Date(2010, 6, 1), interval: 1
}}
  wallColor='transparent'
  enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, DateTime3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Interval customization

Date time intervals can be customized by using the [interval](#) and [intervalType](#) properties of the axis. For example, when you set `interval` as 2 and `intervalType` as `Years`, it considers 2 years as interval.

Datetime axis supports following interval types,

- Auto
- Years
- Months
- Days
- Hours

- Minutes
- Seconds

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, DateTime3D, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: new Date(2000, 6, 11), y: 10 }, { x: new Date(2002, 3, 7), y:
30 },
        { x: new Date(2004, 3, 6), y: 15 }, { x: new Date(2006, 3, 30), y:
65 },
        { x: new Date(2008, 3, 8), y: 90 }, { x: new Date(2010, 3, 8), y: 85
    }
    ];
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
    valueType: 'DateTime',
    intervalType: 'Years'
}}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, DateTime3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, DateTime3D, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: new Date(2000, 6, 11), y: 10 }, { x: new Date(2002, 3, 7), y:
30 },
        { x: new Date(2004, 3, 6), y: 15 }, { x: new Date(2006, 3, 30), y:
65 },
        { x: new Date(2008, 3, 8), y: 90 }, { x: new Date(2010, 3, 8), y: 85
    }
    ]
}
```

```

];
return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
  valueType: 'DateTime',
  intervalType: 'Years'
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
  <Inject services={[ColumnSeries3D, DateTime3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
  <Chart3DSeriesCollectionDirective >
    <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Applying padding to the range

Padding can be applied to the minimum and maximum extremes of the range by using the [rangePadding](#) property. Date time axis supports the following types of padding,

- None
- Round
- Additional

DateTime - None

When the [rangePadding](#) is set to **None**, minimum and maximum of an axis is based on the data.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, DateTime3D, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: new Date(2000, 6, 11), y: 10 }, { x: new Date(2002, 3, 7), y:
30 },
    { x: new Date(2004, 3, 6), y: 15 }, { x: new Date(2006, 3, 30), y:
65 },
    { x: new Date(2008, 3, 8), y: 90 }, { x: new Date(2010, 3, 8), y: 85
  }
];
return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
  valueType: 'DateTime',
  //Range padding as none
  rangePadding: 'None'
}}

```

```

    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, DateTime3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, DateTime3D, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: new Date(2000, 6, 11), y: 10 }, { x: new Date(2002, 3, 7), y:
30 },
        { x: new Date(2004, 3, 6), y: 15 }, { x: new Date(2006, 3, 30), y:
65 },
        { x: new Date(2008, 3, 8), y: 90 }, { x: new Date(2010, 3, 8), y: 85
    }
    ];
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
        valueType: 'DateTime',
        //Range padding as none
        rangePadding: 'None'
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, DateTime3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

DateTime - Round

When the [rangePadding](#) is set to **Round**, minimum and maximum will be rounded to the nearest possible value, which is divisible by interval. For example, when the minimum is **15th Jan**, interval is **1** and interval type is **Month**, then the axis minimum will be **Jan 1st**.

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, DateTime3D, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: new Date(2000, 6, 11), y: 10 }, { x: new Date(2002, 3, 7), y:
30 },
        { x: new Date(2004, 3, 6), y: 15 }, { x: new Date(2006, 3, 30), y:
65 },
        { x: new Date(2008, 3, 8), y: 90 }, { x: new Date(2010, 3, 8), y: 85
    }
    ];
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
        valueType: 'DateTime',
        rangePadding: 'Round'
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, DateTime3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, DateTime3D, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: new Date(2000, 6, 11), y: 10 }, { x: new Date(2002, 3, 7), y:
30 },
        { x: new Date(2004, 3, 6), y: 15 }, { x: new Date(2006, 3, 30), y:
65 },
    ];
```



```

    { x: new Date(2008, 3, 8), y: 90 }, { x: new Date(2010, 3, 8), y: 85
  }
];
return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
  valueType: 'DateTime',
  rangePadding: 'Round'
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
  <Inject services={[ColumnSeries3D, DateTime3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
  <Chart3DSeriesCollectionDirective >
    <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

DateTime - Additional

When the [rangePadding](#) is set to **Additional**, interval of an axis will be padded to the minimum and maximum of the axis.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, DateTime3D, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: new Date(2000, 6, 11), y: 10 }, { x: new Date(2002, 3, 7), y:
30 },
    { x: new Date(2004, 3, 6), y: 15 }, { x: new Date(2006, 3, 30), y:
65 },
    { x: new Date(2008, 3, 8), y: 90 }, { x: new Date(2010, 3, 8), y: 85
  }
];
return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
  valueType: 'DateTime',
  rangePadding: 'Additional'
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
  <Inject services={[ColumnSeries3D, DateTime3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
  <Chart3DSeriesCollectionDirective >
    <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>

```

```

        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, DateTime3D, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: new Date(2000, 6, 11), y: 10 }, { x: new Date(2002, 3, 7), y:
30 },
    { x: new Date(2004, 3, 6), y: 15 }, { x: new Date(2006, 3, 30), y:
65 },
    { x: new Date(2008, 3, 8), y: 90 }, { x: new Date(2010, 3, 8), y: 85
  }
  ];
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
  valueType: 'DateTime',
  rangePadding: 'Additional'
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
<Inject services={[ColumnSeries3D, DateTime3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
<Chart3DSeriesCollectionDirective >
  <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
    </Chart3DSeriesDirective>
  </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Label format

The date can be formatted and parsed to all globalize format using the [labelFormat](#) property in an axis.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, DateTime3D, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {

```

```

const data = [
  { x: new Date(2000, 6, 11), y: 10 }, { x: new Date(2002, 3, 7), y:
30 },
  { x: new Date(2004, 3, 6), y: 15 }, { x: new Date(2006, 3, 30), y:
65 },
  { x: new Date(2008, 3, 8), y: 90 }, { x: new Date(2010, 3, 8), y: 85
}
];
return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
  valueType: 'DateTime',
  //Label format as yMd
  labelFormat: 'yMd'
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
  <Inject services={[ColumnSeries3D, DateTime3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
  <Chart3DSeriesCollectionDirective >
    <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, DateTime3D, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: new Date(2000, 6, 11), y: 10 }, { x: new Date(2002, 3, 7), y:
30 },
    { x: new Date(2004, 3, 6), y: 15 }, { x: new Date(2006, 3, 30), y:
65 },
    { x: new Date(2008, 3, 8), y: 90 }, { x: new Date(2010, 3, 8), y: 85
  }
  ];
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
  valueType: 'DateTime',
  //Label format as yMd
  labelFormat: 'yMd'
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, DateTime3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />

```

```

    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
    </Chart3DSeriesDirective>
  </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

<!-- markdownlint-disable MD033 -->

Label Value	Label Format Property Value	Result	Description
new Date(2000, 03, 10)	EEEE	Monday	The Date is displayed in day format.
new Date(2000, 03, 10)	yMd	04/10/2000	The Date is displayed in month/date/year format.
new Date(2000, 03, 10)	MMM	Apr	The Shorthand month for the date is displayed.
new Date(2000, 03, 10)	hm	12:00 AM	Time of the date value is displayed as label.
new Date(2000, 03, 10)	hms	12:00:00 AM	The Label is displayed in hours:minutes:seconds format.

Custom label format

Axis also supports custom label format using placeholder like {value}°C, in which the value represent the axis label e.g 20°C.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, DateTime3D, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: new Date(2000, 6, 11), y: 10 }, { x: new Date(2002, 3, 7), y:
30 },
    { x: new Date(2004, 3, 6), y: 15 }, { x: new Date(2006, 3, 30), y:
65 },
    { x: new Date(2008, 3, 8), y: 90 }, { x: new Date(2010, 3, 8), y: 85
  }
  ];
  const yAxis= { labelFormat: '${value}K' };
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
    valueType: 'DateTime'

```

```

    }}
    primaryYAxis={yAxis}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, DateTime3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, DateTime3D, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: new Date(2000, 6, 11), y: 10 }, { x: new Date(2002, 3, 7), y:
30 },
        { x: new Date(2004, 3, 6), y: 15 }, { x: new Date(2006, 3, 30), y:
65 },
        { x: new Date(2008, 3, 8), y: 90 }, { x: new Date(2010, 3, 8), y: 85
    }
    ];
    const yAxis= { labelFormat: '${value}K' };
    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}>
        primaryXAxis={{
            valueType: 'DateTime'
        }}
        primaryYAxis={yAxis}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, DateTime3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
                </Chart3DSeriesDirective>
            </Chart3DSeriesCollectionDirective>
        </Chart3DComponent>;
    };
    export default App;
    ReactDOM.render(<App />, document.getElementById("charts"));
}

```

Logarithmic axis in React 3D Chart control

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).

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective, Logarithmic3D,
Chart3DSeriesDirective, Inject, DateTime3D, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: new Date(1995, 0, 1), y: 80 }, { x: new Date(1996, 0, 1), y:
200 },
        { x: new Date(1997, 0, 1), y: 400 }, { x: new Date(1998, 0, 1), y:
600 },
        { x: new Date(1999, 0, 1), y: 700 }, { x: new Date(2000, 0, 1), y:
1400 },
        { x: new Date(2001, 0, 1), y: 2000 }, { x: new Date(2002, 0, 1), y:
4000 },
        { x: new Date(2003, 0, 1), y: 6000 }, { x: new Date(2004, 0, 1), y:
8000 },
        { x: new Date(2005, 0, 1), y: 11000 }
    ];

    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
        // Date time scale in primary X Axis
        valueType: 'DateTime',
    }}
primaryYAxis={{
        // Logarithmic scale in primary X Axis
        valueType: 'Logarithmic',
    }}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Logarithmic3D, DateTime3D,
Legend3D, Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
                </Chart3DSeriesDirective>
            </Chart3DSeriesCollectionDirective>
        </Chart3DComponent>;
    }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective, Logarithmic3D,
Chart3DSeriesDirective, Inject, DateTime3D, Legend3D, DataLabel3D,
```

```

Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: new Date(1995, 0, 1), y: 80 }, { x: new Date(1996, 0, 1), y:
200 },
        { x: new Date(1997, 0, 1), y: 400 }, { x: new Date(1998, 0, 1), y:
600 },
        { x: new Date(1999, 0, 1), y: 700 }, { x: new Date(2000, 0, 1), y:
1400 },
        { x: new Date(2001, 0, 1), y: 2000 }, { x: new Date(2002, 0, 1), y:
4000 },
        { x: new Date(2003, 0, 1), y: 6000 }, { x: new Date(2004, 0, 1), y:
8000 },
        { x: new Date(2005, 0, 1), y: 11000 }
    ];

    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
    // Date time scale in primary X Axis
    valueType: 'DateTime',
}}
primaryYAxis={{
    // Logarithmic scale in primary X Axis
    valueType: 'Logarithmic',
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
<Inject services={[ColumnSeries3D, Logarithmic3D, DateTime3D,
Legend3D, Tooltip3D, DataLabel3D, Highlight3D]} />
<Chart3DSeriesCollectionDirective >
    <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Note: To use logarithmic axis, we need to inject **Logarithmic3D** module into the services and set the [valueType](#) of the axis to **Logarithmic**.

Range

The range of an axis will be calculated automatically based on the provided data and it can also be customized by using the [minimum](#), [maximum](#) and [interval](#) properties of the axis.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective, Logarithmic3D,
Chart3DSeriesDirective, Inject, DateTime3D, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";

```

```

import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: new Date(1995, 0, 1), y: 80 }, { x: new Date(1996, 0, 1), y:
200 },
        { x: new Date(1997, 0, 1), y: 400 }, { x: new Date(1998, 0, 1), y:
600 },
        { x: new Date(1999, 0, 1), y: 700 }, { x: new Date(2000, 0, 1), y:
1400 },
        { x: new Date(2001, 0, 1), y: 2000 }, { x: new Date(2002, 0, 1), y:
4000 },
        { x: new Date(2003, 0, 1), y: 6000 }, { x: new Date(2004, 0, 1), y:
8000 },
        { x: new Date(2005, 0, 1), y: 11000 }
    ];

    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
    valueType: 'DateTime',
}}
primaryYAxis={{
    valueType: 'Logarithmic',
    minimum: 100,
    maximum: 10000,
    interval: 1000
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Logarithmic3D, DateTime3D,
Legend3D, Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective, Logarithmic3D,
Chart3DSeriesDirective, Inject, DateTime3D, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: new Date(1995, 0, 1), y: 80 }, { x: new Date(1996, 0, 1), y:
200 },
        { x: new Date(1997, 0, 1), y: 400 }, { x: new Date(1998, 0, 1), y:
600 },

```



```

    { x: new Date(1999, 0, 1), y: 700 }, { x: new Date(2000, 0, 1), y:
1400 },
    { x: new Date(2001, 0, 1), y: 2000 }, { x: new Date(2002, 0, 1), y:
4000 },
    { x: new Date(2003, 0, 1), y: 6000 }, { x: new Date(2004, 0, 1), y:
8000 },
    { x: new Date(2005, 0, 1), y: 11000 }
  ];

  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
    primaryXAxis={{
      valueType: 'DateTime',
    }}
    primaryYAxis={{
      valueType: 'Logarithmic',
      minimum: 100,
      maximum: 10000,
      interval: 1000
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Logarithmic3D, DateTime3D,
Legend3D, Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Logarithmic base

Logarithmic base can be customized by using the [logBase](#) property of the axis. For example when the `logBase` is 5, the axis values follows 5^{-2} , 5^{-1} , 5^0 , 5^1 , 5^2 etc.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective, Logarithmic3D,
Chart3DSeriesDirective, Inject, DateTime3D, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: new Date(1995, 0, 1), y: 80 }, { x: new Date(1996, 0, 1), y:
200 },
    { x: new Date(1997, 0, 1), y: 400 }, { x: new Date(1998, 0, 1), y:
600 },
    { x: new Date(1999, 0, 1), y: 700 }, { x: new Date(2000, 0, 1), y:
1400 },
    { x: new Date(2001, 0, 1), y: 2000 }, { x: new Date(2002, 0, 1), y:
4000 },

```

```

    { x: new Date(2003, 0, 1), y: 6000 }, { x: new Date(2004, 0, 1), y:
8000 },
    { x: new Date(2005, 0, 1), y: 11000 }
  ];

  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
    primaryXAxis={{
      valueType: 'DateTime',
    }}
    primaryYAxis={{
      valueType: 'Logarithmic',
      logBase: 2
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Logarithmic3D, DateTime3D,
Legend3D, Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective, Logarithmic3D,
Chart3DSeriesDirective, Inject, DateTime3D, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: new Date(1995, 0, 1), y: 80 }, { x: new Date(1996, 0, 1), y:
200 },
    { x: new Date(1997, 0, 1), y: 400 }, { x: new Date(1998, 0, 1), y:
600 },
    { x: new Date(1999, 0, 1), y: 700 }, { x: new Date(2000, 0, 1), y:
1400 },
    { x: new Date(2001, 0, 1), y: 2000 }, { x: new Date(2002, 0, 1), y:
4000 },
    { x: new Date(2003, 0, 1), y: 6000 }, { x: new Date(2004, 0, 1), y:
8000 },
    { x: new Date(2005, 0, 1), y: 11000 }
  ];

  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
    primaryXAxis={{
      valueType: 'DateTime',
    }}
    primaryYAxis={{

```

```

        valueType: 'Logarithmic',
        logBase: 2
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Logarithmic3D, DateTime3D,
    Legend3D, Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Logarithmic interval

The interval of the logarithmic axis can be customized by using the [interval](#) property in the axis. When the logarithmic base is 10 and logarithmic **interval** is 2, then the axis labels are placed at an interval of 10². The default value of the [interval](#) is 1.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective, Logarithmic3D,
Chart3DSeriesDirective, Inject, DateTime3D, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: new Date(1995, 0, 1), y: 80 }, { x: new Date(1996, 0, 1), y:
200 },
        { x: new Date(1997, 0, 1), y: 400 }, { x: new Date(1998, 0, 1), y:
600 },
        { x: new Date(1999, 0, 1), y: 700 }, { x: new Date(2000, 0, 1), y:
1400 },
        { x: new Date(2001, 0, 1), y: 2000 }, { x: new Date(2002, 0, 1), y:
4000 },
        { x: new Date(2003, 0, 1), y: 6000 }, { x: new Date(2004, 0, 1), y:
8000 },
        { x: new Date(2005, 0, 1), y: 11000 }
    ];

    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
        valueType: 'DateTime',
    }}
primaryYAxis={{
        valueType: 'Logarithmic',
        interval: 2
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>

```

```

    <Inject services={[ColumnSeries3D, Logarithmic3D, DateTime3D,
Legend3D, Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective, Logarithmic3D,
Chart3DSeriesDirective, Inject, DateTime3D, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: new Date(1995, 0, 1), y: 80 }, { x: new Date(1996, 0, 1), y:
200 },
        { x: new Date(1997, 0, 1), y: 400 }, { x: new Date(1998, 0, 1), y:
600 },
        { x: new Date(1999, 0, 1), y: 700 }, { x: new Date(2000, 0, 1), y:
1400 },
        { x: new Date(2001, 0, 1), y: 2000 }, { x: new Date(2002, 0, 1), y:
4000 },
        { x: new Date(2003, 0, 1), y: 6000 }, { x: new Date(2004, 0, 1), y:
8000 },
        { x: new Date(2005, 0, 1), y: 11000 }
    ];

    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
        valueType: 'DateTime',
    }}
primaryYAxis={{
        valueType: 'Logarithmic',
        interval: 2
    }}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Logarithmic3D, DateTime3D,
Legend3D, Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
                </Chart3DSeriesDirective>
            </Chart3DSeriesCollectionDirective>
        </Chart3DComponent>;
};
export default App;

```

```
ReactDOM.render(<App />, document.getElementById("charts"));
```

Axis labels in React 3D Chart control

Axis labels are the labels that are positioned adjacent to the y-axis and beneath the x-axis. It provides descriptive information about the axis.

Smart axis labels

When the axis labels overlap with each other, [labelIntersectAction](#) property in the axis can be used to place them smartly.

Case 1: When setting `labelIntersectAction` as `Hide`.

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: "South Korea", y: 39.4 },
    { x: "India", y: 61.3 },
    { x: "Pakistan", y: 20.4 },
    { x: "Germany", y: 65.1 },
    { x: "Australia", y: 15.8 },
    { x: "Italy", y: 29.2 },
    { x: "United Kingdom", y: 44.6 },
    { x: "Saudi Arabia", y: 9.7 },
    { x: "Russia", y: 40.8 },
    { x: "Mexico", y: 31 },
    { x: "Brazil", y: 75.9 },
    { x: "China", y: 51.4 }
  ];

  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
    primaryXAxis={{
      valueType: 'Category',
      labelIntersectAction: 'Hide'
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: "South Korea", y: 39.4 },
        { x: "India", y: 61.3 },
        { x: "Pakistan", y: 20.4 },
        { x: "Germany", y: 65.1 },
        { x: "Australia", y: 15.8 },
        { x: "Italy", y: 29.2 },
        { x: "United Kingdom", y: 44.6 },
        { x: "Saudi Arabia", y: 9.7 },
        { x: "Russia", y: 40.8 },
        { x: "Mexico", y: 31 },
        { x: "Brazil", y: 75.9 },
        { x: "China", y: 51.4 }
    ];

    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
    primaryXAxis={{
        valueType: 'Category',
        labelIntersectAction: 'Hide'
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Case 2: When setting `labelIntersectAction` as `Rotate45`.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: "South Korea", y: 39.4 },
        { x: "India", y: 61.3 },

```

```

    { x: "Pakistan", y: 20.4 },
    { x: "Germany", y: 65.1 },
    { x: "Australia", y: 15.8 },
    { x: "Italy", y: 29.2 },
    { x: "United Kingdom", y: 44.6 },
    { x: "Saudi Arabia", y: 9.7 },
    { x: "Russia", y: 40.8 },
    { x: "Mexico", y: 31 },
    { x: "Brazil", y: 75.9 },
    { x: "China", y: 51.4 }
  ];

  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
    primaryXAxis={{
      valueType: 'Category',
      labelIntersectAction: 'Rotate45'
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D, Tooltip3D,
    DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: "South Korea", y: 39.4 },
    { x: "India", y: 61.3 },
    { x: "Pakistan", y: 20.4 },
    { x: "Germany", y: 65.1 },
    { x: "Australia", y: 15.8 },
    { x: "Italy", y: 29.2 },
    { x: "United Kingdom", y: 44.6 },
    { x: "Saudi Arabia", y: 9.7 },
    { x: "Russia", y: 40.8 },
    { x: "Mexico", y: 31 },
    { x: "Brazil", y: 75.9 },
    { x: "China", y: 51.4 }
  ];
};

```

```

return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
  valueType: 'Category',
  labelIntersectAction: 'Rotate45'
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
<Inject services={[ColumnSeries3D, Category3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
<Chart3DSeriesCollectionDirective >
  <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
    </Chart3DSeriesDirective>
  </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Case 3: When setting `labelIntersectAction` as `Rotate90`.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: "South Korea", y: 39.4 },
    { x: "India", y: 61.3 },
    { x: "Pakistan", y: 20.4 },
    { x: "Germany", y: 65.1 },
    { x: "Australia", y: 15.8 },
    { x: "Italy", y: 29.2 },
    { x: "United Kingdom", y: 44.6 },
    { x: "Saudi Arabia", y: 9.7 },
    { x: "Russia", y: 40.8 },
    { x: "Mexico", y: 31 },
    { x: "Brazil", y: 75.9 },
    { x: "China", y: 51.4 }
  ];

  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
  valueType: 'Category',
  labelIntersectAction: 'Rotate90'
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
<Inject services={[ColumnSeries3D, Category3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
  <Chart3DSeriesCollectionDirective >

```



```

        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: "South Korea", y: 39.4 },
        { x: "India", y: 61.3 },
        { x: "Pakistan", y: 20.4 },
        { x: "Germany", y: 65.1 },
        { x: "Australia", y: 15.8 },
        { x: "Italy", y: 29.2 },
        { x: "United Kingdom", y: 44.6 },
        { x: "Saudi Arabia", y: 9.7 },
        { x: "Russia", y: 40.8 },
        { x: "Mexico", y: 31 },
        { x: "Brazil", y: 75.9 },
        { x: "China", y: 51.4 }
    ];

    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
    valueType: 'Category',
    labelIntersectAction: 'Rotate90'
}}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
                </Chart3DSeriesDirective>
            </Chart3DSeriesCollectionDirective>
        </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Edge label placement

Labels with long text at the edges of an axis may appear partially in the 3D chart. To avoid this,

use the [edgeLabelPlacement](#) property in axis, which moves the label inside the chart area for better appearance or hides it.

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: "South Korea", y: 39.4 },
        { x: "India", y: 61.3 },
        { x: "Pakistan", y: 20.4 },
        { x: "Germany", y: 65.1 },
        { x: "Australia", y: 15.8 },
        { x: "Italy", y: 29.2 },
        { x: "United Kingdom", y: 44.6 },
        { x: "Saudi Arabia", y: 9.7 },
        { x: "Russia", y: 40.8 },
        { x: "Mexico", y: 31 },
        { x: "Brazil", y: 75.9 },
        { x: "China", y: 51.4 }
    ];

    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
    primaryXAxis={{
        valueType: 'Category',
        edgeLabelPlacement: 'Shift',
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
```

```

const data = [
  { x: "South Korea", y: 39.4 },
  { x: "India", y: 61.3 },
  { x: "Pakistan", y: 20.4 },
  { x: "Germany", y: 65.1 },
  { x: "Australia", y: 15.8 },
  { x: "Italy", y: 29.2 },
  { x: "United Kingdom", y: 44.6 },
  { x: "Saudi Arabia", y: 9.7 },
  { x: "Russia", y: 40.8 },
  { x: "Mexico", y: 31 },
  { x: "Brazil", y: 75.9 },
  { x: "China", y: 51.4 }
];

return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
  primaryXAxis={{
    valueType: 'Category',
    edgeLabelPlacement: 'Shift',
  }}
  wallColor='transparent'
  enableRotation={true} rotation={7} tilt={10} depth={100}>
  <Inject services={[ColumnSeries3D, Category3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
  <Chart3DSeriesCollectionDirective >
    <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Column'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Maximum labels

The labels will be rendered based on the count in the [maximumLabels](#) property per 100 pixel. If the range (minimum, maximum, interval) and [maximumLabels](#) are set, then the priority goes to range. If the range is not set, then the priority goes to [maximumLabels](#) property.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  var series1 = [];
  var value = 80;
  for (var i = 1; i < 50; i++) {
    if (Math.random() > .5) {
      value += (1 + Math.random()*10);
    } else {
      value -= (1 + Math.random()*10);
    }
  }
}

```

```

        series1.push( { x: i, y: value.toFixed(1) } );
    }
    const xAxis = {
        title: 'Years', edgeLabelPlacement: 'Shift', majorGridLines: {
width: 0 }, maximumLabels: 1
    }

    return <Chart3DComponent id='charts'
primaryXAxis = {xAxis}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={series1} xName='x' yName='y'
type='Column'>
                </Chart3DSeriesDirective>
            </Chart3DSeriesCollectionDirective>
        </Chart3DComponent>;
    }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Inject, Legend3D, DataLabel3D, Category3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
export let series1: Object[] = [];
export let value = 80;
function App() {
    for (var i = 1; i < 50; i++) {
        if (Math.random() > .5) {
            value += (1 + Math.random()*10);
        } else {
            value -= (1 + Math.random()*10);
        }
        series1.push( { x: i, y: value.toFixed(1) } );
    }
    const xAxis = {
        title: 'Years', edgeLabelPlacement: 'Shift', majorGridLines: {
width: 0 }, maximumLabels: 1
    }

    return <Chart3DComponent id='charts'
primaryXAxis = {xAxis}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >

```

```

        <Chart3DSeriesDirective dataSource={series1} xName='x' yName='y'
type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Axis customization in React 3D Chart control

Title

The title for the axis can be added by using the [title](#) property. It helps 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

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, DateTime3D, Legend3D,
DataLabel3D, Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-
react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];
    const xAxis = {
        valueType: "Category", titleStyle: { size: '16px', color: 'grey',
fontFamily : 'Segoe UI', fontWeight : 'bold' }, title: 'Countries'
    };

    return <Chart3DComponent id='charts' primaryXAxis={xAxis}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, DateTime3D, Legend3D,
DataLabel3D, Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-
react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];
    const xAxis = {
        valueType: "Category", titleStyle: { size: '16px', color: 'grey',
fontFamily : 'Segoe UI', fontWeight : 'bold' }, title: 'Countries'
    };

    return <Chart3DComponent id='charts' primaryXAxis={xAxis}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Title rotation

The title can be rotated from 0 to 360 degree by using the [titleRotation](#) property.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, DateTime3D, Legend3D,
DataLabel3D, Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-
react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },

```

```

    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  const xAxis = {
    valueType: "Category",
    title: 'Countries',
    titleRotation: 90,
    titleStyle: {
      size: '16px', color: 'grey',
      fontFamily : 'Segoe UI', fontWeight : 'bold'
    }
  };

  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
    primaryXAxis={xAxis}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, DateTime3D, Legend3D,
DataLabel3D, Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-
react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];
  const xAxis = {
    valueType: "Category",
    title: 'Countries',
    titleRotation: 90,
    titleStyle: {

```

```

        size: '16px', color: 'grey',
        fontFamily : 'Segoe UI', fontWeight : 'bold'
    }
};

return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={xAxis}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Tick lines customization

The [width](#), [color](#) and [height](#) of the minor and major tick lines can be customized by using the [majorTickLines](#) and [minorTickLines](#) properties in the axis.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, DateTime3D, Legend3D,
DataLabel3D, Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-
react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];
    const xAxis = {
        valueType: "Category",
        majorTickLines : {
            color : 'blue',
            width : 5
        }
    };
};

return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={xAxis}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>

```



```

        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
                </Chart3DSeriesDirective>
            </Chart3DSeriesCollectionDirective>
        </Chart3DComponent>;
    }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, DateTime3D, Legend3D,
DataLabel3D, Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-
react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];
    const xAxis = {
        valueType: "Category",
        majorTickLines : {
            color : 'blue',
            width : 5
        }
    };

    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={xAxis}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
                </Chart3DSeriesDirective>
            </Chart3DSeriesCollectionDirective>
        </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Grid lines customization

The [width](#) and [color](#) of the minor and major grid lines can be customized by using the [majorGridLines](#) and [minorGridLines](#) properties in the axis.

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, DateTime3D, Legend3D,
DataLabel3D, Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];
    const xAxis = {
        valueType: "Category",
        majorGridLines : {
            color : 'blue',
            width : 1
        }
    };

    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
    primaryXAxis={xAxis}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
        Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
                </Chart3DSeriesDirective>
            </Chart3DSeriesCollectionDirective>
        </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, DateTime3D, Legend3D,
DataLabel3D, Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
```

```
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  const xAxis = {
    valueType: "Category",
    majorGridLines : {
      color : 'blue',
      width : 1
    }
  };

  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
    primaryXAxis={xAxis}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Multiple axis

In addition to primary X and Y axis, n number of axis can be added to the chart. Series can be associated with this axis, by mapping with axis's unique name.

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Chart3DAxesDirective,
Chart3DAxisDirective, Legend3D, DataLabel3D, Tooltip3D, ColumnSeries3D,
Highlight3D } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
  ];
```

```

    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
    primaryXAxis={{
      valueType: "Category",
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Chart3DAxesDirective>
      <Chart3DAxisDirective rowIndex={0} name="yAxis"
        opposedPosition={true} title='Temperature (Celsius)'>
        </Chart3DAxisDirective>
      </Chart3DAxesDirective>
      <Inject services={[ColumnSeries3D, Category3D, Legend3D,
        Tooltip3D, DataLabel3D, Highlight3D]} />
      <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
          yName='gold' type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
          yName='silver' yAxisName="yAxis" type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
  Chart3DSeriesDirective, Category3D, Inject, Chart3DAxesDirective,
  Chart3DAxisDirective, Legend3D, DataLabel3D, Tooltip3D, ColumnSeries3D,
  Highlight3D } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
    primaryXAxis={{
      valueType: "Category",
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>

```

```

    <Chart3DAxesDirective>
      <Chart3DAxisDirective rowIndex={0} name="yAxis"
opposedPosition={true} title='Temperature (Celsius)'>
    </Chart3DAxisDirective>
  </Chart3DAxesDirective>
  <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
  <Chart3DSeriesCollectionDirective >
    <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
    </Chart3DSeriesDirective>
    <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' yAxisName="yAxis" type='Column'>
    </Chart3DSeriesDirective>
  </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Inversed axis

<!-- markdownlint-disable MD033 -->

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 the [isInversed](#) property to **true**.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, DateTime3D, Legend3D,
DataLabel3D, Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-
react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
    primaryXAxis={{
      valueType: "Category",
      isInversed: true
    }}
    primaryYAxis={{
      isInversed: true
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>

```

```

        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
                </Chart3DSeriesDirective>
            </Chart3DSeriesCollectionDirective>
        </Chart3DComponent>;
    }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, DateTime3D, Legend3D,
DataLabel3D, Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-
react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
        primaryXAxis={{
            valueType: "Category",
            isInversed: true
        }}
        primaryYAxis={{
            isInversed: true
        }}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
                </Chart3DSeriesDirective>
            </Chart3DSeriesCollectionDirective>
        </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Opposed position

To place an axis opposite from its original position, set the [opposedPosition](#) property to **true**.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
        primaryXAxis={{
            valueType: "Category",
        }}
        primaryYAxis={{
            opposedPosition: true
        }}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
        Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
                </Chart3DSeriesDirective>
            </Chart3DSeriesCollectionDirective>
        </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },

```

```

    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
    primaryXAxis={{
      valueType: "Category",
    }}
    primaryYAxis={{
      opposedPosition: true
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Multiple panes in React 3D Chart control

The chart area can be divided into multiple panes using [rows](#) and [columns](#).

Rows

To split the chart area vertically into number of rows, use [rows](#) property of the 3D chart.

- The space for each row can be allocated by using the [height](#) property. The value can be either in percentage or in pixel.
- To associate a vertical axis to a particular row, specify its index to [rowIndex](#) property of the axis.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Chart3DRowDirective,
Chart3DAxisDirective, Chart3DAxesDirective, Chart3DRowsDirective, Inject,
Legend3D, DataLabel3D, Tooltip3D, ColumnSeries3D, Highlight3D } from
'@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: 'Jan', y: 15, y1: 33 }, { x: 'Feb', y: 20, y1: 31 }, { x:
'Mar', y: 35, y1: 30 },

```



```

    { x: 'Apr', y: 40, y1: 28 }, { x: 'May', y: 80, y1: 29 }, { x:
    'Jun', y: 70, y1: 30 },
    { x: 'Jul', y: 65, y1: 33 }, { x: 'Aug', y: 55, y1: 32 }, { x:
    'Sep', y: 50, y1: 34 },
    { x: 'Oct', y: 30, y1: 32 }, { x: 'Nov', y: 35, y1: 32 }, { x:
    'Dec', y: 35, y1: 31 }
  ];
  const yAxis = { minimum: 0, maximum: 90, labelFormat: '{value}°F',
  title: 'Temperature (Fahrenheit)', interval: 20 };
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
  primaryXAxis={{
    valueType: "Category",
  }}
  primaryYAxis={yAxis} wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DRowsDirective>
      <Chart3DRowDirective height="50%"></Chart3DRowDirective>
      <Chart3DRowDirective height="50%"></Chart3DRowDirective>
    </Chart3DRowsDirective>
    <Chart3DAxesDirective>
      <Chart3DAxisDirective rowIndex={1} name='yAxis1'
    opposedPosition={true} title='Temperature (Celsius)' labelFormat='{value}°C'
    minimum={24} maximum={36} interval={2} majorGridLines={{width:
    0}}></Chart3DAxisDirective>
    </Chart3DAxesDirective>
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x'
    yName='y' type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
    yName='y1' type='Column' yAxisName='yAxis1'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Chart3DRowDirective,
Chart3DAxisDirective, Chart3DAxesDirective, Chart3DRowsDirective, Inject,
Legend3D, DataLabel3D, Tooltip3D, ColumnSeries3D, Highlight3D } from
'@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: 'Jan', y: 15, y1: 33 }, { x: 'Feb', y: 20, y1: 31 }, { x:
    'Mar', y: 35, y1: 30 },
    { x: 'Apr', y: 40, y1: 28 }, { x: 'May', y: 80, y1: 29 }, { x:
    'Jun', y: 70, y1: 30 },

```

```

    { x: 'Jul', y: 65, y1: 33 }, { x: 'Aug', y: 55, y1: 32 }, { x:
'Sep', y: 50, y1: 34 },
    { x: 'Oct', y: 30, y1: 32 }, { x: 'Nov', y: 35, y1: 32 }, { x:
'Dec', y: 35, y1: 31 }
  ];
  const yAxis = { minimum: 0, maximum: 90, labelFormat: '{value}°F',
title: 'Temperature (Fahrenheit)', interval: 20 };
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
    valueType: "Category",
  }}
  primaryYAxis={yAxis} wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DRowsDirective>
      <Chart3DRowDirective height="50%"></Chart3DRowDirective>
      <Chart3DRowDirective height="50%"></Chart3DRowDirective>
    </Chart3DRowsDirective>
    <Chart3DAxesDirective>
      <Chart3DAxisDirective rowIndex={1} name='yAxis1'
opposedPosition={true} title='Temperature (Celsius)' labelFormat='{value}°C'
minimum={24} maximum={36} interval={2} majorGridLines={{width:
0}}></Chart3DAxisDirective>
    </Chart3DAxesDirective>
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='Column'>
    </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='Column' yAxisName="yAxis1">
    </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

For spanning the vertical axis along multiple rows, use [span](#) property of an axis.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Chart3DRowDirective,
Chart3DAxisDirective, Chart3DAxesDirective, Chart3DRowsDirective, Inject,
Legend3D, DataLabel3D, Tooltip3D, ColumnSeries3D, Highlight3D } from
'@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: 'Jan', y: 15, y1: 33 }, { x: 'Feb', y: 20, y1: 31 }, { x:
'Mar', y: 35, y1: 30 },
    { x: 'Apr', y: 40, y1: 28 }, { x: 'May', y: 80, y1: 29 }, { x:
'Jun', y: 70, y1: 30 },

```

```

    { x: 'Jul', y: 65, y1: 33 }, { x: 'Aug', y: 55, y1: 32 }, { x:
'Sep', y: 50, y1: 34 },
    { x: 'Oct', y: 30, y1: 32 }, { x: 'Nov', y: 35, y1: 32 }, { x:
'Dec', y: 35, y1: 31 }
  ];
  const yAxis = { minimum: 0, maximum: 90, interval: 20,
    title: 'Temperature (Fahrenheit)',
    labelFormat: '{value}°F',
    span: 2 };
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
    valueType: "Category",
  }}
  primaryYAxis={yAxis} wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DRowsDirective>
      <Chart3DRowDirective height="50%"></Chart3DRowDirective>
      <Chart3DRowDirective height="50%"></Chart3DRowDirective>
    </Chart3DRowsDirective>
    <Chart3DAxesDirective>
      <Chart3DAxisDirective rowIndex={1} name='yAxis1'
opposedPosition={true} title='Temperature (Celsius)' labelFormat='{value}°C'
minimum={24} maximum={36} interval={2} majorGridLines={{width:
0}}></Chart3DAxisDirective>
    </Chart3DAxesDirective>
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='Column'>
        </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='Column' yAxisName="yAxis1">
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Chart3DRowDirective,
Chart3DAxisDirective, Chart3DAxesDirective, Chart3DRowsDirective, Inject,
Legend3D, DataLabel3D, Tooltip3D, ColumnSeries3D, Highlight3D } from
'@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: 'Jan', y: 15, y1: 33 }, { x: 'Feb', y: 20, y1: 31 }, { x:
'Mar', y: 35, y1: 30 },
    { x: 'Apr', y: 40, y1: 28 }, { x: 'May', y: 80, y1: 29 }, { x:
'Jun', y: 70, y1: 30 },

```

```

    { x: 'Jul', y: 65, y1: 33 }, { x: 'Aug', y: 55, y1: 32 }, { x:
'Sep', y: 50, y1: 34 },
    { x: 'Oct', y: 30, y1: 32 }, { x: 'Nov', y: 35, y1: 32 }, { x:
'Dec', y: 35, y1: 31 }
  ];
  const yAxis = { minimum: 0, maximum: 90, interval: 20,
    title: 'Temperature (Fahrenheit)',
    labelFormat: '{value}°F',
    span: 2 };
  return <Chart3DComponent id='charts' style={{ textAlign: "center" }}
primaryXAxis={{
    valueType: "Category",
  }}
  primaryYAxis={yAxis} wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DRowsDirective>
      <Chart3DRowDirective height="50%"></Chart3DRowDirective>
      <Chart3DRowDirective height="50%"></Chart3DRowDirective>
    </Chart3DRowsDirective>
    <Chart3DAxesDirective>
      <Chart3DAxisDirective rowIndex={1} name='yAxis1'
opposedPosition={true} title='Temperature (Celsius)' labelFormat='{value}°C'
minimum={24} maximum={36} interval={2} majorGridLines={{width:
0}}></Chart3DAxisDirective>
    </Chart3DAxesDirective>
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='Column'>
        </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='Column' yAxisName="yAxis1">
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
  };
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Columns

To split the chart area horizontally into number of columns, use [columns](#) property of the 3D chart.

- The space for each column can be allocated by using the [width](#) property. The given width can be either in percentage or in pixel.
- To associate a horizontal axis to a particular column, specify its index to [columnIndex](#) property of the axis.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Chart3DColumnDirective,
Chart3DAxisDirective, Chart3DAxesDirective, Chart3DColumnsDirective, Inject,

```

```

Legend3D, DataLabel3D, Tooltip3D, ColumnSeries3D, Highlight3D } from
'@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: 'Jan', y: 15, y1: 33 }, { x: 'Feb', y: 20, y1: 31 }, { x:
'Mar', y: 35, y1: 30 },
    { x: 'Apr', y: 40, y1: 28 }, { x: 'May', y: 80, y1: 29 }, { x:
'Jun', y: 70, y1: 30 },
    { x: 'Jul', y: 65, y1: 33 }, { x: 'Aug', y: 55, y1: 32 }, { x:
'Sep', y: 50, y1: 34 },
    { x: 'Oct', y: 30, y1: 32 }, { x: 'Nov', y: 35, y1: 32 }, { x:
'Dec', y: 35, y1: 31 }
  ];
  const yAxis = { minimum: 0, maximum: 90, interval: 20,
    title: 'Temperature (Fahrenheit)',
    labelFormat: '{value}°F' };

  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
    primaryYAxis={yAxis}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DColumnsDirective>
      <Chart3DColumnDirective
width="50%"></Chart3DColumnDirective>
      <Chart3DColumnDirective
width="50%"></Chart3DColumnDirective>
    </Chart3DColumnsDirective>
    <Chart3DAxesDirective>
      <Chart3DAxisDirective columnIndex={1} name='xAxis'
valueType="Category"></Chart3DAxisDirective>
    </Chart3DAxesDirective>
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='Column' xAxisName="xAxis">
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Chart3DColumnDirective,
Chart3DAxisDirective, Chart3DAxesDirective, Chart3DColumnsDirective, Inject,

```

```

Legend3D, DataLabel3D, Tooltip3D, ColumnSeries3D, Highlight3D } from
'@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: 'Jan', y: 15, y1: 33 }, { x: 'Feb', y: 20, y1: 31 }, { x:
'Mar', y: 35, y1: 30 },
    { x: 'Apr', y: 40, y1: 28 }, { x: 'May', y: 80, y1: 29 }, { x:
'Jun', y: 70, y1: 30 },
    { x: 'Jul', y: 65, y1: 33 }, { x: 'Aug', y: 55, y1: 32 }, { x:
'Sep', y: 50, y1: 34 },
    { x: 'Oct', y: 30, y1: 32 }, { x: 'Nov', y: 35, y1: 32 }, { x:
'Dec', y: 35, y1: 31 }
  ];
  const yAxis = { minimum: 0, maximum: 90, interval: 20,
    title: 'Temperature (Fahrenheit)',
    labelFormat: '{value}°F' };

  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
    primaryYAxis={yAxis}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DColumnsDirective>
      <Chart3DColumnDirective
width="50%"></Chart3DColumnDirective>
      <Chart3DColumnDirective
width="50%"></Chart3DColumnDirective>
    </Chart3DColumnsDirective>
    <Chart3DAxesDirective>
      <Chart3DAxisDirective columnIndex={1} name='xAxis'
valueType="Category"></Chart3DAxisDirective>
    </Chart3DAxesDirective>
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='Column' xAxisName="xAxis">
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

For spanning the vertical axis along multiple column, you can use [span](#) property of an axis.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Chart3DColumnDirective,

```

```

Chart3DAxisDirective, Chart3DAxesDirective, Chart3DColumnsDirective, Inject,
Legend3D, DataLabel3D, Tooltip3D, ColumnSeries3D, Highlight3D } from
'@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: 'Jan', y: 15, y1: 33 }, { x: 'Feb', y: 20, y1: 31 }, { x:
'Mar', y: 35, y1: 30 },
    { x: 'Apr', y: 40, y1: 28 }, { x: 'May', y: 80, y1: 29 }, { x:
'Jun', y: 70, y1: 30 },
    { x: 'Jul', y: 65, y1: 33 }, { x: 'Aug', y: 55, y1: 32 }, { x:
'Sep', y: 50, y1: 34 },
    { x: 'Oct', y: 30, y1: 32 }, { x: 'Nov', y: 35, y1: 32 }, { x:
'Dec', y: 35, y1: 31 }
  ];
  const yAxis = { minimum: 0, maximum: 90, interval: 20,
    title: 'Temperature (Fahrenheit)',
    labelFormat: '{value}°F' };

  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
    span: 2
  }}
    primaryYAxis={yAxis}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DColumnsDirective>
      <Chart3DColumnDirective
width="50%"></Chart3DColumnDirective>
      <Chart3DColumnDirective
width="50%"></Chart3DColumnDirective>
    </Chart3DColumnsDirective>
    <Chart3DAxesDirective>
      <Chart3DAxisDirective columnIndex={1} name='xAxis'
valueType="Category"></Chart3DAxisDirective>
    </Chart3DAxesDirective>
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='Column' xAxisName="xAxis">
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Chart3DColumnDirective,
Chart3DAxisDirective, Chart3DAxesDirective, Chart3DColumnsDirective, Inject,
Legend3D, DataLabel3D, Tooltip3D, ColumnSeries3D, Highlight3D } from
'@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 'Jan', y: 15, y1: 33 }, { x: 'Feb', y: 20, y1: 31 }, { x:
'Mar', y: 35, y1: 30 },
        { x: 'Apr', y: 40, y1: 28 }, { x: 'May', y: 80, y1: 29 }, { x:
'Jun', y: 70, y1: 30 },
        { x: 'Jul', y: 65, y1: 33 }, { x: 'Aug', y: 55, y1: 32 }, { x:
'Sep', y: 50, y1: 34 },
        { x: 'Oct', y: 30, y1: 32 }, { x: 'Nov', y: 35, y1: 32 }, { x:
'Dec', y: 35, y1: 31 }
    ];
    const yAxis = { minimum: 0, maximum: 90, interval: 20,
        title: 'Temperature (Fahrenheit)',
        labelFormat: '{value}°F' };

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
        span: 2
    }}
        primaryYAxis={yAxis}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DColumnsDirective>
            <Chart3DColumnDirective
width="50%"></Chart3DColumnDirective>
            <Chart3DColumnDirective
width="50%"></Chart3DColumnDirective>
        </Chart3DColumnsDirective>
        <Chart3DAxesDirective>
            <Chart3DAxisDirective columnIndex={1} name='xAxis'
valueType="Category"></Chart3DAxisDirective>
        </Chart3DAxesDirective>
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='Column'>
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='Column' xAxisName="xAxis">
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```


Chart Types

Column Chart in React 3D Chart control

Column chart

To render a column series, use series [type](#) as `Column` and inject `ColumnSeries3D` module into the services.

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
        Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
            yName='gold' type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
```

```

    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Column space and width

The [columnSpacing](#) and [columnWidth](#) properties are used to customize the space between columns.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />

```

```

        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' columnSpacing={0.5} columnWidth={0.75} type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' columnSpacing={0.5} columnWidth={0.75} type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Grouped column

The data points can be grouped in the column type charts by using the [groupName](#) property. Data points with same group name are grouped together.

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: '2012', y: 104, y1: 46, y2: 65, y3: 29 },
    { x: '2016', y: 121, y1: 46, y2: 67, y3: 27 },
    { x: '2020', y: 113, y1: 39, y2: 65, y3: 22 }
  ];

  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='Column' columnSpacing={0.1} groupName="USA">
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='Column' columnSpacing={0.1} groupName="USA">
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' type='Column' columnSpacing={0.1} groupName="UK">
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y3' type='Column' columnSpacing={0.1} groupName="UK">
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
```

```

        { x: '2012', y: 104, y1: 46, y2: 65, y3: 29 },
        { x: '2016', y: 121, y1: 46, y2: 67, y3: 27 },
        { x: '2020', y: 113, y1: 39, y2: 65, y3: 22 }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
        Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x'
            yName='y' type='Column' columnSpacing={0.1} groupName="USA">
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
            yName='y1' type='Column' columnSpacing={0.1} groupName="USA">
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
            yName='y2' type='Column' columnSpacing={0.1} groupName="UK">
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
            yName='y3' type='Column' columnSpacing={0.1} groupName="UK">
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Cylindrical column chart

To render a cylindrical column chart, set the [columnFacet](#) property to **Cylinder** in the chart series.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 'Czechia', y: 1.11 },
        { x: 'Spain', y: 1.66 },
        { x: 'USA', y: 1.56 },
        { x: 'Germany', y: 3.1 },
        { x: 'Russia', y: 1.35 },
        { x: 'Slovakia', y: 1 },
        { x: 'South Korea', y: 3.16 },
        { x: 'France', y: 0.92 }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}

```

```

    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='Column' columnFacet="Cylinder">
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 'Czechia', y: 1.11 },
        { x: 'Spain', y: 1.66 },
        { x: 'USA', y: 1.56 },
        { x: 'Germany', y: 3.1 },
        { x: 'Russia', y: 1.35 },
        { x: 'Slovakia', y: 1 },
        { x: 'South Korea', y: 3.16 },
        { x: 'France', y: 0.92 }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='Column' columnFacet="Cylinder">
                </Chart3DSeriesDirective>
            </Chart3DSeriesCollectionDirective>
        </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Series customization

The following properties can be used to customize the **column** series.

- [fill](#) – Specifies the color of the series.
- [opacity](#) – Specifies the opacity of the [fill](#) color.

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column' fill="red">
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
```

```

    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
      yName='gold' type='Column' fill="red">
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Stacked column chart in React 3D Chart control

Stacked column chart

To render a stacked column series, use series [type](#) as `StackingColumn` and inject `StackingColumnSeries3D` module into the services.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, StackingColumnSeries3D, Highlight3D } from '@syncfusion/ej2-
react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: 2000, y: 0.61, y1: 0.03, y2: 0.48 }, { x: 2001, y: 0.81, y1:
    0.05, y2: 0.53 },
    { x: 2002, y: 0.91, y1: 0.06, y2: 0.57 }, { x: 2003, y: 1, y1: 0.09,
    y2: 0.61 }, { x: 2004, y: 1.19, y1: 0.14, y2: 0.63 },
    { x: 2005, y: 1.47, y1: 0.20, y2: 0.64 }, { x: 2006, y: 1.74, y1:
    0.29, y2: 0.66 }, { x: 2007, y: 1.98, y1: 0.46, y2: 0.76 },
    { x: 2008, y: 1.99, y1: 0.64, y2: 0.77 }, { x: 2009, y: 1.70, y1:
    0.75, y2: 0.55 }
  ];

  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[StackingColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >

```



```

        <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='StackingColumn'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='StackingColumn'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' type='StackingColumn'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, StackingColumnSeries3D, Highlight3D } from '@syncfusion/ej2-
react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 2000, y: 0.61, y1: 0.03, y2: 0.48 }, { x: 2001, y: 0.81, y1:
0.05, y2: 0.53 },
        { x: 2002, y: 0.91, y1: 0.06, y2: 0.57 }, { x: 2003, y: 1, y1: 0.09,
y2: 0.61 }, { x: 2004, y: 1.19, y1: 0.14, y2: 0.63 },
        { x: 2005, y: 1.47, y1: 0.20, y2: 0.64 }, { x: 2006, y: 1.74, y1:
0.29, y2: 0.66 }, { x: 2007, y: 1.98, y1: 0.46, y2: 0.76 },
        { x: 2008, y: 1.99, y1: 0.64, y2: 0.77 }, { x: 2009, y: 1.70, y1:
0.75, y2: 0.55 }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[StackingColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='StackingColumn'>
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='StackingColumn'>
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' type='StackingColumn'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};

```

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

Stacking group

To group the stacked column, the [stackingGroup](#) property can be used. The columns with same group name are stacked on top of each other.

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, StackingColumnSeries3D, Highlight3D } from '@syncfusion/ej2-
react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 2000, y: 0.61, y1: 0.03, y2: 0.48 }, { x: 2001, y: 0.81, y1:
0.05, y2: 0.53 },
        { x: 2002, y: 0.91, y1: 0.06, y2: 0.57 }, { x: 2003, y: 1, y1: 0.09,
y2: 0.61 }, { x: 2004, y: 1.19, y1: 0.14, y2: 0.63 },
        { x: 2005, y: 1.47, y1: 0.20, y2: 0.64 }, { x: 2006, y: 1.74, y1:
0.29, y2: 0.66 }, { x: 2007, y: 1.98, y1: 0.46, y2: 0.76 },
        { x: 2008, y: 1.99, y1: 0.64, y2: 0.77 }, { x: 2009, y: 1.70, y1:
0.75, y2: 0.55 }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[StackingColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='StackingColumn' stackingGroup='UKAndGermany'>
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='StackingColumn' stackingGroup='UKAndGermany'>
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' type='StackingColumn' stackingGroup='FranceAndItaly'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
```

```

Tooltip3D, StackingColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 2000, y: 0.61, y1: 0.03, y2: 0.48 }, { x: 2001, y: 0.81, y1: 0.05, y2: 0.53 },
        { x: 2002, y: 0.91, y1: 0.06, y2: 0.57 }, { x: 2003, y: 1, y1: 0.09, y2: 0.61 },
        { x: 2004, y: 1.19, y1: 0.14, y2: 0.63 }, { x: 2005, y: 1.47, y1: 0.20, y2: 0.64 },
        { x: 2006, y: 1.74, y1: 0.29, y2: 0.66 }, { x: 2007, y: 1.98, y1: 0.46, y2: 0.76 },
        { x: 2008, y: 1.99, y1: 0.64, y2: 0.77 }, { x: 2009, y: 1.70, y1: 0.75, y2: 0.55 }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[StackingColumnSeries3D, Category3D, Legend3D, Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x' yName='y' type='StackingColumn' stackingGroup='UKAndGermany'>
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x' yName='y1' type='StackingColumn' stackingGroup='UKAndGermany'>
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x' yName='y2' type='StackingColumn' stackingGroup='FranceAndItaly'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Cylindrical stacked column chart

To render a cylindrical stacked column chart, set the [columnFacet](#) property to `Cylinder` in the chart series.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, StackingColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 2000, y: 0.61, y1: 0.03, y2: 0.48 }, { x: 2001, y: 0.81, y1: 0.05, y2: 0.53 },

```

```

    { x: 2002, y: 0.91, y1: 0.06, y2: 0.57 }, { x: 2003, y: 1, y1: 0.09,
y2: 0.61 }, { x: 2004, y: 1.19, y1: 0.14, y2: 0.63 },
    { x: 2005, y: 1.47, y1: 0.20, y2: 0.64 }, { x: 2006, y: 1.74, y1:
0.29, y2: 0.66 }, { x: 2007, y: 1.98, y1: 0.46, y2: 0.76 },
    { x: 2008, y: 1.99, y1: 0.64, y2: 0.77 }, { x: 2009, y: 1.70, y1:
0.75, y2: 0.55 }
  ];

  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[StackingColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='StackingColumn' columnFacet="Cylinder">
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='StackingColumn' columnFacet="Cylinder">
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' type='StackingColumn' columnFacet="Cylinder">
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, StackingColumnSeries3D, Highlight3D } from '@syncfusion/ej2-
react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: 2000, y: 0.61, y1: 0.03, y2: 0.48 }, { x: 2001, y: 0.81, y1:
0.05, y2: 0.53 },
    { x: 2002, y: 0.91, y1: 0.06, y2: 0.57 }, { x: 2003, y: 1, y1: 0.09,
y2: 0.61 }, { x: 2004, y: 1.19, y1: 0.14, y2: 0.63 },
    { x: 2005, y: 1.47, y1: 0.20, y2: 0.64 }, { x: 2006, y: 1.74, y1:
0.29, y2: 0.66 }, { x: 2007, y: 1.98, y1: 0.46, y2: 0.76 },
    { x: 2008, y: 1.99, y1: 0.64, y2: 0.77 }, { x: 2009, y: 1.70, y1:
0.75, y2: 0.55 }
  ];

  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
    wallColor='transparent'

```

```

        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[StackingColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='StackingColumn' columnFacet="Cylinder">
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='StackingColumn' columnFacet="Cylinder">
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' type='StackingColumn' columnFacet="Cylinder">
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Series customization

The following properties can be used to customize the **stacked column** series.

- [fill](#) – Specifies the color of the series.
- [opacity](#) – Specifies the opacity of the [fill](#) color.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, StackingColumnSeries3D, Highlight3D } from '@syncfusion/ej2-
react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 2000, y: 0.61, y1: 0.03, y2: 0.48 }, { x: 2001, y: 0.81, y1:
0.05, y2: 0.53 },
        { x: 2002, y: 0.91, y1: 0.06, y2: 0.57 }, { x: 2003, y: 1, y1: 0.09,
y2: 0.61 }, { x: 2004, y: 1.19, y1: 0.14, y2: 0.63 },
        { x: 2005, y: 1.47, y1: 0.20, y2: 0.64 }, { x: 2006, y: 1.74, y1:
0.29, y2: 0.66 }, { x: 2007, y: 1.98, y1: 0.46, y2: 0.76 },
        { x: 2008, y: 1.99, y1: 0.64, y2: 0.77 }, { x: 2009, y: 1.70, y1:
0.75, y2: 0.55 }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[StackingColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='StackingColumn' fill="green">

```

```

        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='StackingColumn' fill="red">
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' type='StackingColumn' fill="yellow">
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, StackingColumnSeries3D, Highlight3D } from '@syncfusion/ej2-
react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 2000, y: 0.61, y1: 0.03, y2: 0.48 }, { x: 2001, y: 0.81, y1:
0.05, y2: 0.53 },
        { x: 2002, y: 0.91, y1: 0.06, y2: 0.57 }, { x: 2003, y: 1, y1: 0.09,
y2: 0.61 }, { x: 2004, y: 1.19, y1: 0.14, y2: 0.63 },
        { x: 2005, y: 1.47, y1: 0.20, y2: 0.64 }, { x: 2006, y: 1.74, y1:
0.29, y2: 0.66 }, { x: 2007, y: 1.98, y1: 0.46, y2: 0.76 },
        { x: 2008, y: 1.99, y1: 0.64, y2: 0.77 }, { x: 2009, y: 1.70, y1:
0.75, y2: 0.55 }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[StackingColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='StackingColumn' fill="green">
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='StackingColumn' fill="red">
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' type='StackingColumn' fill="yellow">
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

100% Stacked column chart in React 3D Chart control

100% Stacked column chart

To render a 100% stacked column series, use series [type](#) as `StackingColumn100` and inject `StackingColumnSeries3D` module into the services.

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, StackingColumnSeries3D, Highlight3D } from '@syncfusion/ej2-
react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 2000, y: 0.61, y1: 0.03, y2: 0.48 }, { x: 2001, y: 0.81, y1:
0.05, y2: 0.53 },
        { x: 2002, y: 0.91, y1: 0.06, y2: 0.57 }, { x: 2003, y: 1, y1: 0.09,
y2: 0.61 }, { x: 2004, y: 1.19, y1: 0.14, y2: 0.63 },
        { x: 2005, y: 1.47, y1: 0.20, y2: 0.64 }, { x: 2006, y: 1.74, y1:
0.29, y2: 0.66 }, { x: 2007, y: 1.98, y1: 0.46, y2: 0.76 },
        { x: 2008, y: 1.99, y1: 0.64, y2: 0.77 }, { x: 2009, y: 1.70, y1:
0.75, y2: 0.55 }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[StackingColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='StackingColumn100'>
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='StackingColumn100'>
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' type='StackingColumn100'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
```

```

Tooltip3D, StackingColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 2000, y: 0.61, y1: 0.03, y2: 0.48 }, { x: 2001, y: 0.81, y1: 0.05, y2: 0.53 },
        { x: 2002, y: 0.91, y1: 0.06, y2: 0.57 }, { x: 2003, y: 1, y1: 0.09, y2: 0.61 },
        { x: 2004, y: 1.19, y1: 0.14, y2: 0.63 },
        { x: 2005, y: 1.47, y1: 0.20, y2: 0.64 }, { x: 2006, y: 1.74, y1: 0.29, y2: 0.66 },
        { x: 2007, y: 1.98, y1: 0.46, y2: 0.76 },
        { x: 2008, y: 1.99, y1: 0.64, y2: 0.77 }, { x: 2009, y: 1.70, y1: 0.75, y2: 0.55 }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[StackingColumnSeries3D, Category3D, Legend3D, Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x' yName='y' type='StackingColumn100'>
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x' yName='y1' type='StackingColumn100'>
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x' yName='y2' type='StackingColumn100'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

100% Cylindrical stacked column chart

To render a 100% cylindrical stacked column chart, set the [columnFacet](#) property to **Cylinder** in the chart series.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, StackingColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: '2013', y: 9628912, y1: 4298390, y2: 2842133, y3: 2006366 },
        { x: '2014', y: 9609326, y1: 4513769, y2: 3016710, y3: 2165566 },
        { x: '2015', y: 7485587, y1: 4543838, y2: 3034081, y3: 2279503 },
    ];

```



```

        { x: '2016', y: 7793066, y1: 4999266, y2: 2945295, y3: 2359756 },
        { x: '2017', y: 6856880, y1: 5235842, y2: 3302336, y3: 2505741 }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
    primaryYAxis={{
        interval: 25
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[StackingColumnSeries3D, Category3D, Legend3D,
        Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='StackingColumn100' columnFacet="Cylinder">
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='StackingColumn100' columnFacet="Cylinder">
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' type='StackingColumn100' columnFacet="Cylinder">
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, StackingColumnSeries3D, Highlight3D } from '@syncfusion/ej2-
react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: '2013', y: 9628912, y1: 4298390, y2: 2842133, y3: 2006366 },
        { x: '2014', y: 9609326, y1: 4513769, y2: 3016710, y3: 2165566 },
        { x: '2015', y: 7485587, y1: 4543838, y2: 3034081, y3: 2279503 },
        { x: '2016', y: 7793066, y1: 4999266, y2: 2945295, y3: 2359756 },
        { x: '2017', y: 6856880, y1: 5235842, y2: 3302336, y3: 2505741 }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
    primaryYAxis={{
        interval: 25
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>

```

```

    <Inject services={[StackingColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='StackingColumn100' columnFacet="Cylinder">
    </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='StackingColumn100' columnFacet="Cylinder">
    </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' type='StackingColumn100' columnFacet="Cylinder">
    </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Series customization

The following properties can be used to customize the **100% stacked column** series.

- [fill](#) – Specifies the color of the series.
- [opacity](#) – Specifies the opacity of the [fill](#) color.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, StackingColumnSeries3D, Highlight3D } from '@syncfusion/ej2-
react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: '2013', y: 9628912, y1: 4298390, y2: 2842133, y3: 2006366 },
    { x: '2014', y: 9609326, y1: 4513769, y2: 3016710, y3: 2165566 },
    { x: '2015', y: 7485587, y1: 4543838, y2: 3034081, y3: 2279503 },
    { x: '2016', y: 7793066, y1: 4999266, y2: 2945295, y3: 2359756 },
    { x: '2017', y: 6856880, y1: 5235842, y2: 3302336, y3: 2505741 }
  ];

  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
    primaryYAxis={{
      interval: 25
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[StackingColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='StackingColumn100' fill="red">
    </Chart3DSeriesDirective>

```

```

        <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='StackingColumn100' fill="green">
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' type='StackingColumn100' fill="yellow">
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, StackingColumnSeries3D, Highlight3D } from '@syncfusion/ej2-
react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: '2013', y: 9628912, y1: 4298390, y2: 2842133, y3: 2006366 },
        { x: '2014', y: 9609326, y1: 4513769, y2: 3016710, y3: 2165566 },
        { x: '2015', y: 7485587, y1: 4543838, y2: 3034081, y3: 2279503 },
        { x: '2016', y: 7793066, y1: 4999266, y2: 2945295, y3: 2359756 },
        { x: '2017', y: 6856880, y1: 5235842, y2: 3302336, y3: 2505741 }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
    primaryYAxis={{
        interval: 25
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[StackingColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='StackingColumn100' fill="red">
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='StackingColumn100' fill="green">
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' type='StackingColumn100' fill="yellow">
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Bar Chart in React 3D Chart component

Bar chart

To render a bar series, use series [type](#) as `Bar` and inject `BarSeries3D` module into the services.

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, BarSeries3D, Highlight3D } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[BarSeries3D, Category3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Bar'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, BarSeries3D, Highlight3D } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];
}
```

```

    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[BarSeries3D, Category3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Bar'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Bar space and width

The [columnSpacing](#) and [columnWidth](#) properties are used to customize the space between bars.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, BarSeries3D, Highlight3D } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[BarSeries3D, Category3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Bar'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' columnSpacing={0.5} columnWidth={0.75} type='Bar'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
};

```

```

        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, BarSeries3D, Highlight3D } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[BarSeries3D, Category3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Bar'>
        </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' columnSpacing={0.5} columnWidth={0.75} type='Bar'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Grouped bar

The data points can be grouped in the bar type charts by using the [groupName](#) property. Data points with same group name are grouped together.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, BarSeries3D, Highlight3D } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: '2012', y: 104, y1: 46, y2: 65, y3: 29 },
        { x: '2016', y: 121, y1: 46, y2: 67, y3: 27 },
        { x: '2020', y: 113, y1: 39, y2: 65, y3: 22 }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[BarSeries3D, Category3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x' yName='y'
type='Bar' columnSpacing={0.1} groupName="USA">
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x' yName='y1'
type='Bar' columnSpacing={0.1} groupName="USA">
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x' yName='y2'
type='Bar' columnSpacing={0.1} groupName="UK">
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x' yName='y3'
type='Bar' columnSpacing={0.1} groupName="UK">
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, BarSeries3D, Highlight3D } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: '2012', y: 104, y1: 46, y2: 65, y3: 29 },
        { x: '2016', y: 121, y1: 46, y2: 67, y3: 27 },
        { x: '2020', y: 113, y1: 39, y2: 65, y3: 22 }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}

```

```

        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[BarSeries3D, Category3D, Legend3D, Tooltip3D,
        DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x'
            yName='y' type='Bar' columnSpacing={0.1} groupName="USA">
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
            yName='y1' type='Bar' columnSpacing={0.1} groupName="USA">
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
            yName='y2' type='Bar' columnSpacing={0.1} groupName="UK">
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
            yName='y3' type='Bar' columnSpacing={0.1} groupName="UK">
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Cylindrical bar chart

To render a cylindrical bar chart, set the [columnFacet](#) property to **Cylinder** in the chart series.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, BarSeries3D, Highlight3D } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[BarSeries3D, Category3D, Legend3D, Tooltip3D,
        DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
            yName='gold' type='Bar' columnFacet="Cylinder">
            </Chart3DSeriesDirective>

```



```

        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, BarSeries3D, Highlight3D } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[BarSeries3D, Category3D, Legend3D, Tooltip3D,
DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Bar' columnFacet="Cylinder">
                </Chart3DSeriesDirective>
            </Chart3DSeriesCollectionDirective>
        </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Series customization

The following properties can be used to customize the **bar** series.

- [fill](#) – Specifies the color of the series.
- [opacity](#) – Specifies the opacity of the [fill](#) color.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, BarSeries3D, Highlight3D } from '@syncfusion/ej2-react-charts';

```

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[BarSeries3D, Category3D, Legend3D, Tooltip3D,
    DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
      yName='gold' type='Bar' fill="red">
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, BarSeries3D, Highlight3D } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>

```

```

    <Inject services={[BarSeries3D, Category3D, Legend3D, Tooltip3D,
    DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
      yName='gold' type='Bar' fill="red">
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Stacked bar chart in React Chart component

Stacked bar chart

To render a stacked bar series, use series [type](#) as `StackingBar` and inject `StackingBar3D` module into the services.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, StackingBarSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: 'Sochi', y: 9, y1: 10, y2: 4 },
    { x: 'Rio', y: 46, y1: 4, y2: 10 },
    { x: 'Pyeongchang', y: 9, y1: 11, y2: 5 },
    { x: 'Tokyo', y: 39, y1: 7, y2: 10 },
    { x: 'Beijing', y: 8, y1: 4, y2: 5 }
  ];

  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
  primaryYAxis={{
    interval: 25
  }}
  wallColor='transparent'
  enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[StackingBarSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x'
      yName='y' type='StackingBar'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
      yName='y1' type='StackingBar'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
      yName='y2' type='StackingBar'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>

```

```

        </Chart3DComponent>;
    }
    ;
    export default App;
    ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, StackingBarSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 'Sochi', y: 9, y1: 10, y2: 4 },
        { x: 'Rio', y: 46, y1: 4, y2: 10 },
        { x: 'Pyeongchang', y: 9, y1: 11, y2: 5 },
        { x: 'Tokyo', y: 39, y1: 7, y2: 10 },
        { x: 'Beijing', y: 8, y1: 4, y2: 5 }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
    primaryYAxis={{
        interval: 25
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[StackingBarSeries3D, Category3D, Legend3D,
        Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='StackingBar'>
                </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='StackingBar'>
                </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' type='StackingBar'>
                </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Stacking group

To group the stacked bar, the [stackingGroup](#) property can be used. The columns with same group name are stacked on top of each other.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, StackingBarSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 'Sochi', y: 9, y1: 10, y2: 4 },
        { x: 'Rio', y: 46, y1: 4, y2: 10 },
        { x: 'Pyeongchang', y: 9, y1: 11, y2: 5 },
        { x: 'Tokyo', y: 39, y1: 7, y2: 10 },
        { x: 'Beijing', y: 8, y1: 4, y2: 5 }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
    primaryYAxis={{
        interval: 25
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[StackingBarSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='StackingBar' stackingGroup='JohnAndAndrew'>
    </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='StackingBar' stackingGroup='JohnAndAndrew'>
    </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' type='StackingBar' stackingGroup='ThomasAndMichael'>
    </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, StackingBarSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 'Sochi', y: 9, y1: 10, y2: 4 },
        { x: 'Rio', y: 46, y1: 4, y2: 10 },
        { x: 'Pyeongchang', y: 9, y1: 11, y2: 5 },
        { x: 'Tokyo', y: 39, y1: 7, y2: 10 },

```

```

    { x: 'Beijing', y: 8, y1: 4, y2: 5 }
  ];

  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
  primaryYAxis={{
    interval: 25
  }}
  wallColor='transparent'
  enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[StackingBarSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='StackingBar' stackingGroup='JohnAndAndrew'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='StackingBar' stackingGroup='JohnAndAndrew'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' type='StackingBar' stackingGroup='ThomasAndMichael'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Cylindrical stacked bar chart

To render a cylindrical stacked bar chart, set the [columnFacet](#) property to `Cylinder` in the chart series.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, StackingBarSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: 'Sochi', y: 9, y1: 10, y2: 4 },
    { x: 'Rio', y: 46, y1: 4, y2: 10 },
    { x: 'Pyeongchang', y: 9, y1: 11, y2: 5 },
    { x: 'Tokyo', y: 39, y1: 7, y2: 10 },
    { x: 'Beijing', y: 8, y1: 4, y2: 5 }
  ];

  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
  primaryYAxis={{
    interval: 25
  }}
  wallColor='transparent'

```

```

        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[StackingBarSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='StackingBar' columnFacet="Cylinder">
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='StackingBar' columnFacet="Cylinder">
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' type='StackingBar' columnFacet="Cylinder">
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, StackingBarSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 'Sochi', y: 9, y1: 10, y2: 4 },
        { x: 'Rio', y: 46, y1: 4, y2: 10 },
        { x: 'Pyeongchang', y: 9, y1: 11, y2: 5 },
        { x: 'Tokyo', y: 39, y1: 7, y2: 10 },
        { x: 'Beijing', y: 8, y1: 4, y2: 5 }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
    primaryYAxis={{
        interval: 25
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[StackingBarSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='StackingBar' columnFacet="Cylinder">
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='StackingBar' columnFacet="Cylinder">
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' type='StackingBar' columnFacet="Cylinder">

```

```

        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
  };
  export default App;
  ReactDOM.render(<App />, document.getElementById("charts"));

```

Series customization

The following properties can be used to customize the **stacked bar** series.

- [fill](#) – Specifies the color of the series.
- [opacity](#) – Specifies the opacity of the [fill](#) color.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, StackingBarSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: 'Sochi', y: 9, y1: 10, y2: 4 },
    { x: 'Rio', y: 46, y1: 4, y2: 10 },
    { x: 'Pyeongchang', y: 9, y1: 11, y2: 5 },
    { x: 'Tokyo', y: 39, y1: 7, y2: 10 },
    { x: 'Beijing', y: 8, y1: 4, y2: 5 }
  ];

  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
  primaryYAxis={{
    interval: 25
  }}
  wallColor='transparent'
  enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[StackingBarSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='StackingBar' fill="red">
        </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='StackingBar' fill="green">
        </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' type='StackingBar' fill="yellow">
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
  }
  ;
  export default App;

```



```
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, StackingBarSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: 'Sochi', y: 9, y1: 10, y2: 4 },
    { x: 'Rio', y: 46, y1: 4, y2: 10 },
    { x: 'Pyeongchang', y: 9, y1: 11, y2: 5 },
    { x: 'Tokyo', y: 39, y1: 7, y2: 10 },
    { x: 'Beijing', y: 8, y1: 4, y2: 5 }
  ];

  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
    primaryYAxis={{
      interval: 25
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[StackingBarSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='StackingBar' fill="red">
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='StackingBar' fill="green">
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' type='StackingBar' fill="yellow">
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

100% Stacked bar chart in React 3D Chart control

100% Stacked bar chart

To render a 100% stacked bar series, use series [type](#) as `StackingBar100` and inject `StackingBarSeries3D` module into the services.

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
```

```

Tooltip3D, StackingBarSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: '2013', y: 9628912, y1: 4298390, y2: 2842133, y3: 2006366 },
    { x: '2014', y: 9609326, y1: 4513769, y2: 3016710, y3: 2165566 },
    { x: '2015', y: 7485587, y1: 4543838, y2: 3034081, y3: 2279503 },
    { x: '2016', y: 7793066, y1: 4999266, y2: 2945295, y3: 2359756 },
    { x: '2017', y: 6856880, y1: 5235842, y2: 3302336, y3: 2505741 }
  ];

  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
  primaryYAxis={{
    interval: 25
  }}
  wallColor='transparent'
  enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[StackingBarSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='StackingBar100'>
        </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='StackingBar100'>
        </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' type='StackingBar100'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, StackingBarSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: '2013', y: 9628912, y1: 4298390, y2: 2842133, y3: 2006366 },
    { x: '2014', y: 9609326, y1: 4513769, y2: 3016710, y3: 2165566 },
    { x: '2015', y: 7485587, y1: 4543838, y2: 3034081, y3: 2279503 },
    { x: '2016', y: 7793066, y1: 4999266, y2: 2945295, y3: 2359756 },
    { x: '2017', y: 6856880, y1: 5235842, y2: 3302336, y3: 2505741 }
  ];

```

```

    return <Chart3DComponent id='charts' primaryXAxis={{
      valueType: "Category",
    }}
    primaryYAxis={{
      interval: 25
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[StackingBarSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='StackingBar100'>
        </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='StackingBar100'>
        </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' type='StackingBar100'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

100% Cylindrical stacked bar chart

To render a cylindrical 100% stacked bar chart, set the [columnFacet](#) property to **Cylinder** in the chart series.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, StackingBarSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: '2013', y: 9628912, y1: 4298390, y2: 2842133, y3: 2006366 },
    { x: '2014', y: 9609326, y1: 4513769, y2: 3016710, y3: 2165566 },
    { x: '2015', y: 7485587, y1: 4543838, y2: 3034081, y3: 2279503 },
    { x: '2016', y: 7793066, y1: 4999266, y2: 2945295, y3: 2359756 },
    { x: '2017', y: 6856880, y1: 5235842, y2: 3302336, y3: 2505741 }
  ];

  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
  primaryYAxis={{
    interval: 25
  }}
  wallColor='transparent'
  enableRotation={true} rotation={7} tilt={10} depth={100}>

```

```

    <Inject services={[StackingBarSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='StackingBar100' columnFacet="Cylinder">
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='StackingBar100' columnFacet="Cylinder">
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' type='StackingBar100' columnFacet="Cylinder">
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, StackingBarSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: '2013', y: 9628912, y1: 4298390, y2: 2842133, y3: 2006366 },
    { x: '2014', y: 9609326, y1: 4513769, y2: 3016710, y3: 2165566 },
    { x: '2015', y: 7485587, y1: 4543838, y2: 3034081, y3: 2279503 },
    { x: '2016', y: 7793066, y1: 4999266, y2: 2945295, y3: 2359756 },
    { x: '2017', y: 6856880, y1: 5235842, y2: 3302336, y3: 2505741 }
  ];

  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
  primaryYAxis={{
    interval: 25
  }}
  wallColor='transparent'
  enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[StackingBarSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='StackingBar100' columnFacet="Cylinder">
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='StackingBar100' columnFacet="Cylinder">
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' type='StackingBar100' columnFacet="Cylinder">
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>

```

```

        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Series customization

The following properties can be used to customize the **100% stacked bar** series.

- [fill](#) – Specifies the color of the series.
- [opacity](#) – Specifies the opacity of the [fill](#) color.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, StackingBarSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: '2013', y: 9628912, y1: 4298390, y2: 2842133, y3: 2006366 },
        { x: '2014', y: 9609326, y1: 4513769, y2: 3016710, y3: 2165566 },
        { x: '2015', y: 7485587, y1: 4543838, y2: 3034081, y3: 2279503 },
        { x: '2016', y: 7793066, y1: 4999266, y2: 2945295, y3: 2359756 },
        { x: '2017', y: 6856880, y1: 5235842, y2: 3302336, y3: 2505741 }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
    primaryYAxis={{
        interval: 25
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[StackingBarSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='StackingBar100' fill="red">
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='StackingBar100' fill="green">
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' type='StackingBar100' fill="yellow">
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y3' type='StackingBar100' fill="blue">
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}

```

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

INDEX.TSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, StackingBarSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: '2013', y: 9628912, y1: 4298390, y2: 2842133, y3: 2006366 },
        { x: '2014', y: 9609326, y1: 4513769, y2: 3016710, y3: 2165566 },
        { x: '2015', y: 7485587, y1: 4543838, y2: 3034081, y3: 2279503 },
        { x: '2016', y: 7793066, y1: 4999266, y2: 2945295, y3: 2359756 },
        { x: '2017', y: 6856880, y1: 5235842, y2: 3302336, y3: 2505741 }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
        primaryYAxis={{
            interval: 25
        }}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[StackingBarSeries3D, Category3D, Legend3D,
        Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='StackingBar100' fill="red">
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' type='StackingBar100' fill="green">
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' type='StackingBar100' fill="yellow">
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y3' type='StackingBar100' fill="blue">
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Data labels in React 3D Chart control

Data labels are fields that includes information about the sample point connected to an output. It can be added to a chart series by enabling the [visible](#) option in the dataLabel. By default, the labels will arrange smartly without overlapping.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts'
    primaryXAxis={{
        valueType: "Category",
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column' dataLabel={{ visible: true }}>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    ]

```

```

    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
      yName='gold' type='Column' dataLabel={{ visible: true }}>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Note: To use data label feature, we need to inject **DataLabel3D** module into the services.

Position

The **position** property is used to place the label either on **Top**, **Middle**, or **Bottom**.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >

```



```

        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'
        dataLabel={{
            visible: true,
            position: 'Top'
        }}>
    </Chart3DSeriesDirective>
</Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts'
primaryXAxis={{
    valueType: "Category",
}}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'
            dataLabel={{
                visible: true,
                position: 'Top'
            }}>
        </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Template

Label content can be formatted by using the template option. Inside the template, the placeholder text `${point.x}` and `${point.y}` can be added to display corresponding data points x & y value. Using [template](#) property, the data label template can be set.

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 }
  ];
  const dataLabel = {
    template: '<div style="border: 1px solid black; padding: 3px 3px 3px 3px"><div>${point.x}</div><div>${point.y}</div></div>', visible: true
  }
  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'
        dataLabel={dataLabel}>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
```

```

    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 }
  ];
  const dataLabel = {
    template: '<div style="border: 1px solid black; padding: 3px 3px 3px 3px"><div>${point.x}</div><div>${point.y}</div></div>', visible: true
  }
  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'
      dataLabel={dataLabel}>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Text mapping

Text from the data source can be mapped using the [name](#) property.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: 'Jan', y: 12, text: 'January : 12°C' },
    { x: 'Feb', y: 8, text: 'February : 8°C' },
    { x: 'Mar', y: 11, text: 'March : 11°C' },
    { x: 'Apr', y: 6, text: 'April : 6°C' }
  ];

  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='Column'

```

```

        dataLabel={{
            visible: true,
            name: "text"
        }}
    </Chart3DSeriesDirective>
</Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 'Jan', y: 12, text: 'January : 12°C' },
        { x: 'Feb', y: 8, text: 'February : 8°C' },
        { x: 'Mar', y: 11, text: 'March : 11°C' },
        { x: 'Apr', y: 6, text: 'April : 6°C' }
    ];

    return <Chart3DComponent id='charts'
    primaryXAxis={{
        valueType: "Category",
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
        Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x'
            yName='y' type='Column'
            dataLabel={{
                visible: true,
                name: "text"
            }}>
                </Chart3DSeriesDirective>
            </Chart3DSeriesCollectionDirective>
        </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Format

Data label for the chart can be formatted using the [format](#) property. The global formatting options can be used as 'n', 'p', and 'c'.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 }
    ];

    return <Chart3DComponent id='charts'
    primaryXAxis={{
        valueType: "Category",
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'
        dataLabel={{
            visible: true,
            format: 'n2'
        }}>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 }
    ];

    return <Chart3DComponent id='charts'
    primaryXAxis={{
        valueType: "Category",

```

```

    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'
        dataLabel={{
            visible: true,
            format: 'n2'
        }}>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Value	Format	Resultant Value	Description
1000	n1	1000.0	The number is rounded to 1 decimal place.
1000	n2	1000.00	The number is rounded to 2 decimal places.
1000	n3	1000.000	The number is rounded to 3 decimal place.
0.01	p1	1.0%	The number is converted to percentage with 1 decimal place.
0.01	p2	1.00%	The number is converted to percentage with 2 decimal place.
0.01	p3	1.000%	The number is converted to percentage with 3 decimal place.
1000	c1	\$1000.0	The currency symbol is appended to number and number is rounded to 1 decimal place.
1000	c2	\$1000.00	The currency symbol is appended to number and number is rounded to 2 decimal place.

Margin

The [margin](#) for data label can be applied by using [left](#), [right](#), [bottom](#) and [top](#) properties.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    ]

```

```

    { country: "Australia", gold: 60, silver: 56, bronze: 40 }
  ];
  const dataLabel = {
    visible: true,
    border:{
      width: 1,
      color : 'red'
    },
    margin:{
      left:5,
      right:5,
      top:5,
      bottom:5
    }
  }
  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'
        dataLabel={dataLabel}>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 }
  ];
  const dataLabel = {
    visible: true,
    border:{
      width: 1,
      color : 'red'
    },
  },

```

```

        margin:{
            left:5,
            right:5,
            top:5,
            bottom:5
        }
    };
    return <Chart3DComponent id='charts'
    primaryXAxis={{
        valueType: "Category",
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
    yName='gold' type='Column'
        dataLabel={dataLabel}>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Customization

The **stroke** and **border** of data label can be customized using [fill](#) and [border](#) properties.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 }
    ];
    const dataLabel = {
        visible: true,
        border:{ width: 2, color : 'red'},
    };
    return <Chart3DComponent id='charts'
    primaryXAxis={{
        valueType: "Category",
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />

```



```

        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'
                dataLabel={dataLabel}>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 }
    ];
    const dataLabel = {
        visible: true,
        border:{ width: 2, color : 'red'},
    };
    return <Chart3DComponent id='charts'
primaryXAxis={{
        valueType: "Category",
    }}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'
                dataLabel={dataLabel}>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Customizing specific label

A specific label can be customized by using the [textRender](#) event. The `textRender` event allows you to change the label text for the point.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 }
    ];
    const textRender = function(args) {
        if (args.point.index === 2) {
            args.text = 'Label';
        }
        else {
            args.cancel = true;
        }
    }

    return <Chart3DComponent id='charts' textRender={textRender}
primaryXAxis={{
    valueType: "Category",
}}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'
                dataLabel={{
                    visible: true
                }}>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },

```

```

    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 }
  ];
  const textRender = function(args) {
    if (args.point.index === 2) {
      args.text = 'Label';
    }
    else {
      args.cancel = true;
    }
  }

  return <Chart3DComponent id='charts' textRender={textRender}
  primaryXAxis={{
    valueType: "Category",
  }}
  wallColor='transparent'
  enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'
      dataLabel={{
        visible: true
      }}>
    </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

<!-- markdownlint-disable MD036 -->

Legend in React 3D Chart control

<!-- markdownlint-disable MD036 -->

Legend provides information about the series rendered in the 3D chart.

Position and alignment

By using the [position](#) property, the legend can be positioned at left, right, top or bottom of the 3D chart. The legend is positioned at the bottom of the 3D chart, by default.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
  ];

```

```

    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }}
    legendSettings={{
      visible: true,
      //Legend position as top
      position: 'Top'
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Gold" type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' name="Silver" type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' name="Bronze" type='Column'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ]

```

```

];

return <Chart3DComponent id='charts'
primaryXAxis={{
  valueType: "Category",
}}
legendSettings={{
  visible: true,
  //Legend position as top
  position: 'Top'
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
  <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
  <Chart3DSeriesCollectionDirective >
    <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Gold" type='Column'>
    </Chart3DSeriesDirective>
    <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' name="Silver" type='Column'>
    </Chart3DSeriesDirective>
    <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' name="Bronze" type='Column'>
    </Chart3DSeriesDirective>
  </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

The custom position helps you to position the legend anywhere in the 3D chart using x and y coordinates.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];
  const legend = {
    visible: true,
    location: { x: 200, y: 20 },

```

```

        position: 'Custom'
    };
    return <Chart3DComponent id='charts'
    primaryXAxis={{
        valueType: "Category",
    }}
    legendSettings = {legend} wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Gold" type='Column'>
                </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' name="Silver" type='Column'>
                </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' name="Bronze" type='Column'>
                </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];
    const legend = {
        visible: true,
        location: { x: 200, y: 20 },
        position: 'Custom'
    };
    return <Chart3DComponent id='charts'
    primaryXAxis={{
        valueType: "Category",
    }}
    legendSettings = {legend} wallColor='transparent'

```

```

        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Gold" type='Column'>
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' name="Silver" type='Column'>
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' name="Bronze" type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Legend reverse

The order of the legend items can be reversed by using the [reverse](#) property. By default, legend for the first series in the collection will be placed first.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts'
primaryXAxis={{
    valueType: "Category",
}}
legendSettings={{
    visible: true,
    reverse: true
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >

```

```

        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Gold" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' name="Silver" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' name="Bronze" type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts'
primaryXAxis={{
    valueType: "Category",
}}
legendSettings={{
    visible: true,
    reverse: true
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Gold" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' name="Silver" type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>

```



```

        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' name="Bronze" type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Legend alignment

The legend can be aligned at near, far or center to the 3D chart using the [alignment](#) property.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts'
primaryXAxis={{
    valueType: "Category",
}}
legendSettings={{
    visible: true,
    position: 'Top',
    alignment: 'Near'
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Gold" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' name="Silver" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' name="Bronze" type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>

```

```

        </Chart3DComponent>;
    }
    ;
    export default App;
    ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts'
    primaryXAxis={{
        valueType: "Category",
    }}
    legendSettings={{
        visible: true,
        position: 'Top',
        alignment: 'Near'
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Gold" type='Column'>
            </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' name="Silver" type='Column'>
            </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' name="Bronze" type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Legend customization

To change the legend icon shape, [legendShape](#) property in the [series](#) can be used. By default, the legend icon shape is [seriesType](#).

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts'
    primaryXAxis={{
        valueType: "Category",
    }}
    legendSettings={{
        visible: true
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' legendShape="Circle" name="Gold" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' legendShape="SeriesType" name="Silver" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' legendShape="Rectangle" name="Bronze" type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
```

```

Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }}
    legendSettings={{
      visible: true
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' legendShape="Circle" name="Gold" type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' legendShape="SeriesType" name="Silver" type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' legendShape="Rectangle" name="Bronze" type='Column'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Legend size

By default, legend takes 20% - 25% of the 3D chart's height horizontally, when it is placed on top or bottom position and 20% - 25% of the 3D chart's width vertically, when it is placed on left or right position. You can change this default legend size by using the [height](#) and [width](#) properties of the [legendSettings](#).

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";

```

```

import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];
  const legend = {
    visible: true,
    width: '500', height: '100',
    border: { width: 1, color: 'pink' }
  };
  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }}
    legendSettings = {legend} wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Gold" type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' name="Silver" type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' name="Bronze" type='Column'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },

```

```

    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];
  const legend = {
    visible: true,
    width: '500', height: '100',
    border: { width: 1, color: 'pink' }
  };
  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }}
    legendSettings = {legend} wallColor='transparent'
      enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
      Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
        yName='gold' name="Gold" type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='country'
        yName='silver' name="Silver" type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='country'
        yName='bronze' name="Bronze" type='Column'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Legend item size

The size of the legend items can be customised by using the [shapeHeight](#) and [shapeWidth](#) properties.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
  Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
  Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
  charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];
};

```

```

    return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }}
    legendSettings={{
      visible: true,
      shapeHeight: 10, shapeWidth: 10
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
    yName='gold' legendShape="Circle" name="Gold" type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='country'
    yName='silver' legendShape="Circle" name="Silver" type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='country'
    yName='bronze' legendShape="Circle" name="Bronze" type='Column'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts'
  primaryXAxis={{
    valueType: "Category",
  }}
  legendSettings={{
    visible: true,

```

```

        shapeHeight: 10, shapeWidth: 10
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' legendShape="Circle" name="Gold" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' legendShape="Circle" name="Silver" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' legendShape="Circle" name="Bronze" type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Paging for legend

Paging will be enabled by default, when the legend items exceeds the legend bounds. Each legend items can be viewed by navigating between the pages using navigation buttons.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];
    const legend = {
        padding: 10, shapePadding: 10,
        visible: true, border: {
            width: 2, color: 'grey'
        },
        width: '200'
    };
    return <Chart3DComponent id='charts'
primaryXAxis={{
    valueType: "Category",
}}

```



```

        legendSettings = {legend} wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Gold" type='Column'>
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' name="Silver" type='Column'>
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' name="Bronze" type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];
    const legend = {
        padding: 10, shapePadding: 10,
        visible: true, border: {
            width: 2, color: 'grey'
        },
        width: '200'
    };
    return <Chart3DComponent id='charts'
primaryXAxis={{
    valueType: "Category",
}}
    legendSettings = {legend} wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >

```

```

        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Gold" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' name="Silver" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' name="Bronze" type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Legend text wrap

When the legend text exceeds the container, the text can be wrapped by using the [textWrap](#) property. End user can also wrap the legend text based on the [maximumLabelWidth](#) property.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts'
primaryXAxis={{
    valueType: "Category",
}}
legendSettings={{
    visible: true,
    position: 'Right',
    textWrap: 'Wrap',
    maximumLabelWidth: 50,
    width: '200'
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >

```

```

        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Gold" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' name="Silver" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' name="Bronze" type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts'
primaryXAxis={{
    valueType: "Category",
}}
legendSettings={{
    visible: true,
    position: 'Right',
    textWrap: 'Wrap',
    maximumLabelWidth: 50,
    width: '200'
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Gold" type='Column'>
        </Chart3DSeriesDirective>

```

```

        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' name="Silver" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' name="Bronze" type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Series selection through legend

By default, you can collapse the series visibility by clicking the legend. On the other hand, turn off the [toggleVisibility](#) property if you must use a legend click to choose a series.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts'
primaryXAxis={{
    valueType: "Category",
}}
legendSettings={{
    visible: true,
    toggleVisibility: false,
    enableHighlight: true
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Gold" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' name="Silver" type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>

```

```

        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' name="Bronze" type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts'
primaryXAxis={{
    valueType: "Category",
}}
legendSettings={{
    visible: true,
    toggleVisibility: false,
    enableHighlight: true
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Gold" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' name="Silver" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' name="Bronze" type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
};

```

```
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, you can give empty string to the series name.

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }}
    legendSettings={{
      visible: true,
      toggleVisibility: true
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Gold" type='Column'>
        </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' name="Silver" type='Column'>
        </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' name="Bronze" type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts'
    primaryXAxis={{
        valueType: "Category",
    }}
    legendSettings={{
        visible: true,
        toggleVisibility: true
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Gold" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' name="Silver" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' name="Bronze" type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Legend title

You can set title for legend using [title](#) property in [legendSettings](#). The [size](#), [color](#), [opacity](#), [fontStyle](#), [fontWeight](#), [fontFamily](#), [textAlignment](#), and [textOverflow](#) of legend title can be customized by using the [titleStyle](#) property in [legendSettings](#). The [titlePosition](#) is used to set the legend position in Top, Left and Right position. The [maximumTitleWidth](#) is used to set the width of the legend title. By default, it will be 100px.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts'
    primaryXAxis={{
        valueType: "Category",
    }}
    legendSettings={{
        visible: true,
        title: 'Medals'
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Gold" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' name="Silver" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' name="Bronze" type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-charts';
import * as React from "react";

```



```
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }}
    legendSettings={{
      visible: true,
      title: 'Medals'
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Gold" type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' name="Silver" type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' name="Bronze" type='Column'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Arrow page navigation

The page number will always be visible while using legend paging. It is now possible to disable the page number and enable page navigation with the left and right arrows. The [enablePages](#) property needs to be set to **false** in order to render the arrow page navigation.

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
```

```

    { x: 'WW', y: 12, y1: 22, y2: 38.3, y3: 50 },
    { x: 'EU', y: 9.9, y1: 26, y2: 45.2, y3: 63.6 },
    { x: 'APAC', y: 4.4, y1: 9.3, y2: 18.2, y3: 20.9 },
    { x: 'LATAM', y: 6.4, y1: 28, y2: 46.7, y3: 65.1 },
    { x: 'MEA', y: 30, y1: 45.7, y2: 61.5, y3: 73 },
    { x: 'NA', y: 25.3, y1: 35.9, y2: 64, y3: 81.4 }
  ];

  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }}
    legendSettings={{
      width: '180',
      height: "20",
      enablePages: false
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' name="December 2007" type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' name="December 2008" type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' name="December 2009" type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y3' name="December 2010" type='Column'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: 'WW', y: 12, y1: 22, y2: 38.3, y3: 50 },
    { x: 'EU', y: 9.9, y1: 26, y2: 45.2, y3: 63.6 },
    { x: 'APAC', y: 4.4, y1: 9.3, y2: 18.2, y3: 20.9 },
    { x: 'LATAM', y: 6.4, y1: 28, y2: 46.7, y3: 65.1 },
    { x: 'MEA', y: 30, y1: 45.7, y2: 61.5, y3: 73 },

```

```

    { x: 'NA', y: 25.3, y1: 35.9, y2: 64, y3: 81.4 }
  ];

  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }}
    legendSettings={{
      width: '180',
      height: "20",
      enablePages: false
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' name="December 2007" type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' name="December 2008" type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' name="December 2009" type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y3' name="December 2010" type='Column'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Legend item padding

The [itemPadding](#) property can be used to adjust the space between the legend items.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: 'WW', y: 12, y1: 22, y2: 38.3, y3: 50 },
    { x: 'EU', y: 9.9, y1: 26, y2: 45.2, y3: 63.6 },
    { x: 'APAC', y: 4.4, y1: 9.3, y2: 18.2, y3: 20.9 },
    { x: 'LATAM', y: 6.4, y1: 28, y2: 46.7, y3: 65.1 },
    { x: 'MEA', y: 30, y1: 45.7, y2: 61.5, y3: 73 },
    { x: 'NA', y: 25.3, y1: 35.9, y2: 64, y3: 81.4 }
  ];
};

```

```

return <Chart3DComponent id='charts'
  primaryXAxis={{
    valueType: "Category",
  }}
  legendSettings={{
    enablePages: false,
    itemPadding: 30
  }}
  wallColor='transparent'
  enableRotation={true} rotation={7} tilt={10} depth={100}>
  <Inject services={[ColumnSeries3D, Category3D, Legend3D,
  Tooltip3D, DataLabel3D, Highlight3D]} />
  <Chart3DSeriesCollectionDirective >
    <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' name="December 2007" type='Column'>
    </Chart3DSeriesDirective>
    <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' name="December 2008" type='Column'>
    </Chart3DSeriesDirective>
    <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' name="December 2009" type='Column'>
    </Chart3DSeriesDirective>
    <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y3' name="December 2010" type='Column'>
    </Chart3DSeriesDirective>
  </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: 'WW', y: 12, y1: 22, y2: 38.3, y3: 50 },
    { x: 'EU', y: 9.9, y1: 26, y2: 45.2, y3: 63.6 },
    { x: 'APAC', y: 4.4, y1: 9.3, y2: 18.2, y3: 20.9 },
    { x: 'LATAM', y: 6.4, y1: 28, y2: 46.7, y3: 65.1 },
    { x: 'MEA', y: 30, y1: 45.7, y2: 61.5, y3: 73 },
    { x: 'NA', y: 25.3, y1: 35.9, y2: 64, y3: 81.4 }
  ];

  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }}
    legendSettings={{
      enablePages: false,

```

```

        itemPadding: 30
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' name="December 2007" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y1' name="December 2008" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y2' name="December 2009" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y3' name="December 2010" type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Note: To use the legend feature, we need to inject the `Legend3D` module into the services.

Tooltip in React 3D Chart control

<!-- markdownlint-disable MD036 -->

The 3D Chart will display details about the points through tooltip, when the mouse is moved over the specific point.

Default tooltip

By default, tooltip is not visible. The tooltip can be enabled by setting the `enable` in `tooltipSettings` to `true` and by injecting the `Tooltip3D` module into the services.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];
}

```

```

    return <Chart3DComponent id='charts' primaryXAxis={{
      valueType: "Category",
    }}
      tooltip={{ enable: true }}
      wallColor='transparent'
      enableRotation={true} rotation={7} tilt={10} depth={100}>
      <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
      <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
          </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
      </Chart3DComponent>;
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
    tooltip={{ enable: true }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;

```

```
ReactDOM.render(<App />, document.getElementById("charts"));
```

<!-- markdownlint-disable MD013 -->

Fixed tooltip

By default, tooltip track the mouse movement, but the tooltip can be set in fixed position by using the [location](#) property.

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];
  const tooltip = { location: { x: 120, y: 20 }, enable: true };
  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
  tooltip={tooltip}
  wallColor='transparent'
  enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
```

```

    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];
  const tooltip = { location: { x: 120, y: 20 }, enable: true };
  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
    tooltip={tooltip}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
    yName='gold' type='Column'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Format the tooltip

<!-- markdownlint-disable MD013 -->

By default, tooltip shows information of x and y value in points. In addition to that, more information can be shown in tooltip by using the [format](#) property. For example the format `${series.name}`: `${point.y}` shows series name and point y value.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];
  const tooltip = { enable: true,
    header: 'Unemployment',

```



```

        format: '<b>${series.name} : ${point.y}</b>' };
    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
    tooltip={tooltip}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Country" type='Column'>
                </Chart3DSeriesDirective>
            </Chart3DSeriesCollectionDirective>
        </Chart3DComponent>;
    }
    ;
    export default App;
    ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];
    const tooltip = { enable: true,
        header: 'Unemployment',
        format: '<b>${series.name} : ${point.y}</b>' };
    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
    tooltip={tooltip}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Country" type='Column'>
                </Chart3DSeriesDirective>
            </Chart3DSeriesCollectionDirective>
        </Chart3DComponent>;

```

```
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Tooltip template

Any HTML elements can be displayed in the tooltip by using the [template](#) property of the tooltip. The `${x}` and `${y}` can be used as place holders in the HTML element to display the x and y values of the corresponding data point.

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];
  const tooltip = { enable: true,
    template: "<div id='templateWrap'><table style='width:100%; border:
1px solid black;'><tr><th colspan='2'
bgcolor='#00FFFF'>Unemployment</th></tr><tr><td
bgcolor='#00FFFF'>${x}</td><td
bgcolor='#00FFFF'>${y}</td></tr></table></div>" };
  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
    tooltip={tooltip}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Country" type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];
    const tooltip = { enable: true,
        template: "<div id='templateWrap'><table style='width:100%; border:
1px solid black;'><tr><th colspan='2'
bgcolor='#00FFFF'>Unemployment</th></tr><tr><td
bgcolor='#00FFFF'>${x}:</td><td
bgcolor='#00FFFF'>${y}</td></tr></table></div>" };
    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
        tooltip={tooltip}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Country" type='Column'>
                </Chart3DSeriesDirective>
            </Chart3DSeriesCollectionDirective>
        </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Customize the appearance of 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

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },

```

```

    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];
  const tooltip = { enable: true,
    format: '${series.name} ${point.x} : ${point.y}',
    fill: '#7bb4eb',
    border: {
      width: 2,
      color: 'grey'
    }
  };
  return <Chart3DComponent id='charts' primaryXAxis={{
    valueType: "Category",
  }}
    tooltip={tooltip}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Country" type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];
  const tooltip = { enable: true,
    format: '${series.name} ${point.x} : ${point.y}',

```

```

        fill: '#7bb4eb',
        border: {
            width: 2,
            color: 'grey'
        }
    };
    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
        tooltip={tooltip}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
        Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
            yName='gold' name="Country" type='Column'>
                </Chart3DSeriesDirective>
            </Chart3DSeriesCollectionDirective>
        </Chart3DComponent>;
    };
    export default App;
    ReactDOM.render(<App />, document.getElementById("charts"));

```

<!-- markdownlint-disable MD036 -->

Selection in React 3D Chart control

The 3D chart provides selection support for the series and its data points on mouse click.

When mouse is clicked on the data points, the corresponding series legend will also be selected.

We have different types of selection mode for selecting a data.

- None
- Point
- Series
- Cluster

Point

To select a point, set the [selectionMode](#) property to **Point**.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective, Selection3D,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    ]

```

```

    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }} selectionMode="Point"
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Selection3D, Category3D,
    Legend3D, Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' type='Column'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective, Selection3D,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }} selectionMode="Point"
    wallColor='transparent'

```

```

        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Selection3D, Category3D,
Legend3D, Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
                </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' type='Column'>
                </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' type='Column'>
                </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Series

To select a series, set the [selectionMode](#) property to **Series**.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective, Selection3D,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts'
primaryXAxis={{
    valueType: "Category",
}} selectionMode="Series"
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Selection3D, Category3D,
Legend3D, Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
                </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' type='Column'>

```

```

        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective, Selection3D,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts'
primaryXAxis={{
    valueType: "Category",
}} selectionMode="Series"
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Selection3D, Category3D,
Legend3D, Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```


Cluster

To select the points that corresponds to the same index in all the series, set the [selectionMode](#) property to **Cluster**.

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective, Selection3D,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }} selectionMode="Cluster"
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Selection3D, Category3D,
Legend3D, Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' type='Column'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective, Selection3D,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
```

```

import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }} selectionMode="Cluster"
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Selection3D, Category3D,
    Legend3D, Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' type='Column'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Selection type

To select multiple points or series, enable the [isMultiSelect](#) property.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective, Selection3D,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

```

```

    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }} selectionMode="Series"
    isMultiSelect={true}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Selection3D, Category3D,
    Legend3D, Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' type='Column'>
      </Chart3DSeriesDirective>
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' type='Column'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective, Selection3D,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }} selectionMode="Series"
    isMultiSelect={true}
    wallColor='transparent'

```

```

        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Selection3D, Category3D,
Legend3D, Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
                </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' type='Column'>
                </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' type='Column'>
                </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Selection during initial loading

In a 3D chart, selecting a point or series during initial loading can only be done programmatically. The [selectedDataIndexes](#) property can be used for this.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective, Selection3D,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts'
primaryXAxis={{
    valueType: "Category",
}} selectionMode="Series"
isMultiSelect={true}
selectedDataIndexes={[
    { series: 0, point: 1}, { series: 2, point: 3}
]}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Selection3D, Category3D,
Legend3D, Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >

```

```

        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective, Selection3D,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts'
primaryXAxis={{
    valueType: "Category",
}} selectionMode="Series"
isMultiSelect={true}
selectedDataIndexes={[
    { series: 0, point: 1}, { series: 2, point: 3}
]]
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Selection3D, Category3D,
Legend3D, Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' type='Column'>
        </Chart3DSeriesDirective>

```

```

        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Selection through legend

To select a point or series through legend use the [toggleVisibility](#) property. Also, use [enableHighlight](#) property for highlighting the series through legend.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective, Selection3D,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts'
primaryXAxis={{
    valueType: "Category",
}}
legendSettings={{
    visible: true,
    toggleVisibility: false,
    enableHighlight: true
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Selection3D, Category3D,
Legend3D, Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Gold" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' name="Silver" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' name="Bronze" type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>

```

```

        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective, Selection3D,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts'
    primaryXAxis={{
        valueType: "Category",
    }}
    legendSettings={{
        visible: true,
        toggleVisibility: false,
        enableHighlight: true
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Selection3D, Category3D,
Legend3D, Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Gold" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' name="Silver" type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' name="Bronze" type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Print and Export in React 3D Chart control

Print

The rendered 3D chart can be printed directly from the browser by calling the public method [print](#). The ID of the 3D chart's div element must be passed as the input parameter to that method.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];
  let chartInstance;
  function clickHandler() {
    chartInstance.print();
  }

  return <div>
    <button value='print'
onClick={clickHandler.bind(this)}>Print</button>
    <Chart3DComponent id='charts' ref={chart => chartInstance = chart}
primaryXAxis={{
      valueType: "Category",
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Country" type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>
  </div>
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
```



```

Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];
    let chartInstance;
    function clickHandler() {
        chartInstance.print();
    }

    return <div>
        <button value='print'
onClick={clickHandler.bind(this)}>Print</button>
        <Chart3DComponent id='charts' ref={chart => chartInstance = chart}
primaryXAxis={{
            valueType: "Category",
        }}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
            <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
            <Chart3DSeriesCollectionDirective >
                <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Country" type='Column'>
                    </Chart3DSeriesDirective>
                </Chart3DSeriesCollectionDirective>
            </Chart3DComponent>
        </div>
    </div>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Export

The rendered 3D chart can be exported to JPEG, PNG, SVG, or PDF format using the [export](#) method. The input parameters for this method are: `type` for format and `fileName` for result.

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective, Export3D,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
```

```

    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];
  let chartInstance;
  function clickHandler() {
    chartInstance.export3DModule.export('PNG', 'sample');
  }

  return <div>
    <button value='print'
      onClick={clickHandler.bind(this)}>Export</button>
    <Chart3DComponent id='charts' ref={chart => chartInstance = chart}
      primaryXAxis={{
        valueType: "Category",
      }}
      wallColor='transparent'
      enableRotation={true} rotation={7} tilt={10} depth={100}>
      <Inject services={[ColumnSeries3D, Export3D, Category3D,
        Legend3D, Tooltip3D, DataLabel3D, Highlight3D]} />
      <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
          yName='gold' name="Country" type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>
  </div>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective, Export3D,
  Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
  Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
  charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

```

```

let chartInstance;
function clickHandler() {
    chartInstance.export3DModule.export('PNG', 'sample');
}

return <div>
    <button value='print'
onClick={clickHandler.bind(this)}>Export</button>
    <Chart3DComponent id='charts' ref={chart => chartInstance = chart}
primaryXAxis={{
    valueType: "Category",
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Export3D, Category3D,
Legend3D, Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' name="Country" type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>
</div>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Appearance in React 3D Chart control

Custom color palette

The default color of series or points can be customized by providing a custom color palette of your choice by using the [palettes](#) property.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}

```

```

        palettes=[["#E94649", "#F6B53F", "#6FAAB0", "#C4C24A"]]
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' type='Column'>
            </Chart3DSeriesDirective>
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
    palettes=[["#E94649", "#F6B53F", "#6FAAB0", "#C4C24A"]]
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='silver' type='Column'>

```

```

        </Chart3DSeriesDirective>
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='bronze' type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Data point customization

The color of an individual data point can be customized using the below options.

Point color mapping

The color for the points can be bound from the [dataSource](#) for the series by utilizing the [pointColorMapping](#) property.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 'Jan', y: 6.96, color: "red" },
        { x: 'Feb', y: 8.9, color: "blue" },
        { x: 'Mar', y: 12, color: "orange" },
        { x: 'Apr', y: 17.5, color: "aqua" },
        { x: 'May', y: 22.1, color: "gray" }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='Column' pointColorMapping="color">
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { x: 'Jan', y: 6.96, color: "red" },
        { x: 'Feb', y: 8.9, color: "blue" },
        { x: 'Mar', y: 12, color: "orange" },
        { x: 'Apr', y: 17.5, color: "aqua" },
        { x: 'May', y: 22.1, color: "gray" }
    ];

    return <Chart3DComponent id='charts' primaryXAxis={{
        valueType: "Category",
    }}
        wallColor='transparent'
        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
        Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='x'
yName='y' type='Column' pointColorMapping="color">
                </Chart3DSeriesDirective>
            </Chart3DSeriesCollectionDirective>
        </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Point level customization

The data label and fill color of each data point can be customized using the [pointRender](#) and [textRender](#) events.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];
    const pointRender = (args) => {

```

```

    let seriesColor = ['#00bdae', '#404041', '#357cd2', '#e56590',
    '#f8b883',
    '#70ad47', '#dd8abd', '#7f84e8', '#7bb4eb', '#ea7a57'];
    args.fill = seriesColor[args.point.index];
  };

  return <Chart3DComponent id='charts' pointRender={pointRender}
primaryXAxis={{
  valueType: "Category",
}}
  wallColor='transparent'
  enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];
  const pointRender = (args) => {
    let seriesColor = ['#00bdae', '#404041', '#357cd2', '#e56590',
    '#f8b883',
    '#70ad47', '#dd8abd', '#7f84e8', '#7bb4eb', '#ea7a57'];
    args.fill = seriesColor[args.point.index];
  };

  return <Chart3DComponent id='charts' pointRender={pointRender}
primaryXAxis={{
  valueType: "Category",
}}
  wallColor='transparent'

```

```

        enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

<!-- markdownlint-disable MD036 -->

Chart area customization

<!-- markdownlint-disable MD036 -->

Customize the chart background

The background color and border of the 3D chart can be customized using the [background](#) and [border](#) properties.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts' background="skyblue"
border={{ color: "#FF0000", width: 2 }}
primaryXAxis={{
        valueType: "Category",
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
        <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>

```



```

    </Chart3DComponent>;
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts' background="skyblue"
border={{ color: "#FF0000", width: 2 }}
primaryXAxis={{
  valueType: "Category",
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
  <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
  <Chart3DSeriesCollectionDirective >
    <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Chart margin

The 3D chart's margin can be set from its container using the [margin](#) property.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";

```

```

import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts' background="skyblue"
    border={{ color: "#FF0000", width: 2 }}
    //Change chart margin to left, right, top and bottom
    margin={{ left: 40, right: 40, top: 40, bottom: 40 }}
    primaryXAxis={{
      valueType: "Category",
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts' background="skyblue"

```

```

border={{ color: "#FF0000", width: 2 }}
//Change chart margin to left, right, top and bottom
margin={{ left: 40, right: 40, top: 40, bottom: 40 }}
primaryXAxis={{
  valueType: "Category",
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
<Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
<Chart3DSeriesCollectionDirective >
  <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
    </Chart3DSeriesDirective>
  </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Animation

To customize the animation for a particular series, the [animation](#) property can be used. It can be enabled or disabled by using the [enable](#) property. The [duration](#) property specifies the duration of an animation and the [delay](#) property allows us to start the animation at desire time.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column' animation={{

```

```

        enable: true,
        duration: 2000,
        delay: 200
    }>
    </Chart3DSeriesDirective>
  </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts'
    primaryXAxis={{
        valueType: "Category",
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column' animation={{
            enable: true,
            duration: 2000,
            delay: 200
        }}>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Chart rotation

The 3D chart can be rotated by using the [enableRotation](#) property in the chart.

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
```

```

    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
    yName='gold' type='Column'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Title

The 3D chart can be given a title by using [title](#) property, to show the information about the data plotted.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }} title="Olympic Medals"
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >

```

```

        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
</Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts'
primaryXAxis={{
    valueType: "Category",
}} title="Olympic Medals"
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Title position

By using the [position](#) property in [titleStyle](#), the [title](#) can be positioned at left, right, top or bottom of the 3D chart. The title is positioned at the top of the 3D chart, by default.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }} title="Olympic Medals"
    titleStyle= {{ position: "Bottom" }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

```



```

];

return <Chart3DComponent id='charts'
primaryXAxis={{
  valueType: "Category",
}} title="Olympic Medals"
titleStyle= {{ position: "Bottom" }}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
  <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
  <Chart3DSeriesCollectionDirective >
    <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
      </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
  </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

The custom option is used to position the title anywhere in the 3D chart using [x](#) and [y](#) coordinates.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts'
primaryXAxis={{
  valueType: "Category",
}} title="Olympic Medals"
titleStyle= {{
  position: 'Custom',
  x: 300,
  y: 60
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
  <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />

```

```

        <Chart3DSeriesCollectionDirective >
            <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts'
primaryXAxis={{
    valueType: "Category",
}} title="Olympic Medals"
titleStyle= {{
    position: 'Custom',
    x: 300,
    y: 60
}}
wallColor='transparent'
enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
        </Chart3DSeriesDirective>
    </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Title alignment

The title can be aligned to the near, far, or center of the 3D chart by using the [textAlignment](#) property.

INDEX.JSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
        { country: "China", gold: 40, silver: 60, bronze: 55 },
        { country: "Japan", gold: 70, silver: 60, bronze: 50 },
        { country: "Australia", gold: 60, silver: 56, bronze: 40 },
        { country: "France", gold: 50, silver: 45, bronze: 35 },
        { country: "Germany", gold: 40, silver: 30, bronze: 22 },
        { country: "Italy", gold: 40, silver: 35, bronze: 37 },
        { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
    ];

    return <Chart3DComponent id='charts'
    primaryXAxis={{
        valueType: "Category",
    }} title="Olympic Medals"
    titleStyle= {{
        textAlignment: 'Far'
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
            </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    const data = [
        { country: "USA", gold: 50, silver: 70, bronze: 45 },
```

```

    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }} title="Olympic Medals"
    titleStyle={{
      textAlign: 'Far'
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Title customization

The [titleStyle](#) property of the 3D chart title provides options to customize the title by using the following properties.

- [size](#) - Specifies the size of the title.
- [color](#) - Specifies the color for the title.
- [fontFamily](#) - Specifies the font family for the title.
- [fontWeight](#) - Specifies the font weight of the title.
- [fontStyle](#) - Specifies the font style for the title.
- [opacity](#) - Specifies the opacity for the color of the title.
- [textAlignment](#) - Specifies the alignment of the title.
- [textOverflow](#) - Specifies the overflow of the title.

INDEX.JSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },

```

```

    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }} title="Olympic Medals"
    titleStyle= {{
      size: '18px', color: 'Red', textOverflow: 'Wrap'
    }}
    wallColor='transparent'
    enableRotation={true} rotation={7} tilt={10} depth={100}>
    <Inject services={[ColumnSeries3D, Category3D, Legend3D,
    Tooltip3D, DataLabel3D, Highlight3D]} />
    <Chart3DSeriesCollectionDirective >
      <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
        </Chart3DSeriesDirective>
      </Chart3DSeriesCollectionDirective>
    </Chart3DComponent>;
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { Chart3DComponent, Chart3DSeriesCollectionDirective,
Chart3DSeriesDirective, Category3D, Inject, Legend3D, DataLabel3D,
Tooltip3D, ColumnSeries3D, Highlight3D } from '@syncfusion/ej2-react-
charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { country: "USA", gold: 50, silver: 70, bronze: 45 },
    { country: "China", gold: 40, silver: 60, bronze: 55 },
    { country: "Japan", gold: 70, silver: 60, bronze: 50 },
    { country: "Australia", gold: 60, silver: 56, bronze: 40 },
    { country: "France", gold: 50, silver: 45, bronze: 35 },
    { country: "Germany", gold: 40, silver: 30, bronze: 22 },
    { country: "Italy", gold: 40, silver: 35, bronze: 37 },
    { country: "Sweden", gold: 30, silver: 25, bronze: 27 }
  ];

  return <Chart3DComponent id='charts'
    primaryXAxis={{
      valueType: "Category",
    }} title="Olympic Medals"
    titleStyle= {{

```

```

        size: '18px', color: 'Red', textOverflow: 'Wrap'
      }}
      wallColor='transparent'
      enableRotation={true} rotation={7} tilt={10} depth={100}>
      <Inject services={[ColumnSeries3D, Category3D, Legend3D,
Tooltip3D, DataLabel3D, Highlight3D]} />
      <Chart3DSeriesCollectionDirective >
        <Chart3DSeriesDirective dataSource={data} xName='country'
yName='gold' type='Column'>
          </Chart3DSeriesDirective>
        </Chart3DSeriesCollectionDirective>
      </Chart3DComponent>;
    };
    export default App;
    ReactDOM.render(<App />, document.getElementById("charts"));

```

Accessibility in React 3D Chart control

Accessibility is achieved in the 3D chart control through WAI-ARIA standard and keyboard navigation. The 3D chart features can be effectively accessed through assistive technologies such as screen readers.

WAI-ARIA

WAI-ARIA (Accessibility Initiative – Accessible Rich Internet Applications) defines a way to increase the accessibility of web pages, dynamic content, and user interface components developed with AJAX, HTML, JavaScript, and related technologies. ARIA provides additional semantics to describe the role, state, and functionality of web components.

- img (role)
- button (role)
- region (role)
- aria-label (attribute)
- aria-hidden (attribute)
- aria-pressed (attribute)

Keyboard navigation

All the 3D chart actions can be controlled via keyboard keys. The applicable key combinations and their relative functionalities are listed below.

Interaction Keys | Description

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

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

DownArrow | Moves the focus to the data point left side from the selected point.

UpArrow | Moves the focus to the data point right side from the selected point.

Left Arrow | Moves the focus to the next series in the chart.

Right Arrow | Moves the focus to the previous series in the chart.

ESC | Cancel the tooltip for the data point.

Enter/Space | Selects the data point in the series.

Down/Left Arrow | Moves the focus to the legend left side from the selected legend.

Up/Right Arrow | Moves the focus to the legend right side from the selected legend.

Enter/Space | Toggles the visibility of the corresponding series.

Ctrl + P | Print the chart.

Accordion

Getting Started

This section briefly explains you the steps required to create a simple [React Accordion](#) and demonstrate the basic usage of the Accordion control.

Dependencies

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

```
`js
|-- @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 Accordion 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.ts` file.

Initialize the Accordion using Items

The React Accordion can be rendered by defining an array of [items](#).

- Import the Accordion component to your `src/App.ts` file using following code.

```
`ts
```



```
import { AccordionComponent, AccordionItemDirective, AccordionItemsDirective } from
 '@syncfusion/ej2-react-navigations';
```

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

```
const ReactApp = () => {
```

```
  const aspContent = () => {
```

```
    return (<div>
```

Microsoft ASP.NET is a set of technologies in the Microsoft .NET Framework for building Web applications and XML Web services.

```
    </div>);
```

```
  }
```

```
  const mvcContent = () => {
```

```
    return (<div>
```

The Model-View-Controller (MVC) architectural pattern separates an application into three main components: the model, the view, and the controller.

```
    </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.

```
    </div>);
```

```
  }
```

```
  return (
```

```
    <AccordionComponent>
```

```
      <AccordionItemsDirective>
```

```
        <AccordionItemDirective header='ASP.NET' content={aspContent} />
```

```
        <AccordionItemDirective header='ASP.NET MVC' content={mvcContent} />
```

```
        <AccordionItemDirective header='JavaScript' content={jsContent} />
```

```
      </AccordionItemsDirective>
```

```
    </AccordionComponent>
```

```
  );
```

```
}
```

```
export default 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 Accordion is initialized.

Initialize the Accordion using HTML elements

The React Accordion component can be rendered based on the given HTML element using `id` as `target` property.

You need to follow the below structure of HTML elements to render the Accordion.

,

`<div id='accordion/html/markup'>` --> Root Accordion Element

`<div>` --> Accordion Item Container

`<div>` --> Accordion Header Container

`<div> </div>` --> Accordion Header

`</div>`

`<div>` --> Accordion Panel Container

`<div> </div>` --> Accordion Content

`</div>`

`</div>`

`</div>`

,

INDEX.JSX

```
import { AccordionComponent } from '@syncfusion/ej2-react-navigations';
import * as React from "react";
import * as ReactDOM from "react-dom";
const ReactApp = () => {
    return (<AccordionComponent>
        <div>
            <div>
                <div> ASP.NET </div>
            </div>
            <div>
                <div>
                    Microsoft ASP.NET is a set of technologies in the Microsoft .NET
                    Framework for building Web applications and XML Web services.
                </div>
            </div>
        </div>
        <div>
            <div>
```

```

        <div> ASP.NET MVC </div>
    </div>
    <div>
        <div>
            The Model-View-Controller (MVC) architectural pattern separates
            an application into three main components: the model, the view, and the
            controller.
        </div>
    </div>
</div>
<div>
    <div>
        <div> JavaScript </div>
    </div>
    <div>
        <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.
        </div>
    </div>
</div>
</AccordionComponent>);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

INDEX.TSX

```

import { AccordionComponent } from '@syncfusion/ej2-react-navigations';
import * as React from "react";
import * as ReactDOM from "react-dom";
const ReactApp = () => {
    return (
        <AccordionComponent>
            <div>
                <div>
                    <div> ASP.NET </div>
                </div>
                <div>
                    <div>
                        Microsoft ASP.NET is a set of technologies in the Microsoft .NET
                        Framework for building Web applications and XML Web services.
                    </div>
                </div>
            </div>
            <div>
                <div>
                    <div> ASP.NET MVC </div>
                </div>
                <div>
                    <div>
                        The Model-View-Controller (MVC) architectural pattern separates
                        an application into three main components: the model, the view, and the
                        controller.
                    </div>
                </div>
            </div>
        </div>
    )
}

```

```

        </div>
      </div>
      <div>
        <div>
          <div> JavaScript </div>
        </div>
        <div>
          <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.
          </div>
        </div>
      </div>
    </AccordionComponent>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Essential JS 2 React Accordion Sample</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></sc
ript>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
  </style>
</head>
<body>
  <div id='element'>
    <div id='loader'>Loading....</div></div>
</body>
</html>

```

You can add the custom class into Accordion component using [cssClass](#) property which is used to customize the Accordion component.

See Also

- [How to load accordion items dynamically](#)

Note: You can also explore our [React Accordion example](#) that shows you how to configure the Accordion in React.

Expand mode in React Accordion component

The Accordion supports the two listed types of expand modes while expanding or collapsing the item.

- Single
- Multiple

Single

The property enables to expand only one Accordion item at a time. If you expand any new item, the previously expanded one is collapsed and new item changed to expanded state.

You can also choose which accordion pane is expanded state at initial rendering by enabling the [expanded](#) property on accordion items.

INDEX.JSX

```
import * as ReactDOM from "react-dom";
import * as React from "react";
import { AccordionComponent, AccordionItemDirective, AccordionItemsDirective } from '@syncfusion/ej2-react-navigations';
const ReactApp = () => {
  const aspContent = () => {
    return (<div>
      Microsoft ASP.NET is a set of technologies in the Microsoft .NET Framework for building Web applications and XML Web services.
    </div>);
  }
  const mvcContent = () => {
    return (<div>
      The Model-View-Controller (MVC) architectural pattern separates an application into three main components: the model, the view, and the controller.
    </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.
    </div>);
  }
  return (
    <AccordionComponent expandMode='Single'>
      <AccordionItemsDirective>
```

```

        <AccordionItemDirective expanded={true} header='ASP.NET'
content={aspContent} />
        <AccordionItemDirective header='ASP.NET MVC' content={mvcContent} />
        <AccordionItemDirective header='JavaScript' content={jsContent} />
    </AccordionItemsDirective>
</AccordionComponent>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccordionComponent, AccordionItemDirective, AccordionItemsDirective
} from '@syncfusion/ej2-react-navigations';
const ReactApp = () => {
    const aspContent = () => {
        return <div>
            Microsoft ASP.NET is a set of technologies in the Microsoft .NET
Framework for building Web applications and XML Web services.
        </div>;
    }
    const mvcContent = () => {
        return <div>
            The Model-View-Controller (MVC) architectural pattern separates an
application into three main components: the model, the view, and the
controller.
        </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.
        </div>;
    };
    return (
        <AccordionComponent expandMode='Single'>
            <AccordionItemsDirective>
                <AccordionItemDirective expanded={true} header='ASP.NET'
content={aspContent} />
                <AccordionItemDirective header='ASP.NET MVC' content={mvcContent} />
                <AccordionItemDirective header='JavaScript' content={jsContent} />
            </AccordionItemsDirective>
        </AccordionComponent>
    );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

INDEX.HTML

```

<!DOCTYPE html>

```

```

<html lang="en">
<head>
  <title>Essential JS 2 React Accordion Sample</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></sc
ript>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
  </style>
</head>
<body>
  <div id='element'>
    <div id='loader'>Loading....</div></div>
</body>
</html>

```

Multiple

Default [expandMode](#) of the Accordion is **Multiple**. It enables you to expand more than one Accordion item at a time. Expand/collapse action can also be toggled by clicking on it again. For example, expanded item is collapsed when you click on it again.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccordionComponent, AccordionItemDirective, AccordionItemsDirective
} from '@syncfusion/ej2-react-navigations';
const ReactApp = () => {
  const aspContent = () => {
    return (<div>
      Microsoft ASP.NET is a set of technologies in the Microsoft .NET
      Framework for building Web applications and XML Web services.
    </div>);
  }
  const mvcContent = () => {
    return (<div>
      The Model-View-Controller (MVC) architectural pattern separates an
      application into three main components: the model, the view, and the
      controller.
    </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.
  </div>);
};
return (
  <AccordionComponent expandMode='Multiple'>
    <AccordionItemsDirective>
      <AccordionItemDirective expanded={true} header='ASP.NET'
content={aspContent} />
      <AccordionItemDirective header='ASP.NET MVC' content={mvcContent} />
      <AccordionItemDirective header='JavaScript' content={jsContent} />
    </AccordionItemsDirective>
  </AccordionComponent>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

INDEX.TSX

```

import * as React from "react";
import { AccordionComponent, AccordionItemDirective, AccordionItemsDirective
} from '@syncfusion/ej2-react-navigations';
import * as ReactDOM from "react-dom";
const ReactApp = () => {
  const aspContent = () => {
    return (<div>
      Microsoft ASP.NET is a set of technologies in the Microsoft .NET
      Framework for building Web applications and XML Web services.
    </div>);
  }
  const mvcContent = () => {
    return (<div>
      The Model-View-Controller (MVC) architectural pattern separates an
      application into three main components: the model, the view, and the
      controller.
    </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.
    </div>);
  }
  return (
    <AccordionComponent expandMode='Multiple'>
      <AccordionItemsDirective>
        <AccordionItemDirective expanded={true} header='ASP.NET'
content={aspContent} />
        <AccordionItemDirective header='ASP.NET MVC' content={mvcContent} />
      </AccordionItemsDirective>
    </AccordionComponent>
  );
}

```



```

        <AccordionItemDirective header='JavaScript' content={jsContent} />
      </AccordionItemsDirective>
    </AccordionComponent>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Essential JS 2 React Accordion Sample</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" />
  <script
src="https://cdn.jsdelivr.net/npm/systemjs@0.19.38/system.js"></sc
ript>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008c8f;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
  </style>
</head>
<body>
  <div id='element'>
    <div id='loader'>Loading....</div></div>
</body>
</html>

```

See Also

- [How to keep single pane open always](#)

Accessibility in React Accordion component

The Accordion component has been designed keeping in mind the [WAI-ARIA](#) specifications, by applying the prompt WAI-ARIA roles, states, and properties along with the keyboard support. Thus, making it usable for people who use assistive WAI-ARIA Accessibility supports that is achieved through the attributes like `aria-labelledby`. It helps to provides information about the elements in a document for assistive technology. The component implements the keyboard navigation support by following the [WAI-ARIA practices](#) and tested in major screen readers.

The accessibility compliance for the Accordion 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

| **Roles and Attributes** | **Functionalities**

|

----- -----

| role | **Button:** Attribute is set to the Accordion header elements to indicate that the element can be used to toggle the visibility of the associated content section, describing the actual role of the element.
 Region: Attribute is set to the Accordion panel elements to create a landmark region that contains the currently expanded accordion panel, describing the actual role of the element.
|

| aria-labelledby | Attribute is set to content (panel) and it points to the corresponding Accordion header. |

| aria-controls | Attribute is set to the header and it points to the corresponding Accordion content. |

| aria-expanded | Attribute is set to the Accordion header elements to indicates the expand state of the Accordion Item. Default value of this attribute is **false**. If an item is expanded, the attribute value changes to 'true'. |

| aria-hidden | Attribute is set to the Accordion panel elements to indicates the content visible state of the Accordion Item. Default value of this attribute is **true**. If an item content is visible, the attribute value changes to **false**. |

| aria-disabled | It indicates the disabled state of the Accordion and its items. |

Keyboard interaction

Keyboard navigation is enabled by default. Possible keys are:

Key Description
----- -----

| Space or Enter | When focus is on the Accordion header, click on the focused element makes the element to expand and collapse. |

| Down Arrow | Focus the next Accordion header. |

| Up Arrow | Focus the previous Accordion header. |

| Home | Focus the first Accordion header. |

| End | Focus the last Accordion header. |

| Tab | To Move focus through the interactive elements. |

| Shift + Tab | To Move focus through the interactive elements. |

Ensuring accessibility

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

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

See also

- [Accessibility in Syncfusion React components](#)

Style in React Accordion component

The following content provides the exact CSS structure that can be used to modify the control's appearance based on user preference.

Customizing Accordion

Use the following CSS to customize the Accordion.

```
`css
.e-accordion {
border: 5px solid rgb(173, 255, 47);
}
`
```

Customizing the list items

Use the following CSS to customize the items of Accordion.

```
`css
.e-accordion .e-acrdn-item {
text-align: center;
color: pink;
background-color: #2fa1ff;
}
`
```

Customizing Accordion's header

Use the following CSS to customize the header of Accordion control.

```
`css
.e-accordion .e-acrdn-item.e-select > .e-acrdn-header {
background: #2fa1ff !important;
justify-content: center;
}
`
```

Customizing Accordion's expand and collapse icons

Use the following CSS to customize the expand and collapse icons of Accordion control.

```
`css
.e-accordion .e-acrdn-item .e-acrdn-header .e-toggle-icon .e-icons {
color: pink;
}
```

```
}  
,
```

Customizing the hover state of Accordion control

Use the following CSS to customize the accordion item when hovering.

```
`css  
  
.e-accordion .e-acrdn-item .e-acrdn-header:hover {  
border: 2px solid gray;  
}  
,
```

Customizing selected item of Accordion control

Use the following CSS to customize the selected accordion item.

```
`css  
  
.e-accordion .e-acrdn-item.e-select.e-active>.e-acrdn-header,  
.e-accordion .e-acrdn-item.e-select.e-item-focus>.e-acrdn-header {  
background-color: rgb(0, 15, 100) !important;  
}  
,
```

Use the following CSS to customize the selected accordion item text.

```
`css  
  
.e-accordion .e-acrdn-item.e-select.e-active>.e-acrdn-header .e-acrdn-header-content,  
.e-accordion .e-acrdn-item.e-select.e-item-focus>.e-acrdn-header .e-acrdn-header-content {  
color: #2fa1ff !important;  
}  
,
```

How To

Set the nested accordion in React Accordion component

Accordion supports to render **nested** level of Accordion by using [content](#) property. You can give nested Accordion content inside the parent Accordion content property by using **id** of nested element. The nested Accordion can be rendered with the use of provided events, such as [clicked](#) and [expanding](#).

INDEX.JSX

```
import * as React from "react";  
import * as ReactDOM from "react-dom";  
import { AccordionComponent, AccordionItemsDirective, AccordionItemDirective  
} from '@syncfusion/ej2-react-navigations';  
const nestedMusicNew = () => {  
  return (  
    <div>
```

```

        <AccordionComponent>
          <AccordionItemsDirective>
            <AccordionItemDirective header='Track1' />
            <AccordionItemDirective header='Track2' />
          </AccordionItemsDirective>
        </AccordionComponent>
      </div>
    );
  }
  const nestedVideo = () => {
    return (
      <div>
        <AccordionComponent>
          <AccordionItemsDirective>
            <AccordionItemDirective header='Video Track1' />
            <AccordionItemDirective header='Video Track2' />
          </AccordionItemsDirective>
        </AccordionComponent>
      </div>
    );
  }
  const nestedMusic = () => {
    return (
      <div>
        <AccordionComponent>
          <AccordionItemsDirective>
            <AccordionItemDirective header='Music Track1' />
            <AccordionItemDirective header='Music Track2' />
            <AccordionItemDirective header='Music New'
content={nestedMusicNew} />
          </AccordionItemsDirective>
        </AccordionComponent>
      </div>
    );
  }
  const nestedImages = () => {
    return (
      <div>
        <AccordionComponent>
          <AccordionItemsDirective>
            <AccordionItemDirective header='Track1' />
            <AccordionItemDirective header='Track2' />
          </AccordionItemsDirective>
        </AccordionComponent>
      </div>
    );
  }
  const ReactApp = () => {
    return (
      <AccordionComponent>
        <AccordionItemsDirective>
          <AccordionItemDirective header='Video' content={nestedVideo} />
          <AccordionItemDirective header='Music' content={nestedMusic} />
          <AccordionItemDirective header='Images' content={nestedImages} />
        </AccordionItemsDirective>
      </AccordionComponent>
    );
  }

```

```
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from 'react-dom';
import { AccordionComponent, AccordionItemsDirective, AccordionItemDirective } from '@syncfusion/ej2-react-navigations';
const nestedMusicNew = () => {
    return (
        <div>
            <AccordionComponent>
                <AccordionItemsDirective>
                    <AccordionItemDirective header='Track1' />
                    <AccordionItemDirective header='Track2' />
                </AccordionItemsDirective>
            </AccordionComponent>
        </div>
    );
}
const nestedVideo = () => {
    return (
        <div>
            <AccordionComponent>
                <AccordionItemsDirective>
                    <AccordionItemDirective header='Video Track1' />
                    <AccordionItemDirective header='Video Track2' />
                </AccordionItemsDirective>
            </AccordionComponent>
        </div>
    );
}
const nestedMusic = () => {
    return (
        <div>
            <AccordionComponent>
                <AccordionItemsDirective>
                    <AccordionItemDirective header='Music Track1' />
                    <AccordionItemDirective header='Music Track2' />
                    <AccordionItemDirective header='Music New'
content={nestedMusicNew} />
                </AccordionItemsDirective>
            </AccordionComponent>
        </div>
    );
}
const nestedImages = () => {
    return (
        <div>
            <AccordionComponent>
                <AccordionItemsDirective>
                    <AccordionItemDirective header='Track1' />
                    <AccordionItemDirective header='Track2' />
                </AccordionItemsDirective>
            </AccordionComponent>
        </div>
    );
}
```

```

        </AccordionComponent>
      </div>
    );
  }
  const ReactApp = () => {
    return (
      <AccordionComponent>
        <AccordionItemsDirective>
          <AccordionItemDirective header='Video' content={nestedVideo} />
          <AccordionItemDirective header='Music' content={nestedMusic} />
          <AccordionItemDirective header='Images' content={nestedImages} />
        </AccordionItemsDirective>
      </AccordionComponent>
    );
  }
  const root = ReactDOM.createRoot(document.getElementById('element'));
  root.render(<ReactApp />);

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Essential JS 2 React Accordion Sample</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></sc
ript>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
  </style>
</head>
<body>
  <div id='element'>
    <div id='loader'>Loading....</div></div>
</body>
</html>

```

Load content through post in React Accordion component

Accordion supports to load external contents through **AJAX** library. Refer the below steps.

- Import the `Ajax` module from `ej2-base` and initialize with URL path.
- Get data from the Ajax Success event to initialize Accordion with retrieved external path data.

INDEX.JSX

```
import * as React from "react";
import { useState, useEffect } from "react";
import { Ajax } from "@syncfusion/ej2-base";
import * as ReactDOM from "react-dom";
import { AccordionComponent, AccordionItemsDirective, AccordionItemDirective } from "@syncfusion/ej2-react-navigations";

const ReactApp = () => {
  const [employeeDetails, setEmployeeDetails] = useState("");
  useEffect(() => {
    const ajax = new Ajax("./ajax.html", "GET", true);
    ajax.send().then();
    ajax.onSuccess = (dataSt) => {
      setEmployeeDetails(dataSt);
    };
  }, []);
  const deptContent = () => {
    return (
      <div>
        <ul style={{ margin: '0px', padding: '0px 16px', listStyleType: 'none' }}>
          <li>Testing</li>
          <li>Development</li>
        </ul>
      </div>
    );
  };
  const platformContent = () => {
    return (
      <div>
        <ul style={{ margin: '0px', padding: '0px 16px', listStyleType: 'none' }}>
          <li>Mobile</li>
          <li>Web</li>
        </ul>
      </div>
    );
  };
  return (
    <AccordionComponent>
      <AccordionItemsDirective>
        <AccordionItemDirective header="Department" content={deptContent} />
        <AccordionItemDirective header="Platform" content={platformContent} />
        <AccordionItemDirective header="Employee Details" content={employeeDetails} />
      </AccordionItemsDirective>
    </AccordionComponent>
  );
};

const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);
```

INDEX.TSX

```
import * as React from "react";
import { useState, useEffect } from "react";
import { Ajax } from "@syncfusion/ej2-base";
import * as ReactDOM from "react-dom";
import { AccordionComponent, AccordionItemsDirective, AccordionItemDirective } from "@syncfusion/ej2-react-navigations";
const ReactApp = () => {
  const [employeeDetails, setEmployeeDetails] = useState("");
  useEffect(() => {
    const ajax = new Ajax("./ajax.html", "GET", true);
    ajax.send().then();
    ajax.onSuccess = (dataSt: string): void => {
      setEmployeeDetails(dataSt);
    };
  }, []);
  const deptContent = () => {
    return (
      <div>
        <ul style={{ margin: '0px', padding: '0px 16px', listStyleType: 'none' }}>
          <li>Testing</li>
          <li>Development</li>
        </ul>
      </div>
    );
  };
  const platformContent = () => {
    return (
      <div>
        <ul style={{ margin: '0px', padding: '0px 16px', listStyleType: 'none' }}>
          <li>Mobile</li>
          <li>Web</li>
        </ul>
      </div>
    );
  };
  return (
    <AccordionComponent>
      <AccordionItemsDirective>
        <AccordionItemDirective header="Department" content={deptContent} />
        <AccordionItemDirective header="Platform" content={platformContent} />
        <AccordionItemDirective header="Employee Details" content={employeeDetails} />
      </AccordionItemsDirective>
    </AccordionComponent>
  );
};
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);
```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Essential JS 2 Accordion Ajax Sample</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></sc
ript>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
    div#headername {
      margin-left: 100px;
    }
  </style>
</head>
<body>
  <div id='element'>
    <div id='loader'>Loading....</div></div>
</body>
</html>

```

Set custom animation in React Accordion component

Accordion supports custom animations for both expand and collapse actions from the provided animation option of `Animation` library. The `animation` property also allows you to set `easing`, `duration`, and various other effects of your choice.

Default animation is given as `SlideDown` for expanding the panel using `expand` animation property and `SlideUp` for collapsing the panel using `collapse` animation property. You can also disable the animation by setting animation `effect` as `none`.

The sample demonstrates some types of animation that suits for Accordion. You can check all the animation effects [here](#).

INDEX.JSX

```

import * as React from "react";
import { useState, useEffect, useRef } from 'react';
import * as ReactDOM from 'react-dom';
import { Effect } from '@syncfusion/ej2-base';

```

```
import { ChangeEventArgs, DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { AccordionComponent, AccordionItemDirective, AccordionItemsDirective } from '@syncfusion/ej2-react-navigations';
const ReactApp = () => {
    const acrdnInstance = useRef(null);
    const expandInstance = useRef(null);
    const collapseInstance = useRef(null);
    const [collapseAnimationEffect, setCollapseAnimationEffect] =
useState('');
    const [expandAnimationEffect, setExpandAnimationEffect] = useState('');
    const expandAnimation = ['SlideDown', 'SlideUp', 'FadeIn', 'FadeOut',
'FadeZoomIn', 'FadeZoomOut', 'ZoomIn', 'ZoomOut', 'None'];
    useEffect(() => {
        acrdnInstance.current.animation.collapse = { effect:
collapseAnimationEffect };
    }, [collapseAnimationEffect]);
    useEffect(() => {
        acrdnInstance.current.animation.expand = { effect: expandAnimationEffect
};
    }, [expandAnimationEffect]);
    const aspContent = () => {
        return <div>
            Microsoft ASP.NET is a set of technologies in the Microsoft .NET
Framework for building Web applications and XML Web services. ASP.NET pages
execute on the server and generate markup such as HTML, WML, or XML that is
sent to a desktop or mobile browser. ASP.NET pages use a compiled, event-
driven programming model that improves performance and enables the
separation of application logic and user interface.
        </div>;
    }
    const mvcContent = () => {
        return <div>
            The Model-View-Controller (MVC) architectural pattern separates an
application into three main components: the model, the view, and the
controller. The ASP.NET MVC framework provides an alternative to the ASP.NET
Web Forms pattern for creating Web applications. The ASP.NET MVC framework
is a lightweight, highly testable presentation framework that (as with Web
Forms-based applications) is integrated with existing ASP.NET features, such
as master pages and membership-based authentication.
        </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 collapseAnimationChange = (e) => {
        setCollapseAnimationEffect(collapseInstance.current.value);
    }
    const expandAnimationChange = (e) => {
        setExpandAnimationEffect(expandInstance.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> Expand Animation </label>
          </div>
          <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
            <DropDownListComponent ref={expandInstance} index={0}
change={expandAnimationChange} dataSource={expandAnimation}
popupHeight="250px" placeholder="Select Expand animation" />
          </div>
        </div>
        <div className='row'>
          <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
            <label> Collapse Animation </label>
          </div>
          <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
            <DropDownListComponent ref={collapseInstance} index={1}
change={collapseAnimationChange} dataSource={expandAnimation}
popupHeight="250px" placeholder="Select Collapse animation" />
          </div>
        </div>
        <AccordionComponent ref={acrdnInstance}>
          <AccordionItemsDirective>
            <AccordionItemDirective header='ASP.NET' content={aspContent} />
            <AccordionItemDirective header='ASP.NET MVC' content={mvcContent} />
          </AccordionItemsDirective>
          <AccordionItemDirective header='Javascript' content={jsContent} />
        </AccordionComponent>
      </div>
    );
  }
  const root = ReactDOM.createRoot(document.getElementById('element'));
  root.render(<ReactApp />);

```

INDEX.TSX

```

import * as React from "react";
import { useState, useEffect, useRef } from 'react';
import * as ReactDOM from 'react-dom';
import { Effect } from '@syncfusion/ej2-base';
import { ChangeEventArgs, DropDownListComponent } from '@syncfusion/ej2-react-dropdowns';
import { AccordionComponent, AccordionItemDirective, AccordionItemsDirective } from '@syncfusion/ej2-react-navigations';
const ReactApp = () => {
  const acrdnInstance = useRef(null);
  const expandInstance = useRef(null);
  const collapseInstance = useRef(null);
  const [collapseAnimationEffect, setCollapseAnimationEffect] =
useState('');
  const [expandAnimationEffect, setExpandAnimationEffect] = useState('');

```

```

const expandAnimation = ['SlideDown', 'SlideUp', 'FadeIn', 'FadeOut',
'FadeZoomIn', 'FadeZoomOut', 'ZoomIn', 'ZoomOut', 'None'];
useEffect(() => {
  acrdnInstance.current.animation.collapse = { effect:
collapseAnimationEffect };
}, [collapseAnimationEffect]);
useEffect(() => {
  acrdnInstance.current.animation.expand = { effect: expandAnimationEffect
};
}, [expandAnimationEffect]);
const aspContent = () => {
  return <div>
    Microsoft ASP.NET is a set of technologies in the Microsoft .NET
Framework for building Web applications and XML Web services. ASP.NET pages
execute on the server and generate markup such as HTML, WML, or XML that is
sent to a desktop or mobile browser. ASP.NET pages use a compiled, event-
driven programming model that improves performance and enables the
separation of application logic and user interface.
  </div>;
}
const mvcContent = () => {
  return <div>
    The Model-View-Controller (MVC) architectural pattern separates an
application into three main components: the model, the view, and the
controller. The ASP.NET MVC framework provides an alternative to the ASP.NET
Web Forms pattern for creating Web applications. The ASP.NET MVC framework
is a lightweight, highly testable presentation framework that (as with Web
Forms-based applications) is integrated with existing ASP.NET features, such
as master pages and membership-based authentication.
  </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 collapseAnimationChange = (e: ChangeEventArgs) => {
  setCollapseAnimationEffect(collapseInstance.current.value as Effect);
}
const expandAnimationChange = (e: ChangeEventArgs) => {
  setExpandAnimationEffect(expandInstance.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> Expand Animation </label>
      </div>
      <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">

```

```

        <DropDownListComponent ref={expandInstance} index={0}
change={expandAnimationChange} dataSource={expandAnimation}
popupHeight="250px" placeholder="Select Expand animation" />
    </div>
</div>
<div className='row'>
    <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
        <label> Collapse Animation </label>
    </div>
    <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
        <DropDownListComponent ref={collapseInstance} index={1}
change={collapseAnimationChange} dataSource={expandAnimation}
popupHeight="250px" placeholder="Select Collapse animation" />
    </div>
</div>
<AccordionComponent ref={acrdnInstance}>
    <AccordionItemsDirective>
        <AccordionItemDirective header='ASP.NET' content={aspContent} />
        <AccordionItemDirective header='ASP.NET MVC' content={mvcContent}
/>
        <AccordionItemDirective header='Javascript' content={jsContent} />
    </AccordionItemsDirective>
</AccordionComponent>
</div>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Essential JS 2 Accordion React Custom Animation Sample</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 rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
>
    <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"></sc
ript>
    <script src="systemjs.config.js"></script>
    <style>
        #loader {
            color: #008cff;
            height: 40px;
            left: 45%;
            position: absolute;
            top: 45%;

```

```

        width: 30%;
    }
</style>
</head>
<body>
    <div id='element' style='padding-top: 25px'>
        <div id='loader'>Loading....</div>
    </div>
</body>
</html>

```

To keep single pane open always in React Accordion component

By default, all Accordion panels are collapsible. You can customize the Accordion to keep one panel as expanded state always. This is applicable for **Single** expand mode.

INDEX.JSX

```

import * as React from "react";
import { useState, useRef } from "react";
import * as ReactDOM from "react-dom";
import { AccordionComponent, AccordionItemsDirective, AccordionItemDirective } from '@syncfusion/ej2-react-navigations';
const ReactApp = () => {
    const acrdnInstance = useRef(null);
    const [clickEle, setClickEle] = useState(null);
    const expanding = (e) => {
        if (acrdnInstance.current) {
            let expandCount = acrdnInstance.current.element.querySelectorAll('.e-selected').length;
            let ele = acrdnInstance.current.element.querySelectorAll('.e-selected')[0];
            if (ele) {
                ele = ele.firstChild;
            }
            if (expandCount === 1 && ele === clickEle) {
                e.cancel = true;
            }
        }
    }
    const clicked = (e) => {
        setClickEle((e.originalEvent.target).closest('.e-acrdn-header'));
    }
    const aspContent = () => {
        return (<div>
            Microsoft ASP.NET is a set of technologies in the Microsoft .NET Framework for building Web applications and XML Web services.
        </div>);
    }
    const mvcContent = () => {
        return (<div>
            The Model-View-Controller (MVC) architectural pattern separates an application into three main components: the model, the view, and the controller.
        </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.
    </div>);
}
return (
    <AccordionComponent ref={acrdnInstance} expandMode='Single'
    clicked={clicked} expanding={expanding}>
        <AccordionItemsDirective>
            <AccordionItemDirective expanded={true} header='ASP.NET'
            content={aspContent} />
            <AccordionItemDirective header='ASP.NET MVC' content={mvcContent} />
            <AccordionItemDirective header='JavaScript' content={jsContent} />
        </AccordionItemsDirective>
    </AccordionComponent>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

INDEX.TSX

```

import * as React from "react";
import { useState, useRef } from "react";
import * as ReactDOM from "react-dom";
import { AccordionComponent, AccordionItemsDirective, AccordionItemDirective } from '@syncfusion/ej2-react-navigations';
import { ExpandEventArgs, AccordionClickArgs } from '@syncfusion/ej2-navigations';
const ReactApp = () => {
    const acrdnInstance = useRef<AccordionComponent>(null);
    const [clickEle, setClickEle] = useState(null);
    const expanding = (e: ExpandEventArgs) => {
        if (acrdnInstance.current) {
            let expandCount: number =
acrdnInstance.current.element.querySelectorAll('.e-selected').length;
            let ele: HTMLElement =
acrdnInstance.current.element.querySelectorAll('.e-selected')[0] as
HTMLElement;
            if (ele) {
                ele = ele.firstChild as HTMLElement;
            }
            if (expandCount === 1 && ele === clickEle) {
                e.cancel = true;
            }
        }
    }
    const clicked = (e: AccordionClickArgs) => {
        setClickEle((e.originalEvent.target as Element).closest('.e-acrdn-
header'));
    }
    const aspContent = () => {
        return (<div>

```

```

    Microsoft ASP.NET is a set of technologies in the Microsoft .NET
    Framework for building Web applications and XML Web services.
    </div>);
  }
  const mvcContent = () => {
    return (<div>
      The Model-View-Controller (MVC) architectural pattern separates an
      application into three main components: the model, the view, and the
      controller.
    </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.
    </div>);
  }
  return (
    <AccordionComponent ref={acrdnInstance} expandMode='Single'
    clicked={clicked} expanding={expanding}>
      <AccordionItemsDirective>
        <AccordionItemDirective expanded={true} header='ASP.NET'
        content={aspContent} />
        <AccordionItemDirective header='ASP.NET MVC' content={mvcContent} />
        <AccordionItemDirective header='JavaScript' content={jsContent} />
      </AccordionItemsDirective>
    </AccordionComponent>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Essential JS 2 React Accordion Sample</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></sc
ript>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;

```

```

        position: absolute;
        top: 45%;
        width: 30%;
    }
</style>
</head>
<body>
    <div id='element'>
        <div id='loader'>Loading....</div></div>
</body>
</html>

```

Create wizard using accordion in React Accordion component

Accordion items can be disabled dynamically by passing the index and boolean value with the ['enableItem'](#) method and also dynamically expand the item using [expandItem](#) method.

The below Wizard sample is designed for Online Shopping model. In this, each Accordion item is integrated with required components to fill the details and designed for getting user details and making payment at the end. Each field is provided with validation for all mandatory option to enable/disable to next Accordion. In below sample, accordion items can be disabled dynamically with [enableItem](#) method using [created](#) event.

INDEX.JSX

```

import * as React from "react";
import { useRef } from "react";
import * as ReactDOM from "react-dom";
import { DialogComponent } from '@syncfusion/ej2-react-popups';
import { DatePickerComponent } from '@syncfusion/ej2-react-calendars';
import { NumericTextBoxComponent } from '@syncfusion/ej2-react-inputs';
import { AccordionComponent, AccordionItemsDirective, AccordionItemDirective }
} from '@syncfusion/ej2-react-navigations';
const ReactApp = () => {
    let success = "Your payment successfully processed";
    let email_alert = "Enter valid email address";
    let mobile_alert = "Mobile number length should be 10";
    let card_alert = "Card number length should be 16";
    let cvv_alert = "CVV number length should be 3";
    let dlgTarget = document.body;
    const accordion = useRef(null);
    const alertDlg = useRef(null);
    const cardNo = useRef(null);
    const mobile = useRef(null);
    const expiry = useRef(null);
    const cvv = useRef(null);
    const email = useRef(null);
    const password = useRef(null);
    const name = useRef(null);
    const address = useRef(null);
    const cardHolder = useRef(null);
    const err1 = useRef(null);
    const err2 = useRef(null);
    const err3 = useRef(null);
    const dlgButtons = [
        {

```

```

buttonModel: { content: "Ok", isPrimary: true },
click: () => {
  alertDlg.current.hide();
  if (
    accordion.current.expandedIndices[0] === 2 &&
    alertDlg.current.content === success
  ) {
    accordion.current.enableItem(0, true);
    accordion.current.enableItem(1, false);
    accordion.current.enableItem(2, false);
    accordion.current.expandItem(true, 0);
  }
}
];
const dlgCreated = () => {
  alertDlg.current.hide();
}
const acrdnCreated = () => {
  if (accordion !== undefined) {
    accordion.current.enableItem(1, false);
    accordion.current.enableItem(2, false);
  }
}
const btnClick = (e) => {
  switch (e.target.id) {
    case "Continue_Btn":
      if (email.current.value !== "" && password.current.value !== "") {
        if (checkMail(email.current.value)) {
          email.current.value = password.current.value = "";
          accordion.current.enableItem(1, true);
          accordion.current.enableItem(0, false);
          accordion.current.expandItem(true, 1);
        }
        err1.current.classList.remove("show");
      } else {
        err1.current.classList.add("show");
      }
      break;
    case "Continue_BtnAdr":
      if (
        name.current.value !== "" &&
        address.current.value !== "" &&
        mobile.current.value !== null
      ) {
        if (checkMobile(mobile.current.value)) {
          accordion.current.enableItem(2, true);
          accordion.current.enableItem(1, false);
          accordion.current.expandItem(true, 2);
        }
        err2.current.classList.remove("show");
      } else {
        err2.current.classList.add("show");
      }
      break;
    case "Back_Btn":
      accordion.current.enableItem(1, true);

```

```

        accordion.current.enableItem(2, false);
        accordion.current.expandItem(true, 1);
        break;
    case "Save_Btn":
        if (
            cardNo.current.value != null &&
            cardHolder.current.value != "" &&
            expiry.current.value != null &&
            cvv.current.value != null
        ) {
            if (checkCardNo(cardNo.current.value)) {
                if (checkCVV(cvv.current.value)) {
                    alertDlg.current.content = success;
                    alertDlg.current.show();
                }
            }
            err3.current.classList.remove("show");
        } else {
            err3.current.classList.add("show");
        }
        break;
    }
}

const checkMail = (mail) => {
    if (/^\w+([.-]?\w+)*@\w+([.-]?\w+)*(\.\w{2,3})+$/ .test(mail)) {
        return true;
    } else {
        alertDlg.current.content = email_alert;
        alertDlg.current.show();
        return false;
    }
}

const checkMobile = (mobile) => {
    if (/^\d{10}$/ .test(mobile.toString())) {
        return true;
    } else {
        alertDlg.current.content = mobile_alert;
        alertDlg.current.show();
        return false;
    }
}

const checkCardNo = (cardNo) => {
    if (/^\d{16}$/ .test(cardNo.toString())) {
        return true;
    } else {
        alertDlg.current.content = card_alert;
        alertDlg.current.show();
        return false;
    }
}

const checkCVV = (cvv) => {
    if (/^\d{3}$/ .test(cvv.toString())) {
        return true;
    } else {
        alertDlg.current.content = cvv_alert;
        alertDlg.current.show();
        return false;
    }
}

```

```

    }
  }
  const signForm = () => {
    return (
      <div id="Sign_In_Form">
        <form id="formId">
          <div className="form-group">
            <div className="e-float-input">
              <input type="text" ref={email} name="Email" required />
              <span className="e-float-line" />
              <label className="e-float-text">Email</label>
            </div>
            <div className="e-float-input">
              <input ref={password} type="password" name="Password" required
/>
              <span className="e-float-line" />
              <label className="e-float-text" htmlFor="password">
                Password
              </label>
            </div>
          </div>
          </form>
          <div style={{ textAlign: 'center' }}>
            <button
              className="e-btn"
              id="Continue_Btn"
              onClick={btnClick}
            >
              Continue
            </button>
            <div ref={err1}>* Please fill all fields</div>
          </div>
        </div>
      );
    }
  }
  const addressFill = () => {
    return (
      <div id="Address_Fill">
        <form id="formId_Address">
          <div className="form-group">
            <div className="e-float-input">
              <input type="text" ref={name} name="Name" required />
              <span className="e-float-line" />
              <label className="e-float-text" htmlFor="name">
                Name
              </label>
            </div>
          </div>
          <div className="form-group">
            <div className="e-float-input">
              <input type="text" ref={address} name="Address" required />
              <span className="e-float-line" />
              <label className="e-float-text" htmlFor="address">
                Address
              </label>
            </div>
          </div>
        </form>
      </div>
    );
  }

```

```

        <div className="form-group">
            <NumericTextBoxComponent
                ref={mobile}
                format="0"
                placeholder="Mobile"
                floatLabelType="Auto"
                showSpinButton={false}
            />
        </div>
    </form>
    <div style={{ textAlign: 'center' }}>
        <button
            className="e-btn"
            id="Continue_BtnAdr"
            onClick={btnClick}
        >
            Continue
        </button>
        <div ref={err2}>* Please fill all fields</div>
    </div>
</div>
);
}
const cardFill = () => {
    return (
        <div id="Card_Fill">
            <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
                <div className="form-group">
                    <NumericTextBoxComponent
                        ref={cardNo}
                        format="0"
                        placeholder="Card No"
                        floatLabelType="Auto"
                        showSpinButton={false}
                    />
                </div>
            </div>
            <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
                <div className="form-group">
                    <div className="e-float-input">
                        <input type="text" ref={cardHolder} name="cardHolder" required
                    />

                    <span className="e-float-line" />
                    <label className="e-float-text" htmlFor="cardHolder">
                        CardHolder Name
                    </label>
                </div>
            </div>
            <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
                <DatePickerComponent
                    ref={expiry}
                    width="100%"
                    format="MM/yyyy"
                    placeholder="Expiry Date"
                    floatLabelType="Auto"
                />
            </div>
        </div>
    );
}

```

```

    </div>
    <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
      <div className="form-group">
        <NumericTextBoxComponent
          ref={cvv}
          format="0"
          placeholder="CVV"
          floatLabelType="Auto"
          showSpinButton={false}
        />
      </div>
    </div>
    <div style={{ textAlign: 'center' }}>
      <button
        className="e-btn"
        id="Back_Btn"
        onClick={btnClick}
      >
        Back
      </button>
      <button
        className="e-btn"
        id="Save_Btn"
        onClick={btnClick}
      >
        Save
      </button>
      <div ref={err3}>* Please fill all fields</div>
    </div>
  </div>
);
}
return (
  <div>
    <div className="template_title"> Online Shopping Payment Module</div>
    <DialogComponent
      ref={alertDlg}
      header="Alert"
      width={200}
      isModal={true}
      content=""
      target={dlgTarget}
      buttons={dlgButtons}
      created={dlgCreated}
    />
    <AccordionComponent
      ref={accordion}
      created={acrdnCreated}
    >
      <AccordionItemsDirective>
        <AccordionItemDirective
          expanded={true}
          header="Sign In"
          content={signForm}
        />
        <AccordionItemDirective
          header="Delivery Address"

```



```

        content={addressFill}
      />
      <AccordionItemDirective
        header="Card Details"
        content={cardFill}
      />
    </AccordionItemsDirective>
  </AccordionComponent>
</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 { DialogComponent } from '@syncfusion/ej2-react-popups';
import { DatePickerComponent } from '@syncfusion/ej2-react-calendars';
import { NumericTextBoxComponent } from '@syncfusion/ej2-react-inputs';
import { AccordionComponent, AccordionItemsDirective, AccordionItemDirective } from '@syncfusion/ej2-react-navigations';
const ReactApp = () => {
  let success: string = "Your payment successfully processed";
  let email_alert: string = "Enter valid email address";
  let mobile_alert: string = "Mobile number length should be 10";
  let card_alert: string = "Card number length should be 16";
  let cvv_alert: string = "CVV number length should be 3";
  let dlgTarget: HTMLElement = document.body;
  const accordion = useRef<AccordionComponent>(null);
  const alertDlg = useRef<DialogComponent>(null);
  const cardNo = useRef<NumericTextBoxComponent>(null);
  const mobile = useRef<NumericTextBoxComponent>(null);
  const expiry = useRef<DatePickerComponent>(null);
  const cvv = useRef<NumericTextBoxComponent>(null);
  const email = useRef(null);
  const password = useRef(null);
  const name = useRef(null);
  const address = useRef(null);
  const cardHolder = useRef(null);
  const err1 = useRef(null);
  const err2 = useRef(null);
  const err3 = useRef(null);
  const dlgButtons: Object[] = [
    {
      buttonModel: { content: "Ok", isPrimary: true },
      click: () => {
        alertDlg.current.hide();
        if (
          accordion.current.expandedIndices[0] === 2 &&
          alertDlg.current.content === success
        ) {
          accordion.current.enableItem(0, true);
          accordion.current.enableItem(1, false);

```

```

        accordion.current.enableItem(2, false);
        accordion.current.expandItem(true, 0);
    }
}
];
const dlgCreated = (): void => {
    alertDlg.current.hide();
}
const acrdnCreated = (): void => {
    if (accordion !== undefined) {
        accordion.current.enableItem(1, false);
        accordion.current.enableItem(2, false);
    }
}
const btnClick = (e: any): void => {
    switch (e.target.id) {
        case "Continue_Btn":
            if (email.current.value !== "" && password.current.value !== "") {
                if (checkMail(email.current.value)) {
                    email.current.value = password.current.value = "";
                    accordion.current.enableItem(1, true);
                    accordion.current.enableItem(0, false);
                    accordion.current.expandItem(true, 1);
                }
                err1.current.classList.remove("show");
            } else {
                err1.current.classList.add("show");
            }
            break;
        case "Continue_BtnAdr":
            if (
                name.current.value !== "" &&
                address.current.value !== "" &&
                mobile.current.value != null
            ) {
                if (checkMobile(mobile.current.value)) {
                    accordion.current.enableItem(2, true);
                    accordion.current.enableItem(1, false);
                    accordion.current.expandItem(true, 2);
                }
                err2.current.classList.remove("show");
            } else {
                err2.current.classList.add("show");
            }
            break;
        case "Back_Btn":
            accordion.current.enableItem(1, true);
            accordion.current.enableItem(2, false);
            accordion.current.expandItem(true, 1);
            break;
        case "Save_Btn":
            if (
                cardNo.current.value != null &&
                cardHolder.current.value !== "" &&
                expiry.current.value != null &&
                cvv.current.value != null
            ) {

```

```

    } {
        if (checkCardNo(cardNo.current.value)) {
            if (checkCVV(cvv.current.value)) {
                alertDlg.current.content = success;
                alertDlg.current.show();
            }
        }
        err3.current.classList.remove("show");
    } else {
        err3.current.classList.add("show");
    }
    break;
}
}
const checkMail = (mail: string) => {
    if (/^\w+([.-]?\w+)*@\w+([.-]?\w+)*(\.\w{2,3})+$/ .test(mail)) {
        return true;
    } else {
        alertDlg.current.content = email_alert;
        alertDlg.current.show();
        return false;
    }
}
const checkMobile = (mobile: number) => {
    if (/^\d{10}$/ .test(mobile.toString())) {
        return true;
    } else {
        alertDlg.current.content = mobile_alert;
        alertDlg.current.show();
        return false;
    }
}
const checkCardNo = (cardNo: number) => {
    if (/^\d{16}$/ .test(cardNo.toString())) {
        return true;
    } else {
        alertDlg.current.content = card_alert;
        alertDlg.current.show();
        return false;
    }
}
const checkCVV = (cvv: number) => {
    if (/^\d{3}$/ .test(cvv.toString())) {
        return true;
    } else {
        alertDlg.current.content = cvv_alert;
        alertDlg.current.show();
        return false;
    }
}
const signForm = () => {
    return (
        <div id="Sign_In_Form">
            <form id="formId">
                <div className="form-group">
                    <div className="e-float-input">
                        <input type="text" ref={email} name="Email" required />
                    </div>
                </div>
            </form>
        </div>
    );
}

```

```

        <span className="e-float-line" />
        <label className="e-float-text">Email</label>
    </div>
    <div className="e-float-input">
        <input ref={password} type="password" name="Password" required
/>

        <span className="e-float-line" />
        <label className="e-float-text" htmlFor="password">
            Password
        </label>
    </div>
</div>
</form>
<div style={{ textAlign: 'center' }}>
    <button
        className="e-btn"
        id="Continue_Btn"
        onClick={btnClick}
    >
        Continue
    </button>
    <div ref={err1}>* Please fill all fields</div>
</div>
</div>
);
}
const addressFill = () => {
    return (
        <div id="Address_Fill">
            <form id="formId_Address">
                <div className="form-group">
                    <div className="e-float-input">
                        <input type="text" ref={name} name="Name" required />
                        <span className="e-float-line" />
                        <label className="e-float-text" htmlFor="name">
                            Name
                        </label>
                    </div>
                </div>
                <div className="form-group">
                    <div className="e-float-input">
                        <input type="text" ref={address} name="Address" required />
                        <span className="e-float-line" />
                        <label className="e-float-text" htmlFor="address">
                            Address
                        </label>
                    </div>
                </div>
                <div className="form-group">
                    <NumericTextBoxComponent
                        ref={mobile}
                        format="0"
                        placeholder="Mobile"
                        floatLabelType="Auto"
                        showSpinButton={false}
                    />
                </div>
            </form>
        </div>
    );
}

```

```

        </form>
        <div style={{ textAlign: 'center' }}>
            <button
                className="e-btn"
                id="Continue_BtnAdr"
                onClick={btnClick}
            >
                Continue
            </button>
            <div ref={err2}>* Please fill all fields</div>
        </div>
    </div>
    );
}
const cardFill = () => {
    return (
        <div id="Card_Fill">
            <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
                <div className="form-group">
                    <NumericTextBoxComponent
                        ref={cardNo}
                        format="0"
                        placeholder="Card No"
                        floatLabelType="Auto"
                        showSpinButton={false}
                    />
                </div>
            </div>
            <div>
                <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
                    <div className="form-group">
                        <div className="e-float-input">
                            <input type="text" ref={cardHolder} name="cardHolder" required
                                />
                            <span className="e-float-line" />
                            <label className="e-float-text" htmlFor="cardHolder">
                                CardHolder Name
                            </label>
                        </div>
                    </div>
                </div>
            </div>
            <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
                <DatePickerComponent
                    ref={expiry}
                    width="100%"
                    format="MM/yyyy"
                    placeholder="Expiry Date"
                    floatLabelType="Auto"
                />
            </div>
            <div className="col-xs-6 col-sm-6 col-lg-6 col-md-6">
                <div className="form-group">
                    <NumericTextBoxComponent
                        ref={cvv}
                        format="0"
                        placeholder="CVV"
                        floatLabelType="Auto"
                        showSpinButton={false}
                    />
                </div>
            </div>
        </div>
    );
}
    
```

```

        />
      </div>
    </div>
    <div style={{ textAlign: 'center' }}>
      <button
        className="e-btn"
        id="Back_Btn"
        onClick={btnClick}
      >
        Back
      </button>
      <button
        className="e-btn"
        id="Save_Btn"
        onClick={btnClick}
      >
        Save
      </button>
      <div ref={err3}>* Please fill all fields</div>
    </div>
  </div>
);
}
return (
  <div>
    <div className="template_title"> Online Shopping Payment Module</div>
    <DialogComponent
      ref={alertDlg}
      header="Alert"
      width={200}
      isModal={true}
      content=""
      target={dlgTarget}
      buttons={dlgButtons}
      created={dlgCreated}
    />
    <AccordionComponent
      ref={accordion}
      created={acrdnCreated}
    >
      <AccordionItemsDirective>
        <AccordionItemDirective
          expanded={true}
          header="Sign In"
          content={signForm}
        />
        <AccordionItemDirective
          header="Delivery Address"
          content={addressFill}
        />
        <AccordionItemDirective
          header="Card Details"
          content={cardFill}
        />
      </AccordionItemsDirective>
    </AccordionComponent>
  </div>

```

```

    );
  }
  const root = ReactDOM.createRoot(document.getElementById('element'));
  root.render(<ReactApp />);

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Essential JS 2 React Accordion Sample</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://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
rel="stylesheet">
  <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"></sc
ript>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
  </style>
</head>
<body id="container">
  <div id='element'>
    <div id='loader'>Loading....</div>
  </div>
</body>
</html>

```

INDEX.CSS

```

#container {
  max-width: 450px;
  margin: 0 auto;
}
#loader {
  color: #008cff;
  height: 40px;
  left: 45%;
  position: absolute;
  top: 45%;

```

```

        width: 30%;
    }
    .accordion-control-section {
        margin: 0 10% 0 10%;
    }
    @media screen and (max-width: 768px) {
        .accordion-control-section {
            margin: 0;
        }
    }
    #err1, #err2, #err3 {
        display: none;
        color: red;
        margin-top: 10px;
        font-weight: 500;
    }
    #err1.show,
    #err2.show,
    #err3.show {
        display: block;
    }
    .e-dialog {
        max-height: 300px !important; /* csslint allow: important */
    }
    .template_title {
        text-align: center;
        padding: 10px 0;
        margin: 20px 0;
        text-overflow: ellipsis;
        font-weight: bold;
        font-size: 16px;
    }
}

```

Load accordion with data source in React Accordion component

You can bind any data object to Accordion items, by mapping it to [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 Accordion.

INDEX.JSX

```

import * as React from "react";
import { useEffect, useState } from "react";
import * as ReactDOM from "react-dom";
import { DataManager, Query, ODataV4Adaptor } from '@syncfusion/ej2-data';
import { AccordionComponent } from '@syncfusion/ej2-react-navigations';
const SERVICE_URI =
'https://services.odata.org/V4/Northwind/Northwind.svc/Employees';
const ReactApp = () => {
    const [acrdnInstance, setAcrdnInstance] = useState(null);
    useEffect(() => {
        if (acrdnInstance) {
            new DataManager({ url: SERVICE_URI, adaptor: new ODataV4Adaptor() })
                .executeQuery(new Query().range(1, 4)).then((e) => {
                    let itemsData = [];
                    let result = e.result;

```



```

        let mapping = { header: 'FirstName', content: 'Notes' };
        for (let i = 0; i < result.length; i++) {
            itemsData.push({ header: result[i][mapping.header], content:
result[i][mapping.content] });
        }
        acrdnInstance.items = itemsData;
        acrdnInstance.refresh();
    });
}
}, [acrdnInstance]);
const accordCreated = (acrdn) => {
    setAcrdnInstance(acrdn);
};
return (
    <AccordionComponent ref={ (acrdn) => { setAcrdnInstance(acrdn) }}
created={accordCreated}>
    </AccordionComponent>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

INDEX.TSX

```

import * as React from "react";
import { useEffect, useState } from "react";
import * as ReactDOM from "react-dom";
import { DataManager, Query, ODataV4Adaptor, ReturnOption } from
'@syncfusion/ej2-data';
import { AccordionComponent } from '@syncfusion/ej2-react-navigations';
const SERVICE_URI: string =
'https://services.odata.org/V4/Northwind/Northwind.svc/Employees';
const ReactApp = () => {
    const [acrdnInstance, setAcrdnInstance] = useState(null);
    useEffect(() => {
        if (acrdnInstance) {
            new DataManager({ url: SERVICE_URI, adaptor: new ODataV4Adaptor() })
                .executeQuery(new Query().range(1, 4)).then((e: ReturnOption) => {
                    let itemsData: Object[] = [];
                    let result: any = e.result;
                    let mapping = { header: 'FirstName', content: 'Notes' };
                    for (let i: number = 0; i < result.length; i++) {
                        itemsData.push({ header: result[i][mapping.header], content:
result[i][mapping.content] });
                    }
                    acrdnInstance.items = itemsData;
                    acrdnInstance.refresh();
                });
        }
    }, [acrdnInstance]);
    const accordCreated = (acrdn: AccordionComponent): void => {
        setAcrdnInstance(acrdn);
    };
    return (
        <AccordionComponent ref={ (acrdn) => { setAcrdnInstance(acrdn) }}
created={accordCreated}>

```

```

    </AccordionComponent>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Essential JS 2 React Accordion Sample</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></sc
ript>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
  </style>
</head>
<body>
  <div id='element'>
    <div id='loader'>Loading....</div></div>
</body>
</html>

```

Load accordion items dynamically in React Accordion component

Accordion items can be added dynamically by passing the item and index value with the [addItem](#) method.

In the following demo, you can add the accordion content by expanding any accordion header content using [expanded](#) event.

- Data is fetched from the data source.
- The data is formatted as a JSON object with [header](#) and [content](#) fields.
- Here last index is calculated to append the new accordion at the end.

INDEX.JSX

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

```
import { useRef } from 'react';
import * as ReactDOM from 'react-dom';
import { AccordionComponent, AccordionItemDirective, AccordionItemsDirective
} from '@syncfusion/ej2-react-navigations';
let dbFlag= 0;
let dynamicAcrdnCount = 2;
const ReactApp = () => {
  let accordion = [
    {
      header: ' ASP.NET Razor ',
      content: ' Razor is an ASP.NET programming syntax used to create
dynamic web pages with the C# or Visual Basic .NET programming languages.
Razor was in development in June 2010 and was released for Microsoft Visual
Studio 2010 in January 2011. Razor is a simple-syntax view engine and was
released as part of MVC 3 and the WebMatrix tool set. '
    },
    {
      header: ' EmberJS ',
      content: ' EmberJS is an open-source JavaScript web framework, based
on the Model-view-viewmodel (MVVM) pattern. It allows developers to create
scalable single-page web applications by incorporating common idioms and
best practices into the framework. '
    },
    {
      header: ' Hypertext Markup Language ',
      content: ' Hypertext Markup Language (HTML) is the standard markup
language for creating web pages and web applications. With Cascading Style
Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies
for the World Wide Web. '
    },
    {
      header: ' HTML5 ',
      content: ' HTML5 is a markup language used for structuring and
presenting content on the World Wide Web. It is the fifth and current major
version of the HTML standard. '
    },
    {
      header: ' JavaScript ',
      content: ' JavaScript (JS) is an interpreted computer programming
language. '
    },
    {
      header: ' JSP ',
      content: ' JavaServer Pages (JSP) is a technology that helps software
developers create dynamically generated web pages based on HTML, XML, or
other document types. Released in 1999 by Sun Microsystems, JSP is similar
to PHP and ASP, but it uses the Java programming language. '
    },
    {
      header: ' Perl ',
      content: ' Perl is a family of high-level, general-purpose,
interpreted, dynamic programming languages. Perl was originally developed
by Larry Wall in 1987 as a general-purpose Unix scripting language to make
report processing easier. '
    },
    {
      header: ' PHP ',
```

```

        content: ' PHP is a server-side scripting language designed for web
development but also used as a general-purpose programming language.
Originally created by Rasmus Lerdorf in 1994, the PHP reference
implementation is now produced by The PHP Group. PHP originally stood for
Personal Home Page, but it now stands for the recursive acronym PHP:
Hypertext Preprocessor '
    },
    {
        header: ' Ruby ',
        content: ' Ruby is a dynamic, reflective, object-oriented, general-
purpose programming language. It supports multiple programming paradigms,
including functional, object-oriented, and imperative. It also has a dynamic
type system and automatic memory management. '
    },
    {
        header: ' Typescript ',
        content: ' TypeScript is an open-source programming language developed
and maintained by Microsoft. It is a strict syntactical superset of
JavaScript, and adds optional static typing to the language. '
    },
];
const acrdnInstance = useRef(null);
const expanded = (e) => {
    const Elementindex = document.getElementsByClassName("e-expand-state e-
selected e-active")[0];
    if ([...e.element.parentElement.children].indexOf(e.element) ===
[...e.element.parentElement.children].indexOf(Elementindex)) {
        const array = accordion;
        for (let i = 0; i < dynamicAcrdnCount; i++) {
            if (dbFlag === array.length) {
                return;
            }
            acrdnInstance.current.addItem(array[dbFlag],
acrdnInstance.current.items.length);
            ++dbFlag;
        }
    }
}
const aspNetContent = () => {
    return (<div>
        Microsoft ASP.NET is a set of technologies in the Microsoft .NET
Framework for building Web applications and XML Web services.
    </div>);
}
const aspCoreContent = () => {
    return (<div>
        ASP.NET Core is a free and open-source web framework, and the next
generation of ASP.NET, developed by Microsoft and the community. It is a
modular framework that runs on both the full .NET Framework, on Windows, and
the cross-platform .NET Core.
    </div>);
}
const aspMvcContent = () => {
    return (<div>
        The Model-View-Controller (MVC) architectural pattern separates an
application into three main components: the model, the view, and the
controller.
    </div>);
}

```

```

    </div>);
  }
  return (
    <div className='control-pane'>
      <div className='control-section accordion-control-section'>
        <div className='control Accordion-sample'>
          /* Render the Accoridon Component */
          <AccordionComponent expanded={expanded} ref={acrdnInstance}>
            <AccordionItemsDirective>
              <AccordionItemDirective header='ASP.NET'
content={aspNetContent} />
              <AccordionItemDirective header='ASP.NET Core'
content={aspCoreContent} />
              <AccordionItemDirective header='ASP.NET MVC'
content={aspMvcContent} />
            </AccordionItemsDirective>
          </AccordionComponent>
        </div>
      </div>
    </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 { AccordionComponent, AccordionItemDirective,
AccordionItemsDirective, ExpandEventArgs } from '@syncfusion/ej2-react-
navigations';
// @ts-ignore
let dbFlag: number = 0;
let dynamicAcrdnCount: number = 2;
const ReactApp = () => {
  let accordion: Object[] = [
    {
      header: ' ASP.NET Razor ',
      content: ' Razor is an ASP.NET programming syntax used to create
dynamic web pages with the C# or Visual Basic .NET programming languages.
Razor was in development in June 2010 and was released for Microsoft Visual
Studio 2010 in January 2011. Razor is a simple-syntax view engine and was
released as part of MVC 3 and the WebMatrix tool set. '
    },
    {
      header: ' EmberJS ',
      content: ' EmberJS is an open-source JavaScript web framework, based
on the Model-view-viewmodel (MVVM) pattern. It allows developers to create
scalable single-page web applications by incorporating common idioms and
best practices into the framework. '
    },
    {
      header: ' Hypertext Markup Language ',

```

```

        content: ' Hypertext Markup Language (HTML) is the standard markup
language for creating web pages and web applications. With Cascading Style
Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies
for the World Wide Web. '
    },
    {
        header: ' HTML5 ',
        content: ' HTML5 is a markup language used for structuring and
presenting content on the World Wide Web. It is the fifth and current major
version of the HTML standard. '
    },
    {
        header: ' JavaScript ',
        content: ' JavaScript (JS) is an interpreted computer programming
language. '
    },
    {
        header: ' JSP ',
        content: ' JavaServer Pages (JSP) is a technology that helps software
developers create dynamically generated web pages based on HTML, XML, or
other document types. Released in 1999 by Sun Microsystems, JSP is similar
to PHP and ASP, but it uses the Java programming language. '
    },
    {
        header: ' Perl ',
        content: ' Perl is a family of high-level, general-purpose,
interpreted, dynamic programming languages. Perl was originally developed
by Larry Wall in 1987 as a general-purpose Unix scripting language to make
report processing easier. '
    },
    {
        header: ' PHP ',
        content: ' PHP is a server-side scripting language designed for web
development but also used as a general-purpose programming language.
Originally created by Rasmus Lerdorf in 1994, the PHP reference
implementation is now produced by The PHP Group. PHP originally stood for
Personal Home Page, but it now stands for the recursive acronym PHP:
Hypertext Preprocessor '
    },
    {
        header: ' Ruby ',
        content: ' Ruby is a dynamic, reflective, object-oriented, general-
purpose programming language. It supports multiple programming paradigms,
including functional, object-oriented, and imperative. It also has a dynamic
type system and automatic memory management. '
    },
    {
        header: ' Typescript ',
        content: ' TypeScript is an open-source programming language developed
and maintained by Microsoft. It is a strict syntactical superset of
JavaScript, and adds optional static typing to the language. '
    },
    ],
    ];
    const acrdnInstance = useRef<AccordionComponent>(null);
    const expanded = (e: ExpandEventArgs): void => {
        const Elementindex = document.getElementsByClassName("e-expand-state e-
selected e-active")[0];

```

```

    if ([...e.element.parentElement.children].indexOf(e.element) ===
    [...e.element.parentElement.children].indexOf(Elementindex)) {
        const array: object[] = accordion;
        for (let i = 0; i < dynamicAcrdnCount; i++) {
            if (dbFlag === array.length) {
                return;
            }
            acrdnInstance.current.addItem(array[dbFlag],
            acrdnInstance.current.items.length);
            ++dbFlag;
        }
    }
}

const aspNetContent = () => {
    return (<div>
        Microsoft ASP.NET is a set of technologies in the Microsoft .NET
        Framework for building Web applications and XML Web services.
    </div>);
}

const aspCoreContent = () => {
    return (<div>
        ASP.NET Core is a free and open-source web framework, and the next
        generation of ASP.NET, developed by Microsoft and the community. It is a
        modular framework that runs on both the full .NET Framework, on Windows, and
        the cross-platform .NET Core.
    </div>);
}

const aspMvcContent = () => {
    return (<div>
        The Model-View-Controller (MVC) architectural pattern separates an
        application into three main components: the model, the view, and the
        controller.
    </div>);
}

return (
    <div className='control-pane'>
        <div className='control-section accordion-control-section'>
            <div className='control Accordion-sample'>
                { /* Render the Accoridon Component */ }
                <AccordionComponent expanded={expanded} ref={acrdnInstance}>
                    <AccordionItemsDirective>
                        <AccordionItemDirective header='ASP.NET '
content={aspNetContent} />
                        <AccordionItemDirective header='ASP.NET Core '
content={aspCoreContent} />
                        <AccordionItemDirective header='ASP.NET MVC '
content={aspMvcContent} />
                    </AccordionItemsDirective>
                </AccordionComponent>
            </div>
        </div>
    </div>
);
}

const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

INDEX.HTML

```

<!DOCTYPE html>
<html>
<head>
  <title>Syncfusion React Sample</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" />
  <link
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
rel="stylesheet" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></sc
ript>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
    .accordion-control-section {
      margin: 0 10% 0 10%;
    }

    @media screen and (max-width: 768px) {
      .accordion-control-section {
        margin: 0;
      }
    }
  </style>
</head>
<body>
  <div id='element'>
    <div id='loader'>Loading....</div>
  </div>
</body>
</html>

```

Customize expand collapse actions in React Accordion component

Accordion component supports customizing the expand or collapse animation action behavior. You can manually change the expand animation action performed after the collapse animation operation performed on already expand pane when the expand icons are clicked.

By default, the Accordion component pane is expanded or collapsed, when click the expand or collapse icon. It is not affected on already expand pane.

The following sample demonstrates, how to expand the collapsed Accordion item after collapse animation performed on the expanded Accordion item using [created](#), [expanding](#), and [expanded](#) event. In the Expanding event, get the previously expanded item index and prevent the expanding behavior using `args.cancel` option. Expand the Accordion item dynamically based on specifying the `index` value using the [expandItem](#) public method and [expanded](#) event.

INDEX.JSX

```
import * as React from "react";
import { useRef } from "react";
import * as ReactDOM from "react-dom";
import { AccordionComponent, AccordionItemDirective, AccordionItemsDirective } from '@syncfusion/ej2-react-navigations';
const ReactApp = () => {
  let initialLoad = true;
  let isCollapsed = false;
  let expandIndex;
  let acrdnInstance = useRef(null)
  const expanded = (e) => {
    if (!e.isExpanded && !initialLoad && isCollapsed) {
      acrdnInstance.current.expandItem(true, expandIndex);
      isCollapsed = false;
    }
  }
  const expanding = (e) => {
    if (e.isExpanded && !initialLoad && !isCollapsed) {
      e.cancel = true;
      expandIndex = acrdnInstance.current.items.indexOf(e.item);
      isCollapsed = true;
    }
  }
  const created = () => {
    initialLoad = false;
  }
  const aspContent = () => {
    return (<div>
      Microsoft ASP.NET is a set of technologies in the Microsoft .NET Framework for building Web applications and XML Web services. ASP.NET pages execute on the server and generate markup such as HTML, WML, or XML that is sent to a desktop or mobile browser. ASP.NET pages use a compiled, event-driven programming model that improves performance and enables the separation of application logic and user interface.
    </div>);
  }
  const mvcContent = () => {
    return (<div>
      The Model-View-Controller (MVC) architectural pattern separates an application into three main components: the model, the view, and the controller. The ASP.NET MVC framework provides an alternative to the ASP.NET Web Forms pattern for creating Web applications. The ASP.NET MVC framework is a lightweight, highly testable presentation framework that (as with Web Forms-based applications) is integrated with existing ASP.NET features, such as master pages and membership-based authentication.
    </div>);
  }
}
```

```

    </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 (<AccordionComponent expanded={expanded} expanding={expanding}
    created={created} expandMode='Single' ref={acrdnInstance}>
    <AccordionItemsDirective>
      <AccordionItemDirective header='ASP.NET' content={aspContent} />
      <AccordionItemDirective header='ASP.NET MVC' content={mvcContent} />
      <AccordionItemDirective header='Javascript' content={jsContent} />
    </AccordionItemsDirective>
  </AccordionComponent>);
}
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 { AccordionComponent, AccordionItemDirective, AccordionItemsDirective } from '@syncfusion/ej2-react-navigations';
const ReactApp = () => {
  let initialLoad = true;
  let isCollapsed = false;
  let expandIndex: number;
  let acrdnInstance = useRef(null)
  const expanded = (e: any): void => {
    if (!e.isExpanded && !initialLoad && isCollapsed) {
      acrdnInstance.current.expandItem(true, expandIndex);
      isCollapsed = false;
    }
  }
  const expanding = (e: any): void => {
    if (e.isExpanded && !initialLoad && !isCollapsed) {
      e.cancel = true;
      expandIndex = acrdnInstance.current.items.indexOf(e.item);
      isCollapsed = true;
    }
  }
  const created = (): void => {
    initialLoad = false;
  }
  const aspContent = () => {
    return (<div>
      Microsoft ASP.NET is a set of technologies in the Microsoft .NET
      Framework for building Web applications and XML Web services. ASP.NET pages

```

```

execute on the server and generate markup such as HTML, WML, or XML that is
sent to a desktop or mobile browser. ASP.NET pages use a compiled, event-
driven programming model that improves performance and enables the
separation of application logic and user interface.
    </div>);
}
const mvcContent = () => {
    return <div>
        The Model-View-Controller (MVC) architectural pattern separates an
        application into three main components: the model, the view, and the
        controller. The ASP.NET MVC framework provides an alternative to the ASP.NET
        Web Forms pattern for creating Web applications. The ASP.NET MVC framework
        is a lightweight, highly testable presentation framework that (as with Web
        Forms-based applications) is integrated with existing ASP.NET features, such
        as master pages and membership-based authentication.
    </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 <AccordionComponent expanded={expanded} expanding={expanding}
created={created} expandMode='Single' ref={acrdnInstance}>
    <AccordionItemsDirective>
        <AccordionItemDirective header='ASP.NET' content={aspContent} />
        <AccordionItemDirective header='ASP.NET MVC' content={mvcContent} />
        <AccordionItemDirective header='Javascript' content={jsContent} />
    </AccordionItemsDirective>
</AccordionComponent>);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<ReactApp />);

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Essential JS 2 React Accordion Sample</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" />
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></sc
ript>
    <script src="systemjs.config.js"></script>

```

```
<style>
  #loader {
    color: #008cff;
    height: 40px;
    left: 45%;
    position: absolute;
    top: 45%;
    width: 30%;
  }
</style>
</head>
<body>
  <div id='element'>
    <div id='loader'>Loading....</div></div>
</body>
</html>
```

Render other components in accordion using template in React Accordion component

You can render other components inside Accordion using React **template**. Through this, we can add content as other components directly with all functionalities to our Accordion. Follow the below guidelines for using the other components as template in Accordion.

- 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.

INDEX.JSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { AccordionComponent, AccordionItemsDirective, AccordionItemDirective } from '@syncfusion/ej2-react-navigations';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { TreeViewComponent } from '@syncfusion/ej2-react-navigations';
import { DatePickerComponent, CalendarComponent } from '@syncfusion/ej2-react-calendars';
const App = () => {
  const continents = [
    {
      code: 'AF', name: 'Africa', countries: [
        { code: 'NGA', name: 'Nigeria' },
        { code: 'EGY', name: 'Egypt' },
        { code: 'ZAF', name: 'South Africa' }
      ]
    },
    {
      code: 'AS', name: 'Asia', expanded: true, countries: [
        { code: 'CHN', name: 'China' },
        { code: 'IND', name: 'India', selected: true },
        { code: 'JPN', name: 'Japan' }
      ]
    },
    {
      code: 'EU', name: 'Europe', countries: [
```

```

        { code: 'DNK', name: 'Denmark' },
        { code: 'FIN', name: 'Finland' },
        { code: 'AUT', name: 'Austria' }
    ]
},
{
    code: 'NA', name: 'North America', countries: [
        { code: 'USA', name: 'United States of America' },
        { code: 'CUB', name: 'Cuba' },
        { code: 'MEX', name: 'Mexico' }
    ]
},
{
    code: 'SA', name: 'South America', countries: [
        { code: 'BRA', name: 'Brazil' },
        { code: 'COL', name: 'Colombia' },
        { code: 'ARG', name: 'Argentina' }
    ]
},
{
    code: 'OC', name: 'Oceania', countries: [
        { code: 'AUS', name: 'Australia' },
        { code: 'NZL', name: 'New Zealand' },
        { code: 'WSM', name: 'Samoa' }
    ]
},
{
    code: 'AN', name: 'Antarctica', countries: [
        { code: 'BVT', name: 'Bouvet Island' },
        { code: 'ATF', name: 'French Southern Lands' }
    ]
},
];
const field = { dataSource: continents, id: 'code', text: 'name', child:
'countries' }
const buttonTemplate = () => {
    return (<ButtonComponent>Click me</ButtonComponent>);
}
const datePickerTemplate = () => {
    return (<DatePickerComponent></DatePickerComponent>);
}
const calendarTemplate = () => {
    return (<CalendarComponent ></CalendarComponent>);
}
const treeViewTemplate = () => {
    return (<TreeViewComponent fields={field}></TreeViewComponent>);
}
return (
    <div id='container'>
        <AccordionComponent>
            <AccordionItemsDirective>
                <AccordionItemDirective header='Button' content={buttonTemplate}
/>
                <AccordionItemDirective header='DatePicker'
content={datePickerTemplate} />
                <AccordionItemDirective header='Calendar'
content={calendarTemplate} />
            </AccordionItemsDirective>
        </AccordionComponent>
    </div>
);

```

```

        <AccordionItemDirective header='Treeview'
content={treeViewTemplate} />
    </AccordionItemsDirective>
</AccordionComponent>
</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 { AccordionComponent, AccordionItemsDirective, AccordionItemDirective } from '@syncfusion/ej2-react-navigations';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { TreeViewComponent } from '@syncfusion/ej2-react-navigations';
import { DatePickerComponent, CalendarComponent } from '@syncfusion/ej2-react-calendars';
const App = () => {
    const continents: { [key: string]: Object }[] = [
        {
            code: 'AF', name: 'Africa', countries: [
                { code: 'NGA', name: 'Nigeria' },
                { code: 'EGY', name: 'Egypt' },
                { code: 'ZAF', name: 'South Africa' }
            ]
        },
        {
            code: 'AS', name: 'Asia', expanded: true, countries: [
                { code: 'CHN', name: 'China' },
                { code: 'IND', name: 'India', selected: true },
                { code: 'JPN', name: 'Japan' }
            ]
        },
        {
            code: 'EU', name: 'Europe', countries: [
                { code: 'DNK', name: 'Denmark' },
                { code: 'FIN', name: 'Finland' },
                { code: 'AUT', name: 'Austria' }
            ]
        },
        {
            code: 'NA', name: 'North America', countries: [
                { code: 'USA', name: 'United States of America' },
                { code: 'CUB', name: 'Cuba' },
                { code: 'MEX', name: 'Mexico' }
            ]
        },
        {
            code: 'SA', name: 'South America', countries: [
                { code: 'BRA', name: 'Brazil' },
                { code: 'COL', name: 'Colombia' },
                { code: 'ARG', name: 'Argentina' }
            ]
        }
    ]
}

```

```

    },
    {
      code: 'OC', name: 'Oceania', countries: [
        { code: 'AUS', name: 'Australia' },
        { code: 'NZL', name: 'New Zealand' },
        { code: 'WSM', name: 'Samoa' }
      ]
    },
    {
      code: 'AN', name: 'Antarctica', countries: [
        { code: 'BVT', name: 'Bouvet Island' },
        { code: 'ATF', name: 'French Southern Lands' }
      ]
    }
  ],
};

const field: object = { dataSource: continents, id: 'code', text: 'name',
child: 'countries' }
const buttonTemplate = (): JSX.Element => {
  return (<ButtonComponent>Click me</ButtonComponent>);
}
const datePickerTemplate = (): JSX.Element => {
  return (<DatePickerComponent></DatePickerComponent>);
}
const calenderTemplate = (): JSX.Element => {
  return (<CalendarComponent ></CalendarComponent>);
}
const treeViewTemplate4 = (): JSX.Element => {
  return (<TreeViewComponent fields={field}></TreeViewComponent>);
}
return (
  <div id='container'>
    <AccordionComponent>
      <AccordionItemsDirective>
        <AccordionItemDirective header='Button' content={buttonTemplate}
        />
        <AccordionItemDirective header='DatePicker'
content={datePickerTemplate} />
        <AccordionItemDirective header='Calendar'
content={calenderTemplate} />
        <AccordionItemDirective header='Treeview'
content={treeViewTemplate4} />
      </AccordionItemsDirective>
    </AccordionComponent>
  </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"></sc
ript>
    <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 accordion in React Accordion component

You can prevent the React JSX component reload issue on state changes by rendering the React JSX component as children of Accordion component. In the below example, we have prevented the re-rendering of textbox component on state changes in Accordion 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 { AccordionComponent } from '@syncfusion/ej2-react-navigations';
const App = () => {
    const [name, setName] = useState('');
    return (<div>
        <div className="control_wrapper" id="control_wrapper">
            <AccordionComponent>
                <div>
                    <div>
                        <div> TextBox </div>
                    </div>
                    <div>
                        <div>
                            <TextBoxComponent placeholder="Enter Name" value={name}
input={ (e) => setName(e.value) }></TextBoxComponent>
                            <div>My name is {name}</div>
                        </div>
                    </div>
                </AccordionComponent>
            </div>
        </div>);

```



```
}
const root = ReactDOM.createRoot(document.getElementById('container'));
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 { AccordionComponent } from '@syncfusion/ej2-react-navigations';
const App = () => {
  const [name, setName] = useState('');
  return (
    <div>
      <div className="control_wrapper" id="control_wrapper">
        <AccordionComponent>
          <div>
            <div>
              <div> TextBox </div>
            </div>
            <div>
              <div>
                <TextBoxComponent placeholder="Enter Name" value={name}
input={ (e) => setName(e.value)} ></TextBoxComponent>
                <div>My name is {name}</div>
              </div>
            </div>
          </AccordionComponent>
        </div>
      </div>
    );
  }
const root = ReactDOM.createRoot(document.getElementById('container'));
root.render(<App />);
```

Ej1 api migration in React Accordion component

This article describes the API migration process of Accordion 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
 <EJ.Accordion id="Accordion" allowKeyboardNavigation={false}></EJ.Accordion> | Not Applicable |

| Localization | Not Applicable | **Property** : locale
 <AccordionComponent id="Accordion" locale="fr-BE" ></AccordionComponent> |

| Right to left | **Property:** enableRTL
 <EJ.Accordion id="Accordion" enableRTL={true}></EJ.Accordion> | **Property:** enableRTL
 <AccordionComponent id="Accordion" enableRTL={true}></AccordionComponent> |

AjaxSettings

<!-- markdownlint-disable MD033 -->

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

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

| Default | **Property** : ajaxSettings
 <EJ.Accordion id="accordion" ajaxSettings.type="GET"></EJ.Accordion> | Not Applicable |

| Asynchronous | **Property** : ajaxSettings.async
 <EJ.Accordion id="accordion" ajaxSettings.async={true}></EJ.Accordion> | Not Applicable |

| Browser Cache | **Property:** ajaxSettings.cache
 <EJ.Accordion id="accordion" ajaxSettings.cache={false}></EJ.Accordion> | Not Applicable |

| Request type | **Property** : ajaxSettings.contentType
 <EJ.Accordion id="accordion" ajaxSettings.contentType="html "></EJ.Accordion> | Not Applicable |

| Data | **Property** : ajaxSettings.data
 <EJ.Accordion id="accordion" ajaxSettings.data={}></EJ.Accordion> | Not Applicable |

| Response type | **Property** : ajaxSettings.dataType
 <EJ.Accordion id="accordion" ajaxSettings.dataType="html"></EJ.Accordion> | Not Applicable |

| HTTP request type | **Property:** ajaxSettings.type
 <EJ.Accordion id="accordion" ajaxSettings.type="GET"></EJ.Accordion> | Not Applicable |

| AjaxBeforeLoad | **Event:** ajaxBeforeLoad
 <EJ.Accordion id='accordion' ajaxBeforeLoad={this.onajaxBeforeLoad}></EJ.Accordion>
 onajaxBeforeLoad (event){ } | Not Applicable |

| AjaxError | **Event:** AjaxError
 <EJ.Accordion id='accordion' ajaxError={this.onajaxError}></EJ.Accordion>
 onajaxError (event){ } | Not Applicable |

| AjaxLoad | **Event:** ajaxLoad
 <EJ.Accordion id='accordion' ajaxLoad={this.onajaxLoad}></EJ.Accordion>
 onajaxLoad (event){ } | Not Applicable |

| AjaxSuccess | **Event:** ajaxSuccess
 <EJ.Accordion id='accordion' ajaxSuccess={this.onajaxSuccess}></EJ.Accordion>
 onajaxSuccess (event){ } | Not Applicable |

Animation

<!-- markdownlint-disable MD033 -->

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

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

| Default | Not Applicable | **Property** : animation
 <AccordionComponent id='accordion' animation={this.animation}></AccordionComponent>

constructor(props: {}) {

this.animation = {collapse: { effect: 'FlipXDownIn', duration: 600, easing: 'ease' },expand: { effect: 'FlipXUpIn', duration: 600, easing: 'ease' }}; } |

| EnableAnimation | **Property** :animation
 <EJ.Accordion id="Accordion"
enableAnimation={false}></EJ.Accordion>
 | Not Applicable|

| Expand animation | Not Applicable | **Property** :animation.expand

<AccordionComponent

id='accordion'animation={this.animation}></AccordionComponent>

constructor(props:
{}) {
 this.animation = { expand: { effect: 'FlipXUpIn', duration: 600, easing: 'ease' }};} |

| Collapse animation | Not Applicable | **Property** :animation.collapse

<AccordionComponent

id='accordion'animation={this.animation}></AccordionComponent>

constructor(props:
{}) {
 this.animation = { collapse: { effect: 'SlideLeft', duration: 600, easing: 'ease-in' }};} |

| Duration [expand / collapse] | Not Applicable | **Property** :animation.collapse.duration

<AccordionComponent

id='accordion'animation={this.animation}></AccordionComponent>

constructor(props:
{}) {
 this.animation = { expand: { duration: 600 }};} |

| Easing [expand / collapse] | Not Applicable | **Property** :animation.collapse.easing

<AccordionComponent

id='accordion'animation={this.animation}></AccordionComponent>

constructor(props:
{}) {
 this.animation = { expand: { easing: 'ease-in' }};} |

| Effect [expand / collapse] | Not Applicable | **Property** :animation.collapse.effect

<AccordionComponent

id='accordion'animation={this.animation}></AccordionComponent>

constructor(props:
{}) {
 this.animation = { expand: { effect: 'SlideLeft' }};} |

Items

<!-- markdownlint-disable MD033 -->

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

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

| Default | Not Applicable | **Property** : items
 <AccordionComponent id="Accordion"
items={this.items}> </AccordionComponent> |

| Content | Not Applicable | **Property** : items[0].content
 <AccordionComponent
id="Accordion"items={this.items}> </AccordionComponent>

constructor(props: {})
{
 this.items = {content: 'Contents'}}; } |

| Custom class | Not Applicable | **Property** : items[0].cssClass
 <AccordionComponent
id="Accordion"items={this.items}> </AccordionComponent>

 constructor(props: {})
{
 this.items = { cssClass: 'customClass'}}; } |

| Header | Not Applicable | **Property** : items[0].Header
 <AccordionComponent
id="Accordion"items={this.items}> </AccordionComponent>

 constructor(props: {})
{
 this.items = { header: 'Header1'}}; } |

```
| HeaderSize | Property : items[0].headerSize <br/> <EJ.Accordion id="Accordion"
headerSize="50px" ></EJ.Accordion> | <b>Not Applicable</b> |

| Icon class | <b>Not Applicable</b> | Property : items[0].iconCss <br/> <AccordionComponent
id="Accordion"items={this.items}> </AccordionComponent><br/> <br/>constructor(props: {})
{<br/> this.items = { iconCss: 'e-icons'}};|

| IsExpand | <b>Not Applicable</b> | Property : items[0].expanded <br/> <AccordionComponent
id="Accordion"items={this.items}> </AccordionComponent><br/> <br/>constructor(props: {})
{<br/> this.items = { expanded: true }};|

| Collapse Item | Method : collapsePanel(index) <br/> <EJ.Accordion id="Accordion"
#Accordion></EJ.Accordion><br/><br/>var
obj=$('#'#Accordion').ejAccordion('instance')<br/>obj.collapsePanel(0);| Method : expandItem(index,
false) <br/> <AccordionComponent id="Accordion" items={this.items} ref = {{(scope) =>
{this.AccordionObj = scope}}}> </AccordionComponent><br/> <br/> constructor(props: {}) {<br/>
this.AccordionObj.expandItem(0, false); }|

| Expand Item | Method : expandPanel(index) <br/> <EJ.Accordion id="Accordion"
#Accordion></EJ.Accordion><br/> <br/> var
obj=$('#'#Accordion').ejAccordion('instance')<br/>obj.expandPanel(0);| Method : expandItem(index, true)
<br/> <AccordionComponent id="Accordion" ref = {{(scope) => {this.AccordionObj = scope}}
items={this.items}> </AccordionComponent><br/> <br/> constructor(props: {}) {<br/>
this.AccordionObj.expandItem(0, true); }|

| CollapseAll | Method : collapseAll() <br/> <EJ.Accordion id="Accordion"
#Accordion></EJ.Accordion><br/> <br/>var
obj=$('#'#Accordion').ejAccordion('instance')<br/>obj.collapseAll();| <b>Not Applicable</b> |

| ExpandAll | Method : expandAll() <br/> <EJ.Accordion id="Accordion"
#Accordion></EJ.Accordion><br/> <br/>var
obj=$('#'#Accordion').ejAccordion('instance')<br/>obj.expandAll();| <b>Not Applicable</b> |

| Get ItemsCount | Method : getItemsCount() <br/> <EJ.Accordion id="Accordion"
#Accordion></EJ.Accordion><br/> <br/>var
obj=$('#'#Accordion').ejAccordion('instance')<br/>obj.getItemsCount();| <b>Not Applicable</b> |

| AddItem | Method : addItem(text, content, index) <br/> <EJ.Accordion id="Accordion"
#Accordion></EJ.Accordion><br/> <br/> var
obj=$('#'#Accordion').ejAccordion('instance')<br/>obj.addItem("New item", "The accordion content", 2);|
Method : addItem(items, index) <br/> <AccordionComponent id="Accordion"items={this.items}>
</AccordionComponent><br/> <br/> constructor(props: {}) {<br/> this.AccordionObj.addItem({{
header: 'App', content: 'text' }}, 0); }|

| Remove Item | Method : removeItem(index) <br/> <EJ.Accordion id="Accordion"
#Accordion></EJ.Accordion><br/> <br/>var
obj=$('#'#Accordion').ejAccordion('instance')<br/>obj.removeItem(0);| Method : removeItem(index)
<br/> <AccordionComponent id="Accordion"items={this.items}> </AccordionComponent><br/>
<br/> constructor(props: {}) {<br/> this.AccordionObj.removeItem(index); }|
```

| Disable Items | **Property** : disabledItems
 <EJ.Accordion id="Accordion" disabledItems="[0, 1]"></EJ.Accordion> | Not Applicable |

| Enable Items | **Property** : enabledItems
 <EJ.Accordion id="Accordion" enabledItems="[0, 1]"></EJ.Accordion> | Not Applicable |

| Disable Item | **Method** : disableItems([index])
 <EJ.Accordion id="Accordion" #Accordion></EJ.Accordion>

var
obj=\$('#Accordion').ejAccordion('instance')
obj.disableItems([1]); | **Method** : enableItem(index, false)
 <AccordionComponent id="Accordion" ref = {(scope) => {this.AccordionObj = scope}}></AccordionComponent>

constructor(props: {}) {

this.AccordionObj.enableItem(0, false); } |

| Enable Item | **Method** : enableItems([index])
 <EJ.Accordion id="Accordion" #Accordion></EJ.Accordion>

var
obj=\$('#Accordion').ejAccordion('instance')
obj.enableItems([1]); | **Method** : enableItem(index, true)
 <AccordionComponent id="Accordion" ref = {(scope) => {this.AccordionObj = scope}}></AccordionComponent>

constructor(props: {}) {

this.AccordionObj.enableItem(0, true); } |

| Hide Item | Not Applicable | **Method** : hideItem(index, true)
 <AccordionComponent id="Accordion" ref = {(scope) => {this.AccordionObj = scope}}></AccordionComponent>

constructor(props: {}) {
 this.AccordionObj.hideItem(0, true); } |

| SelectedItemIndex | **Property** : selectedItemIndex
 <EJ.Accordion id="Accordion" #Accordion selectedItemIndex={false}></EJ.Accordion>
 | Not Applicable |

| Select | Not Applicable | **Method** : select(index)
 <AccordionComponent id="Accordion" ref = {(scope) => {this.AccordionObj = scope}}></AccordionComponent>

constructor(props: {}) {
 this.AccordionObj.select(0); } |

| BeforeActivate | **Event**: beforeActivate
 <EJ.Accordion id='accordion' beforeActivate={this.onbeforeActivate}></EJ.Accordion>

 onbeforeActivate(event){} |
Event: expanding
 <AccordionComponent id="Accordion" ref = {(scope) => {this.AccordionObj = scope}}
expanding={this.onexpanding.bind(this)}></AccordionComponent>

onexpanding(event){} |

| Activate | **Event**: activate
 <EJ.Accordion id='accordion' activate={this.onActivate}></EJ.Accordion>

 onActivate(event){} | **Event**: expanded

 <AccordionComponent id="Accordion" ref = {(scope) => {this.AccordionObj = scope}}
expanded={onexpanded.bind(this)}></AccordionComponent>

 onexpanded(event){} |

| beforeInActivate | **Event**: beforeInactivate
 <EJ.Accordion id='accordion' beforeInactivate={this.onbeforeInactivate}></EJ.Accordion>

onbeforeInactivate(event){} | Not Applicable |

| InActive | **Event**: inActivate
<EJ.Accordion id='accordion' inActivate={this.onInActivate}></EJ.Accordion>

 onInActivate(event){} | Not Applicable |

|Clicked | **Event:** clicked
 <EJ.Accordion id='accordion'
clicked={this.onclicked}></EJ.Accordion>

 onclicked (event){} | Not Applicable |

Common

<!-- markdownlint-disable MD033 -->

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

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

|Collapsible | **Property** : collapsible
 <EJ.Accordion id="Accordion" collapsible={false}>
</EJ.Accordion>| Not Applicable |

|Collapse speed | **Property** : collapseSpeed
 <EJ.Accordion id="Accordion"
collapseSpeed="500"> </EJ.Accordion>| Not Applicable |

|Custom class | **Property** : cssClass
 <EJ.Accordion id="Accordion" cssClass="custom" >
</EJ.Accordion>| Not Applicable |

|CustomIcon class | **Property:** customIcon
 <EJ.Accordion id="Accordion"
customIcon={this.custom} > </EJ.Accordion>

 public custom: Object = {header: "header-
close", selectedHeader: "header-expand"}; | Not Applicable |

|Enabled | **Property** : enabled
 <EJ.Accordion id="Accordion" enabled={false} >
</EJ.Accordion>| Not Applicable |

|Events | **Property** : events
 <EJ.Accordion id="Accordion" events="mouseover" >
</EJ.Accordion>| Not Applicable |

|Expand speed | **Property** : expandSpeed
 <EJ.Accordion id="Accordion" expandSpeed="300"
> </EJ.Accordion>| Not Applicable |

|Height | **Property** : height
 <EJ.Accordion id="Accordion" height="400" > </EJ.Accordion>|
Property : height
 <AccordionComponent id="Accordion" height="400" >
</AccordionComponent> |

|HeightAdjustMode | **Property** : heightAdjustMode
 <EJ.Accordion id="Accordion"
heightAdjustMode="content" > </EJ.Accordion>| Not Applicable |

|HtmlAttributes | **Property** : htmlAttributes
 <EJ.Accordion id="Accordion"
[htmlAttributes]="attributes" > </EJ.Accordion>

 public attributes: Object = {title:
"Demo"};| Not Applicable |

|MultipleOpen | **Property** : enableMultipleOpen
 <EJ.Accordion id="Accordion"
enableMultipleOpen={true} > </EJ.Accordion>| **Property** : expandMode

<AccordionComponent id="Accordion" [expandMode]="Multiple" > </AccordionComponent> |

|Persistence | **Property** : enablePersistence
 <EJ.Accordion id="Accordion"
enablePersistence={false} > </EJ.Accordion>| **Property** : enablePersistence

<AccordionComponent id="Accordion" enablePersistence={true} > </AccordionComponent> |

|ShowRoundedCorner | **Property** : showRoundedCorner
 <EJ.Accordion id="Accordion"
showRoundedCorner={false} > </EJ.Accordion>| Not Applicable |

```
|Width | Property : width <br/> <EJ.Accordion id="Accordion" [width]="600" > </EJ.Accordion> |
Property : width <br/> <AccordionComponent id="Accordion" [width]="400" >
</AccordionComponent> |
```

```
|Enable | Method : enable() <br/> <EJ.Accordion id="Accordion"
#Accordion></EJ.Accordion><br/> <br/> @ViewChild('Accordion') var
obj=$('#'#Accordion').ejAccordion('instance')<br/>obj.enable(); | <b>Not Applicable</b> |
```

```
|Disable | Method : disable() <br/> <EJ.Accordion id="Accordion"
#Accordion></EJ.Accordion><br/> <br/> var
obj=$('#'#Accordion').ejAccordion('instance')<br/>obj.disable(); | <b>Not Applicable</b> |
```

```
|Show | Method : show() <br/> <EJ.Accordion id="Accordion" #Accordion></EJ.Accordion><br/>
<br/> var obj=$('#'#Accordion').ejAccordion('instance')<br/>obj.show(); | <b>Not Applicable</b> |
```

```
|Hide | Method : hide() <br/> <EJ.Accordion id="Accordion" #Accordion></EJ.Accordion><br/>
<br/> var obj=$('#'#Accordion').ejAccordion('instance')<br/>obj.hide(); | <b>Not Applicable</b> |
```

```
|Destroy | Method : destroy() <br/> <EJ.Accordion id="Accordion"
#Accordion></EJ.Accordion><br/> <br/> var
obj=$('#'#Accordion').ejAccordion('instance')<br/>obj.destroy(); | Method : destroy() <br/>
<AccordionComponent id="Accordion" ref = {{(scope) => {this.AccordionObj =
scope}}} ></AccordionComponent><br/> <br/> constructor(props: {}) {<br/>
this.AccordionObj.destroy(); |
```

```
|Refresh | Method : refresh() <br/> <EJ.Accordion id="Accordion"
#Accordion></EJ.Accordion><br/> <br/> var
obj=$('#'#Accordion').ejAccordion('instance')<br/>obj.refresh(); | Method : refresh() <br/>
<AccordionComponent id="Accordion" ref = {{(scope) => {this.AccordionObj =
scope}}} ></AccordionComponent><br/> <br/> constructor(props: {}) {<br/>
this.AccordionObj.refresh(); |
```

```
|Created | Event: create<br/> <EJ.Accordion id="Accordion" #Accordion
create={this.oncreate}></EJ.Accordion><br/> <br/> oncreate(event) {} | Event: created<br/>
<AccordionComponent id="Accordion" ref = {{(scope) => {this.AccordionObj = scope}}}
created={this.oncreated.bind(this)}></AccordionComponent><br/> <br/> oncreated(event) {} |
```

```
|Destroyed | Event: destroy<br/> <EJ.Accordion id="Accordion" #Accordion
destroy={this.ondestroy}></EJ.Accordion><br/> <br/> ondestroy(event) {} | Link to the Video
destroyed<br/> <AccordionComponent id="Accordion" ref = {{(scope) => {this.AccordionObj =
scope}}} (destroyed)={this.ondestroyed.bind(this)}></AccordionComponent><br/> <br/>
ondestroyed(event) {} |
```

Accumulation Chart

```
<!-- markdownlint-disable MD036 -->
```

Getting Started

In EJ2, accumulation chart is implemented as a separate control to avoid the axis related logics.

The dependencies for accumulation chart is same as chart control.

To get start quickly with React Accumulation Charts, you can check on this video:

Dependencies

Below is the list of minimum dependencies required to use the accumulation chart component.

```
`javascript
|-- @syncfusion/ej2-react-charts
|-- @syncfusion/ej2-base
|-- @syncfusion/ej2-data
|-- @syncfusion/ej2-charts
|-- @syncfusion/ej2-pdf-export
|-- @syncfusion/ej2-file-utils
|-- @syncfusion/ej2-compression
|-- @syncfusion/ej2-react-popups
|-- @syncfusion/ej2-react-buttons
|-- @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 Accumulation Chart to the Project

Now, you can start adding Accumulation Chart component in the application.

For getting started, add the Accumulation Chart component in `src/App.tsx` file using following code.

INDEX.JSX

```
import { AccumulationChartComponent } from '@syncfusion/ej2-react-charts';
import * as React from 'react';
function App() {
  return (<AccumulationChartComponent />);
}
export default App;
```

INDEX.TSX

```
import { AccumulationChartComponent } from '@syncfusion/ej2-react-charts';
import * as React from 'react';
function App() {
  return (<AccumulationChartComponent />);
}
export default App;
```

Pie Series

By default, the pie series will be rendered when assigning the JSON data to the series using the [dataSource](#) property. Map the field names in the JSON data to the [xName](#) and [yName](#) properties of the series.

INDEX.JSX

```
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
  AccumulationSeriesDirective } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { x: 'Jan', y: 3, text: 'Jan: 3' }, { x: 'Feb', y: 3.5, text: 'Feb: 3.5' },
    { x: 'Mar', y: 7, text: 'Mar: 7' }, { x: 'Apr', y: 13.5, text: 'Apr: 13.5' },
    { x: 'May', y: 19, text: 'May: 19' }, { x: 'Jun', y: 23.5, text: 'Jun: 23.5' },
    { x: 'Jul', y: 26, text: 'Jul: 26' }, { x: 'Aug', y: 25, text: 'Aug: 25' },
  ],
```

```

    { x: 'Sep', y: 21, text: 'Sep: 21' }, { x: 'Oct', y: 15, text: 'Oct:
15' }
  ];
  return <AccumulationChartComponent id="charts">
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={data} xName='x' yName='y'
radius='90%' />
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data: any[] = [
    { x: 'Jan', y: 3, text: 'Jan: 3' }, { x: 'Feb', y: 3.5, text: 'Feb: 3.5'
},
    { x: 'Mar', y: 7, text: 'Mar: 7' }, { x: 'Apr', y: 13.5, text: 'Apr:
13.5' },
    { x: 'May', y: 19, text: 'May: 19' }, { x: 'Jun', y: 23.5, text: 'Jun:
23.5' },
    { x: 'Jul', y: 26, text: 'Jul: 26' }, { x: 'Aug', y: 25, text: 'Aug: 25'
},
    { x: 'Sep', y: 21, text: 'Sep: 21' }, { x: 'Oct', y: 15, text: 'Oct: 15'
}
  ];
  return <AccumulationChartComponent id="charts">
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={data} xName='x' yName='y'
radius='90%' />
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

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

,

npm start

,

Pie dough nut in React Accumulation chart component

Pie Chart

To render a pie series, use the series `type` as `Pie` and inject the `PieSeries` module into the `services`. If the `PieSeries` module is not injected, this module will be loaded by default.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, PieSeries, Inject } from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    return <AccumulationChartComponent id='charts'>
        <Inject services={[PieSeries]}/>
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' type='Pie'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, PieSeries, Inject } from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    return <AccumulationChartComponent id='charts'>
        <Inject services={[PieSeries]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' type='Pie'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Radius customization

By default, radius of the pie series will be 80% of the size (minimum of chart width and height).

You can customize this using [radius](#) property of the series.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective } from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
```

```

    return <AccumulationChartComponent id='charts'>
      <AccumulationSeriesCollectionDirective>
        <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' radius='100%'>
        </AccumulationSeriesDirective>
      </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective } from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
  return <AccumulationChartComponent id='charts'>
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' radius='100%'>
      </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Pie Center

The center position of the pie can be changed by Center X and Center Y. By default, center value of the pie series x and y is 50%. You can customize this using [center](#) property of the series.

INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend, PieSeries,
AccumulationTooltip, AccumulationDataLabel } from '@syncfusion/ej2-react-
charts';
import { data1 } from 'datasource.ts';
function App() {
  let pie;
  return <AccumulationChartComponent id='pie-chart' ref={pie => pie = pie}
title='Mobile Browser Statistics' legendSettings={{ visible: false }}
enableSmartLabels={true} enableAnimation={false} center={{ x: '60%', y:
'60%' }} tooltip={{ enable: false, format: '$ {point.x} : <b>${point.y}%</b>'
}}>
    <Inject services={[AccumulationLegend, PieSeries,
AccumulationTooltip, AccumulationDataLabel]}/>
    <AccumulationSeriesCollectionDirective>

```

```

        <AccumulationSeriesDirective dataSource={data1}
name='Browser' xName='x' yName='y' explode={true} explodeOffset='10%'
explodeIndex={0} dataLabel={{
    visible: true,
    position: 'Inside', name: 'text',
    font: {
        fontWeight: '600'
    }
}} radius='70%'>
        </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
</AccumulationChartComponent>;
}
;
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 { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective,
    Inject, AccumulationLegend, PieSeries, AccumulationTooltip,
    AccumulationDataLabel } from '@syncfusion/ej2-react-charts';
import { data1 } from 'datasource.ts';
function App() {
    let pie: AccumulationChartComponent;

    return <AccumulationChartComponent id='pie-chart' ref={pie => pie = pie}
        title='Mobile Browser Statistics'
        legendSettings={{ visible: false }}
        enableSmartLabels={true}
        enableAnimation={false}
        center={{x: '60%', y: '60%'}}
        tooltip={{ enable: false, format: '${point.x} :
<b>${point.y}%</b>' }}
        >
        <Inject services={[AccumulationLegend, PieSeries,
AccumulationTooltip, AccumulationDataLabel]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={data1}
name='Browser' xName='x' yName='y'
                explode={true} explodeOffset='10%' explodeIndex={0}
                dataLabel={{
                    visible: true,
                    position: 'Inside', name: 'text',
                    font: {
                        fontWeight: '600'
                    }
                }}
                radius='70%'
            >
            </AccumulationSeriesDirective>

```

```

        </AccumulationSeriesCollectionDirective>
      </AccumulationChartComponent>
    };
    export default App;
    ReactDOM.render(<App />, document.getElementById("charts"));
    {% endraw %}

```

Various Radius Pie Chart

You can use radius mapping to render the slice with different [radius](#) pie and also use [border](#), fill properties to customize the point. dataLabel is used to represent individual data and its value.

INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, AccumulationLegend, PieSeries,
AccumulationDataLabel, AccumulationTooltip, Inject } from '@syncfusion/ej2-
react-charts';
import { variouspiedata } from 'datasource.ts';
function App() {
  let pie;
  return <AccumulationChartComponent id='pie-chart' ref={pie => pie = pie}
  legendSettings={{
    visible: true
  }} enableSmartLabels={true} enableAnimation={true} tooltip={{
enable: true }}>
    <Inject services={[AccumulationLegend, PieSeries,
AccumulationDataLabel, AccumulationTooltip]}/>
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={variouspiedata}
xName='x' yName='y' innerRadius='20%' dataLabel={{
  visible: true, position: 'Outside', name: 'x'
}} radius='r'>
    </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>;
}
;
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 { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, AccumulationLegend, PieSeries,
AccumulationDataLabel, AccumulationTooltip, Inject } from '@syncfusion/ej2-
react-charts';
import { variouspiedata } from 'datasource.ts';
function App() {

```

```

let pie: AccumulationChartComponent;

return <AccumulationChartComponent id='pie-chart' ref={pie => pie = pie}
  legendSettings={{
    visible: true
  }}
  enableSmartLabels={true}
  enableAnimation={true}
  tooltip={{ enable: true }}
>
  <Inject services={[AccumulationLegend, PieSeries,
AccumulationDataLabel, AccumulationTooltip]} />
  <AccumulationSeriesCollectionDirective>
    <AccumulationSeriesDirective dataSource={variouspiedata}
xName='x' yName='y' innerRadius='20%'
      dataLabel={{
        visible: true, position: 'Outside', name: 'x'
      }}
      radius='r'
    >
    </AccumulationSeriesDirective>
  </AccumulationSeriesCollectionDirective>
</AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

Doughnut Chart

To achieve a doughnut in pie series, customize the [innerRadius](#) property of the series. By setting value greater than 0%, a doughnut will appear. The [innerRadius](#) property takes value from 0% to 100% of the pie radius.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective } from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
  return <AccumulationChartComponent id='charts'>
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' innerRadius='40%'>
    </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective } from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    return <AccumulationChartComponent id='charts'>
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' innerRadius='40%'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Start and End angles

You can customize the start and end angle of the pie series using the [startAngle](#) and [endAngle](#) properties. The default value of `startAngle` is 0 degree, and `endAngle` is 360 degrees. By customizing this, you can achieve a semi pie series.

INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationDataLabel } from
'@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    return <AccumulationChartComponent id='charts'>
        <Inject services={[AccumulationDataLabel]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' startAngle='270' endAngle='90' dataLabel={{ visible: true, name:
'text', position: 'Outside' }}>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
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 { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationDataLabel }
from '@syncfusion/ej2-react-charts';
```



```
import { accData } from 'datasource.ts';
function App() {
  return <AccumulationChartComponent id='charts'>
    <Inject services={[AccumulationDataLabel]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' startAngle='270' endAngle='90' dataLabel={{ visible: true, name:
'text', position: 'Outside' }}>
    </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

Color & Text Mapping

The fill color and the text in the data source can be mapped to the chart using `pointColorMapping` in series and `name` in `datalabel` respectively.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationDataLabel } from
'@syncfusion/ej2-react-charts';
import { dataMapping } from 'datasource.ts';
function App() {
  const datalabel = { visible: true, name: 'text', position: 'Outside' };
  return <AccumulationChartComponent id='charts'>
    <Inject services={[AccumulationDataLabel]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={dataMapping} xName='x'
yName='y' pointColorMapping='fill'
dataLabel={datalabel}></AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective,
Inject, AccumulationDataLabel, AccumulationDataLabelSettingsModel }
from '@syncfusion/ej2-react-charts';
import { dataMapping } from 'datasource.ts';
function App() {
  const datalabel: AccumulationDataLabelSettingsModel = { visible: true,
name: 'text', position: 'Outside' };
}
```

```

return <AccumulationChartComponent id='charts'>
  <Inject services={[AccumulationDataLabel]} />
  <AccumulationSeriesCollectionDirective>
    <AccumulationSeriesDirective dataSource={dataMapping} xName='x'
yName='y' pointColorMapping='fill'
dataLabel={datalabel}></AccumulationSeriesDirective>
  </AccumulationSeriesCollectionDirective>
</AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Customization

Individual points can be customized using the `pointRender` event.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationDataLabel } from
'@syncfusion/ej2-react-charts';
import { dataMapping } from 'datasource.ts';
function App() {
  const onPointRender = (args) => {
    if (args.point.x.indexOf('Apr') > -1) {
      args.fill = '#f4bc42';
    }
    else {
      args.fill = '#597cf9';
    }
  };
  return <AccumulationChartComponent id='charts'
pointRender={onPointRender}>
    <Inject services={[AccumulationDataLabel]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={dataMapping} xName='x'
yName='y'>
        </AccumulationSeriesDirective>
      </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent,
AccumulationSeriesCollectionDirective, IAccPointRenderEventArgs,
AccumulationSeriesDirective, Inject, AccumulationDataLabel}
from '@syncfusion/ej2-react-charts';
import { dataMapping } from 'datasource.ts';

```

```
function App() {
  const onPointRender: EmitType<IAccPointRenderEventArgs> = (args:
IAccPointRenderEventArgs): void => {
    if ((args.point.x as string).indexOf('Apr') > -1) {
      args.fill = '#f4bc42';
    }
    else {
      args.fill = '#597cf9';
    }
  };
  return <AccumulationChartComponent id='charts'
pointRender={onPointRender}>
    <Inject services={[AccumulationDataLabel]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={dataMapping} xName='x'
yName='y'>
    </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Hide pie or doughnut border

By default, the border will appear in the pie/doughnut chart while mouse hover on the chart. You can disable the the border by setting `enableBorderOnMouseMove` property is `false`.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationDataLabel } from
'@syncfusion/ej2-react-charts';
import { dataMapping } from 'datasource.ts';
function App() {
  return <AccumulationChartComponent id='charts'
enableBorderOnMouseMove={false}>
    <Inject services={[AccumulationDataLabel]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={dataMapping} xName='x'
yName='y'>
    </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

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

```
import { AccumulationChartComponent,
AccumulationSeriesCollectionDirective, IAccPointRenderEventArgs,
AccumulationSeriesDirective, Inject, AccumulationDataLabel }
from '@syncfusion/ej2-react-charts';
import { dataMapping } from 'datasource.ts';
function App() {

    return <AccumulationChartComponent id='charts'
    enableBorderOnMouseMove={false}>
        <Inject services={[AccumulationDataLabel]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={dataMapping} xName='x'
yName='y'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Multi-level pie chart

You can achieve a multi-level drill down in pie and doughnut charts using [pointClick](#) event. If user clicks any point in the chart, that corresponding data will be shown in the next level and so on based on point clicked.

You can also achieve drill-up (back to the initial state) by using [chartMouseClicked](#) event. In below sample, you can drill-up by clicking back button in the center of the chart.

DATASOURCE.JSX

```
export let productData = [
    {
        'ProductID': 1,
        'ProductName': 'Chai',
        'SupplierID': 1,
        'QuantityPerUnit': '10 boxes x 20 bags',
        'UnitPrice': 18.00,
        'UnitsInStock': 39,
        'Discontinued': true
    },
    {
        'ProductID': 2,
        'ProductName': 'Chang',
        'SupplierID': 1,
        'QuantityPerUnit': '24 - 12 oz bottles',
        'UnitPrice': 19.00,
        'UnitsInStock': 17,
        'Discontinued': true
    },
    {
        'ProductID': 3,
        'ProductName': 'Aniseed Syrup',
        'SupplierID': 1,
        'QuantityPerUnit': '12 - 550 ml bottles',
        'UnitPrice': 10.00,
        'UnitsInStock': 13,
```

```

    'Discontinued': true
  },
  {
    'ProductID': 4,
    'ProductName': 'Chef Anton\'s Cajun Seasoning',
    'SupplierID': 2,
    'QuantityPerUnit': '48 - 6 oz jars',
    'UnitPrice': 22.00,
    'UnitsInStock': 53,
    'Discontinued': true
  },
  {
    'ProductID': 5,
    'ProductName': 'Chef Anton\'s Gumbo Mix',
    'SupplierID': 2,
    'QuantityPerUnit': '36 boxes',
    'UnitPrice': 21.35,
    'UnitsInStock': 0,
    'Discontinued': true
  },
  {
    'ProductID': 6,
    'ProductName': 'Grandma\'s Boysenberry Spread',
    'SupplierID': 3,
    'QuantityPerUnit': '12 - 8 oz jars',
    'UnitPrice': 25.00,
    'UnitsInStock': 120,
    'Discontinued': false
  },
  {
    'ProductID': 7,
    'ProductName': 'Uncle Bob\'s Organic Dried Pears',
    'SupplierID': 3,
    'QuantityPerUnit': '12 - 1 lb pkgs.',
    'UnitPrice': 30.00,
    'UnitsInStock': 15,
    'Discontinued': false
  },
  {
    'ProductID': 8,
    'ProductName': 'Northwoods Cranberry Sauce',
    'SupplierID': 3,
    'QuantityPerUnit': '12 - 12 oz jars',
    'UnitPrice': 40.00,
    'UnitsInStock': 6,
    'Discontinued': false
  },
  {
    'ProductID': 9,
    'ProductName': 'Mishi Kobe Niku',
    'SupplierID': 4,
    'QuantityPerUnit': '18 - 500 g pkgs.',
    'UnitPrice': 97.00,
    'UnitsInStock': 29,
    'Discontinued': true
  },
  {

```

```

    'ProductID': 10,
    'ProductName': 'Ikura',
    'SupplierID': 4,
    'QuantityPerUnit': '12 - 200 ml jars',
    'UnitPrice': 31.00,
    'UnitsInStock': 31,
    'Discontinued': false
  },
  {
    'ProductID': 11,
    'ProductName': 'Queso Cabrales',
    'SupplierID': 5,
    'QuantityPerUnit': '1 kg pkg.',
    'UnitPrice': 21.00,
    'UnitsInStock': 22,
    'Discontinued': false
  },
  {
    'ProductID': 12,
    'ProductName': 'Queso Manchego La Pastora',
    'SupplierID': 5,
    'QuantityPerUnit': '10 - 500 g pkgs.',
    'UnitPrice': 38.00,
    'UnitsInStock': 86,
    'Discontinued': false
  },
  {
    'ProductID': 13,
    'ProductName': 'Konbu',
    'SupplierID': 6,
    'QuantityPerUnit': '2 kg box',
    'UnitPrice': 6.00,
    'UnitsInStock': 24,
    'Discontinued': true
  },
  {
    'ProductID': 14,
    'ProductName': 'Tofu',
    'SupplierID': 6,
    'QuantityPerUnit': '40 - 100 g pkgs.',
    'UnitPrice': 23.25,
    'UnitsInStock': 35,
    'Discontinued': true
  },
  {
    'ProductID': 15,
    'ProductName': 'Genen Shouyu',
    'SupplierID': 6,
    'QuantityPerUnit': '24 - 250 ml bottles',
    'UnitPrice': 15.50,
    'UnitsInStock': 39,
    'Discontinued': true
  },
  {
    'ProductID': 16,
    'ProductName': 'Pavlova',
    'SupplierID': 7,

```

```

    'QuantityPerUnit': '32 - 500 g boxes',
    'UnitPrice': 17.45,
    'UnitsInStock': 29,
    'Discontinued': true
  },
  {
    'ProductID': 17,
    'ProductName': 'Alice Mutton',
    'SupplierID': 7,
    'QuantityPerUnit': '20 - 1 kg tins',
    'UnitPrice': 39.00,
    'UnitsInStock': 0,
    'Discontinued': true
  },
  {
    'ProductID': 18,
    'ProductName': 'Carnarvon Tigers',
    'SupplierID': 7,
    'QuantityPerUnit': '16 kg pkg.',
    'UnitPrice': 62.50,
    'UnitsInStock': 42,
    'Discontinued': false
  },
  {
    'ProductID': 19,
    'ProductName': 'Teatime Chocolate Biscuits',
    'SupplierID': 8,
    'QuantityPerUnit': '10 boxes x 12 pieces',
    'UnitPrice': 9.20,
    'UnitsInStock': 25,
    'Discontinued': false
  },
  {
    'ProductID': 20,
    'ProductName': 'Sir Rodney\'s Marmalade',
    'SupplierID': 8,
    'QuantityPerUnit': '30 gift boxes',
    'UnitPrice': 81.00,
    'UnitsInStock': 40,
    'Discontinued': false
  },
  {
    'ProductID': 21,
    'ProductName': 'Sir Rodney\'s Scones',
    'SupplierID': 8,
    'QuantityPerUnit': '24 pkgs. x 4 pieces',
    'UnitPrice': 10.00,
    'UnitsInStock': 3,
    'Discontinued': false
  },
  {
    'ProductID': 22,
    'ProductName': 'Gustaf\'s Knäckebröd',
    'SupplierID': 9,
    'QuantityPerUnit': '24 - 500 g pkgs.',
    'UnitPrice': 21.00,
    'UnitsInStock': 104,

```

```

    'Discontinued': false
  },
  {
    'ProductID': 23,
    'ProductName': 'Tunnbröd',
    'SupplierID': 9,
    'QuantityPerUnit': '12 - 250 g pkgs.',
    'UnitPrice': 9.00,
    'UnitsInStock': 61,
    'Discontinued': false
  },
  {
    'ProductID': 24,
    'ProductName': 'Guaraná Fantástica',
    'SupplierID': 10,
    'QuantityPerUnit': '12 - 355 ml cans',
    'UnitPrice': 4.50,
    'UnitsInStock': 20,
    'Discontinued': true
  },
  {
    'ProductID': 25,
    'ProductName': 'NuNuCa Nuß-Nougat-Creme',
    'SupplierID': 11,
    'QuantityPerUnit': '20 - 450 g glasses',
    'UnitPrice': 14.00,
    'UnitsInStock': 76,
    'Discontinued': false
  },
  {
    'ProductID': 26,
    'ProductName': 'Gumbär Gummibärchen',
    'SupplierID': 11,
    'QuantityPerUnit': '100 - 250 g bags',
    'UnitPrice': 31.23,
    'UnitsInStock': 15,
    'Discontinued': true
  },
  {
    'ProductID': 27,
    'ProductName': 'Schoggi Schokolade',
    'SupplierID': 11,
    'QuantityPerUnit': '100 - 100 g pieces',
    'UnitPrice': 43.90,
    'UnitsInStock': 49,
    'Discontinued': true
  },
  {
    'ProductID': 28,
    'ProductName': 'Rössle Sauerkraut',
    'SupplierID': 12,
    'QuantityPerUnit': '25 - 825 g cans',
    'UnitPrice': 45.60,
    'UnitsInStock': 26,
    'Discontinued': true
  },
  {

```



```

    'ProductID': 29,
    'ProductName': 'Thüringer Rostbratwurst',
    'SupplierID': 12,
    'QuantityPerUnit': '50 bags x 30 sausgs.',
    'UnitPrice': 123.79,
    'UnitsInStock': 0,
    'Discontinued': true
  },
  {
    'ProductID': 30,
    'ProductName': 'Nord-Ost Matjeshering',
    'SupplierID': 13,
    'QuantityPerUnit': '10 - 200 g glasses',
    'UnitPrice': 25.89,
    'UnitsInStock': 10,
    'Discontinued': true
  },
  {
    'ProductID': 31,
    'ProductName': 'Gorgonzola Telino',
    'SupplierID': 14,
    'QuantityPerUnit': '12 - 100 g pkgs',
    'UnitPrice': 12.50,
    'UnitsInStock': 0,
    'Discontinued': true
  },
  {
    'ProductID': 32,
    'ProductName': 'Mascarpone Fabioli',
    'SupplierID': 14,
    'QuantityPerUnit': '24 - 200 g pkgs.',
    'UnitPrice': 32.00,
    'UnitsInStock': 9,
    'Discontinued': false
  },
  {
    'ProductID': 33,
    'ProductName': 'Geitost',
    'SupplierID': 15,
    'QuantityPerUnit': '500 g',
    'UnitPrice': 2.50,
    'UnitsInStock': 112,
    'Discontinued': false
  },
  {
    'ProductID': 34,
    'ProductName': 'Sasquatch Ale',
    'SupplierID': 16,
    'QuantityPerUnit': '24 - 12 oz bottles',
    'UnitPrice': 14.00,
    'UnitsInStock': 111,
    'Discontinued': false
  },
  {
    'ProductID': 35,
    'ProductName': 'Steeleye Stout',
    'SupplierID': 16,

```

```

    'QuantityPerUnit': '24 - 12 oz bottles',
    'UnitPrice': 18.00,
    'UnitsInStock': 20,
    'Discontinued': false
  },
  {
    'ProductID': 36,
    'ProductName': 'Inlagd Sill',
    'SupplierID': 17,
    'QuantityPerUnit': '24 - 250 g jars',
    'UnitPrice': 19.00,
    'UnitsInStock': 112,
    'Discontinued': false
  },
  {
    'ProductID': 37,
    'ProductName': 'Gravad lax',
    'SupplierID': 17,
    'QuantityPerUnit': '12 - 500 g pkgs.',
    'UnitPrice': 26.00,
    'UnitsInStock': 11,
    'Discontinued': false
  },
  {
    'ProductID': 38,
    'ProductName': 'Côte de Blaye',
    'SupplierID': 18,
    'QuantityPerUnit': '12 - 75 cl bottles',
    'UnitPrice': 263.50,
    'UnitsInStock': 17,
    'Discontinued': false
  },
  {
    'ProductID': 39,
    'ProductName': 'Chartreuse verte',
    'SupplierID': 18,
    'QuantityPerUnit': '750 cc per bottle',
    'UnitPrice': 18.00,
    'UnitsInStock': 69,
    'Discontinued': true
  },
  {
    'ProductID': 40,
    'ProductName': 'Boston Crab Meat',
    'SupplierID': 19,
    'QuantityPerUnit': '24 - 4 oz tins',
    'UnitPrice': 18.40,
    'UnitsInStock': 123,
    'Discontinued': true
  },
  {
    'ProductID': 41,
    'ProductName': 'Jack\'s New England Clam Chowder',
    'SupplierID': 19,
    'QuantityPerUnit': '12 - 12 oz cans',
    'UnitPrice': 9.65,
    'UnitsInStock': 85,

```

```

    'Discontinued': false
  },
  {
    'ProductID': 42,
    'ProductName': 'Singaporean Hokkien Fried Mee',
    'SupplierID': 20,
    'QuantityPerUnit': '32 - 1 kg pkgs.',
    'UnitPrice': 14.00,
    'UnitsInStock': 26,
    'Discontinued': true
  },
  {
    'ProductID': 43,
    'ProductName': 'Ipoh Coffee',
    'SupplierID': 20,
    'QuantityPerUnit': '16 - 500 g tins',
    'UnitPrice': 46.00,
    'UnitsInStock': 17,
    'Discontinued': false
  },
  {
    'ProductID': 44,
    'ProductName': 'Gula Malacca',
    'SupplierID': 20,
    'QuantityPerUnit': '20 - 2 kg bags',
    'UnitPrice': 19.45,
    'UnitsInStock': 27,
    'Discontinued': false
  },
  {
    'ProductID': 45,
    'ProductName': 'Rogede sild',
    'SupplierID': 21,
    'QuantityPerUnit': '1k pkg.',
    'UnitPrice': 9.50,
    'UnitsInStock': 5,
    'Discontinued': true
  },
  {
    'ProductID': 46,
    'ProductName': 'Spegesild',
    'SupplierID': 21,
    'QuantityPerUnit': '4 - 450 g glasses',
    'UnitPrice': 12.00,
    'UnitsInStock': 95,
    'Discontinued': true
  },
  {
    'ProductID': 47,
    'ProductName': 'Zaanse koeken',
    'SupplierID': 22,
    'QuantityPerUnit': '10 - 4 oz boxes',
    'UnitPrice': 9.50,
    'UnitsInStock': 36,
    'Discontinued': true
  },
  {

```

```

    'ProductID': 48,
    'ProductName': 'Chocolade',
    'SupplierID': 22,
    'QuantityPerUnit': '10 pkgs.',
    'UnitPrice': 12.75,
    'UnitsInStock': 15,
    'Discontinued': true
  },
  {
    'ProductID': 49,
    'ProductName': 'Maxilaku',
    'SupplierID': 23,
    'QuantityPerUnit': '24 - 50 g pkgs.',
    'UnitPrice': 20.00,
    'UnitsInStock': 10,
    'Discontinued': false
  },
  {
    'ProductID': 50,
    'ProductName': 'Valkoinen suklaa',
    'SupplierID': 23,
    'QuantityPerUnit': '12 - 100 g bars',
    'UnitPrice': 16.25,
    'UnitsInStock': 65,
    'Discontinued': false
  },
  {
    'ProductID': 51,
    'ProductName': 'Manjimup Dried Apples',
    'SupplierID': 24,
    'QuantityPerUnit': '50 - 300 g pkgs.',
    'UnitPrice': 53.00,
    'UnitsInStock': 20,
    'Discontinued': false
  },
  {
    'ProductID': 52,
    'ProductName': 'Filo Mix',
    'SupplierID': 24,
    'QuantityPerUnit': '16 - 2 kg boxes',
    'UnitPrice': 7.00,
    'UnitsInStock': 38,
    'Discontinued': true
  },
  {
    'ProductID': 53,
    'ProductName': 'Perth Pasties',
    'SupplierID': 24,
    'QuantityPerUnit': '48 pieces',
    'UnitPrice': 32.80,
    'UnitsInStock': 0,
    'Discontinued': true
  },
  {
    'ProductID': 54,
    'ProductName': 'Tourtière',
    'SupplierID': 25,

```

```

      'QuantityPerUnit': '16 pies',
      'UnitPrice': 7.45,
      'UnitsInStock': 21,
      'Discontinued': true
    },
    {
      'ProductID': 55,
      'ProductName': 'Pâté chinois',
      'SupplierID': 25,
      'QuantityPerUnit': '24 boxes x 2 pies',
      'UnitPrice': 24.00,
      'UnitsInStock': 115,
      'Discontinued': true
    },
    {
      'ProductID': 56,
      'ProductName': 'Gnocchi di nonna Alice',
      'SupplierID': 26,
      'QuantityPerUnit': '24 - 250 g pkgs.',
      'UnitPrice': 38.00,
      'UnitsInStock': 21,
      'Discontinued': false
    },
    {
      'ProductID': 57,
      'ProductName': 'Ravioli Angelo',
      'SupplierID': 26,
      'QuantityPerUnit': '24 - 250 g pkgs.',
      'UnitPrice': 19.50,
      'UnitsInStock': 36,
      'Discontinued': false
    },
    {
      'ProductID': 58,
      'ProductName': 'Escargots de Bourgogne',
      'SupplierID': 27,
      'QuantityPerUnit': '24 pieces',
      'UnitPrice': 13.25,
      'UnitsInStock': 62,
      'Discontinued': false
    },
    {
      'ProductID': 59,
      'ProductName': 'Raclette Courdavault',
      'SupplierID': 28,
      'QuantityPerUnit': '5 kg pkg.',
      'UnitPrice': 55.00,
      'UnitsInStock': 79,
      'Discontinued': false
    },
    {
      'ProductID': 60,
      'ProductName': 'Camembert Pierrot',
      'SupplierID': 28,
      'QuantityPerUnit': '15 - 300 g rounds',
      'UnitPrice': 34.00,
      'UnitsInStock': 19,

```

```

    'Discontinued': false
  },
  {
    'ProductID': 61,
    'ProductName': 'Sirop d\''érable',
    'SupplierID': 29,
    'QuantityPerUnit': '24 - 500 ml bottles',
    'UnitPrice': 28.50,
    'UnitsInStock': 113,
    'Discontinued': false
  },
  {
    'ProductID': 62,
    'ProductName': 'Tarte au sucre',
    'SupplierID': 29,
    'QuantityPerUnit': '48 pies',
    'UnitPrice': 49.30,
    'UnitsInStock': 17,
    'Discontinued': false
  },
  {
    'ProductID': 63,
    'ProductName': 'Veggie-spread',
    'SupplierID': 7,
    'QuantityPerUnit': '15 - 625 g jars',
    'UnitPrice': 43.90,
    'UnitsInStock': 24,
    'Discontinued': true
  },
  {
    'ProductID': 64,
    'ProductName': 'Wimmers gute Semmelknödel',
    'SupplierID': 12,
    'QuantityPerUnit': '20 bags x 4 pieces',
    'UnitPrice': 33.25,
    'UnitsInStock': 22,
    'Discontinued': true
  },
  {
    'ProductID': 65,
    'ProductName': 'Louisiana Fiery Hot Pepper Sauce',
    'SupplierID': 2,
    'QuantityPerUnit': '32 - 8 oz bottles',
    'UnitPrice': 21.05,
    'UnitsInStock': 76,
    'Discontinued': true
  },
  {
    'ProductID': 66,
    'ProductName': 'Louisiana Hot Spiced Okra',
    'SupplierID': 2,
    'QuantityPerUnit': '24 - 8 oz jars',
    'UnitPrice': 17.00,
    'UnitsInStock': 4,
    'Discontinued': false
  },
  {

```

```

    'ProductID': 67,
    'ProductName': 'Laughing Lumberjack Lager',
    'SupplierID': 16,
    'QuantityPerUnit': '24 - 12 oz bottles',
    'UnitPrice': 14.00,
    'UnitsInStock': 52,
    'Discontinued': false
  },
  {
    'ProductID': 68,
    'ProductName': 'Scottish Longbreads',
    'SupplierID': 8,
    'QuantityPerUnit': '10 boxes x 8 pieces',
    'UnitPrice': 12.50,
    'UnitsInStock': 6,
    'Discontinued': false
  },
  {
    'ProductID': 69,
    'ProductName': 'Gudbrandsdalsost',
    'SupplierID': 15,
    'QuantityPerUnit': '10 kg pkg.',
    'UnitPrice': 36.00,
    'UnitsInStock': 26,
    'Discontinued': false
  },
  {
    'ProductID': 70,
    'ProductName': 'Outback Lager',
    'SupplierID': 7,
    'QuantityPerUnit': '24 - 355 ml bottles',
    'UnitPrice': 15.00,
    'UnitsInStock': 15,
    'Discontinued': false
  },
  {
    'ProductID': 71,
    'ProductName': 'Flotemysost',
    'SupplierID': 15,
    'QuantityPerUnit': '10 - 500 g pkgs.',
    'UnitPrice': 21.50,
    'UnitsInStock': 26,
    'Discontinued': true
  },
  {
    'ProductID': 72,
    'ProductName': 'Mozzarella di Giovanni',
    'SupplierID': 14,
    'QuantityPerUnit': '24 - 200 g pkgs.',
    'UnitPrice': 34.80,
    'UnitsInStock': 14,
    'Discontinued': true
  },
  {
    'ProductID': 73,
    'ProductName': 'Röd Kaviar',
    'SupplierID': 17,

```

```

      'QuantityPerUnit': '24 - 150 g jars',
      'UnitPrice': 15.00,
      'UnitsInStock': 101,
      'Discontinued': true
    },
    {
      'ProductID': 74,
      'ProductName': 'Longlife Tofu',
      'SupplierID': 4,
      'QuantityPerUnit': '5 kg pkg.',
      'UnitPrice': 10.00,
      'UnitsInStock': 4,
      'Discontinued': true
    },
    {
      'ProductID': 75,
      'ProductName': 'Rhönbräu Klosterbier',
      'SupplierID': 12,
      'QuantityPerUnit': '24 - 0.5 l bottles',
      'UnitPrice': 7.75,
      'UnitsInStock': 125,
      'Discontinued': true
    },
    {
      'ProductID': 76,
      'ProductName': 'Lakkalikööri',
      'SupplierID': 23,
      'QuantityPerUnit': '500 ml',
      'UnitPrice': 18.00,
      'UnitsInStock': 57,
      'Discontinued': true
    },
    {
      'ProductID': 77,
      'ProductName': 'Original Frankfurter grüne Soße',
      'SupplierID': 12,
      'QuantityPerUnit': '12 boxes',
      'UnitPrice': 13.00,
      'UnitsInStock': 32,
      'Discontinued': false
    }
  ];
export let employeeData = [{
  'EmployeeID': 1,
  'LastName': 'Davolio',
  'FirstName': 'Nancy',
  'Title': 'Sales Representative',
  'TitleOfCourtesy': 'Ms.',
  'BirthDate': new Date(-664743600000),
  'HireDate': new Date(704692800000),
  'Address': '507 - 20th Ave. E.\r\nApt. 2A',
  'City': 'Seattle',
  'Region': 'WA',
  'PostalCode': '98122',
  'Country': 'USA',
  'HomePhone': '(206) 555-9857',
  'Extension': '5467',

```



```

    'Photo': { 'Length': 21626 },
    'Notes': 'Education includes a BA in psychology from Colorado State
University in 1970. She also completed\
\'The Art of the Cold Call.\' Nancy is a member of Toastmasters
International.',
    'ReportsTo': 2,
    'PhotoPath': 'http://accweb/emmployees/davolio.bmp'
  },
  {
    'EmployeeID': 2,
    'LastName': 'Fuller',
    'FirstName': 'Andrew',
    'Title': 'Vice President, Sales',
    'TitleOfCourtesy': 'Dr.',
    'BirthDate': new Date(-563828400000),
    'HireDate': new Date(713764800000),
    'Address': '908 W. Capital Way',
    'City': 'Tacoma',
    'Region': 'WA',
    'PostalCode': '98401',
    'Country': 'USA',
    'HomePhone': '(206) 555-9482',
    'Extension': '3457',
    'Photo': { 'Length': 21626 },
    'Notes': '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.',
    'ReportsTo': 0,
    'PhotoPath': 'http://accweb/emmployees/fuller.bmp'
  },
  {
    'EmployeeID': 3,
    'LastName': 'Leverling',
    'FirstName': 'Janet',
    'Title': 'Sales Representative',
    'TitleOfCourtesy': 'Ms.',
    'BirthDate': new Date(-200088000000),
    'HireDate': new Date(702104400000),
    'Address': '722 Moss Bay Blvd.',
    'City': 'Kirkland',
    'Region': 'WA',
    'PostalCode': '98033',
    'Country': 'USA',
    'HomePhone': '(206) 555-3412',
    'Extension': '3355',
    'Photo': { 'Length': 21722 },
    'Notes': '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.',

```

```

      'ReportsTo': 2,
      'PhotoPath': 'http://accweb/emmployees/leverling.bmp'
    },
    {
      'EmployeeID': 4,
      'LastName': 'Peacock',
      'FirstName': 'Margaret',
      'Title': 'Sales Representative',
      'TitleOfCourtesy': 'Mrs.',
      'BirthDate': new Date(-1018814400000),
      'HireDate': new Date(736401600000),
      'Address': '4110 Old Redmond Rd.',
      'City': 'Redmond',
      'Region': 'WA',
      'PostalCode': '98052',
      'Country': 'USA',
      'HomePhone': '(206) 555-8122',
      'Extension': '5176',
      'Photo': { 'Length': 21626 },
      'Notes': '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.',
      'ReportsTo': 2,
      'PhotoPath': 'http://accweb/emmployees/peacock.bmp'
    },
    {
      'EmployeeID': 5,
      'LastName': 'Buchanan',
      'FirstName': 'Steven',
      'Title': 'Sales Manager',
      'TitleOfCourtesy': 'Mr.',
      'BirthDate': new Date(-468010800000),
      'HireDate': new Date(750830400000),
      'Address': '14 Garrett Hill',
      'City': 'London',
      'Region': null,
      'PostalCode': 'SW1 8JR',
      'Country': 'UK',
      'HomePhone': '(71) 555-4848',
      'Extension': '3453',
      'Photo': { 'Length': 21626 },
      'Notes': 'Steven Buchanan graduated from St. Andrews University, Scotland, with a BSC degree in 1976. Upon joining the company as \
        a sales representative in 1992, he spent 6 months in an orientation program at the Seattle office and then returned to his permanent \
        post in London. He was promoted to sales manager in March 1993. Mr. Buchanan has completed the courses \'Successful \
        Telemarketing\' and \'International Sales Management.\' He is fluent in French.',
      'ReportsTo': 2,
      'PhotoPath': 'http://accweb/emmployees/buchanan.bmp'
    },
    {
      'EmployeeID': 6,
      'LastName': 'Suyama',
      'FirstName': 'Michael',

```

```

        'Title': 'Sales Representative',
        'TitleOfCourtesy': 'Mr.',
        'BirthDate': new Date(-205185600000),
        'HireDate': new Date(750830400000),
        'Address': 'Coventry House\r\nMiner Rd.',
        'City': 'London',
        'Region': null,
        'PostalCode': 'EC2 7JR',
        'Country': 'UK',
        'HomePhone': '(71) 555-7773',
        'Extension': '428',
        'Photo': { 'Length': 21626 },
        'Notes': 'Michael is a graduate of Sussex University (MA, economics,
1983) and the University of California at Los Angeles \
(MBA, marketing, 1986). He has also taken the courses \'Multi-Cultural
Selling\' and \'Time Management for the Sales Professional.\' \
He is fluent in Japanese and can read and write French, Portuguese, and
Spanish.',
        'ReportsTo': 5,
        'PhotoPath': 'http://accweb/emmployees/davolio.bmp'
    },
    {
        'EmployeeID': 7,
        'LastName': 'King',
        'FirstName': 'Robert',
        'Title': 'Sales Representative',
        'TitleOfCourtesy': 'Mr.',
        'BirthDate': new Date(-302731200000),
        'HireDate': new Date(757486800000),
        'Address': 'Edgeham Hollow\r\nWinchester Way',
        'City': 'London',
        'Region': null,
        'PostalCode': 'RG1 9SP',
        'Country': 'UK',
        'HomePhone': '(71) 555-5598',
        'Extension': '465',
        'Photo': { 'Length': 21626 },
        'Notes': 'Robert King served in the Peace Corps and traveled
extensively before completing his degree in English at the \
University of Michigan in 1992, the year he joined the company. After
completing a course entitled \'Selling in Europe,\' \
he was transferred to the London office in March 1993.',
        'ReportsTo': 5,
        'PhotoPath': 'http://accweb/emmployees/davolio.bmp'
    },
    {
        'EmployeeID': 8,
        'LastName': 'Callahan',
        'FirstName': 'Laura',
        'Title': 'Inside Sales Coordinator',
        'TitleOfCourtesy': 'Ms.',
        'BirthDate': new Date(-377982000000),
        'HireDate': new Date(762843600000),
        'Address': '4726 - 11th Ave. N.E.',
        'City': 'Seattle',
        'Region': 'WA',
        'PostalCode': '98105',

```

```

        'Country': 'USA',
        'HomePhone': '(206) 555-1189',
        'Extension': '2344',
        'Photo': { 'Length': 21626 },
        'Notes': 'Laura received a BA in psychology from the University of
Washington. She has also completed a course in business \
French. She reads and writes French.',
        'ReportsTo': 2,
        'PhotoPath': 'http://accweb/emmployees/davolio.bmp'
    },
    {
        'EmployeeID': 9,
        'LastName': 'Dodsworth',
        'FirstName': 'Anne',
        'Title': 'Sales Representative',
        'TitleOfCourtesy': 'Ms.',
        'BirthDate': new Date(-123966000000),
        'HireDate': new Date(784875600000),
        'Address': '7 Houndstooth Rd.',
        'City': 'London',
        'Region': null,
        'PostalCode': 'WG2 7LT',
        'Country': 'UK',
        'HomePhone': '(71) 555-4444',
        'Extension': '452',
        'Photo': { 'Length': 21626 },
        'Notes': 'Anne has a BA degree in English from St. Lawrence College.
She is fluent in French and German.',
        'ReportsTo': 5,
        'PhotoPath': 'http://accweb/emmployees/davolio.bmp'
    }
    ]];
let stringData = JSON.stringify([
    {
        'OrderID': 10248,
        'CustomerID': 'VINET',
        'OrderDate': '1996-07-04T00:00:00.000Z',
        'ShippedDate': '1996-07-16T00:00:00.000Z',
        'Freight': 32.38,
        'ShipName': 'Vins et alcools Chevalier',
        'ShipAddress': '59 rue de l\'Abbaye',
        'ShipCity': 'Reims',
        'ShipRegion': null,
        'ShipCountry': 'France'
    },
    {
        'OrderID': 10249,
        'CustomerID': 'TOMSP',
        'OrderDate': '1996-07-05T00:00:00.000Z',
        'ShippedDate': '1996-07-10T00:00:00.000Z',
        'Freight': 11.61,
        'ShipName': 'Toms Spezialitäten',
        'ShipAddress': 'Luisenstr. 48',
        'ShipCity': 'Münster',
        'ShipRegion': null,
        'ShipCountry': 'Germany'
    },
    {

```

```

      'OrderID': 10250,
      'CustomerID': 'HANAR',
      'OrderDate': '1996-07-08T00:00:00.000Z',
      'ShippedDate': '1996-07-12T00:00:00.000Z',
      'Freight': 65.83,
      'ShipName': 'Hanari Carnes',
      'ShipAddress': 'Rua do Paço, 67',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10251,
      'CustomerID': 'VICTE',
      'OrderDate': '1996-07-08T00:00:00.000Z',
      'ShippedDate': '1996-07-15T00:00:00.000Z',
      'Freight': 41.34,
      'ShipName': 'Victuailles en stock',
      'ShipAddress': '2, rue du Commerce',
      'ShipCity': 'Lyon',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10252,
      'CustomerID': 'SUPRD',
      'OrderDate': '1996-07-09T00:00:00.000Z',
      'ShippedDate': '1996-07-11T00:00:00.000Z',
      'Freight': 51.3,
      'ShipName': 'Suprêmes délices',
      'ShipAddress': 'Boulevard Tirou, 255',
      'ShipCity': 'Charleroi',
      'ShipRegion': null,
      'ShipCountry': 'Belgium'
    },
    {
      'OrderID': 10253,
      'CustomerID': 'HANAR',
      'OrderDate': '1996-07-10T00:00:00.000Z',
      'ShippedDate': '1996-07-16T00:00:00.000Z',
      'Freight': 58.17,
      'ShipName': 'Hanari Carnes',
      'ShipAddress': 'Rua do Paço, 67',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10254,
      'CustomerID': 'CHOPS',
      'OrderDate': '1996-07-11T00:00:00.000Z',
      'ShippedDate': '1996-07-23T00:00:00.000Z',
      'Freight': 22.98,
      'ShipName': 'Chop-suey Chinese',
      'ShipAddress': 'Hauptstr. 31',
      'ShipCity': 'Bern',
      'ShipRegion': null,

```

```

    'ShipCountry': 'Switzerland'
  },
  {
    'OrderID': 10255,
    'CustomerID': 'RICSU',
    'OrderDate': '1996-07-12T00:00:00.000Z',
    'ShippedDate': '1996-07-15T00:00:00.000Z',
    'Freight': 148.33,
    'ShipName': 'Richter Supermarkt',
    'ShipAddress': 'Starenweg 5',
    'ShipCity': 'Genève',
    'ShipRegion': null,
    'ShipCountry': 'Switzerland'
  },
  {
    'OrderID': 10256,
    'CustomerID': 'WELLI',
    'OrderDate': '1996-07-15T00:00:00.000Z',
    'ShippedDate': '1996-07-17T00:00:00.000Z',
    'Freight': 13.97,
    'ShipName': 'Wellington Importadora',
    'ShipAddress': 'Rua do Mercado, 12',
    'ShipCity': 'Resende',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10257,
    'CustomerID': 'HILAA',
    'OrderDate': '1996-07-16T00:00:00.000Z',
    'ShippedDate': '1996-07-22T00:00:00.000Z',
    'Freight': 81.91,
    'ShipName': 'HILARION-Abastos',
    'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
    'ShipCity': 'San Cristóbal',
    'ShipRegion': 'Táchira',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10258,
    'CustomerID': 'ERNSH',
    'OrderDate': '1996-07-17T00:00:00.000Z',
    'ShippedDate': '1996-07-23T00:00:00.000Z',
    'Freight': 140.51,
    'ShipName': 'Ernst Handel',
    'ShipAddress': 'Kirchgasse 6',
    'ShipCity': 'Graz',
    'ShipRegion': null,
    'ShipCountry': 'Austria'
  },
  {
    'OrderID': 10259,
    'CustomerID': 'CENTC',
    'OrderDate': '1996-07-18T00:00:00.000Z',
    'ShippedDate': '1996-07-25T00:00:00.000Z',
    'Freight': 3.25,
    'ShipName': 'Centro comercial Moctezuma',

```

```

      'ShipAddress': 'Sierras de Granada 9993',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10260,
      'CustomerID': 'OTTIK',
      'OrderDate': '1996-07-19T00:00:00.000Z',
      'ShippedDate': '1996-07-29T00:00:00.000Z',
      'Freight': 55.09,
      'ShipName': 'Ottilies Käseladen',
      'ShipAddress': 'Mehrheimerstr. 369',
      'ShipCity': 'Köln',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10261,
      'CustomerID': 'QUEDE',
      'OrderDate': '1996-07-19T00:00:00.000Z',
      'ShippedDate': '1996-07-30T00:00:00.000Z',
      'Freight': 3.05,
      'ShipName': 'Que Delícia',
      'ShipAddress': 'Rua da Panificadora, 12',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10262,
      'CustomerID': 'RATTC',
      'OrderDate': '1996-07-22T00:00:00.000Z',
      'ShippedDate': '1996-07-25T00:00:00.000Z',
      'Freight': 48.29,
      'ShipName': 'Rattlesnake Canyon Grocery',
      'ShipAddress': '2817 Milton Dr.',
      'ShipCity': 'Albuquerque',
      'ShipRegion': 'NM',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10263,
      'CustomerID': 'ERNSH',
      'OrderDate': '1996-07-23T00:00:00.000Z',
      'ShippedDate': '1996-07-31T00:00:00.000Z',
      'Freight': 146.06,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10264,
      'CustomerID': 'FOLKO',
      'OrderDate': '1996-07-24T00:00:00.000Z',

```

```

      'ShippedDate': '1996-08-23T00:00:00.000Z',
      'Freight': 3.67,
      'ShipName': 'Folk och fä HB',
      'ShipAddress': 'Åkergatan 24',
      'ShipCity': 'Bräcke',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 10265,
      'CustomerID': 'BLONP',
      'OrderDate': '1996-07-25T00:00:00.000Z',
      'ShippedDate': '1996-08-12T00:00:00.000Z',
      'Freight': 55.28,
      'ShipName': 'Blondel père et fils',
      'ShipAddress': '24, place Kléber',
      'ShipCity': 'Strasbourg',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10266,
      'CustomerID': 'WARTH',
      'OrderDate': '1996-07-26T00:00:00.000Z',
      'ShippedDate': '1996-07-31T00:00:00.000Z',
      'Freight': 25.73,
      'ShipName': 'Wartian Herkku',
      'ShipAddress': 'Torikatu 38',
      'ShipCity': 'Oulu',
      'ShipRegion': null,
      'ShipCountry': 'Finland'
    },
    {
      'OrderID': 10267,
      'CustomerID': 'FRANK',
      'OrderDate': '1996-07-29T00:00:00.000Z',
      'ShippedDate': '1996-08-06T00:00:00.000Z',
      'Freight': 208.58,
      'ShipName': 'Frankenversand',
      'ShipAddress': 'Berliner Platz 43',
      'ShipCity': 'München',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10268,
      'CustomerID': 'GROSR',
      'OrderDate': '1996-07-30T00:00:00.000Z',
      'ShippedDate': '1996-08-02T00:00:00.000Z',
      'Freight': 66.29,
      'ShipName': 'GROSELLA-Restaurante',
      'ShipAddress': '5a Ave. Los Palos Grandes',
      'ShipCity': 'Caracas',
      'ShipRegion': 'DF',
      'ShipCountry': 'Venezuela'
    }
  ],
  {

```



```

      'OrderID': 10269,
      'CustomerID': 'WHITC',
      'OrderDate': '1996-07-31T00:00:00.000Z',
      'ShippedDate': '1996-08-09T00:00:00.000Z',
      'Freight': 4.56,
      'ShipName': 'White Clover Markets',
      'ShipAddress': '1029 - 12th Ave. S.',
      'ShipCity': 'Seattle',
      'ShipRegion': 'WA',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10270,
      'CustomerID': 'WARTH',
      'OrderDate': '1996-08-01T00:00:00.000Z',
      'ShippedDate': '1996-08-02T00:00:00.000Z',
      'Freight': 136.54,
      'ShipName': 'Wartian Herkku',
      'ShipAddress': 'Torikatu 38',
      'ShipCity': 'Oulu',
      'ShipRegion': null,
      'ShipCountry': 'Finland'
    },
    {
      'OrderID': 10271,
      'CustomerID': 'SPLIR',
      'OrderDate': '1996-08-01T00:00:00.000Z',
      'ShippedDate': '1996-08-30T00:00:00.000Z',
      'Freight': 4.54,
      'ShipName': 'Split Rail Beer & Ale',
      'ShipAddress': 'P.O. Box 555',
      'ShipCity': 'Lander',
      'ShipRegion': 'WY',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10272,
      'CustomerID': 'RATTC',
      'OrderDate': '1996-08-02T00:00:00.000Z',
      'ShippedDate': '1996-08-06T00:00:00.000Z',
      'Freight': 98.03,
      'ShipName': 'Rattlesnake Canyon Grocery',
      'ShipAddress': '2817 Milton Dr.',
      'ShipCity': 'Albuquerque',
      'ShipRegion': 'NM',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10273,
      'CustomerID': 'QUICK',
      'OrderDate': '1996-08-05T00:00:00.000Z',
      'ShippedDate': '1996-08-12T00:00:00.000Z',
      'Freight': 76.07,
      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,

```

```

    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10274,
    'CustomerID': 'VINET',
    'OrderDate': '1996-08-06T00:00:00.000Z',
    'ShippedDate': '1996-08-16T00:00:00.000Z',
    'Freight': 6.01,
    'ShipName': 'Vins et alcools Chevalier',
    'ShipAddress': '59 rue de l\'Abbaye',
    'ShipCity': 'Reims',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10275,
    'CustomerID': 'MAGAA',
    'OrderDate': '1996-08-07T00:00:00.000Z',
    'ShippedDate': '1996-08-09T00:00:00.000Z',
    'Freight': 26.93,
    'ShipName': 'Magazzini Alimentari Riuniti',
    'ShipAddress': 'Via Ludovico il Moro 22',
    'ShipCity': 'Bergamo',
    'ShipRegion': null,
    'ShipCountry': 'Italy'
  },
  {
    'OrderID': 10276,
    'CustomerID': 'TORTU',
    'OrderDate': '1996-08-08T00:00:00.000Z',
    'ShippedDate': '1996-08-14T00:00:00.000Z',
    'Freight': 13.84,
    'ShipName': 'Tortuga Restaurante',
    'ShipAddress': 'Avda. Azteca 123',
    'ShipCity': 'México D.F.',
    'ShipRegion': null,
    'ShipCountry': 'Mexico'
  },
  {
    'OrderID': 10277,
    'CustomerID': 'MORGK',
    'OrderDate': '1996-08-09T00:00:00.000Z',
    'ShippedDate': '1996-08-13T00:00:00.000Z',
    'Freight': 125.77,
    'ShipName': 'Morgenstern Gesundkost',
    'ShipAddress': 'Heerstr. 22',
    'ShipCity': 'Leipzig',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10278,
    'CustomerID': 'BERGS',
    'OrderDate': '1996-08-12T00:00:00.000Z',
    'ShippedDate': '1996-08-16T00:00:00.000Z',
    'Freight': 92.69,
    'ShipName': 'Bergrlunds snabbköp',

```

```

      'ShipAddress': 'Berguvsvägen 8',
      'ShipCity': 'Luleå',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 10279,
      'CustomerID': 'LEHMS',
      'OrderDate': '1996-08-13T00:00:00.000Z',
      'ShippedDate': '1996-08-16T00:00:00.000Z',
      'Freight': 25.83,
      'ShipName': 'Lehmanns Marktstand',
      'ShipAddress': 'Magazinweg 7',
      'ShipCity': 'Frankfurt a.M.',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10280,
      'CustomerID': 'BERGS',
      'OrderDate': '1996-08-14T00:00:00.000Z',
      'ShippedDate': '1996-09-12T00:00:00.000Z',
      'Freight': 8.98,
      'ShipName': 'Berglunds snabbköp',
      'ShipAddress': 'Berguvsvägen 8',
      'ShipCity': 'Luleå',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 10281,
      'CustomerID': 'ROMEY',
      'OrderDate': '1996-08-14T00:00:00.000Z',
      'ShippedDate': '1996-08-21T00:00:00.000Z',
      'Freight': 2.94,
      'ShipName': 'Romero y tomillo',
      'ShipAddress': 'Gran Vía, 1',
      'ShipCity': 'Madrid',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    },
    {
      'OrderID': 10282,
      'CustomerID': 'ROMEY',
      'OrderDate': '1996-08-15T00:00:00.000Z',
      'ShippedDate': '1996-08-21T00:00:00.000Z',
      'Freight': 12.69,
      'ShipName': 'Romero y tomillo',
      'ShipAddress': 'Gran Vía, 1',
      'ShipCity': 'Madrid',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    },
    {
      'OrderID': 10283,
      'CustomerID': 'LILAS',
      'OrderDate': '1996-08-16T00:00:00.000Z',

```

```

      'ShippedDate': '1996-08-23T00:00:00.000Z',
      'Freight': 84.81,
      'ShipName': 'LILA-Supermercado',
      'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
      'ShipCity': 'Barquisimeto',
      'ShipRegion': 'Lara',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10284,
      'CustomerID': 'LEHMS',
      'OrderDate': '1996-08-19T00:00:00.000Z',
      'ShippedDate': '1996-08-27T00:00:00.000Z',
      'Freight': 76.56,
      'ShipName': 'Lehmanns Marktstand',
      'ShipAddress': 'Magazinweg 7',
      'ShipCity': 'Frankfurt a.M.',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10285,
      'CustomerID': 'QUICK',
      'OrderDate': '1996-08-20T00:00:00.000Z',
      'ShippedDate': '1996-08-26T00:00:00.000Z',
      'Freight': 76.83,
      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10286,
      'CustomerID': 'QUICK',
      'OrderDate': '1996-08-21T00:00:00.000Z',
      'ShippedDate': '1996-08-30T00:00:00.000Z',
      'Freight': 229.24,
      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10287,
      'CustomerID': 'RICAR',
      'OrderDate': '1996-08-22T00:00:00.000Z',
      'ShippedDate': '1996-08-28T00:00:00.000Z',
      'Freight': 12.76,
      'ShipName': 'Ricardo Adocicados',
      'ShipAddress': 'Av. Copacabana, 267',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    }
  ],
  {

```

```

      'OrderID': 10288,
      'CustomerID': 'REGGC',
      'OrderDate': '1996-08-23T00:00:00.000Z',
      'ShippedDate': '1996-09-03T00:00:00.000Z',
      'Freight': 7.45,
      'ShipName': 'Reggiani Caseifici',
      'ShipAddress': 'Strada Provinciale 124',
      'ShipCity': 'Reggio Emilia',
      'ShipRegion': null,
      'ShipCountry': 'Italy'
    },
    {
      'OrderID': 10289,
      'CustomerID': 'BSBEV',
      'OrderDate': '1996-08-26T00:00:00.000Z',
      'ShippedDate': '1996-08-28T00:00:00.000Z',
      'Freight': 22.77,
      'ShipName': 'B\ ' Beverages',
      'ShipAddress': 'Fauntleroy Circus',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10290,
      'CustomerID': 'COMMI',
      'OrderDate': '1996-08-27T00:00:00.000Z',
      'ShippedDate': '1996-09-03T00:00:00.000Z',
      'Freight': 79.7,
      'ShipName': 'Comércio Mineiro',
      'ShipAddress': 'Av. dos Lusíadas, 23',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10291,
      'CustomerID': 'QUEDE',
      'OrderDate': '1996-08-27T00:00:00.000Z',
      'ShippedDate': '1996-09-04T00:00:00.000Z',
      'Freight': 6.4,
      'ShipName': 'Que Delícia',
      'ShipAddress': 'Rua da Panificadora, 12',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10292,
      'CustomerID': 'TRADH',
      'OrderDate': '1996-08-28T00:00:00.000Z',
      'ShippedDate': '1996-09-02T00:00:00.000Z',
      'Freight': 1.35,
      'ShipName': 'Tradição Hipermercados',
      'ShipAddress': 'Av. Inês de Castro, 414',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',

```

```

    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10293,
    'CustomerID': 'TORTU',
    'OrderDate': '1996-08-29T00:00:00.000Z',
    'ShippedDate': '1996-09-11T00:00:00.000Z',
    'Freight': 21.18,
    'ShipName': 'Tortuga Restaurante',
    'ShipAddress': 'Avda. Azteca 123',
    'ShipCity': 'México D.F.',
    'ShipRegion': null,
    'ShipCountry': 'Mexico'
  },
  {
    'OrderID': 10294,
    'CustomerID': 'RATTC',
    'OrderDate': '1996-08-30T00:00:00.000Z',
    'ShippedDate': '1996-09-05T00:00:00.000Z',
    'Freight': 147.26,
    'ShipName': 'Rattlesnake Canyon Grocery',
    'ShipAddress': '2817 Milton Dr.',
    'ShipCity': 'Albuquerque',
    'ShipRegion': 'NM',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10295,
    'CustomerID': 'VINET',
    'OrderDate': '1996-09-02T00:00:00.000Z',
    'ShippedDate': '1996-09-10T00:00:00.000Z',
    'Freight': 1.15,
    'ShipName': 'Vins et alcools Chevalier',
    'ShipAddress': '59 rue de l\'Abbaye',
    'ShipCity': 'Reims',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10296,
    'CustomerID': 'LILAS',
    'OrderDate': '1996-09-03T00:00:00.000Z',
    'ShippedDate': '1996-09-11T00:00:00.000Z',
    'Freight': 0.12,
    'ShipName': 'LILA-Supermercado',
    'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
    'ShipCity': 'Barquisimeto',
    'ShipRegion': 'Lara',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10297,
    'CustomerID': 'BLONP',
    'OrderDate': '1996-09-04T00:00:00.000Z',
    'ShippedDate': '1996-09-10T00:00:00.000Z',
    'Freight': 5.74,
    'ShipName': 'Blondel père et fils',

```

```

      'ShipAddress': '24, place Kléber',
      'ShipCity': 'Strasbourg',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10298,
      'CustomerID': 'HUNGO',
      'OrderDate': '1996-09-05T00:00:00.000Z',
      'ShippedDate': '1996-09-11T00:00:00.000Z',
      'Freight': 168.22,
      'ShipName': 'Hungry Owl All-Night Grocers',
      'ShipAddress': '8 Johnstown Road',
      'ShipCity': 'Cork',
      'ShipRegion': 'Co. Cork',
      'ShipCountry': 'Ireland'
    },
    {
      'OrderID': 10299,
      'CustomerID': 'RICAR',
      'OrderDate': '1996-09-06T00:00:00.000Z',
      'ShippedDate': '1996-09-13T00:00:00.000Z',
      'Freight': 29.76,
      'ShipName': 'Ricardo Adocicados',
      'ShipAddress': 'Av. Copacabana, 267',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10300,
      'CustomerID': 'MAGAA',
      'OrderDate': '1996-09-09T00:00:00.000Z',
      'ShippedDate': '1996-09-18T00:00:00.000Z',
      'Freight': 17.68,
      'ShipName': 'Magazzini Alimentari Riuniti',
      'ShipAddress': 'Via Ludovico il Moro 22',
      'ShipCity': 'Bergamo',
      'ShipRegion': null,
      'ShipCountry': 'Italy'
    },
    {
      'OrderID': 10301,
      'CustomerID': 'WANDK',
      'OrderDate': '1996-09-09T00:00:00.000Z',
      'ShippedDate': '1996-09-17T00:00:00.000Z',
      'Freight': 45.08,
      'ShipName': 'Die Wandernde Kuh',
      'ShipAddress': 'Adenauerallee 900',
      'ShipCity': 'Stuttgart',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10302,
      'CustomerID': 'SUPRD',
      'OrderDate': '1996-09-10T00:00:00.000Z',

```

```

    'ShippedDate': '1996-10-09T00:00:00.000Z',
    'Freight': 6.27,
    'ShipName': 'Suprêmes délices',
    'ShipAddress': 'Boulevard Tirou, 255',
    'ShipCity': 'Charleroi',
    'ShipRegion': null,
    'ShipCountry': 'Belgium'
  },
  {
    'OrderID': 10303,
    'CustomerID': 'GODOS',
    'OrderDate': '1996-09-11T00:00:00.000Z',
    'ShippedDate': '1996-09-18T00:00:00.000Z',
    'Freight': 107.83,
    'ShipName': 'Godos Cocina Típica',
    'ShipAddress': 'C/ Romero, 33',
    'ShipCity': 'Sevilla',
    'ShipRegion': null,
    'ShipCountry': 'Spain'
  },
  {
    'OrderID': 10304,
    'CustomerID': 'TORTU',
    'OrderDate': '1996-09-12T00:00:00.000Z',
    'ShippedDate': '1996-09-17T00:00:00.000Z',
    'Freight': 63.79,
    'ShipName': 'Tortuga Restaurante',
    'ShipAddress': 'Avda. Azteca 123',
    'ShipCity': 'México D.F.',
    'ShipRegion': null,
    'ShipCountry': 'Mexico'
  },
  {
    'OrderID': 10305,
    'CustomerID': 'OLDWO',
    'OrderDate': '1996-09-13T00:00:00.000Z',
    'ShippedDate': '1996-10-09T00:00:00.000Z',
    'Freight': 257.62,
    'ShipName': 'Old World Delicatessen',
    'ShipAddress': '2743 Bering St.',
    'ShipCity': 'Anchorage',
    'ShipRegion': 'AK',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10306,
    'CustomerID': 'ROMEY',
    'OrderDate': '1996-09-16T00:00:00.000Z',
    'ShippedDate': '1996-09-23T00:00:00.000Z',
    'Freight': 7.56,
    'ShipName': 'Romero y tomillo',
    'ShipAddress': 'Gran Vía, 1',
    'ShipCity': 'Madrid',
    'ShipRegion': null,
    'ShipCountry': 'Spain'
  },
  {

```



```

    'OrderID': 10307,
    'CustomerID': 'LONEP',
    'OrderDate': '1996-09-17T00:00:00.000Z',
    'ShippedDate': '1996-09-25T00:00:00.000Z',
    'Freight': 0.56,
    'ShipName': 'Lonesome Pine Restaurant',
    'ShipAddress': '89 Chiaroscuro Rd.',
    'ShipCity': 'Portland',
    'ShipRegion': 'OR',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10308,
    'CustomerID': 'ANATR',
    'OrderDate': '1996-09-18T00:00:00.000Z',
    'ShippedDate': '1996-09-24T00:00:00.000Z',
    'Freight': 1.61,
    'ShipName': 'Ana Trujillo Emparedados y helados',
    'ShipAddress': 'Avda. de la Constitución 2222',
    'ShipCity': 'México D.F.',
    'ShipRegion': null,
    'ShipCountry': 'Mexico'
  },
  {
    'OrderID': 10309,
    'CustomerID': 'HUNGO',
    'OrderDate': '1996-09-19T00:00:00.000Z',
    'ShippedDate': '1996-10-23T00:00:00.000Z',
    'Freight': 47.3,
    'ShipName': 'Hungry Owl All-Night Grocers',
    'ShipAddress': '8 Johnstown Road',
    'ShipCity': 'Cork',
    'ShipRegion': 'Co. Cork',
    'ShipCountry': 'Ireland'
  },
  {
    'OrderID': 10310,
    'CustomerID': 'THEBI',
    'OrderDate': '1996-09-20T00:00:00.000Z',
    'ShippedDate': '1996-09-27T00:00:00.000Z',
    'Freight': 17.52,
    'ShipName': 'The Big Cheese',
    'ShipAddress': '89 Jefferson Way Suite 2',
    'ShipCity': 'Portland',
    'ShipRegion': 'OR',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10311,
    'CustomerID': 'DUMON',
    'OrderDate': '1996-09-20T00:00:00.000Z',
    'ShippedDate': '1996-09-26T00:00:00.000Z',
    'Freight': 24.69,
    'ShipName': 'Du monde entier',
    'ShipAddress': '67, rue des Cinquante Otages',
    'ShipCity': 'Nantes',
    'ShipRegion': null,

```

```

    'ShipCountry': 'France'
  },
  {
    'OrderID': 10312,
    'CustomerID': 'WANDK',
    'OrderDate': '1996-09-23T00:00:00.000Z',
    'ShippedDate': '1996-10-03T00:00:00.000Z',
    'Freight': 40.26,
    'ShipName': 'Die Wandernde Kuh',
    'ShipAddress': 'Adenauerallee 900',
    'ShipCity': 'Stuttgart',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10313,
    'CustomerID': 'QUICK',
    'OrderDate': '1996-09-24T00:00:00.000Z',
    'ShippedDate': '1996-10-04T00:00:00.000Z',
    'Freight': 1.96,
    'ShipName': 'QUICK-Stop',
    'ShipAddress': 'Taucherstraße 10',
    'ShipCity': 'Cunewalde',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10314,
    'CustomerID': 'RATTC',
    'OrderDate': '1996-09-25T00:00:00.000Z',
    'ShippedDate': '1996-10-04T00:00:00.000Z',
    'Freight': 74.16,
    'ShipName': 'Rattlesnake Canyon Grocery',
    'ShipAddress': '2817 Milton Dr.',
    'ShipCity': 'Albuquerque',
    'ShipRegion': 'NM',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10315,
    'CustomerID': 'ISLAT',
    'OrderDate': '1996-09-26T00:00:00.000Z',
    'ShippedDate': '1996-10-03T00:00:00.000Z',
    'Freight': 41.76,
    'ShipName': 'Island Trading',
    'ShipAddress': 'Garden House Crowther Way',
    'ShipCity': 'Cowes',
    'ShipRegion': 'Isle of Wight',
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10316,
    'CustomerID': 'RATTC',
    'OrderDate': '1996-09-27T00:00:00.000Z',
    'ShippedDate': '1996-10-08T00:00:00.000Z',
    'Freight': 150.15,
    'ShipName': 'Rattlesnake Canyon Grocery',

```

```

      'ShipAddress': '2817 Milton Dr.',
      'ShipCity': 'Albuquerque',
      'ShipRegion': 'NM',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10317,
      'CustomerID': 'LONEP',
      'OrderDate': '1996-09-30T00:00:00.000Z',
      'ShippedDate': '1996-10-10T00:00:00.000Z',
      'Freight': 12.69,
      'ShipName': 'Lonesome Pine Restaurant',
      'ShipAddress': '89 Chiaroscuro Rd.',
      'ShipCity': 'Portland',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10318,
      'CustomerID': 'ISLAT',
      'OrderDate': '1996-10-01T00:00:00.000Z',
      'ShippedDate': '1996-10-04T00:00:00.000Z',
      'Freight': 4.73,
      'ShipName': 'Island Trading',
      'ShipAddress': 'Garden House Crowther Way',
      'ShipCity': 'Cowes',
      'ShipRegion': 'Isle of Wight',
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10319,
      'CustomerID': 'TORTU',
      'OrderDate': '1996-10-02T00:00:00.000Z',
      'ShippedDate': '1996-10-11T00:00:00.000Z',
      'Freight': 64.5,
      'ShipName': 'Tortuga Restaurante',
      'ShipAddress': 'Avda. Azteca 123',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10320,
      'CustomerID': 'WARTH',
      'OrderDate': '1996-10-03T00:00:00.000Z',
      'ShippedDate': '1996-10-18T00:00:00.000Z',
      'Freight': 34.57,
      'ShipName': 'Wartian Herkku',
      'ShipAddress': 'Torikatu 38',
      'ShipCity': 'Oulu',
      'ShipRegion': null,
      'ShipCountry': 'Finland'
    },
    {
      'OrderID': 10321,
      'CustomerID': 'ISLAT',
      'OrderDate': '1996-10-03T00:00:00.000Z',

```

```

      'ShippedDate': '1996-10-11T00:00:00.000Z',
      'Freight': 3.43,
      'ShipName': 'Island Trading',
      'ShipAddress': 'Garden House Crowther Way',
      'ShipCity': 'Cowes',
      'ShipRegion': 'Isle of Wight',
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10322,
      'CustomerID': 'PERIC',
      'OrderDate': '1996-10-04T00:00:00.000Z',
      'ShippedDate': '1996-10-23T00:00:00.000Z',
      'Freight': 0.4,
      'ShipName': 'Pericles Comidas clásicas',
      'ShipAddress': 'Calle Dr. Jorge Cash 321',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10323,
      'CustomerID': 'KOENE',
      'OrderDate': '1996-10-07T00:00:00.000Z',
      'ShippedDate': '1996-10-14T00:00:00.000Z',
      'Freight': 4.88,
      'ShipName': 'Königlich Essen',
      'ShipAddress': 'Maubelstr. 90',
      'ShipCity': 'Brandenburg',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10324,
      'CustomerID': 'SAVEA',
      'OrderDate': '1996-10-08T00:00:00.000Z',
      'ShippedDate': '1996-10-10T00:00:00.000Z',
      'Freight': 214.27,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10325,
      'CustomerID': 'KOENE',
      'OrderDate': '1996-10-09T00:00:00.000Z',
      'ShippedDate': '1996-10-14T00:00:00.000Z',
      'Freight': 64.86,
      'ShipName': 'Königlich Essen',
      'ShipAddress': 'Maubelstr. 90',
      'ShipCity': 'Brandenburg',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    }
  ],
  {

```

```

      'OrderID': 10326,
      'CustomerID': 'BOLID',
      'OrderDate': '1996-10-10T00:00:00.000Z',
      'ShippedDate': '1996-10-14T00:00:00.000Z',
      'Freight': 77.92,
      'ShipName': 'Bólido Comidas preparadas',
      'ShipAddress': 'C/ Araquil, 67',
      'ShipCity': 'Madrid',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    },
    {
      'OrderID': 10327,
      'CustomerID': 'FOLKO',
      'OrderDate': '1996-10-11T00:00:00.000Z',
      'ShippedDate': '1996-10-14T00:00:00.000Z',
      'Freight': 63.36,
      'ShipName': 'Folk och fä HB',
      'ShipAddress': 'Åkergatan 24',
      'ShipCity': 'Bräcke',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 10328,
      'CustomerID': 'FURIB',
      'OrderDate': '1996-10-14T00:00:00.000Z',
      'ShippedDate': '1996-10-17T00:00:00.000Z',
      'Freight': 87.03,
      'ShipName': 'Furia Bacalhau e Frutos do Mar',
      'ShipAddress': 'Jardim das rosas n. 32',
      'ShipCity': 'Lisboa',
      'ShipRegion': null,
      'ShipCountry': 'Portugal'
    },
    {
      'OrderID': 10329,
      'CustomerID': 'SPLIR',
      'OrderDate': '1996-10-15T00:00:00.000Z',
      'ShippedDate': '1996-10-23T00:00:00.000Z',
      'Freight': 191.67,
      'ShipName': 'Split Rail Beer & Ale',
      'ShipAddress': 'P.O. Box 555',
      'ShipCity': 'Lander',
      'ShipRegion': 'WY',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10330,
      'CustomerID': 'LILAS',
      'OrderDate': '1996-10-16T00:00:00.000Z',
      'ShippedDate': '1996-10-28T00:00:00.000Z',
      'Freight': 12.75,
      'ShipName': 'LILA-Supermercado',
      'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
      'ShipCity': 'Barquisimeto',
      'ShipRegion': 'Lara',

```

```

    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10331,
    'CustomerID': 'BONAP',
    'OrderDate': '1996-10-16T00:00:00.000Z',
    'ShippedDate': '1996-10-21T00:00:00.000Z',
    'Freight': 10.19,
    'ShipName': 'Bon app',
    'ShipAddress': '12, rue des Bouchers',
    'ShipCity': 'Marseille',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10332,
    'CustomerID': 'MEREP',
    'OrderDate': '1996-10-17T00:00:00.000Z',
    'ShippedDate': '1996-10-21T00:00:00.000Z',
    'Freight': 52.84,
    'ShipName': 'Mère Paillarde',
    'ShipAddress': '43 rue St. Laurent',
    'ShipCity': 'Montréal',
    'ShipRegion': 'Québec',
    'ShipCountry': 'Canada'
  },
  {
    'OrderID': 10333,
    'CustomerID': 'WARTH',
    'OrderDate': '1996-10-18T00:00:00.000Z',
    'ShippedDate': '1996-10-25T00:00:00.000Z',
    'Freight': 0.59,
    'ShipName': 'Wartian Herkku',
    'ShipAddress': 'Torikatu 38',
    'ShipCity': 'Oulu',
    'ShipRegion': null,
    'ShipCountry': 'Finland'
  },
  {
    'OrderID': 10334,
    'CustomerID': 'VICTE',
    'OrderDate': '1996-10-21T00:00:00.000Z',
    'ShippedDate': '1996-10-28T00:00:00.000Z',
    'Freight': 8.56,
    'ShipName': 'Victuailles en stock',
    'ShipAddress': '2, rue du Commerce',
    'ShipCity': 'Lyon',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10335,
    'CustomerID': 'HUNGO',
    'OrderDate': '1996-10-22T00:00:00.000Z',
    'ShippedDate': '1996-10-24T00:00:00.000Z',
    'Freight': 42.11,
    'ShipName': 'Hungry Owl All-Night Grocers',

```

```

      'ShipAddress': '8 Johnstown Road',
      'ShipCity': 'Cork',
      'ShipRegion': 'Co. Cork',
      'ShipCountry': 'Ireland'
    },
    {
      'OrderID': 10336,
      'CustomerID': 'PRINI',
      'OrderDate': '1996-10-23T00:00:00.000Z',
      'ShippedDate': '1996-10-25T00:00:00.000Z',
      'Freight': 15.51,
      'ShipName': 'Princesa Isabel Vinhos',
      'ShipAddress': 'Estrada da saúde n. 58',
      'ShipCity': 'Lisboa',
      'ShipRegion': null,
      'ShipCountry': 'Portugal'
    },
    {
      'OrderID': 10337,
      'CustomerID': 'FRANK',
      'OrderDate': '1996-10-24T00:00:00.000Z',
      'ShippedDate': '1996-10-29T00:00:00.000Z',
      'Freight': 108.26,
      'ShipName': 'Frankenversand',
      'ShipAddress': 'Berliner Platz 43',
      'ShipCity': 'München',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10338,
      'CustomerID': 'OLDWO',
      'OrderDate': '1996-10-25T00:00:00.000Z',
      'ShippedDate': '1996-10-29T00:00:00.000Z',
      'Freight': 84.21,
      'ShipName': 'Old World Delicatessen',
      'ShipAddress': '2743 Bering St.',
      'ShipCity': 'Anchorage',
      'ShipRegion': 'AK',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10339,
      'CustomerID': 'MEREP',
      'OrderDate': '1996-10-28T00:00:00.000Z',
      'ShippedDate': '1996-11-04T00:00:00.000Z',
      'Freight': 15.66,
      'ShipName': 'Mère Paillarde',
      'ShipAddress': '43 rue St. Laurent',
      'ShipCity': 'Montréal',
      'ShipRegion': 'Québec',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10340,
      'CustomerID': 'BONAP',
      'OrderDate': '1996-10-29T00:00:00.000Z',

```

```

    'ShippedDate': '1996-11-08T00:00:00.000Z',
    'Freight': 166.31,
    'ShipName': 'Bon app',
    'ShipAddress': '12, rue des Bouchers',
    'ShipCity': 'Marseille',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10341,
    'CustomerID': 'SIMOB',
    'OrderDate': '1996-10-29T00:00:00.000Z',
    'ShippedDate': '1996-11-05T00:00:00.000Z',
    'Freight': 26.78,
    'ShipName': 'Simons bistro',
    'ShipAddress': 'Vinbæltet 34',
    'ShipCity': 'Kobenhavn',
    'ShipRegion': null,
    'ShipCountry': 'Denmark'
  },
  {
    'OrderID': 10342,
    'CustomerID': 'FRANK',
    'OrderDate': '1996-10-30T00:00:00.000Z',
    'ShippedDate': '1996-11-04T00:00:00.000Z',
    'Freight': 54.83,
    'ShipName': 'Frankenversand',
    'ShipAddress': 'Berliner Platz 43',
    'ShipCity': 'München',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10343,
    'CustomerID': 'LEHMS',
    'OrderDate': '1996-10-31T00:00:00.000Z',
    'ShippedDate': '1996-11-06T00:00:00.000Z',
    'Freight': 110.37,
    'ShipName': 'Lehmanns Marktstand',
    'ShipAddress': 'Magazinweg 7',
    'ShipCity': 'Frankfurt a.M.',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10344,
    'CustomerID': 'WHITC',
    'OrderDate': '1996-11-01T00:00:00.000Z',
    'ShippedDate': '1996-11-05T00:00:00.000Z',
    'Freight': 23.29,
    'ShipName': 'White Clover Markets',
    'ShipAddress': '1029 - 12th Ave. S.',
    'ShipCity': 'Seattle',
    'ShipRegion': 'WA',
    'ShipCountry': 'USA'
  },
  {

```



```

    'OrderID': 10345,
    'CustomerID': 'QUICK',
    'OrderDate': '1996-11-04T00:00:00.000Z',
    'ShippedDate': '1996-11-11T00:00:00.000Z',
    'Freight': 249.06,
    'ShipName': 'QUICK-Stop',
    'ShipAddress': 'Taucherstraße 10',
    'ShipCity': 'Cunewalde',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10346,
    'CustomerID': 'RATTC',
    'OrderDate': '1996-11-05T00:00:00.000Z',
    'ShippedDate': '1996-11-08T00:00:00.000Z',
    'Freight': 142.08,
    'ShipName': 'Rattlesnake Canyon Grocery',
    'ShipAddress': '2817 Milton Dr.',
    'ShipCity': 'Albuquerque',
    'ShipRegion': 'NM',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10347,
    'CustomerID': 'FAMIA',
    'OrderDate': '1996-11-06T00:00:00.000Z',
    'ShippedDate': '1996-11-08T00:00:00.000Z',
    'Freight': 3.1,
    'ShipName': 'Familia Arquibaldo',
    'ShipAddress': 'Rua Orós, 92',
    'ShipCity': 'Sao Paulo',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10348,
    'CustomerID': 'WANDK',
    'OrderDate': '1996-11-07T00:00:00.000Z',
    'ShippedDate': '1996-11-15T00:00:00.000Z',
    'Freight': 0.78,
    'ShipName': 'Die Wandernde Kuh',
    'ShipAddress': 'Adenauerallee 900',
    'ShipCity': 'Stuttgart',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10349,
    'CustomerID': 'SPLIR',
    'OrderDate': '1996-11-08T00:00:00.000Z',
    'ShippedDate': '1996-11-15T00:00:00.000Z',
    'Freight': 8.63,
    'ShipName': 'Split Rail Beer & Ale',
    'ShipAddress': 'P.O. Box 555',
    'ShipCity': 'Lander',
    'ShipRegion': 'WY',
  }

```

```

    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10350,
    'CustomerID': 'LAMAI',
    'OrderDate': '1996-11-11T00:00:00.000Z',
    'ShippedDate': '1996-12-03T00:00:00.000Z',
    'Freight': 64.19,
    'ShipName': 'La maison d\'Asie',
    'ShipAddress': '1 rue Alsace-Lorraine',
    'ShipCity': 'Toulouse',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10351,
    'CustomerID': 'ERNSH',
    'OrderDate': '1996-11-11T00:00:00.000Z',
    'ShippedDate': '1996-11-20T00:00:00.000Z',
    'Freight': 162.33,
    'ShipName': 'Ernst Handel',
    'ShipAddress': 'Kirchgasse 6',
    'ShipCity': 'Graz',
    'ShipRegion': null,
    'ShipCountry': 'Austria'
  },
  {
    'OrderID': 10352,
    'CustomerID': 'FURIB',
    'OrderDate': '1996-11-12T00:00:00.000Z',
    'ShippedDate': '1996-11-18T00:00:00.000Z',
    'Freight': 1.3,
    'ShipName': 'Furia Bacalhau e Frutos do Mar',
    'ShipAddress': 'Jardim das rosas n. 32',
    'ShipCity': 'Lisboa',
    'ShipRegion': null,
    'ShipCountry': 'Portugal'
  },
  {
    'OrderID': 10353,
    'CustomerID': 'PICCO',
    'OrderDate': '1996-11-13T00:00:00.000Z',
    'ShippedDate': '1996-11-25T00:00:00.000Z',
    'Freight': 360.63,
    'ShipName': 'Piccolo und mehr',
    'ShipAddress': 'Geislweg 14',
    'ShipCity': 'Salzburg',
    'ShipRegion': null,
    'ShipCountry': 'Austria'
  },
  {
    'OrderID': 10354,
    'CustomerID': 'PERIC',
    'OrderDate': '1996-11-14T00:00:00.000Z',
    'ShippedDate': '1996-11-20T00:00:00.000Z',
    'Freight': 53.8,
    'ShipName': 'Pericles Comidas clásicas',

```

```

      'ShipAddress': 'Calle Dr. Jorge Cash 321',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10355,
      'CustomerID': 'AROUT',
      'OrderDate': '1996-11-15T00:00:00.000Z',
      'ShippedDate': '1996-11-20T00:00:00.000Z',
      'Freight': 41.95,
      'ShipName': 'Around the Horn',
      'ShipAddress': 'Brook Farm Stratford St. Mary',
      'ShipCity': 'Colchester',
      'ShipRegion': 'Essex',
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10356,
      'CustomerID': 'WANDK',
      'OrderDate': '1996-11-18T00:00:00.000Z',
      'ShippedDate': '1996-11-27T00:00:00.000Z',
      'Freight': 36.71,
      'ShipName': 'Die Wandernde Kuh',
      'ShipAddress': 'Adenauerallee 900',
      'ShipCity': 'Stuttgart',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10357,
      'CustomerID': 'LILAS',
      'OrderDate': '1996-11-19T00:00:00.000Z',
      'ShippedDate': '1996-12-02T00:00:00.000Z',
      'Freight': 34.88,
      'ShipName': 'LILA-Supermercado',
      'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
      'ShipCity': 'Barquisimeto',
      'ShipRegion': 'Lara',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10358,
      'CustomerID': 'LAMAI',
      'OrderDate': '1996-11-20T00:00:00.000Z',
      'ShippedDate': '1996-11-27T00:00:00.000Z',
      'Freight': 19.64,
      'ShipName': 'La maison d\'Asie',
      'ShipAddress': '1 rue Alsace-Lorraine',
      'ShipCity': 'Toulouse',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10359,
      'CustomerID': 'SEVES',
      'OrderDate': '1996-11-21T00:00:00.000Z',

```

```

      'ShippedDate': '1996-11-26T00:00:00.000Z',
      'Freight': 288.43,
      'ShipName': 'Seven Seas Imports',
      'ShipAddress': '90 Wadhurst Rd.',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10360,
      'CustomerID': 'BLONP',
      'OrderDate': '1996-11-22T00:00:00.000Z',
      'ShippedDate': '1996-12-02T00:00:00.000Z',
      'Freight': 131.7,
      'ShipName': 'Blondel père et fils',
      'ShipAddress': '24, place Kléber',
      'ShipCity': 'Strasbourg',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10361,
      'CustomerID': 'QUICK',
      'OrderDate': '1996-11-22T00:00:00.000Z',
      'ShippedDate': '1996-12-03T00:00:00.000Z',
      'Freight': 183.17,
      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10362,
      'CustomerID': 'BONAP',
      'OrderDate': '1996-11-25T00:00:00.000Z',
      'ShippedDate': '1996-11-28T00:00:00.000Z',
      'Freight': 96.04,
      'ShipName': 'Bon app',
      'ShipAddress': '12, rue des Bouchers',
      'ShipCity': 'Marseille',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10363,
      'CustomerID': 'DRACD',
      'OrderDate': '1996-11-26T00:00:00.000Z',
      'ShippedDate': '1996-12-04T00:00:00.000Z',
      'Freight': 30.54,
      'ShipName': 'Drachenblut Delikatessen',
      'ShipAddress': 'Walserweg 21',
      'ShipCity': 'Aachen',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    }
  ],
  {

```

```

      'OrderID': 10364,
      'CustomerID': 'EASTC',
      'OrderDate': '1996-11-26T00:00:00.000Z',
      'ShippedDate': '1996-12-04T00:00:00.000Z',
      'Freight': 71.97,
      'ShipName': 'Eastern Connection',
      'ShipAddress': '35 King George',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10365,
      'CustomerID': 'ANTON',
      'OrderDate': '1996-11-27T00:00:00.000Z',
      'ShippedDate': '1996-12-02T00:00:00.000Z',
      'Freight': 22,
      'ShipName': 'Antonio Moreno Taquería',
      'ShipAddress': 'Mataderos 2312',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10366,
      'CustomerID': 'GALED',
      'OrderDate': '1996-11-28T00:00:00.000Z',
      'ShippedDate': '1996-12-30T00:00:00.000Z',
      'Freight': 10.14,
      'ShipName': 'Galería del gastronómo',
      'ShipAddress': 'Rambla de Cataluña, 23',
      'ShipCity': 'Barcelona',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    },
    {
      'OrderID': 10367,
      'CustomerID': 'VAFFE',
      'OrderDate': '1996-11-28T00:00:00.000Z',
      'ShippedDate': '1996-12-02T00:00:00.000Z',
      'Freight': 13.55,
      'ShipName': 'Vaffeljernet',
      'ShipAddress': 'Smagsloget 45',
      'ShipCity': 'Århus',
      'ShipRegion': null,
      'ShipCountry': 'Denmark'
    },
    {
      'OrderID': 10368,
      'CustomerID': 'ERNSH',
      'OrderDate': '1996-11-29T00:00:00.000Z',
      'ShippedDate': '1996-12-02T00:00:00.000Z',
      'Freight': 101.95,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,

```

```

    'ShipCountry': 'Austria'
  },
  {
    'OrderID': 10369,
    'CustomerID': 'SPLIR',
    'OrderDate': '1996-12-02T00:00:00.000Z',
    'ShippedDate': '1996-12-09T00:00:00.000Z',
    'Freight': 195.68,
    'ShipName': 'Split Rail Beer & Ale',
    'ShipAddress': 'P.O. Box 555',
    'ShipCity': 'Lander',
    'ShipRegion': 'WY',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10370,
    'CustomerID': 'CHOPS',
    'OrderDate': '1996-12-03T00:00:00.000Z',
    'ShippedDate': '1996-12-27T00:00:00.000Z',
    'Freight': 1.17,
    'ShipName': 'Chop-suey Chinese',
    'ShipAddress': 'Hauptstr. 31',
    'ShipCity': 'Bern',
    'ShipRegion': null,
    'ShipCountry': 'Switzerland'
  },
  {
    'OrderID': 10371,
    'CustomerID': 'LAMAI',
    'OrderDate': '1996-12-03T00:00:00.000Z',
    'ShippedDate': '1996-12-24T00:00:00.000Z',
    'Freight': 0.45,
    'ShipName': 'La maison d\'Asie',
    'ShipAddress': '1 rue Alsace-Lorraine',
    'ShipCity': 'Toulouse',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10372,
    'CustomerID': 'QUEEN',
    'OrderDate': '1996-12-04T00:00:00.000Z',
    'ShippedDate': '1996-12-09T00:00:00.000Z',
    'Freight': 890.78,
    'ShipName': 'Queen Cozinha',
    'ShipAddress': 'Alameda dos Canários, 891',
    'ShipCity': 'Sao Paulo',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10373,
    'CustomerID': 'HUNGO',
    'OrderDate': '1996-12-05T00:00:00.000Z',
    'ShippedDate': '1996-12-11T00:00:00.000Z',
    'Freight': 124.12,
    'ShipName': 'Hungry Owl All-Night Grocers',

```

```

      'ShipAddress': '8 Johnstown Road',
      'ShipCity': 'Cork',
      'ShipRegion': 'Co. Cork',
      'ShipCountry': 'Ireland'
    },
    {
      'OrderID': 10374,
      'CustomerID': 'WOLZA',
      'OrderDate': '1996-12-05T00:00:00.000Z',
      'ShippedDate': '1996-12-09T00:00:00.000Z',
      'Freight': 3.94,
      'ShipName': 'Wolski Zajazd',
      'ShipAddress': 'ul. Filtrowa 68',
      'ShipCity': 'Warszawa',
      'ShipRegion': null,
      'ShipCountry': 'Poland'
    },
    {
      'OrderID': 10375,
      'CustomerID': 'HUNGC',
      'OrderDate': '1996-12-06T00:00:00.000Z',
      'ShippedDate': '1996-12-09T00:00:00.000Z',
      'Freight': 20.12,
      'ShipName': 'Hungry Coyote Import Store',
      'ShipAddress': 'City Center Plaza 516 Main St.',
      'ShipCity': 'Elgin',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10376,
      'CustomerID': 'MEREP',
      'OrderDate': '1996-12-09T00:00:00.000Z',
      'ShippedDate': '1996-12-13T00:00:00.000Z',
      'Freight': 20.39,
      'ShipName': 'Mère Paillarde',
      'ShipAddress': '43 rue St. Laurent',
      'ShipCity': 'Montréal',
      'ShipRegion': 'Québec',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10377,
      'CustomerID': 'SEVES',
      'OrderDate': '1996-12-09T00:00:00.000Z',
      'ShippedDate': '1996-12-13T00:00:00.000Z',
      'Freight': 22.21,
      'ShipName': 'Seven Seas Imports',
      'ShipAddress': '90 Wadhurst Rd.',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10378,
      'CustomerID': 'FOLKO',
      'OrderDate': '1996-12-10T00:00:00.000Z',

```

```

    'ShippedDate': '1996-12-19T00:00:00.000Z',
    'Freight': 5.44,
    'ShipName': 'Folk och fä HB',
    'ShipAddress': 'Åkergatan 24',
    'ShipCity': 'Bräcke',
    'ShipRegion': null,
    'ShipCountry': 'Sweden'
  },
  {
    'OrderID': 10379,
    'CustomerID': 'QUEDE',
    'OrderDate': '1996-12-11T00:00:00.000Z',
    'ShippedDate': '1996-12-13T00:00:00.000Z',
    'Freight': 45.03,
    'ShipName': 'Que Delícia',
    'ShipAddress': 'Rua da Panificadora, 12',
    'ShipCity': 'Rio de Janeiro',
    'ShipRegion': 'RJ',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10380,
    'CustomerID': 'HUNGO',
    'OrderDate': '1996-12-12T00:00:00.000Z',
    'ShippedDate': '1997-01-16T00:00:00.000Z',
    'Freight': 35.03,
    'ShipName': 'Hungry Owl All-Night Grocers',
    'ShipAddress': '8 Johnstown Road',
    'ShipCity': 'Cork',
    'ShipRegion': 'Co. Cork',
    'ShipCountry': 'Ireland'
  },
  {
    'OrderID': 10381,
    'CustomerID': 'LILAS',
    'OrderDate': '1996-12-12T00:00:00.000Z',
    'ShippedDate': '1996-12-13T00:00:00.000Z',
    'Freight': 7.99,
    'ShipName': 'LILA-Supermercado',
    'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
    'ShipCity': 'Barquisimeto',
    'ShipRegion': 'Lara',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10382,
    'CustomerID': 'ERNSH',
    'OrderDate': '1996-12-13T00:00:00.000Z',
    'ShippedDate': '1996-12-16T00:00:00.000Z',
    'Freight': 94.77,
    'ShipName': 'Ernst Handel',
    'ShipAddress': 'Kirchgasse 6',
    'ShipCity': 'Graz',
    'ShipRegion': null,
    'ShipCountry': 'Austria'
  },
  {

```



```

      'OrderID': 10383,
      'CustomerID': 'AROUT',
      'OrderDate': '1996-12-16T00:00:00.000Z',
      'ShippedDate': '1996-12-18T00:00:00.000Z',
      'Freight': 34.24,
      'ShipName': 'Around the Horn',
      'ShipAddress': 'Brook Farm Stratford St. Mary',
      'ShipCity': 'Colchester',
      'ShipRegion': 'Essex',
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10384,
      'CustomerID': 'BERGS',
      'OrderDate': '1996-12-16T00:00:00.000Z',
      'ShippedDate': '1996-12-20T00:00:00.000Z',
      'Freight': 168.64,
      'ShipName': 'Berglunds snabbköp',
      'ShipAddress': 'Berguvsvägen 8',
      'ShipCity': 'Luleå',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 10385,
      'CustomerID': 'SPLIR',
      'OrderDate': '1996-12-17T00:00:00.000Z',
      'ShippedDate': '1996-12-23T00:00:00.000Z',
      'Freight': 30.96,
      'ShipName': 'Split Rail Beer & Ale',
      'ShipAddress': 'P.O. Box 555',
      'ShipCity': 'Lander',
      'ShipRegion': 'WY',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10386,
      'CustomerID': 'FAMIA',
      'OrderDate': '1996-12-18T00:00:00.000Z',
      'ShippedDate': '1996-12-25T00:00:00.000Z',
      'Freight': 13.99,
      'ShipName': 'Familia Arquibaldo',
      'ShipAddress': 'Rua Orós, 92',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10387,
      'CustomerID': 'SANTG',
      'OrderDate': '1996-12-18T00:00:00.000Z',
      'ShippedDate': '1996-12-20T00:00:00.000Z',
      'Freight': 93.63,
      'ShipName': 'Santé Gourmet',
      'ShipAddress': 'Erling Skakkes gate 78',
      'ShipCity': 'Stavern',
      'ShipRegion': null,

```

```

      'ShipCountry': 'Norway'
    },
    {
      'OrderID': 10388,
      'CustomerID': 'SEVES',
      'OrderDate': '1996-12-19T00:00:00.000Z',
      'ShippedDate': '1996-12-20T00:00:00.000Z',
      'Freight': 34.86,
      'ShipName': 'Seven Seas Imports',
      'ShipAddress': '90 Wadhurst Rd.',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10389,
      'CustomerID': 'BOTTM',
      'OrderDate': '1996-12-20T00:00:00.000Z',
      'ShippedDate': '1996-12-24T00:00:00.000Z',
      'Freight': 47.42,
      'ShipName': 'Bottom-Dollar Markets',
      'ShipAddress': '23 Tsawassen Blvd.',
      'ShipCity': 'Tsawassen',
      'ShipRegion': 'BC',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10390,
      'CustomerID': 'ERNSH',
      'OrderDate': '1996-12-23T00:00:00.000Z',
      'ShippedDate': '1996-12-26T00:00:00.000Z',
      'Freight': 126.38,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10391,
      'CustomerID': 'DRACD',
      'OrderDate': '1996-12-23T00:00:00.000Z',
      'ShippedDate': '1996-12-31T00:00:00.000Z',
      'Freight': 5.45,
      'ShipName': 'Drachenblut Delikatessen',
      'ShipAddress': 'Walserweg 21',
      'ShipCity': 'Aachen',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10392,
      'CustomerID': 'PICCO',
      'OrderDate': '1996-12-24T00:00:00.000Z',
      'ShippedDate': '1997-01-01T00:00:00.000Z',
      'Freight': 122.46,
      'ShipName': 'Piccolo und mehr',

```

```

      'ShipAddress': 'Geislweg 14',
      'ShipCity': 'Salzburg',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10393,
      'CustomerID': 'SAVEA',
      'OrderDate': '1996-12-25T00:00:00.000Z',
      'ShippedDate': '1997-01-03T00:00:00.000Z',
      'Freight': 126.56,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10394,
      'CustomerID': 'HUNGC',
      'OrderDate': '1996-12-25T00:00:00.000Z',
      'ShippedDate': '1997-01-03T00:00:00.000Z',
      'Freight': 30.34,
      'ShipName': 'Hungry Coyote Import Store',
      'ShipAddress': 'City Center Plaza 516 Main St.',
      'ShipCity': 'Elgin',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10395,
      'CustomerID': 'HILAA',
      'OrderDate': '1996-12-26T00:00:00.000Z',
      'ShippedDate': '1997-01-03T00:00:00.000Z',
      'Freight': 184.41,
      'ShipName': 'HILARION-Abastos',
      'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
      'ShipCity': 'San Cristóbal',
      'ShipRegion': 'Táchira',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10396,
      'CustomerID': 'FRANK',
      'OrderDate': '1996-12-27T00:00:00.000Z',
      'ShippedDate': '1997-01-06T00:00:00.000Z',
      'Freight': 135.35,
      'ShipName': 'Frankenversand',
      'ShipAddress': 'Berliner Platz 43',
      'ShipCity': 'München',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10397,
      'CustomerID': 'PRINI',
      'OrderDate': '1996-12-27T00:00:00.000Z',

```

```

    'ShippedDate': '1997-01-02T00:00:00.000Z',
    'Freight': 60.26,
    'ShipName': 'Princesa Isabel Vinhos',
    'ShipAddress': 'Estrada da saúde n. 58',
    'ShipCity': 'Lisboa',
    'ShipRegion': null,
    'ShipCountry': 'Portugal'
  },
  {
    'OrderID': 10398,
    'CustomerID': 'SAVEA',
    'OrderDate': '1996-12-30T00:00:00.000Z',
    'ShippedDate': '1997-01-09T00:00:00.000Z',
    'Freight': 89.16,
    'ShipName': 'Save-a-lot Markets',
    'ShipAddress': '187 Suffolk Ln.',
    'ShipCity': 'Boise',
    'ShipRegion': 'ID',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10399,
    'CustomerID': 'VAFFE',
    'OrderDate': '1996-12-31T00:00:00.000Z',
    'ShippedDate': '1997-01-08T00:00:00.000Z',
    'Freight': 27.36,
    'ShipName': 'Vaffeljernet',
    'ShipAddress': 'Smagsloget 45',
    'ShipCity': 'Århus',
    'ShipRegion': null,
    'ShipCountry': 'Denmark'
  },
  {
    'OrderID': 10400,
    'CustomerID': 'EASTC',
    'OrderDate': '1997-01-01T00:00:00.000Z',
    'ShippedDate': '1997-01-16T00:00:00.000Z',
    'Freight': 83.93,
    'ShipName': 'Eastern Connection',
    'ShipAddress': '35 King George',
    'ShipCity': 'London',
    'ShipRegion': null,
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10401,
    'CustomerID': 'RATTC',
    'OrderDate': '1997-01-01T00:00:00.000Z',
    'ShippedDate': '1997-01-10T00:00:00.000Z',
    'Freight': 12.51,
    'ShipName': 'Rattlesnake Canyon Grocery',
    'ShipAddress': '2817 Milton Dr.',
    'ShipCity': 'Albuquerque',
    'ShipRegion': 'NM',
    'ShipCountry': 'USA'
  },
  {

```

```

      'OrderID': 10402,
      'CustomerID': 'ERNSH',
      'OrderDate': '1997-01-02T00:00:00.000Z',
      'ShippedDate': '1997-01-10T00:00:00.000Z',
      'Freight': 67.88,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10403,
      'CustomerID': 'ERNSH',
      'OrderDate': '1997-01-03T00:00:00.000Z',
      'ShippedDate': '1997-01-09T00:00:00.000Z',
      'Freight': 73.79,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10404,
      'CustomerID': 'MAGAA',
      'OrderDate': '1997-01-03T00:00:00.000Z',
      'ShippedDate': '1997-01-08T00:00:00.000Z',
      'Freight': 155.97,
      'ShipName': 'Magazzini Alimentari Riuniti',
      'ShipAddress': 'Via Ludovico il Moro 22',
      'ShipCity': 'Bergamo',
      'ShipRegion': null,
      'ShipCountry': 'Italy'
    },
    {
      'OrderID': 10405,
      'CustomerID': 'LINOD',
      'OrderDate': '1997-01-06T00:00:00.000Z',
      'ShippedDate': '1997-01-22T00:00:00.000Z',
      'Freight': 34.82,
      'ShipName': 'LINO-Delicateses',
      'ShipAddress': 'Ave. 5 de Mayo Porlamar',
      'ShipCity': 'I. de Margarita',
      'ShipRegion': 'Nueva Esparta',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10406,
      'CustomerID': 'QUEEN',
      'OrderDate': '1997-01-07T00:00:00.000Z',
      'ShippedDate': '1997-01-13T00:00:00.000Z',
      'Freight': 108.04,
      'ShipName': 'Queen Cozinha',
      'ShipAddress': 'Alameda dos Canários, 891',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',

```

```

    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10407,
    'CustomerID': 'OTTIK',
    'OrderDate': '1997-01-07T00:00:00.000Z',
    'ShippedDate': '1997-01-30T00:00:00.000Z',
    'Freight': 91.48,
    'ShipName': 'Ottilies Käseladen',
    'ShipAddress': 'Mehrheimerstr. 369',
    'ShipCity': 'Köln',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10408,
    'CustomerID': 'FOLIG',
    'OrderDate': '1997-01-08T00:00:00.000Z',
    'ShippedDate': '1997-01-14T00:00:00.000Z',
    'Freight': 11.26,
    'ShipName': 'Folies gourmandes',
    'ShipAddress': '184, chaussée de Tournai',
    'ShipCity': 'Lille',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10409,
    'CustomerID': 'OCEAN',
    'OrderDate': '1997-01-09T00:00:00.000Z',
    'ShippedDate': '1997-01-14T00:00:00.000Z',
    'Freight': 29.83,
    'ShipName': 'Océano Atlántico Ltda.',
    'ShipAddress': 'Ing. Gustavo Moncada 8585 Piso 20-A',
    'ShipCity': 'Buenos Aires',
    'ShipRegion': null,
    'ShipCountry': 'Argentina'
  },
  {
    'OrderID': 10410,
    'CustomerID': 'BOTTM',
    'OrderDate': '1997-01-10T00:00:00.000Z',
    'ShippedDate': '1997-01-15T00:00:00.000Z',
    'Freight': 2.4,
    'ShipName': 'Bottom-Dollar Markets',
    'ShipAddress': '23 Tsawassen Blvd.',
    'ShipCity': 'Tsawassen',
    'ShipRegion': 'BC',
    'ShipCountry': 'Canada'
  },
  {
    'OrderID': 10411,
    'CustomerID': 'BOTTM',
    'OrderDate': '1997-01-10T00:00:00.000Z',
    'ShippedDate': '1997-01-21T00:00:00.000Z',
    'Freight': 23.65,
    'ShipName': 'Bottom-Dollar Markets',

```

```

      'ShipAddress': '23 Tsawassen Blvd.',
      'ShipCity': 'Tsawassen',
      'ShipRegion': 'BC',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10412,
      'CustomerID': 'WARTH',
      'OrderDate': '1997-01-13T00:00:00.000Z',
      'ShippedDate': '1997-01-15T00:00:00.000Z',
      'Freight': 3.77,
      'ShipName': 'Wartian Herkku',
      'ShipAddress': 'Torikatu 38',
      'ShipCity': 'Oulu',
      'ShipRegion': null,
      'ShipCountry': 'Finland'
    },
    {
      'OrderID': 10413,
      'CustomerID': 'LAMAI',
      'OrderDate': '1997-01-14T00:00:00.000Z',
      'ShippedDate': '1997-01-16T00:00:00.000Z',
      'Freight': 95.66,
      'ShipName': 'La maison d\'Asie',
      'ShipAddress': '1 rue Alsace-Lorraine',
      'ShipCity': 'Toulouse',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10414,
      'CustomerID': 'FAMIA',
      'OrderDate': '1997-01-14T00:00:00.000Z',
      'ShippedDate': '1997-01-17T00:00:00.000Z',
      'Freight': 21.48,
      'ShipName': 'Familia Arquibaldo',
      'ShipAddress': 'Rua Orós, 92',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10415,
      'CustomerID': 'HUNGC',
      'OrderDate': '1997-01-15T00:00:00.000Z',
      'ShippedDate': '1997-01-24T00:00:00.000Z',
      'Freight': 0.2,
      'ShipName': 'Hungry Coyote Import Store',
      'ShipAddress': 'City Center Plaza 516 Main St.',
      'ShipCity': 'Elgin',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10416,
      'CustomerID': 'WARTH',
      'OrderDate': '1997-01-16T00:00:00.000Z',

```

```

    'ShippedDate': '1997-01-27T00:00:00.000Z',
    'Freight': 22.72,
    'ShipName': 'Wartian Herkku',
    'ShipAddress': 'Torikatu 38',
    'ShipCity': 'Oulu',
    'ShipRegion': null,
    'ShipCountry': 'Finland'
  },
  {
    'OrderID': 10417,
    'CustomerID': 'SIMOB',
    'OrderDate': '1997-01-16T00:00:00.000Z',
    'ShippedDate': '1997-01-28T00:00:00.000Z',
    'Freight': 70.29,
    'ShipName': 'Simons bistro',
    'ShipAddress': 'Vinbæltet 34',
    'ShipCity': 'Kobenhavn',
    'ShipRegion': null,
    'ShipCountry': 'Denmark'
  },
  {
    'OrderID': 10418,
    'CustomerID': 'QUICK',
    'OrderDate': '1997-01-17T00:00:00.000Z',
    'ShippedDate': '1997-01-24T00:00:00.000Z',
    'Freight': 17.55,
    'ShipName': 'QUICK-Stop',
    'ShipAddress': 'Taucherstraße 10',
    'ShipCity': 'Cunewalde',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10419,
    'CustomerID': 'RICSU',
    'OrderDate': '1997-01-20T00:00:00.000Z',
    'ShippedDate': '1997-01-30T00:00:00.000Z',
    'Freight': 137.35,
    'ShipName': 'Richter Supermarkt',
    'ShipAddress': 'Starenweg 5',
    'ShipCity': 'Genève',
    'ShipRegion': null,
    'ShipCountry': 'Switzerland'
  },
  {
    'OrderID': 10420,
    'CustomerID': 'WELLI',
    'OrderDate': '1997-01-21T00:00:00.000Z',
    'ShippedDate': '1997-01-27T00:00:00.000Z',
    'Freight': 44.12,
    'ShipName': 'Wellington Importadora',
    'ShipAddress': 'Rua do Mercado, 12',
    'ShipCity': 'Resende',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  }
]

```



```

      'OrderID': 10421,
      'CustomerID': 'QUEDE',
      'OrderDate': '1997-01-21T00:00:00.000Z',
      'ShippedDate': '1997-01-27T00:00:00.000Z',
      'Freight': 99.23,
      'ShipName': 'Que Delícia',
      'ShipAddress': 'Rua da Panificadora, 12',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10422,
      'CustomerID': 'FRANS',
      'OrderDate': '1997-01-22T00:00:00.000Z',
      'ShippedDate': '1997-01-31T00:00:00.000Z',
      'Freight': 3.02,
      'ShipName': 'Franchi S.p.A.',
      'ShipAddress': 'Via Monte Bianco 34',
      'ShipCity': 'Torino',
      'ShipRegion': null,
      'ShipCountry': 'Italy'
    },
    {
      'OrderID': 10423,
      'CustomerID': 'GOURL',
      'OrderDate': '1997-01-23T00:00:00.000Z',
      'ShippedDate': '1997-02-24T00:00:00.000Z',
      'Freight': 24.5,
      'ShipName': 'Gourmet Lanchonetes',
      'ShipAddress': 'Av. Brasil, 442',
      'ShipCity': 'Campinas',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10424,
      'CustomerID': 'MEREP',
      'OrderDate': '1997-01-23T00:00:00.000Z',
      'ShippedDate': '1997-01-27T00:00:00.000Z',
      'Freight': 370.61,
      'ShipName': 'Mère Paillarde',
      'ShipAddress': '43 rue St. Laurent',
      'ShipCity': 'Montréal',
      'ShipRegion': 'Québec',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10425,
      'CustomerID': 'LAMAI',
      'OrderDate': '1997-01-24T00:00:00.000Z',
      'ShippedDate': '1997-02-14T00:00:00.000Z',
      'Freight': 7.93,
      'ShipName': 'La maison d\'Asie',
      'ShipAddress': '1 rue Alsace-Lorraine',
      'ShipCity': 'Toulouse',
      'ShipRegion': null,

```

```

    'ShipCountry': 'France'
  },
  {
    'OrderID': 10426,
    'CustomerID': 'GALED',
    'OrderDate': '1997-01-27T00:00:00.000Z',
    'ShippedDate': '1997-02-06T00:00:00.000Z',
    'Freight': 18.69,
    'ShipName': 'Galería del gastronómo',
    'ShipAddress': 'Rambla de Cataluña, 23',
    'ShipCity': 'Barcelona',
    'ShipRegion': null,
    'ShipCountry': 'Spain'
  },
  {
    'OrderID': 10427,
    'CustomerID': 'PICCO',
    'OrderDate': '1997-01-27T00:00:00.000Z',
    'ShippedDate': '1997-03-03T00:00:00.000Z',
    'Freight': 31.29,
    'ShipName': 'Piccolo und mehr',
    'ShipAddress': 'Geislweg 14',
    'ShipCity': 'Salzburg',
    'ShipRegion': null,
    'ShipCountry': 'Austria'
  },
  {
    'OrderID': 10428,
    'CustomerID': 'REGGC',
    'OrderDate': '1997-01-28T00:00:00.000Z',
    'ShippedDate': '1997-02-04T00:00:00.000Z',
    'Freight': 11.09,
    'ShipName': 'Reggiani Caseifici',
    'ShipAddress': 'Strada Provinciale 124',
    'ShipCity': 'Reggio Emilia',
    'ShipRegion': null,
    'ShipCountry': 'Italy'
  },
  {
    'OrderID': 10429,
    'CustomerID': 'HUNGO',
    'OrderDate': '1997-01-29T00:00:00.000Z',
    'ShippedDate': '1997-02-07T00:00:00.000Z',
    'Freight': 56.63,
    'ShipName': 'Hungry Owl All-Night Grocers',
    'ShipAddress': '8 Johnstown Road',
    'ShipCity': 'Cork',
    'ShipRegion': 'Co. Cork',
    'ShipCountry': 'Ireland'
  },
  {
    'OrderID': 10430,
    'CustomerID': 'ERNSH',
    'OrderDate': '1997-01-30T00:00:00.000Z',
    'ShippedDate': '1997-02-03T00:00:00.000Z',
    'Freight': 458.78,
    'ShipName': 'Ernst Handel',

```

```

      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10431,
      'CustomerID': 'BOTTM',
      'OrderDate': '1997-01-30T00:00:00.000Z',
      'ShippedDate': '1997-02-07T00:00:00.000Z',
      'Freight': 44.17,
      'ShipName': 'Bottom-Dollar Markets',
      'ShipAddress': '23 Tsawassen Blvd.',
      'ShipCity': 'Tsawassen',
      'ShipRegion': 'BC',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10432,
      'CustomerID': 'SPLIR',
      'OrderDate': '1997-01-31T00:00:00.000Z',
      'ShippedDate': '1997-02-07T00:00:00.000Z',
      'Freight': 4.34,
      'ShipName': 'Split Rail Beer & Ale',
      'ShipAddress': 'P.O. Box 555',
      'ShipCity': 'Lander',
      'ShipRegion': 'WY',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10433,
      'CustomerID': 'PRINI',
      'OrderDate': '1997-02-03T00:00:00.000Z',
      'ShippedDate': '1997-03-04T00:00:00.000Z',
      'Freight': 73.83,
      'ShipName': 'Princesa Isabel Vinhos',
      'ShipAddress': 'Estrada da saúde n. 58',
      'ShipCity': 'Lisboa',
      'ShipRegion': null,
      'ShipCountry': 'Portugal'
    },
    {
      'OrderID': 10434,
      'CustomerID': 'FOLKO',
      'OrderDate': '1997-02-03T00:00:00.000Z',
      'ShippedDate': '1997-02-13T00:00:00.000Z',
      'Freight': 17.92,
      'ShipName': 'Folk och fä HB',
      'ShipAddress': 'Åkergatan 24',
      'ShipCity': 'Bräcke',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 10435,
      'CustomerID': 'CONSH',
      'OrderDate': '1997-02-04T00:00:00.000Z',

```

```

      'ShippedDate': '1997-02-07T00:00:00.000Z',
      'Freight': 9.21,
      'ShipName': 'Consolidated Holdings',
      'ShipAddress': 'Berkeley Gardens 12 Brewery',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10436,
      'CustomerID': 'BLONP',
      'OrderDate': '1997-02-05T00:00:00.000Z',
      'ShippedDate': '1997-02-11T00:00:00.000Z',
      'Freight': 156.66,
      'ShipName': 'Blondel père et fils',
      'ShipAddress': '24, place Kléber',
      'ShipCity': 'Strasbourg',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10437,
      'CustomerID': 'WARTH',
      'OrderDate': '1997-02-05T00:00:00.000Z',
      'ShippedDate': '1997-02-12T00:00:00.000Z',
      'Freight': 19.97,
      'ShipName': 'Wartian Herkku',
      'ShipAddress': 'Torikatu 38',
      'ShipCity': 'Oulu',
      'ShipRegion': null,
      'ShipCountry': 'Finland'
    },
    {
      'OrderID': 10438,
      'CustomerID': 'TOMSP',
      'OrderDate': '1997-02-06T00:00:00.000Z',
      'ShippedDate': '1997-02-14T00:00:00.000Z',
      'Freight': 8.24,
      'ShipName': 'Toms Spezialitäten',
      'ShipAddress': 'Luisenstr. 48',
      'ShipCity': 'Münster',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10439,
      'CustomerID': 'MEREP',
      'OrderDate': '1997-02-07T00:00:00.000Z',
      'ShippedDate': '1997-02-10T00:00:00.000Z',
      'Freight': 4.07,
      'ShipName': 'Mère Paillarde',
      'ShipAddress': '43 rue St. Laurent',
      'ShipCity': 'Montréal',
      'ShipRegion': 'Québec',
      'ShipCountry': 'Canada'
    }
  ],
  {

```

```

      'OrderID': 10440,
      'CustomerID': 'SAVEA',
      'OrderDate': '1997-02-10T00:00:00.000Z',
      'ShippedDate': '1997-02-28T00:00:00.000Z',
      'Freight': 86.53,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10441,
      'CustomerID': 'OLDWO',
      'OrderDate': '1997-02-10T00:00:00.000Z',
      'ShippedDate': '1997-03-14T00:00:00.000Z',
      'Freight': 73.02,
      'ShipName': 'Old World Delicatessen',
      'ShipAddress': '2743 Bering St.',
      'ShipCity': 'Anchorage',
      'ShipRegion': 'AK',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10442,
      'CustomerID': 'ERNSH',
      'OrderDate': '1997-02-11T00:00:00.000Z',
      'ShippedDate': '1997-02-18T00:00:00.000Z',
      'Freight': 47.94,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10443,
      'CustomerID': 'REGGC',
      'OrderDate': '1997-02-12T00:00:00.000Z',
      'ShippedDate': '1997-02-14T00:00:00.000Z',
      'Freight': 13.95,
      'ShipName': 'Reggiani Caseifici',
      'ShipAddress': 'Strada Provinciale 124',
      'ShipCity': 'Reggio Emilia',
      'ShipRegion': null,
      'ShipCountry': 'Italy'
    },
    {
      'OrderID': 10444,
      'CustomerID': 'BERGS',
      'OrderDate': '1997-02-12T00:00:00.000Z',
      'ShippedDate': '1997-02-21T00:00:00.000Z',
      'Freight': 3.5,
      'ShipName': 'Berglunds snabbköp',
      'ShipAddress': 'Berguvsvägen 8',
      'ShipCity': 'Luleå',
      'ShipRegion': null,

```

```

    'ShipCountry': 'Sweden'
  },
  {
    'OrderID': 10445,
    'CustomerID': 'BERGS',
    'OrderDate': '1997-02-13T00:00:00.000Z',
    'ShippedDate': '1997-02-20T00:00:00.000Z',
    'Freight': 9.3,
    'ShipName': 'Berglunds snabbköp',
    'ShipAddress': 'Berguvsvägen 8',
    'ShipCity': 'Luleå',
    'ShipRegion': null,
    'ShipCountry': 'Sweden'
  },
  {
    'OrderID': 10446,
    'CustomerID': 'TOMSP',
    'OrderDate': '1997-02-14T00:00:00.000Z',
    'ShippedDate': '1997-02-19T00:00:00.000Z',
    'Freight': 14.68,
    'ShipName': 'Toms Spezialitäten',
    'ShipAddress': 'Luisenstr. 48',
    'ShipCity': 'Münster',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10447,
    'CustomerID': 'RICAR',
    'OrderDate': '1997-02-14T00:00:00.000Z',
    'ShippedDate': '1997-03-07T00:00:00.000Z',
    'Freight': 68.66,
    'ShipName': 'Ricardo Adocicados',
    'ShipAddress': 'Av. Copacabana, 267',
    'ShipCity': 'Rio de Janeiro',
    'ShipRegion': 'RJ',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10448,
    'CustomerID': 'RANCH',
    'OrderDate': '1997-02-17T00:00:00.000Z',
    'ShippedDate': '1997-02-24T00:00:00.000Z',
    'Freight': 38.82,
    'ShipName': 'Rancho grande',
    'ShipAddress': 'Av. del Libertador 900',
    'ShipCity': 'Buenos Aires',
    'ShipRegion': null,
    'ShipCountry': 'Argentina'
  },
  {
    'OrderID': 10449,
    'CustomerID': 'BLONP',
    'OrderDate': '1997-02-18T00:00:00.000Z',
    'ShippedDate': '1997-02-27T00:00:00.000Z',
    'Freight': 53.3,
    'ShipName': 'Blondel père et fils',

```

```

    'ShipAddress': '24, place Kléber',
    'ShipCity': 'Strasbourg',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10450,
    'CustomerID': 'VICTE',
    'OrderDate': '1997-02-19T00:00:00.000Z',
    'ShippedDate': '1997-03-11T00:00:00.000Z',
    'Freight': 7.23,
    'ShipName': 'Victuailles en stock',
    'ShipAddress': '2, rue du Commerce',
    'ShipCity': 'Lyon',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10451,
    'CustomerID': 'QUICK',
    'OrderDate': '1997-02-19T00:00:00.000Z',
    'ShippedDate': '1997-03-12T00:00:00.000Z',
    'Freight': 189.09,
    'ShipName': 'QUICK-Stop',
    'ShipAddress': 'Taucherstraße 10',
    'ShipCity': 'Cunewalde',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10452,
    'CustomerID': 'SAVEA',
    'OrderDate': '1997-02-20T00:00:00.000Z',
    'ShippedDate': '1997-02-26T00:00:00.000Z',
    'Freight': 140.26,
    'ShipName': 'Save-a-lot Markets',
    'ShipAddress': '187 Suffolk Ln.',
    'ShipCity': 'Boise',
    'ShipRegion': 'ID',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10453,
    'CustomerID': 'AROUT',
    'OrderDate': '1997-02-21T00:00:00.000Z',
    'ShippedDate': '1997-02-26T00:00:00.000Z',
    'Freight': 25.36,
    'ShipName': 'Around the Horn',
    'ShipAddress': 'Brook Farm Stratford St. Mary',
    'ShipCity': 'Colchester',
    'ShipRegion': 'Essex',
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10454,
    'CustomerID': 'LAMAI',
    'OrderDate': '1997-02-21T00:00:00.000Z',

```

```

      'ShippedDate': '1997-02-25T00:00:00.000Z',
      'Freight': 2.74,
      'ShipName': 'La maison d\'Asie',
      'ShipAddress': '1 rue Alsace-Lorraine',
      'ShipCity': 'Toulouse',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10455,
      'CustomerID': 'WARTH',
      'OrderDate': '1997-02-24T00:00:00.000Z',
      'ShippedDate': '1997-03-03T00:00:00.000Z',
      'Freight': 180.45,
      'ShipName': 'Wartian Herkku',
      'ShipAddress': 'Torikatu 38',
      'ShipCity': 'Oulu',
      'ShipRegion': null,
      'ShipCountry': 'Finland'
    },
    {
      'OrderID': 10456,
      'CustomerID': 'KOENE',
      'OrderDate': '1997-02-25T00:00:00.000Z',
      'ShippedDate': '1997-02-28T00:00:00.000Z',
      'Freight': 8.12,
      'ShipName': 'Königlich Essen',
      'ShipAddress': 'Maubelstr. 90',
      'ShipCity': 'Brandenburg',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10457,
      'CustomerID': 'KOENE',
      'OrderDate': '1997-02-25T00:00:00.000Z',
      'ShippedDate': '1997-03-03T00:00:00.000Z',
      'Freight': 11.57,
      'ShipName': 'Königlich Essen',
      'ShipAddress': 'Maubelstr. 90',
      'ShipCity': 'Brandenburg',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10458,
      'CustomerID': 'SUPRD',
      'OrderDate': '1997-02-26T00:00:00.000Z',
      'ShippedDate': '1997-03-04T00:00:00.000Z',
      'Freight': 147.06,
      'ShipName': 'Suprêmes délices',
      'ShipAddress': 'Boulevard Tirou, 255',
      'ShipCity': 'Charleroi',
      'ShipRegion': null,
      'ShipCountry': 'Belgium'
    }
  ],
  {

```



```

    'OrderID': 10459,
    'CustomerID': 'VICTE',
    'OrderDate': '1997-02-27T00:00:00.000Z',
    'ShippedDate': '1997-02-28T00:00:00.000Z',
    'Freight': 25.09,
    'ShipName': 'Victuailles en stock',
    'ShipAddress': '2, rue du Commerce',
    'ShipCity': 'Lyon',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10460,
    'CustomerID': 'FOLKO',
    'OrderDate': '1997-02-28T00:00:00.000Z',
    'ShippedDate': '1997-03-03T00:00:00.000Z',
    'Freight': 16.27,
    'ShipName': 'Folk och fä HB',
    'ShipAddress': 'Åkergatan 24',
    'ShipCity': 'Bräcke',
    'ShipRegion': null,
    'ShipCountry': 'Sweden'
  },
  {
    'OrderID': 10461,
    'CustomerID': 'LILAS',
    'OrderDate': '1997-02-28T00:00:00.000Z',
    'ShippedDate': '1997-03-05T00:00:00.000Z',
    'Freight': 148.61,
    'ShipName': 'LILA-Supermercado',
    'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
    'ShipCity': 'Barquisimeto',
    'ShipRegion': 'Lara',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10462,
    'CustomerID': 'CONSH',
    'OrderDate': '1997-03-03T00:00:00.000Z',
    'ShippedDate': '1997-03-18T00:00:00.000Z',
    'Freight': 6.17,
    'ShipName': 'Consolidated Holdings',
    'ShipAddress': 'Berkeley Gardens 12 Brewery',
    'ShipCity': 'London',
    'ShipRegion': null,
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10463,
    'CustomerID': 'SUPRD',
    'OrderDate': '1997-03-04T00:00:00.000Z',
    'ShippedDate': '1997-03-06T00:00:00.000Z',
    'Freight': 14.78,
    'ShipName': 'Suprêmes délices',
    'ShipAddress': 'Boulevard Tirou, 255',
    'ShipCity': 'Charleroi',
    'ShipRegion': null,

```

```

    'ShipCountry': 'Belgium'
  },
  {
    'OrderID': 10464,
    'CustomerID': 'FURIB',
    'OrderDate': '1997-03-04T00:00:00.000Z',
    'ShippedDate': '1997-03-14T00:00:00.000Z',
    'Freight': 89,
    'ShipName': 'Furia Bacalhau e Frutos do Mar',
    'ShipAddress': 'Jardim das rosas n. 32',
    'ShipCity': 'Lisboa',
    'ShipRegion': null,
    'ShipCountry': 'Portugal'
  },
  {
    'OrderID': 10465,
    'CustomerID': 'VAFFE',
    'OrderDate': '1997-03-05T00:00:00.000Z',
    'ShippedDate': '1997-03-14T00:00:00.000Z',
    'Freight': 145.04,
    'ShipName': 'Vaffeljernet',
    'ShipAddress': 'Smagsloget 45',
    'ShipCity': 'Århus',
    'ShipRegion': null,
    'ShipCountry': 'Denmark'
  },
  {
    'OrderID': 10466,
    'CustomerID': 'COMMI',
    'OrderDate': '1997-03-06T00:00:00.000Z',
    'ShippedDate': '1997-03-13T00:00:00.000Z',
    'Freight': 11.93,
    'ShipName': 'Comércio Mineiro',
    'ShipAddress': 'Av. dos Lusíadas, 23',
    'ShipCity': 'Sao Paulo',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10467,
    'CustomerID': 'MAGAA',
    'OrderDate': '1997-03-06T00:00:00.000Z',
    'ShippedDate': '1997-03-11T00:00:00.000Z',
    'Freight': 4.93,
    'ShipName': 'Magazzini Alimentari Riuniti',
    'ShipAddress': 'Via Ludovico il Moro 22',
    'ShipCity': 'Bergamo',
    'ShipRegion': null,
    'ShipCountry': 'Italy'
  },
  {
    'OrderID': 10468,
    'CustomerID': 'KOENE',
    'OrderDate': '1997-03-07T00:00:00.000Z',
    'ShippedDate': '1997-03-12T00:00:00.000Z',
    'Freight': 44.12,
    'ShipName': 'Königlich Essen',

```

```

      'ShipAddress': 'Maubelstr. 90',
      'ShipCity': 'Brandenburg',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10469,
      'CustomerID': 'WHITC',
      'OrderDate': '1997-03-10T00:00:00.000Z',
      'ShippedDate': '1997-03-14T00:00:00.000Z',
      'Freight': 60.18,
      'ShipName': 'White Clover Markets',
      'ShipAddress': '1029 - 12th Ave. S.',
      'ShipCity': 'Seattle',
      'ShipRegion': 'WA',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10470,
      'CustomerID': 'BONAP',
      'OrderDate': '1997-03-11T00:00:00.000Z',
      'ShippedDate': '1997-03-14T00:00:00.000Z',
      'Freight': 64.56,
      'ShipName': 'Bon app',
      'ShipAddress': '12, rue des Bouchers',
      'ShipCity': 'Marseille',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10471,
      'CustomerID': 'BSBEV',
      'OrderDate': '1997-03-11T00:00:00.000Z',
      'ShippedDate': '1997-03-18T00:00:00.000Z',
      'Freight': 45.59,
      'ShipName': 'B\ Beverages',
      'ShipAddress': 'Fauntleroy Circus',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10472,
      'CustomerID': 'SEVES',
      'OrderDate': '1997-03-12T00:00:00.000Z',
      'ShippedDate': '1997-03-19T00:00:00.000Z',
      'Freight': 4.2,
      'ShipName': 'Seven Seas Imports',
      'ShipAddress': '90 Wadhurst Rd.',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10473,
      'CustomerID': 'ISLAT',
      'OrderDate': '1997-03-13T00:00:00.000Z',

```

```

    'ShippedDate': '1997-03-21T00:00:00.000Z',
    'Freight': 16.37,
    'ShipName': 'Island Trading',
    'ShipAddress': 'Garden House Crowther Way',
    'ShipCity': 'Cowes',
    'ShipRegion': 'Isle of Wight',
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10474,
    'CustomerID': 'PERIC',
    'OrderDate': '1997-03-13T00:00:00.000Z',
    'ShippedDate': '1997-03-21T00:00:00.000Z',
    'Freight': 83.49,
    'ShipName': 'Pericles Comidas clásicas',
    'ShipAddress': 'Calle Dr. Jorge Cash 321',
    'ShipCity': 'México D.F.',
    'ShipRegion': null,
    'ShipCountry': 'Mexico'
  },
  {
    'OrderID': 10475,
    'CustomerID': 'SUPRD',
    'OrderDate': '1997-03-14T00:00:00.000Z',
    'ShippedDate': '1997-04-04T00:00:00.000Z',
    'Freight': 68.52,
    'ShipName': 'Suprêmes délices',
    'ShipAddress': 'Boulevard Tirou, 255',
    'ShipCity': 'Charleroi',
    'ShipRegion': null,
    'ShipCountry': 'Belgium'
  },
  {
    'OrderID': 10476,
    'CustomerID': 'HILAA',
    'OrderDate': '1997-03-17T00:00:00.000Z',
    'ShippedDate': '1997-03-24T00:00:00.000Z',
    'Freight': 4.41,
    'ShipName': 'HILARION-Abastos',
    'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
    'ShipCity': 'San Cristóbal',
    'ShipRegion': 'Táchira',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10477,
    'CustomerID': 'PRINI',
    'OrderDate': '1997-03-17T00:00:00.000Z',
    'ShippedDate': '1997-03-25T00:00:00.000Z',
    'Freight': 13.02,
    'ShipName': 'Princesa Isabel Vinhos',
    'ShipAddress': 'Estrada da saúde n. 58',
    'ShipCity': 'Lisboa',
    'ShipRegion': null,
    'ShipCountry': 'Portugal'
  },
  {

```

```

    'OrderID': 10478,
    'CustomerID': 'VICTE',
    'OrderDate': '1997-03-18T00:00:00.000Z',
    'ShippedDate': '1997-03-26T00:00:00.000Z',
    'Freight': 4.81,
    'ShipName': 'Victuailles en stock',
    'ShipAddress': '2, rue du Commerce',
    'ShipCity': 'Lyon',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10479,
    'CustomerID': 'RATTC',
    'OrderDate': '1997-03-19T00:00:00.000Z',
    'ShippedDate': '1997-03-21T00:00:00.000Z',
    'Freight': 708.95,
    'ShipName': 'Rattlesnake Canyon Grocery',
    'ShipAddress': '2817 Milton Dr.',
    'ShipCity': 'Albuquerque',
    'ShipRegion': 'NM',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10480,
    'CustomerID': 'FOLIG',
    'OrderDate': '1997-03-20T00:00:00.000Z',
    'ShippedDate': '1997-03-24T00:00:00.000Z',
    'Freight': 1.35,
    'ShipName': 'Folies gourmandes',
    'ShipAddress': '184, chaussée de Tournai',
    'ShipCity': 'Lille',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10481,
    'CustomerID': 'RICAR',
    'OrderDate': '1997-03-20T00:00:00.000Z',
    'ShippedDate': '1997-03-25T00:00:00.000Z',
    'Freight': 64.33,
    'ShipName': 'Ricardo Adocicados',
    'ShipAddress': 'Av. Copacabana, 267',
    'ShipCity': 'Rio de Janeiro',
    'ShipRegion': 'RJ',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10482,
    'CustomerID': 'LAZYK',
    'OrderDate': '1997-03-21T00:00:00.000Z',
    'ShippedDate': '1997-04-10T00:00:00.000Z',
    'Freight': 7.48,
    'ShipName': 'Lazy K Kountry Store',
    'ShipAddress': '12 Orchestra Terrace',
    'ShipCity': 'Walla Walla',
    'ShipRegion': 'WA',
  }

```

```

    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10483,
    'CustomerID': 'WHITC',
    'OrderDate': '1997-03-24T00:00:00.000Z',
    'ShippedDate': '1997-04-25T00:00:00.000Z',
    'Freight': 15.28,
    'ShipName': 'White Clover Markets',
    'ShipAddress': '1029 - 12th Ave. S.',
    'ShipCity': 'Seattle',
    'ShipRegion': 'WA',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10484,
    'CustomerID': 'BSBEV',
    'OrderDate': '1997-03-24T00:00:00.000Z',
    'ShippedDate': '1997-04-01T00:00:00.000Z',
    'Freight': 6.88,
    'ShipName': 'B\ Beverages',
    'ShipAddress': 'Fauntleroy Circus',
    'ShipCity': 'London',
    'ShipRegion': null,
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10485,
    'CustomerID': 'LINOD',
    'OrderDate': '1997-03-25T00:00:00.000Z',
    'ShippedDate': '1997-03-31T00:00:00.000Z',
    'Freight': 64.45,
    'ShipName': 'LINO-Delicateses',
    'ShipAddress': 'Ave. 5 de Mayo Porlamar',
    'ShipCity': 'I. de Margarita',
    'ShipRegion': 'Nueva Esparta',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10486,
    'CustomerID': 'HILAA',
    'OrderDate': '1997-03-26T00:00:00.000Z',
    'ShippedDate': '1997-04-02T00:00:00.000Z',
    'Freight': 30.53,
    'ShipName': 'HILARION-Abastos',
    'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
    'ShipCity': 'San Cristóbal',
    'ShipRegion': 'Táchira',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10487,
    'CustomerID': 'QUEEN',
    'OrderDate': '1997-03-26T00:00:00.000Z',
    'ShippedDate': '1997-03-28T00:00:00.000Z',
    'Freight': 71.07,
    'ShipName': 'Queen Cozinha',

```

```

      'ShipAddress': 'Alameda dos Canários, 891',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10488,
      'CustomerID': 'FRANK',
      'OrderDate': '1997-03-27T00:00:00.000Z',
      'ShippedDate': '1997-04-02T00:00:00.000Z',
      'Freight': 4.93,
      'ShipName': 'Frankenversand',
      'ShipAddress': 'Berliner Platz 43',
      'ShipCity': 'München',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10489,
      'CustomerID': 'PICCO',
      'OrderDate': '1997-03-28T00:00:00.000Z',
      'ShippedDate': '1997-04-09T00:00:00.000Z',
      'Freight': 5.29,
      'ShipName': 'Piccolo und mehr',
      'ShipAddress': 'Geislweg 14',
      'ShipCity': 'Salzburg',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10490,
      'CustomerID': 'HILAA',
      'OrderDate': '1997-03-31T00:00:00.000Z',
      'ShippedDate': '1997-04-03T00:00:00.000Z',
      'Freight': 210.19,
      'ShipName': 'HILARION-Abastos',
      'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
      'ShipCity': 'San Cristóbal',
      'ShipRegion': 'Táchira',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10491,
      'CustomerID': 'FURIB',
      'OrderDate': '1997-03-31T00:00:00.000Z',
      'ShippedDate': '1997-04-08T00:00:00.000Z',
      'Freight': 16.96,
      'ShipName': 'Furia Bacalhau e Frutos do Mar',
      'ShipAddress': 'Jardim das rosas n. 32',
      'ShipCity': 'Lisboa',
      'ShipRegion': null,
      'ShipCountry': 'Portugal'
    },
    {
      'OrderID': 10492,
      'CustomerID': 'BOTTM',
      'OrderDate': '1997-04-01T00:00:00.000Z',

```

```

    'ShippedDate': '1997-04-11T00:00:00.000Z',
    'Freight': 62.89,
    'ShipName': 'Bottom-Dollar Markets',
    'ShipAddress': '23 Tsawassen Blvd.',
    'ShipCity': 'Tsawassen',
    'ShipRegion': 'BC',
    'ShipCountry': 'Canada'
  },
  {
    'OrderID': 10493,
    'CustomerID': 'LAMAI',
    'OrderDate': '1997-04-02T00:00:00.000Z',
    'ShippedDate': '1997-04-10T00:00:00.000Z',
    'Freight': 10.64,
    'ShipName': 'La maison d\'Asie',
    'ShipAddress': '1 rue Alsace-Lorraine',
    'ShipCity': 'Toulouse',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10494,
    'CustomerID': 'COMMI',
    'OrderDate': '1997-04-02T00:00:00.000Z',
    'ShippedDate': '1997-04-09T00:00:00.000Z',
    'Freight': 65.99,
    'ShipName': 'Comércio Mineiro',
    'ShipAddress': 'Av. dos Lusíadas, 23',
    'ShipCity': 'Sao Paulo',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10495,
    'CustomerID': 'LAUGB',
    'OrderDate': '1997-04-03T00:00:00.000Z',
    'ShippedDate': '1997-04-11T00:00:00.000Z',
    'Freight': 4.65,
    'ShipName': 'Laughing Bacchus Wine Cellars',
    'ShipAddress': '2319 Elm St.',
    'ShipCity': 'Vancouver',
    'ShipRegion': 'BC',
    'ShipCountry': 'Canada'
  },
  {
    'OrderID': 10496,
    'CustomerID': 'TRADH',
    'OrderDate': '1997-04-04T00:00:00.000Z',
    'ShippedDate': '1997-04-07T00:00:00.000Z',
    'Freight': 46.77,
    'ShipName': 'Tradição Hipermercados',
    'ShipAddress': 'Av. Inês de Castro, 414',
    'ShipCity': 'Sao Paulo',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  },
  {

```



```

    'OrderID': 10497,
    'CustomerID': 'LEHMS',
    'OrderDate': '1997-04-04T00:00:00.000Z',
    'ShippedDate': '1997-04-07T00:00:00.000Z',
    'Freight': 36.21,
    'ShipName': 'Lehmanns Marktstand',
    'ShipAddress': 'Magazinweg 7',
    'ShipCity': 'Frankfurt a.M.',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10498,
    'CustomerID': 'HILAA',
    'OrderDate': '1997-04-07T00:00:00.000Z',
    'ShippedDate': '1997-04-11T00:00:00.000Z',
    'Freight': 29.75,
    'ShipName': 'HILARION-Abastos',
    'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
    'ShipCity': 'San Cristóbal',
    'ShipRegion': 'Táchira',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10499,
    'CustomerID': 'LILAS',
    'OrderDate': '1997-04-08T00:00:00.000Z',
    'ShippedDate': '1997-04-16T00:00:00.000Z',
    'Freight': 102.02,
    'ShipName': 'LILA-Supermercado',
    'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
    'ShipCity': 'Barquisimeto',
    'ShipRegion': 'Lara',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10500,
    'CustomerID': 'LAMAI',
    'OrderDate': '1997-04-09T00:00:00.000Z',
    'ShippedDate': '1997-04-17T00:00:00.000Z',
    'Freight': 42.68,
    'ShipName': 'La maison d\'Asie',
    'ShipAddress': '1 rue Alsace-Lorraine',
    'ShipCity': 'Toulouse',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10501,
    'CustomerID': 'BLAUS',
    'OrderDate': '1997-04-09T00:00:00.000Z',
    'ShippedDate': '1997-04-16T00:00:00.000Z',
    'Freight': 8.85,
    'ShipName': 'Blauer See Delikatessen',
    'ShipAddress': 'Forsterstr. 57',
    'ShipCity': 'Mannheim',
    'ShipRegion': null,

```

```

    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10502,
    'CustomerID': 'PERIC',
    'OrderDate': '1997-04-10T00:00:00.000Z',
    'ShippedDate': '1997-04-29T00:00:00.000Z',
    'Freight': 69.32,
    'ShipName': 'Pericles Comidas clásicas',
    'ShipAddress': 'Calle Dr. Jorge Cash 321',
    'ShipCity': 'México D.F.',
    'ShipRegion': null,
    'ShipCountry': 'Mexico'
  },
  {
    'OrderID': 10503,
    'CustomerID': 'HUNGO',
    'OrderDate': '1997-04-11T00:00:00.000Z',
    'ShippedDate': '1997-04-16T00:00:00.000Z',
    'Freight': 16.74,
    'ShipName': 'Hungry Owl All-Night Grocers',
    'ShipAddress': '8 Johnstown Road',
    'ShipCity': 'Cork',
    'ShipRegion': 'Co. Cork',
    'ShipCountry': 'Ireland'
  },
  {
    'OrderID': 10504,
    'CustomerID': 'WHITC',
    'OrderDate': '1997-04-11T00:00:00.000Z',
    'ShippedDate': '1997-04-18T00:00:00.000Z',
    'Freight': 59.13,
    'ShipName': 'White Clover Markets',
    'ShipAddress': '1029 - 12th Ave. S.',
    'ShipCity': 'Seattle',
    'ShipRegion': 'WA',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10505,
    'CustomerID': 'MEREPE',
    'OrderDate': '1997-04-14T00:00:00.000Z',
    'ShippedDate': '1997-04-21T00:00:00.000Z',
    'Freight': 7.13,
    'ShipName': 'Mère Paillarde',
    'ShipAddress': '43 rue St. Laurent',
    'ShipCity': 'Montréal',
    'ShipRegion': 'Québec',
    'ShipCountry': 'Canada'
  },
  {
    'OrderID': 10506,
    'CustomerID': 'KOENE',
    'OrderDate': '1997-04-15T00:00:00.000Z',
    'ShippedDate': '1997-05-02T00:00:00.000Z',
    'Freight': 21.19,
    'ShipName': 'Königlich Essen',

```

```

      'ShipAddress': 'Maubelstr. 90',
      'ShipCity': 'Brandenburg',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10507,
      'CustomerID': 'ANTON',
      'OrderDate': '1997-04-15T00:00:00.000Z',
      'ShippedDate': '1997-04-22T00:00:00.000Z',
      'Freight': 47.45,
      'ShipName': 'Antonio Moreno Taquería',
      'ShipAddress': 'Mataderos 2312',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10508,
      'CustomerID': 'OTTIK',
      'OrderDate': '1997-04-16T00:00:00.000Z',
      'ShippedDate': '1997-05-13T00:00:00.000Z',
      'Freight': 4.99,
      'ShipName': 'Ottilies Käseladen',
      'ShipAddress': 'Mehrheimerstr. 369',
      'ShipCity': 'Köln',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10509,
      'CustomerID': 'BLAUS',
      'OrderDate': '1997-04-17T00:00:00.000Z',
      'ShippedDate': '1997-04-29T00:00:00.000Z',
      'Freight': 0.15,
      'ShipName': 'Blauer See Delikatessen',
      'ShipAddress': 'Forsterstr. 57',
      'ShipCity': 'Mannheim',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10510,
      'CustomerID': 'SAVEA',
      'OrderDate': '1997-04-18T00:00:00.000Z',
      'ShippedDate': '1997-04-28T00:00:00.000Z',
      'Freight': 367.63,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10511,
      'CustomerID': 'BONAP',
      'OrderDate': '1997-04-18T00:00:00.000Z',

```

```

      'ShippedDate': '1997-04-21T00:00:00.000Z',
      'Freight': 350.64,
      'ShipName': 'Bon app',
      'ShipAddress': '12, rue des Bouchers',
      'ShipCity': 'Marseille',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10512,
      'CustomerID': 'FAMIA',
      'OrderDate': '1997-04-21T00:00:00.000Z',
      'ShippedDate': '1997-04-24T00:00:00.000Z',
      'Freight': 3.53,
      'ShipName': 'Familia Arquibaldo',
      'ShipAddress': 'Rua Orós, 92',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10513,
      'CustomerID': 'WANDK',
      'OrderDate': '1997-04-22T00:00:00.000Z',
      'ShippedDate': '1997-04-28T00:00:00.000Z',
      'Freight': 105.65,
      'ShipName': 'Die Wandernde Kuh',
      'ShipAddress': 'Adenauerallee 900',
      'ShipCity': 'Stuttgart',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10514,
      'CustomerID': 'ERNSH',
      'OrderDate': '1997-04-22T00:00:00.000Z',
      'ShippedDate': '1997-05-16T00:00:00.000Z',
      'Freight': 789.95,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10515,
      'CustomerID': 'QUICK',
      'OrderDate': '1997-04-23T00:00:00.000Z',
      'ShippedDate': '1997-05-23T00:00:00.000Z',
      'Freight': 204.47,
      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    }
  ],
  {

```

```

      'OrderID': 10516,
      'CustomerID': 'HUNGO',
      'OrderDate': '1997-04-24T00:00:00.000Z',
      'ShippedDate': '1997-05-01T00:00:00.000Z',
      'Freight': 62.78,
      'ShipName': 'Hungry Owl All-Night Grocers',
      'ShipAddress': '8 Johnstown Road',
      'ShipCity': 'Cork',
      'ShipRegion': 'Co. Cork',
      'ShipCountry': 'Ireland'
    },
    {
      'OrderID': 10517,
      'CustomerID': 'NORTS',
      'OrderDate': '1997-04-24T00:00:00.000Z',
      'ShippedDate': '1997-04-29T00:00:00.000Z',
      'Freight': 32.07,
      'ShipName': 'North/South',
      'ShipAddress': 'South House 300 Queensbridge',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10518,
      'CustomerID': 'TORTU',
      'OrderDate': '1997-04-25T00:00:00.000Z',
      'ShippedDate': '1997-05-05T00:00:00.000Z',
      'Freight': 218.15,
      'ShipName': 'Tortuga Restaurante',
      'ShipAddress': 'Avda. Azteca 123',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10519,
      'CustomerID': 'CHOPS',
      'OrderDate': '1997-04-28T00:00:00.000Z',
      'ShippedDate': '1997-05-01T00:00:00.000Z',
      'Freight': 91.76,
      'ShipName': 'Chop-suey Chinese',
      'ShipAddress': 'Hauptstr. 31',
      'ShipCity': 'Bern',
      'ShipRegion': null,
      'ShipCountry': 'Switzerland'
    },
    {
      'OrderID': 10520,
      'CustomerID': 'SANTG',
      'OrderDate': '1997-04-29T00:00:00.000Z',
      'ShippedDate': '1997-05-01T00:00:00.000Z',
      'Freight': 13.37,
      'ShipName': 'Santé Gourmet',
      'ShipAddress': 'Erling Skakkes gate 78',
      'ShipCity': 'Stavern',
      'ShipRegion': null,

```

```

    'ShipCountry': 'Norway'
  },
  {
    'OrderID': 10521,
    'CustomerID': 'CACTU',
    'OrderDate': '1997-04-29T00:00:00.000Z',
    'ShippedDate': '1997-05-02T00:00:00.000Z',
    'Freight': 17.22,
    'ShipName': 'Cactus Comidas para llevar',
    'ShipAddress': 'Cerrito 333',
    'ShipCity': 'Buenos Aires',
    'ShipRegion': null,
    'ShipCountry': 'Argentina'
  },
  {
    'OrderID': 10522,
    'CustomerID': 'LEHMS',
    'OrderDate': '1997-04-30T00:00:00.000Z',
    'ShippedDate': '1997-05-06T00:00:00.000Z',
    'Freight': 45.33,
    'ShipName': 'Lehmanns Marktstand',
    'ShipAddress': 'Magazinweg 7',
    'ShipCity': 'Frankfurt a.M.',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10523,
    'CustomerID': 'SEVES',
    'OrderDate': '1997-05-01T00:00:00.000Z',
    'ShippedDate': '1997-05-30T00:00:00.000Z',
    'Freight': 77.63,
    'ShipName': 'Seven Seas Imports',
    'ShipAddress': '90 Wadhurst Rd.',
    'ShipCity': 'London',
    'ShipRegion': null,
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10524,
    'CustomerID': 'BERGS',
    'OrderDate': '1997-05-01T00:00:00.000Z',
    'ShippedDate': '1997-05-07T00:00:00.000Z',
    'Freight': 244.79,
    'ShipName': 'Berglunds snabbköp',
    'ShipAddress': 'Berguvsvägen 8',
    'ShipCity': 'Luleå',
    'ShipRegion': null,
    'ShipCountry': 'Sweden'
  },
  {
    'OrderID': 10525,
    'CustomerID': 'BONAP',
    'OrderDate': '1997-05-02T00:00:00.000Z',
    'ShippedDate': '1997-05-23T00:00:00.000Z',
    'Freight': 11.06,
    'ShipName': 'Bon app',

```

```

      'ShipAddress': '12, rue des Bouchers',
      'ShipCity': 'Marseille',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10526,
      'CustomerID': 'WARTH',
      'OrderDate': '1997-05-05T00:00:00.000Z',
      'ShippedDate': '1997-05-15T00:00:00.000Z',
      'Freight': 58.59,
      'ShipName': 'Wartian Herkku',
      'ShipAddress': 'Torikatu 38',
      'ShipCity': 'Oulu',
      'ShipRegion': null,
      'ShipCountry': 'Finland'
    },
    {
      'OrderID': 10527,
      'CustomerID': 'QUICK',
      'OrderDate': '1997-05-05T00:00:00.000Z',
      'ShippedDate': '1997-05-07T00:00:00.000Z',
      'Freight': 41.9,
      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10528,
      'CustomerID': 'GREAL',
      'OrderDate': '1997-05-06T00:00:00.000Z',
      'ShippedDate': '1997-05-09T00:00:00.000Z',
      'Freight': 3.35,
      'ShipName': 'Great Lakes Food Market',
      'ShipAddress': '2732 Baker Blvd.',
      'ShipCity': 'Eugene',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10529,
      'CustomerID': 'MAISD',
      'OrderDate': '1997-05-07T00:00:00.000Z',
      'ShippedDate': '1997-05-09T00:00:00.000Z',
      'Freight': 66.69,
      'ShipName': 'Maison Dewey',
      'ShipAddress': 'Rue Joseph-Bens 532',
      'ShipCity': 'Bruxelles',
      'ShipRegion': null,
      'ShipCountry': 'Belgium'
    },
    {
      'OrderID': 10530,
      'CustomerID': 'PICCO',
      'OrderDate': '1997-05-08T00:00:00.000Z',

```

```

    'ShippedDate': '1997-05-12T00:00:00.000Z',
    'Freight': 339.22,
    'ShipName': 'Piccolo und mehr',
    'ShipAddress': 'Geislweg 14',
    'ShipCity': 'Salzburg',
    'ShipRegion': null,
    'ShipCountry': 'Austria'
  },
  {
    'OrderID': 10531,
    'CustomerID': 'OCEAN',
    'OrderDate': '1997-05-08T00:00:00.000Z',
    'ShippedDate': '1997-05-19T00:00:00.000Z',
    'Freight': 8.12,
    'ShipName': 'Océano Atlántico Ltda.',
    'ShipAddress': 'Ing. Gustavo Moncada 8585 Piso 20-A',
    'ShipCity': 'Buenos Aires',
    'ShipRegion': null,
    'ShipCountry': 'Argentina'
  },
  {
    'OrderID': 10532,
    'CustomerID': 'EASTC',
    'OrderDate': '1997-05-09T00:00:00.000Z',
    'ShippedDate': '1997-05-12T00:00:00.000Z',
    'Freight': 74.46,
    'ShipName': 'Eastern Connection',
    'ShipAddress': '35 King George',
    'ShipCity': 'London',
    'ShipRegion': null,
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10533,
    'CustomerID': 'FOLKO',
    'OrderDate': '1997-05-12T00:00:00.000Z',
    'ShippedDate': '1997-05-22T00:00:00.000Z',
    'Freight': 188.04,
    'ShipName': 'Folk och fä HB',
    'ShipAddress': 'Åkergatan 24',
    'ShipCity': 'Bräcke',
    'ShipRegion': null,
    'ShipCountry': 'Sweden'
  },
  {
    'OrderID': 10534,
    'CustomerID': 'LEHMS',
    'OrderDate': '1997-05-12T00:00:00.000Z',
    'ShippedDate': '1997-05-14T00:00:00.000Z',
    'Freight': 27.94,
    'ShipName': 'Lehmanns Marktstand',
    'ShipAddress': 'Magazinweg 7',
    'ShipCity': 'Frankfurt a.M.',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {

```



```

      'OrderID': 10535,
      'CustomerID': 'ANTON',
      'OrderDate': '1997-05-13T00:00:00.000Z',
      'ShippedDate': '1997-05-21T00:00:00.000Z',
      'Freight': 15.64,
      'ShipName': 'Antonio Moreno Taquería',
      'ShipAddress': 'Mataderos 2312',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10536,
      'CustomerID': 'LEHMS',
      'OrderDate': '1997-05-14T00:00:00.000Z',
      'ShippedDate': '1997-06-06T00:00:00.000Z',
      'Freight': 58.88,
      'ShipName': 'Lehmanns Marktstand',
      'ShipAddress': 'Magazinweg 7',
      'ShipCity': 'Frankfurt a.M.',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10537,
      'CustomerID': 'RICSU',
      'OrderDate': '1997-05-14T00:00:00.000Z',
      'ShippedDate': '1997-05-19T00:00:00.000Z',
      'Freight': 78.85,
      'ShipName': 'Richter Supermarkt',
      'ShipAddress': 'Starenweg 5',
      'ShipCity': 'Genève',
      'ShipRegion': null,
      'ShipCountry': 'Switzerland'
    },
    {
      'OrderID': 10538,
      'CustomerID': 'BSBEV',
      'OrderDate': '1997-05-15T00:00:00.000Z',
      'ShippedDate': '1997-05-16T00:00:00.000Z',
      'Freight': 4.87,
      'ShipName': 'B\ Beverages',
      'ShipAddress': 'Fauntleroy Circus',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10539,
      'CustomerID': 'BSBEV',
      'OrderDate': '1997-05-16T00:00:00.000Z',
      'ShippedDate': '1997-05-23T00:00:00.000Z',
      'Freight': 12.36,
      'ShipName': 'B\ Beverages',
      'ShipAddress': 'Fauntleroy Circus',
      'ShipCity': 'London',
      'ShipRegion': null,

```

```

    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10540,
    'CustomerID': 'QUICK',
    'OrderDate': '1997-05-19T00:00:00.000Z',
    'ShippedDate': '1997-06-13T00:00:00.000Z',
    'Freight': 1007.64,
    'ShipName': 'QUICK-Stop',
    'ShipAddress': 'Taucherstraße 10',
    'ShipCity': 'Cunewalde',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10541,
    'CustomerID': 'HANAR',
    'OrderDate': '1997-05-19T00:00:00.000Z',
    'ShippedDate': '1997-05-29T00:00:00.000Z',
    'Freight': 68.65,
    'ShipName': 'Hanari Carnes',
    'ShipAddress': 'Rua do Paço, 67',
    'ShipCity': 'Rio de Janeiro',
    'ShipRegion': 'RJ',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10542,
    'CustomerID': 'KOENE',
    'OrderDate': '1997-05-20T00:00:00.000Z',
    'ShippedDate': '1997-05-26T00:00:00.000Z',
    'Freight': 10.95,
    'ShipName': 'Königlich Essen',
    'ShipAddress': 'Maubelstr. 90',
    'ShipCity': 'Brandenburg',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10543,
    'CustomerID': 'LILAS',
    'OrderDate': '1997-05-21T00:00:00.000Z',
    'ShippedDate': '1997-05-23T00:00:00.000Z',
    'Freight': 48.17,
    'ShipName': 'LILA-Supermercado',
    'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
    'ShipCity': 'Barquisimeto',
    'ShipRegion': 'Lara',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10544,
    'CustomerID': 'LONEP',
    'OrderDate': '1997-05-21T00:00:00.000Z',
    'ShippedDate': '1997-05-30T00:00:00.000Z',
    'Freight': 24.91,
    'ShipName': 'Lonesome Pine Restaurant',

```

```

      'ShipAddress': '89 Chiaroscuro Rd.',
      'ShipCity': 'Portland',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10545,
      'CustomerID': 'LAZYK',
      'OrderDate': '1997-05-22T00:00:00.000Z',
      'ShippedDate': '1997-06-26T00:00:00.000Z',
      'Freight': 11.92,
      'ShipName': 'Lazy K Kountry Store',
      'ShipAddress': '12 Orchestra Terrace',
      'ShipCity': 'Walla Walla',
      'ShipRegion': 'WA',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10546,
      'CustomerID': 'VICTE',
      'OrderDate': '1997-05-23T00:00:00.000Z',
      'ShippedDate': '1997-05-27T00:00:00.000Z',
      'Freight': 194.72,
      'ShipName': 'Victuailles en stock',
      'ShipAddress': '2, rue du Commerce',
      'ShipCity': 'Lyon',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10547,
      'CustomerID': 'SEVES',
      'OrderDate': '1997-05-23T00:00:00.000Z',
      'ShippedDate': '1997-06-02T00:00:00.000Z',
      'Freight': 178.43,
      'ShipName': 'Seven Seas Imports',
      'ShipAddress': '90 Wadhurst Rd.',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10548,
      'CustomerID': 'TOMSP',
      'OrderDate': '1997-05-26T00:00:00.000Z',
      'ShippedDate': '1997-06-02T00:00:00.000Z',
      'Freight': 1.43,
      'ShipName': 'Toms Spezialitäten',
      'ShipAddress': 'Luisenstr. 48',
      'ShipCity': 'Münster',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10549,
      'CustomerID': 'QUICK',
      'OrderDate': '1997-05-27T00:00:00.000Z',

```

```

    'ShippedDate': '1997-05-30T00:00:00.000Z',
    'Freight': 171.24,
    'ShipName': 'QUICK-Stop',
    'ShipAddress': 'Taucherstraße 10',
    'ShipCity': 'Cunewalde',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10550,
    'CustomerID': 'GODOS',
    'OrderDate': '1997-05-28T00:00:00.000Z',
    'ShippedDate': '1997-06-06T00:00:00.000Z',
    'Freight': 4.32,
    'ShipName': 'Godos Cocina Típica',
    'ShipAddress': 'C/ Romero, 33',
    'ShipCity': 'Sevilla',
    'ShipRegion': null,
    'ShipCountry': 'Spain'
  },
  {
    'OrderID': 10551,
    'CustomerID': 'FURIB',
    'OrderDate': '1997-05-28T00:00:00.000Z',
    'ShippedDate': '1997-06-06T00:00:00.000Z',
    'Freight': 72.95,
    'ShipName': 'Furia Bacalhau e Frutos do Mar',
    'ShipAddress': 'Jardim das rosas n. 32',
    'ShipCity': 'Lisboa',
    'ShipRegion': null,
    'ShipCountry': 'Portugal'
  },
  {
    'OrderID': 10552,
    'CustomerID': 'HILAA',
    'OrderDate': '1997-05-29T00:00:00.000Z',
    'ShippedDate': '1997-06-05T00:00:00.000Z',
    'Freight': 83.22,
    'ShipName': 'HILARION-Abastos',
    'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
    'ShipCity': 'San Cristóbal',
    'ShipRegion': 'Táchira',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10553,
    'CustomerID': 'WARTH',
    'OrderDate': '1997-05-30T00:00:00.000Z',
    'ShippedDate': '1997-06-03T00:00:00.000Z',
    'Freight': 149.49,
    'ShipName': 'Wartian Herkku',
    'ShipAddress': 'Torikatu 38',
    'ShipCity': 'Oulu',
    'ShipRegion': null,
    'ShipCountry': 'Finland'
  },
  {

```

```

      'OrderID': 10554,
      'CustomerID': 'OTTIK',
      'OrderDate': '1997-05-30T00:00:00.000Z',
      'ShippedDate': '1997-06-05T00:00:00.000Z',
      'Freight': 120.97,
      'ShipName': 'Ottilies Käseladen',
      'ShipAddress': 'Mehrheimerstr. 369',
      'ShipCity': 'Köln',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10555,
      'CustomerID': 'SAVEA',
      'OrderDate': '1997-06-02T00:00:00.000Z',
      'ShippedDate': '1997-06-04T00:00:00.000Z',
      'Freight': 252.49,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10556,
      'CustomerID': 'SIMOB',
      'OrderDate': '1997-06-03T00:00:00.000Z',
      'ShippedDate': '1997-06-13T00:00:00.000Z',
      'Freight': 9.8,
      'ShipName': 'Simons bistro',
      'ShipAddress': 'Vinbæltet 34',
      'ShipCity': 'Kobenhavn',
      'ShipRegion': null,
      'ShipCountry': 'Denmark'
    },
    {
      'OrderID': 10557,
      'CustomerID': 'LEHMS',
      'OrderDate': '1997-06-03T00:00:00.000Z',
      'ShippedDate': '1997-06-06T00:00:00.000Z',
      'Freight': 96.72,
      'ShipName': 'Lehmanns Marktstand',
      'ShipAddress': 'Magazinweg 7',
      'ShipCity': 'Frankfurt a.M.',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10558,
      'CustomerID': 'AROUT',
      'OrderDate': '1997-06-04T00:00:00.000Z',
      'ShippedDate': '1997-06-10T00:00:00.000Z',
      'Freight': 72.97,
      'ShipName': 'Around the Horn',
      'ShipAddress': 'Brook Farm Stratford St. Mary',
      'ShipCity': 'Colchester',
      'ShipRegion': 'Essex',

```

```

    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10559,
    'CustomerID': 'BLONP',
    'OrderDate': '1997-06-05T00:00:00.000Z',
    'ShippedDate': '1997-06-13T00:00:00.000Z',
    'Freight': 8.05,
    'ShipName': 'Blondel père et fils',
    'ShipAddress': '24, place Kléber',
    'ShipCity': 'Strasbourg',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10560,
    'CustomerID': 'FRANK',
    'OrderDate': '1997-06-06T00:00:00.000Z',
    'ShippedDate': '1997-06-09T00:00:00.000Z',
    'Freight': 36.65,
    'ShipName': 'Frankenversand',
    'ShipAddress': 'Berliner Platz 43',
    'ShipCity': 'München',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10561,
    'CustomerID': 'FOLKO',
    'OrderDate': '1997-06-06T00:00:00.000Z',
    'ShippedDate': '1997-06-09T00:00:00.000Z',
    'Freight': 242.21,
    'ShipName': 'Folk och fä HB',
    'ShipAddress': 'Åkergatan 24',
    'ShipCity': 'Bräcke',
    'ShipRegion': null,
    'ShipCountry': 'Sweden'
  },
  {
    'OrderID': 10562,
    'CustomerID': 'REGGC',
    'OrderDate': '1997-06-09T00:00:00.000Z',
    'ShippedDate': '1997-06-12T00:00:00.000Z',
    'Freight': 22.95,
    'ShipName': 'Reggiani Caseifici',
    'ShipAddress': 'Strada Provinciale 124',
    'ShipCity': 'Reggio Emilia',
    'ShipRegion': null,
    'ShipCountry': 'Italy'
  },
  {
    'OrderID': 10563,
    'CustomerID': 'RICAR',
    'OrderDate': '1997-06-10T00:00:00.000Z',
    'ShippedDate': '1997-06-24T00:00:00.000Z',
    'Freight': 60.43,
    'ShipName': 'Ricardo Adocicados',

```

```

      'ShipAddress': 'Av. Copacabana, 267',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10564,
      'CustomerID': 'RATTC',
      'OrderDate': '1997-06-10T00:00:00.000Z',
      'ShippedDate': '1997-06-16T00:00:00.000Z',
      'Freight': 13.75,
      'ShipName': 'Rattlesnake Canyon Grocery',
      'ShipAddress': '2817 Milton Dr.',
      'ShipCity': 'Albuquerque',
      'ShipRegion': 'NM',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10565,
      'CustomerID': 'MEREP',
      'OrderDate': '1997-06-11T00:00:00.000Z',
      'ShippedDate': '1997-06-18T00:00:00.000Z',
      'Freight': 7.15,
      'ShipName': 'Mère Paillarde',
      'ShipAddress': '43 rue St. Laurent',
      'ShipCity': 'Montréal',
      'ShipRegion': 'Québec',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10566,
      'CustomerID': 'BLONP',
      'OrderDate': '1997-06-12T00:00:00.000Z',
      'ShippedDate': '1997-06-18T00:00:00.000Z',
      'Freight': 88.4,
      'ShipName': 'Blondel père et fils',
      'ShipAddress': '24, place Kléber',
      'ShipCity': 'Strasbourg',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10567,
      'CustomerID': 'HUNGO',
      'OrderDate': '1997-06-12T00:00:00.000Z',
      'ShippedDate': '1997-06-17T00:00:00.000Z',
      'Freight': 33.97,
      'ShipName': 'Hungry Owl All-Night Grocers',
      'ShipAddress': '8 Johnstown Road',
      'ShipCity': 'Cork',
      'ShipRegion': 'Co. Cork',
      'ShipCountry': 'Ireland'
    },
    {
      'OrderID': 10568,
      'CustomerID': 'GALED',
      'OrderDate': '1997-06-13T00:00:00.000Z',

```

```

    'ShippedDate': '1997-07-09T00:00:00.000Z',
    'Freight': 6.54,
    'ShipName': 'Galería del gastronómo',
    'ShipAddress': 'Rambla de Cataluña, 23',
    'ShipCity': 'Barcelona',
    'ShipRegion': null,
    'ShipCountry': 'Spain'
  },
  {
    'OrderID': 10569,
    'CustomerID': 'RATTC',
    'OrderDate': '1997-06-16T00:00:00.000Z',
    'ShippedDate': '1997-07-11T00:00:00.000Z',
    'Freight': 58.98,
    'ShipName': 'Rattlesnake Canyon Grocery',
    'ShipAddress': '2817 Milton Dr.',
    'ShipCity': 'Albuquerque',
    'ShipRegion': 'NM',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10570,
    'CustomerID': 'MEREP',
    'OrderDate': '1997-06-17T00:00:00.000Z',
    'ShippedDate': '1997-06-19T00:00:00.000Z',
    'Freight': 188.99,
    'ShipName': 'Mère Paillarde',
    'ShipAddress': '43 rue St. Laurent',
    'ShipCity': 'Montréal',
    'ShipRegion': 'Québec',
    'ShipCountry': 'Canada'
  },
  {
    'OrderID': 10571,
    'CustomerID': 'ERNSH',
    'OrderDate': '1997-06-17T00:00:00.000Z',
    'ShippedDate': '1997-07-04T00:00:00.000Z',
    'Freight': 26.06,
    'ShipName': 'Ernst Handel',
    'ShipAddress': 'Kirchgasse 6',
    'ShipCity': 'Graz',
    'ShipRegion': null,
    'ShipCountry': 'Austria'
  },
  {
    'OrderID': 10572,
    'CustomerID': 'BERGS',
    'OrderDate': '1997-06-18T00:00:00.000Z',
    'ShippedDate': '1997-06-25T00:00:00.000Z',
    'Freight': 116.43,
    'ShipName': 'Berglunds snabbköp',
    'ShipAddress': 'Berguvsvägen 8',
    'ShipCity': 'Luleå',
    'ShipRegion': null,
    'ShipCountry': 'Sweden'
  },
  {

```



```

      'OrderID': 10573,
      'CustomerID': 'ANTON',
      'OrderDate': '1997-06-19T00:00:00.000Z',
      'ShippedDate': '1997-06-20T00:00:00.000Z',
      'Freight': 84.84,
      'ShipName': 'Antonio Moreno Taquería',
      'ShipAddress': 'Mataderos 2312',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10574,
      'CustomerID': 'TRAIH',
      'OrderDate': '1997-06-19T00:00:00.000Z',
      'ShippedDate': '1997-06-30T00:00:00.000Z',
      'Freight': 37.6,
      'ShipName': 'Trail\' Head Gourmet Provisioners',
      'ShipAddress': '722 DaVinci Blvd.',
      'ShipCity': 'Kirkland',
      'ShipRegion': 'WA',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10575,
      'CustomerID': 'MORGK',
      'OrderDate': '1997-06-20T00:00:00.000Z',
      'ShippedDate': '1997-06-30T00:00:00.000Z',
      'Freight': 127.34,
      'ShipName': 'Morgenstern Gesundkost',
      'ShipAddress': 'Heerstr. 22',
      'ShipCity': 'Leipzig',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10576,
      'CustomerID': 'TORTU',
      'OrderDate': '1997-06-23T00:00:00.000Z',
      'ShippedDate': '1997-06-30T00:00:00.000Z',
      'Freight': 18.56,
      'ShipName': 'Tortuga Restaurante',
      'ShipAddress': 'Avda. Azteca 123',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10577,
      'CustomerID': 'TRAIH',
      'OrderDate': '1997-06-23T00:00:00.000Z',
      'ShippedDate': '1997-06-30T00:00:00.000Z',
      'Freight': 25.41,
      'ShipName': 'Trail\' Head Gourmet Provisioners',
      'ShipAddress': '722 DaVinci Blvd.',
      'ShipCity': 'Kirkland',
      'ShipRegion': 'WA',

```

```

      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10578,
      'CustomerID': 'BSBEV',
      'OrderDate': '1997-06-24T00:00:00.000Z',
      'ShippedDate': '1997-07-25T00:00:00.000Z',
      'Freight': 29.6,
      'ShipName': 'B\ ' Beverages',
      'ShipAddress': 'Fauntleroy Circus',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10579,
      'CustomerID': 'LETSS',
      'OrderDate': '1997-06-25T00:00:00.000Z',
      'ShippedDate': '1997-07-04T00:00:00.000Z',
      'Freight': 13.73,
      'ShipName': 'Let\ ' Stop N Shop',
      'ShipAddress': '87 Polk St. Suite 5',
      'ShipCity': 'San Francisco',
      'ShipRegion': 'CA',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10580,
      'CustomerID': 'OTTIK',
      'OrderDate': '1997-06-26T00:00:00.000Z',
      'ShippedDate': '1997-07-01T00:00:00.000Z',
      'Freight': 75.89,
      'ShipName': 'Ottilies Käseladen',
      'ShipAddress': 'Mehrheimerstr. 369',
      'ShipCity': 'Köln',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10581,
      'CustomerID': 'FAMIA',
      'OrderDate': '1997-06-26T00:00:00.000Z',
      'ShippedDate': '1997-07-02T00:00:00.000Z',
      'Freight': 3.01,
      'ShipName': 'Familia Arquibaldo',
      'ShipAddress': 'Rua Orós, 92',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10582,
      'CustomerID': 'BLAUS',
      'OrderDate': '1997-06-27T00:00:00.000Z',
      'ShippedDate': '1997-07-14T00:00:00.000Z',
      'Freight': 27.71,
      'ShipName': 'Blauer See Delikatessen',

```

```

      'ShipAddress': 'Forsterstr. 57',
      'ShipCity': 'Mannheim',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10583,
      'CustomerID': 'WARTH',
      'OrderDate': '1997-06-30T00:00:00.000Z',
      'ShippedDate': '1997-07-04T00:00:00.000Z',
      'Freight': 7.28,
      'ShipName': 'Wartian Herkku',
      'ShipAddress': 'Torikatu 38',
      'ShipCity': 'Oulu',
      'ShipRegion': null,
      'ShipCountry': 'Finland'
    },
    {
      'OrderID': 10584,
      'CustomerID': 'BLONP',
      'OrderDate': '1997-06-30T00:00:00.000Z',
      'ShippedDate': '1997-07-04T00:00:00.000Z',
      'Freight': 59.14,
      'ShipName': 'Blondel père et fils',
      'ShipAddress': '24, place Kléber',
      'ShipCity': 'Strasbourg',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10585,
      'CustomerID': 'WELLI',
      'OrderDate': '1997-07-01T00:00:00.000Z',
      'ShippedDate': '1997-07-10T00:00:00.000Z',
      'Freight': 13.41,
      'ShipName': 'Wellington Importadora',
      'ShipAddress': 'Rua do Mercado, 12',
      'ShipCity': 'Resende',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10586,
      'CustomerID': 'REGGC',
      'OrderDate': '1997-07-02T00:00:00.000Z',
      'ShippedDate': '1997-07-09T00:00:00.000Z',
      'Freight': 0.48,
      'ShipName': 'Reggiani Caseifici',
      'ShipAddress': 'Strada Provinciale 124',
      'ShipCity': 'Reggio Emilia',
      'ShipRegion': null,
      'ShipCountry': 'Italy'
    },
    {
      'OrderID': 10587,
      'CustomerID': 'QUEDE',
      'OrderDate': '1997-07-02T00:00:00.000Z',

```

```

      'ShippedDate': '1997-07-09T00:00:00.000Z',
      'Freight': 62.52,
      'ShipName': 'Que Delícia',
      'ShipAddress': 'Rua da Panificadora, 12',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10588,
      'CustomerID': 'QUICK',
      'OrderDate': '1997-07-03T00:00:00.000Z',
      'ShippedDate': '1997-07-10T00:00:00.000Z',
      'Freight': 194.67,
      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10589,
      'CustomerID': 'GREAL',
      'OrderDate': '1997-07-04T00:00:00.000Z',
      'ShippedDate': '1997-07-14T00:00:00.000Z',
      'Freight': 4.42,
      'ShipName': 'Great Lakes Food Market',
      'ShipAddress': '2732 Baker Blvd.',
      'ShipCity': 'Eugene',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10590,
      'CustomerID': 'MEREP',
      'OrderDate': '1997-07-07T00:00:00.000Z',
      'ShippedDate': '1997-07-14T00:00:00.000Z',
      'Freight': 44.77,
      'ShipName': 'Mère Paillarde',
      'ShipAddress': '43 rue St. Laurent',
      'ShipCity': 'Montréal',
      'ShipRegion': 'Québec',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10591,
      'CustomerID': 'VAFFE',
      'OrderDate': '1997-07-07T00:00:00.000Z',
      'ShippedDate': '1997-07-16T00:00:00.000Z',
      'Freight': 55.92,
      'ShipName': 'Vaffeljernet',
      'ShipAddress': 'Smagsloget 45',
      'ShipCity': 'Århus',
      'ShipRegion': null,
      'ShipCountry': 'Denmark'
    }
  ],
  {

```

```

      'OrderID': 10592,
      'CustomerID': 'LEHMS',
      'OrderDate': '1997-07-08T00:00:00.000Z',
      'ShippedDate': '1997-07-16T00:00:00.000Z',
      'Freight': 32.1,
      'ShipName': 'Lehmanns Marktstand',
      'ShipAddress': 'Magazinweg 7',
      'ShipCity': 'Frankfurt a.M.',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10593,
      'CustomerID': 'LEHMS',
      'OrderDate': '1997-07-09T00:00:00.000Z',
      'ShippedDate': '1997-08-13T00:00:00.000Z',
      'Freight': 174.2,
      'ShipName': 'Lehmanns Marktstand',
      'ShipAddress': 'Magazinweg 7',
      'ShipCity': 'Frankfurt a.M.',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10594,
      'CustomerID': 'OLDWO',
      'OrderDate': '1997-07-09T00:00:00.000Z',
      'ShippedDate': '1997-07-16T00:00:00.000Z',
      'Freight': 5.24,
      'ShipName': 'Old World Delicatessen',
      'ShipAddress': '2743 Bering St.',
      'ShipCity': 'Anchorage',
      'ShipRegion': 'AK',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10595,
      'CustomerID': 'ERNSH',
      'OrderDate': '1997-07-10T00:00:00.000Z',
      'ShippedDate': '1997-07-14T00:00:00.000Z',
      'Freight': 96.78,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10596,
      'CustomerID': 'WHITC',
      'OrderDate': '1997-07-11T00:00:00.000Z',
      'ShippedDate': '1997-08-12T00:00:00.000Z',
      'Freight': 16.34,
      'ShipName': 'White Clover Markets',
      'ShipAddress': '1029 - 12th Ave. S.',
      'ShipCity': 'Seattle',
      'ShipRegion': 'WA',

```

```

    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10597,
    'CustomerID': 'PICCO',
    'OrderDate': '1997-07-11T00:00:00.000Z',
    'ShippedDate': '1997-07-18T00:00:00.000Z',
    'Freight': 35.12,
    'ShipName': 'Piccolo und mehr',
    'ShipAddress': 'Geislweg 14',
    'ShipCity': 'Salzburg',
    'ShipRegion': null,
    'ShipCountry': 'Austria'
  },
  {
    'OrderID': 10598,
    'CustomerID': 'RATTC',
    'OrderDate': '1997-07-14T00:00:00.000Z',
    'ShippedDate': '1997-07-18T00:00:00.000Z',
    'Freight': 44.42,
    'ShipName': 'Rattlesnake Canyon Grocery',
    'ShipAddress': '2817 Milton Dr.',
    'ShipCity': 'Albuquerque',
    'ShipRegion': 'NM',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10599,
    'CustomerID': 'BSBEV',
    'OrderDate': '1997-07-15T00:00:00.000Z',
    'ShippedDate': '1997-07-21T00:00:00.000Z',
    'Freight': 29.98,
    'ShipName': 'B Beverages',
    'ShipAddress': 'Fauntleroy Circus',
    'ShipCity': 'London',
    'ShipRegion': null,
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10600,
    'CustomerID': 'HUNGC',
    'OrderDate': '1997-07-16T00:00:00.000Z',
    'ShippedDate': '1997-07-21T00:00:00.000Z',
    'Freight': 45.13,
    'ShipName': 'Hungry Coyote Import Store',
    'ShipAddress': 'City Center Plaza 516 Main St.',
    'ShipCity': 'Elgin',
    'ShipRegion': 'OR',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10601,
    'CustomerID': 'HILAA',
    'OrderDate': '1997-07-16T00:00:00.000Z',
    'ShippedDate': '1997-07-22T00:00:00.000Z',
    'Freight': 58.3,
    'ShipName': 'HILARION-Abastos',

```

```

    'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
    'ShipCity': 'San Cristóbal',
    'ShipRegion': 'Táchira',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10602,
    'CustomerID': 'VAFFE',
    'OrderDate': '1997-07-17T00:00:00.000Z',
    'ShippedDate': '1997-07-22T00:00:00.000Z',
    'Freight': 2.92,
    'ShipName': 'Vaffeljernet',
    'ShipAddress': 'Smagsloget 45',
    'ShipCity': 'Århus',
    'ShipRegion': null,
    'ShipCountry': 'Denmark'
  },
  {
    'OrderID': 10603,
    'CustomerID': 'SAVEA',
    'OrderDate': '1997-07-18T00:00:00.000Z',
    'ShippedDate': '1997-08-08T00:00:00.000Z',
    'Freight': 48.77,
    'ShipName': 'Save-a-lot Markets',
    'ShipAddress': '187 Suffolk Ln.',
    'ShipCity': 'Boise',
    'ShipRegion': 'ID',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10604,
    'CustomerID': 'FURIB',
    'OrderDate': '1997-07-18T00:00:00.000Z',
    'ShippedDate': '1997-07-29T00:00:00.000Z',
    'Freight': 7.46,
    'ShipName': 'Furia Bacalhau e Frutos do Mar',
    'ShipAddress': 'Jardim das rosas n. 32',
    'ShipCity': 'Lisboa',
    'ShipRegion': null,
    'ShipCountry': 'Portugal'
  },
  {
    'OrderID': 10605,
    'CustomerID': 'MEREPE',
    'OrderDate': '1997-07-21T00:00:00.000Z',
    'ShippedDate': '1997-07-29T00:00:00.000Z',
    'Freight': 379.13,
    'ShipName': 'Mère Paillarde',
    'ShipAddress': '43 rue St. Laurent',
    'ShipCity': 'Montréal',
    'ShipRegion': 'Québec',
    'ShipCountry': 'Canada'
  },
  {
    'OrderID': 10606,
    'CustomerID': 'TRADH',
    'OrderDate': '1997-07-22T00:00:00.000Z',

```

```

      'ShippedDate': '1997-07-31T00:00:00.000Z',
      'Freight': 79.4,
      'ShipName': 'Tradição Hipermercados',
      'ShipAddress': 'Av. Inês de Castro, 414',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10607,
      'CustomerID': 'SAVEA',
      'OrderDate': '1997-07-22T00:00:00.000Z',
      'ShippedDate': '1997-07-25T00:00:00.000Z',
      'Freight': 200.24,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10608,
      'CustomerID': 'TOMSP',
      'OrderDate': '1997-07-23T00:00:00.000Z',
      'ShippedDate': '1997-08-01T00:00:00.000Z',
      'Freight': 27.79,
      'ShipName': 'Toms Spezialitäten',
      'ShipAddress': 'Luisenstr. 48',
      'ShipCity': 'Münster',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10609,
      'CustomerID': 'DUMON',
      'OrderDate': '1997-07-24T00:00:00.000Z',
      'ShippedDate': '1997-07-30T00:00:00.000Z',
      'Freight': 1.85,
      'ShipName': 'Du monde entier',
      'ShipAddress': '67, rue des Cinquante Otages',
      'ShipCity': 'Nantes',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10610,
      'CustomerID': 'LAMAI',
      'OrderDate': '1997-07-25T00:00:00.000Z',
      'ShippedDate': '1997-08-06T00:00:00.000Z',
      'Freight': 26.78,
      'ShipName': 'La maison d\'Asie',
      'ShipAddress': '1 rue Alsace-Lorraine',
      'ShipCity': 'Toulouse',
      'ShipRegion': null,
      'ShipCountry': 'France'
    }
  ],
  {

```



```

      'OrderID': 10611,
      'CustomerID': 'WOLZA',
      'OrderDate': '1997-07-25T00:00:00.000Z',
      'ShippedDate': '1997-08-01T00:00:00.000Z',
      'Freight': 80.65,
      'ShipName': 'Wolski Zajazd',
      'ShipAddress': 'ul. Filtrowa 68',
      'ShipCity': 'Warszawa',
      'ShipRegion': null,
      'ShipCountry': 'Poland'
    },
    {
      'OrderID': 10612,
      'CustomerID': 'SAVEA',
      'OrderDate': '1997-07-28T00:00:00.000Z',
      'ShippedDate': '1997-08-01T00:00:00.000Z',
      'Freight': 544.08,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10613,
      'CustomerID': 'HILAA',
      'OrderDate': '1997-07-29T00:00:00.000Z',
      'ShippedDate': '1997-08-01T00:00:00.000Z',
      'Freight': 8.11,
      'ShipName': 'HILARION-Abastos',
      'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
      'ShipCity': 'San Cristóbal',
      'ShipRegion': 'Táchira',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10614,
      'CustomerID': 'BLAUS',
      'OrderDate': '1997-07-29T00:00:00.000Z',
      'ShippedDate': '1997-08-01T00:00:00.000Z',
      'Freight': 1.93,
      'ShipName': 'Blauer See Delikatessen',
      'ShipAddress': 'Forsterstr. 57',
      'ShipCity': 'Mannheim',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10615,
      'CustomerID': 'WILMK',
      'OrderDate': '1997-07-30T00:00:00.000Z',
      'ShippedDate': '1997-08-06T00:00:00.000Z',
      'Freight': 0.75,
      'ShipName': 'Wilman Kala',
      'ShipAddress': 'Keskuskatu 45',
      'ShipCity': 'Helsinki',
      'ShipRegion': null,

```

```

    'ShipCountry': 'Finland'
  },
  {
    'OrderID': 10616,
    'CustomerID': 'GREAL',
    'OrderDate': '1997-07-31T00:00:00.000Z',
    'ShippedDate': '1997-08-05T00:00:00.000Z',
    'Freight': 116.53,
    'ShipName': 'Great Lakes Food Market',
    'ShipAddress': '2732 Baker Blvd.',
    'ShipCity': 'Eugene',
    'ShipRegion': 'OR',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10617,
    'CustomerID': 'GREAL',
    'OrderDate': '1997-07-31T00:00:00.000Z',
    'ShippedDate': '1997-08-04T00:00:00.000Z',
    'Freight': 18.53,
    'ShipName': 'Great Lakes Food Market',
    'ShipAddress': '2732 Baker Blvd.',
    'ShipCity': 'Eugene',
    'ShipRegion': 'OR',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10618,
    'CustomerID': 'MEREP',
    'OrderDate': '1997-08-01T00:00:00.000Z',
    'ShippedDate': '1997-08-08T00:00:00.000Z',
    'Freight': 154.68,
    'ShipName': 'Mère Paillarde',
    'ShipAddress': '43 rue St. Laurent',
    'ShipCity': 'Montréal',
    'ShipRegion': 'Québec',
    'ShipCountry': 'Canada'
  },
  {
    'OrderID': 10619,
    'CustomerID': 'MEREP',
    'OrderDate': '1997-08-04T00:00:00.000Z',
    'ShippedDate': '1997-08-07T00:00:00.000Z',
    'Freight': 91.05,
    'ShipName': 'Mère Paillarde',
    'ShipAddress': '43 rue St. Laurent',
    'ShipCity': 'Montréal',
    'ShipRegion': 'Québec',
    'ShipCountry': 'Canada'
  },
  {
    'OrderID': 10620,
    'CustomerID': 'LAUGB',
    'OrderDate': '1997-08-05T00:00:00.000Z',
    'ShippedDate': '1997-08-14T00:00:00.000Z',
    'Freight': 0.94,
    'ShipName': 'Laughing Bacchus Wine Cellars',

```

```

      'ShipAddress': '2319 Elm St.',
      'ShipCity': 'Vancouver',
      'ShipRegion': 'BC',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10621,
      'CustomerID': 'ISLAT',
      'OrderDate': '1997-08-05T00:00:00.000Z',
      'ShippedDate': '1997-08-11T00:00:00.000Z',
      'Freight': 23.73,
      'ShipName': 'Island Trading',
      'ShipAddress': 'Garden House Crowther Way',
      'ShipCity': 'Cowes',
      'ShipRegion': 'Isle of Wight',
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10622,
      'CustomerID': 'RICAR',
      'OrderDate': '1997-08-06T00:00:00.000Z',
      'ShippedDate': '1997-08-11T00:00:00.000Z',
      'Freight': 50.97,
      'ShipName': 'Ricardo Adocicados',
      'ShipAddress': 'Av. Copacabana, 267',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10623,
      'CustomerID': 'FRANK',
      'OrderDate': '1997-08-07T00:00:00.000Z',
      'ShippedDate': '1997-08-12T00:00:00.000Z',
      'Freight': 97.18,
      'ShipName': 'Frankenversand',
      'ShipAddress': 'Berliner Platz 43',
      'ShipCity': 'München',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10624,
      'CustomerID': 'THECR',
      'OrderDate': '1997-08-07T00:00:00.000Z',
      'ShippedDate': '1997-08-19T00:00:00.000Z',
      'Freight': 94.8,
      'ShipName': 'The Cracker Box',
      'ShipAddress': '55 Grizzly Peak Rd.',
      'ShipCity': 'Butte',
      'ShipRegion': 'MT',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10625,
      'CustomerID': 'ANATR',
      'OrderDate': '1997-08-08T00:00:00.000Z',

```

```

      'ShippedDate': '1997-08-14T00:00:00.000Z',
      'Freight': 43.9,
      'ShipName': 'Ana Trujillo Emparedados y helados',
      'ShipAddress': 'Avda. de la Constitución 2222',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10626,
      'CustomerID': 'BERGS',
      'OrderDate': '1997-08-11T00:00:00.000Z',
      'ShippedDate': '1997-08-20T00:00:00.000Z',
      'Freight': 138.69,
      'ShipName': 'Berglunds snabbköp',
      'ShipAddress': 'Berguvsvägen 8',
      'ShipCity': 'Luleå',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 10627,
      'CustomerID': 'SAVEA',
      'OrderDate': '1997-08-11T00:00:00.000Z',
      'ShippedDate': '1997-08-21T00:00:00.000Z',
      'Freight': 107.46,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10628,
      'CustomerID': 'BLONP',
      'OrderDate': '1997-08-12T00:00:00.000Z',
      'ShippedDate': '1997-08-20T00:00:00.000Z',
      'Freight': 30.36,
      'ShipName': 'Blondel père et fils',
      'ShipAddress': '24, place Kléber',
      'ShipCity': 'Strasbourg',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10629,
      'CustomerID': 'GODOS',
      'OrderDate': '1997-08-12T00:00:00.000Z',
      'ShippedDate': '1997-08-20T00:00:00.000Z',
      'Freight': 85.46,
      'ShipName': 'Godos Cocina Típica',
      'ShipAddress': 'C/ Romero, 33',
      'ShipCity': 'Sevilla',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    }
  ],
  {

```

```

      'OrderID': 10630,
      'CustomerID': 'KOENE',
      'OrderDate': '1997-08-13T00:00:00.000Z',
      'ShippedDate': '1997-08-19T00:00:00.000Z',
      'Freight': 32.35,
      'ShipName': 'Königlich Essen',
      'ShipAddress': 'Maubelstr. 90',
      'ShipCity': 'Brandenburg',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10631,
      'CustomerID': 'LAMAI',
      'OrderDate': '1997-08-14T00:00:00.000Z',
      'ShippedDate': '1997-08-15T00:00:00.000Z',
      'Freight': 0.87,
      'ShipName': 'La maison d\'Asie',
      'ShipAddress': '1 rue Alsace-Lorraine',
      'ShipCity': 'Toulouse',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10632,
      'CustomerID': 'WANDK',
      'OrderDate': '1997-08-14T00:00:00.000Z',
      'ShippedDate': '1997-08-19T00:00:00.000Z',
      'Freight': 41.38,
      'ShipName': 'Die Wandernde Kuh',
      'ShipAddress': 'Adenauerallee 900',
      'ShipCity': 'Stuttgart',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10633,
      'CustomerID': 'ERNSH',
      'OrderDate': '1997-08-15T00:00:00.000Z',
      'ShippedDate': '1997-08-18T00:00:00.000Z',
      'Freight': 477.9,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10634,
      'CustomerID': 'FOLIG',
      'OrderDate': '1997-08-15T00:00:00.000Z',
      'ShippedDate': '1997-08-21T00:00:00.000Z',
      'Freight': 487.38,
      'ShipName': 'Folies gourmandes',
      'ShipAddress': '184, chaussée de Tournai',
      'ShipCity': 'Lille',
      'ShipRegion': null,

```

```

    'ShipCountry': 'France'
  },
  {
    'OrderID': 10635,
    'CustomerID': 'MAGAA',
    'OrderDate': '1997-08-18T00:00:00.000Z',
    'ShippedDate': '1997-08-21T00:00:00.000Z',
    'Freight': 47.46,
    'ShipName': 'Magazzini Alimentari Riuniti',
    'ShipAddress': 'Via Ludovico il Moro 22',
    'ShipCity': 'Bergamo',
    'ShipRegion': null,
    'ShipCountry': 'Italy'
  },
  {
    'OrderID': 10636,
    'CustomerID': 'WARTH',
    'OrderDate': '1997-08-19T00:00:00.000Z',
    'ShippedDate': '1997-08-26T00:00:00.000Z',
    'Freight': 1.15,
    'ShipName': 'Wartian Herkku',
    'ShipAddress': 'Torikatu 38',
    'ShipCity': 'Oulu',
    'ShipRegion': null,
    'ShipCountry': 'Finland'
  },
  {
    'OrderID': 10637,
    'CustomerID': 'QUEEN',
    'OrderDate': '1997-08-19T00:00:00.000Z',
    'ShippedDate': '1997-08-26T00:00:00.000Z',
    'Freight': 201.29,
    'ShipName': 'Queen Cozinha',
    'ShipAddress': 'Alameda dos Canários, 891',
    'ShipCity': 'Sao Paulo',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10638,
    'CustomerID': 'LINOD',
    'OrderDate': '1997-08-20T00:00:00.000Z',
    'ShippedDate': '1997-09-01T00:00:00.000Z',
    'Freight': 158.44,
    'ShipName': 'LINO-Delicateses',
    'ShipAddress': 'Ave. 5 de Mayo Porlamar',
    'ShipCity': 'I. de Margarita',
    'ShipRegion': 'Nueva Esparta',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10639,
    'CustomerID': 'SANTG',
    'OrderDate': '1997-08-20T00:00:00.000Z',
    'ShippedDate': '1997-08-27T00:00:00.000Z',
    'Freight': 38.64,
    'ShipName': 'Santé Gourmet',

```

```

      'ShipAddress': 'Erling Skakkes gate 78',
      'ShipCity': 'Stavern',
      'ShipRegion': null,
      'ShipCountry': 'Norway'
    },
    {
      'OrderID': 10640,
      'CustomerID': 'WANDK',
      'OrderDate': '1997-08-21T00:00:00.000Z',
      'ShippedDate': '1997-08-28T00:00:00.000Z',
      'Freight': 23.55,
      'ShipName': 'Die Wandernde Kuh',
      'ShipAddress': 'Adenauerallee 900',
      'ShipCity': 'Stuttgart',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10641,
      'CustomerID': 'HILAA',
      'OrderDate': '1997-08-22T00:00:00.000Z',
      'ShippedDate': '1997-08-26T00:00:00.000Z',
      'Freight': 179.61,
      'ShipName': 'HILARION-Abastos',
      'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
      'ShipCity': 'San Cristóbal',
      'ShipRegion': 'Táchira',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10642,
      'CustomerID': 'SIMOB',
      'OrderDate': '1997-08-22T00:00:00.000Z',
      'ShippedDate': '1997-09-05T00:00:00.000Z',
      'Freight': 41.89,
      'ShipName': 'Simons bistro',
      'ShipAddress': 'Vinbæltet 34',
      'ShipCity': 'Kobenhavn',
      'ShipRegion': null,
      'ShipCountry': 'Denmark'
    },
    {
      'OrderID': 10643,
      'CustomerID': 'ALFKI',
      'OrderDate': '1997-08-25T00:00:00.000Z',
      'ShippedDate': '1997-09-02T00:00:00.000Z',
      'Freight': 29.46,
      'ShipName': 'Alfreds Futterkiste',
      'ShipAddress': 'Obere Str. 57',
      'ShipCity': 'Berlin',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10644,
      'CustomerID': 'WELLI',
      'OrderDate': '1997-08-25T00:00:00.000Z',

```

```

      'ShippedDate': '1997-09-01T00:00:00.000Z',
      'Freight': 0.14,
      'ShipName': 'Wellington Importadora',
      'ShipAddress': 'Rua do Mercado, 12',
      'ShipCity': 'Resende',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10645,
      'CustomerID': 'HANAR',
      'OrderDate': '1997-08-26T00:00:00.000Z',
      'ShippedDate': '1997-09-02T00:00:00.000Z',
      'Freight': 12.41,
      'ShipName': 'Hanari Carnes',
      'ShipAddress': 'Rua do Paço, 67',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10646,
      'CustomerID': 'HUNGO',
      'OrderDate': '1997-08-27T00:00:00.000Z',
      'ShippedDate': '1997-09-03T00:00:00.000Z',
      'Freight': 142.33,
      'ShipName': 'Hungry Owl All-Night Grocers',
      'ShipAddress': '8 Johnstown Road',
      'ShipCity': 'Cork',
      'ShipRegion': 'Co. Cork',
      'ShipCountry': 'Ireland'
    },
    {
      'OrderID': 10647,
      'CustomerID': 'QUEDE',
      'OrderDate': '1997-08-27T00:00:00.000Z',
      'ShippedDate': '1997-09-03T00:00:00.000Z',
      'Freight': 45.54,
      'ShipName': 'Que Delícia',
      'ShipAddress': 'Rua da Panificadora, 12',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10648,
      'CustomerID': 'RICAR',
      'OrderDate': '1997-08-28T00:00:00.000Z',
      'ShippedDate': '1997-09-09T00:00:00.000Z',
      'Freight': 14.25,
      'ShipName': 'Ricardo Adocicados',
      'ShipAddress': 'Av. Copacabana, 267',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    }
  ],
  {

```



```

      'OrderID': 10649,
      'CustomerID': 'MAISD',
      'OrderDate': '1997-08-28T00:00:00.000Z',
      'ShippedDate': '1997-08-29T00:00:00.000Z',
      'Freight': 6.2,
      'ShipName': 'Maison Dewey',
      'ShipAddress': 'Rue Joseph-Bens 532',
      'ShipCity': 'Bruxelles',
      'ShipRegion': null,
      'ShipCountry': 'Belgium'
    },
    {
      'OrderID': 10650,
      'CustomerID': 'FAMIA',
      'OrderDate': '1997-08-29T00:00:00.000Z',
      'ShippedDate': '1997-09-03T00:00:00.000Z',
      'Freight': 176.81,
      'ShipName': 'Familia Arquibaldo',
      'ShipAddress': 'Rua Orós, 92',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10651,
      'CustomerID': 'WANDK',
      'OrderDate': '1997-09-01T00:00:00.000Z',
      'ShippedDate': '1997-09-11T00:00:00.000Z',
      'Freight': 20.6,
      'ShipName': 'Die Wandernde Kuh',
      'ShipAddress': 'Adenauerallee 900',
      'ShipCity': 'Stuttgart',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10652,
      'CustomerID': 'GOURL',
      'OrderDate': '1997-09-01T00:00:00.000Z',
      'ShippedDate': '1997-09-08T00:00:00.000Z',
      'Freight': 7.14,
      'ShipName': 'Gourmet Lanchonetes',
      'ShipAddress': 'Av. Brasil, 442',
      'ShipCity': 'Campinas',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10653,
      'CustomerID': 'FRANK',
      'OrderDate': '1997-09-02T00:00:00.000Z',
      'ShippedDate': '1997-09-19T00:00:00.000Z',
      'Freight': 93.25,
      'ShipName': 'Frankenversand',
      'ShipAddress': 'Berliner Platz 43',
      'ShipCity': 'München',
      'ShipRegion': null,

```

```

    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10654,
    'CustomerID': 'BERGS',
    'OrderDate': '1997-09-02T00:00:00.000Z',
    'ShippedDate': '1997-09-11T00:00:00.000Z',
    'Freight': 55.26,
    'ShipName': 'Berglunds snabbköp',
    'ShipAddress': 'Berguvsvägen 8',
    'ShipCity': 'Luleå',
    'ShipRegion': null,
    'ShipCountry': 'Sweden'
  },
  {
    'OrderID': 10655,
    'CustomerID': 'REGGC',
    'OrderDate': '1997-09-03T00:00:00.000Z',
    'ShippedDate': '1997-09-11T00:00:00.000Z',
    'Freight': 4.41,
    'ShipName': 'Reggiani Caseifici',
    'ShipAddress': 'Strada Provinciale 124',
    'ShipCity': 'Reggio Emilia',
    'ShipRegion': null,
    'ShipCountry': 'Italy'
  },
  {
    'OrderID': 10656,
    'CustomerID': 'GREAL',
    'OrderDate': '1997-09-04T00:00:00.000Z',
    'ShippedDate': '1997-09-10T00:00:00.000Z',
    'Freight': 57.15,
    'ShipName': 'Great Lakes Food Market',
    'ShipAddress': '2732 Baker Blvd.',
    'ShipCity': 'Eugene',
    'ShipRegion': 'OR',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10657,
    'CustomerID': 'SAVEA',
    'OrderDate': '1997-09-04T00:00:00.000Z',
    'ShippedDate': '1997-09-15T00:00:00.000Z',
    'Freight': 352.69,
    'ShipName': 'Save-a-lot Markets',
    'ShipAddress': '187 Suffolk Ln.',
    'ShipCity': 'Boise',
    'ShipRegion': 'ID',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10658,
    'CustomerID': 'QUICK',
    'OrderDate': '1997-09-05T00:00:00.000Z',
    'ShippedDate': '1997-09-08T00:00:00.000Z',
    'Freight': 364.15,
    'ShipName': 'QUICK-Stop',

```

```

      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10659,
      'CustomerID': 'QUEEN',
      'OrderDate': '1997-09-05T00:00:00.000Z',
      'ShippedDate': '1997-09-10T00:00:00.000Z',
      'Freight': 105.81,
      'ShipName': 'Queen Cozinha',
      'ShipAddress': 'Alameda dos Canários, 891',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10660,
      'CustomerID': 'HUNGC',
      'OrderDate': '1997-09-08T00:00:00.000Z',
      'ShippedDate': '1997-10-15T00:00:00.000Z',
      'Freight': 111.29,
      'ShipName': 'Hungry Coyote Import Store',
      'ShipAddress': 'City Center Plaza 516 Main St.',
      'ShipCity': 'Elgin',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10661,
      'CustomerID': 'HUNGO',
      'OrderDate': '1997-09-09T00:00:00.000Z',
      'ShippedDate': '1997-09-15T00:00:00.000Z',
      'Freight': 17.55,
      'ShipName': 'Hungry Owl All-Night Grocers',
      'ShipAddress': '8 Johnstown Road',
      'ShipCity': 'Cork',
      'ShipRegion': 'Co. Cork',
      'ShipCountry': 'Ireland'
    },
    {
      'OrderID': 10662,
      'CustomerID': 'LONEP',
      'OrderDate': '1997-09-09T00:00:00.000Z',
      'ShippedDate': '1997-09-18T00:00:00.000Z',
      'Freight': 1.28,
      'ShipName': 'Lonesome Pine Restaurant',
      'ShipAddress': '89 Chiaroscuro Rd.',
      'ShipCity': 'Portland',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10663,
      'CustomerID': 'BONAP',
      'OrderDate': '1997-09-10T00:00:00.000Z',

```

```

    'ShippedDate': '1997-10-03T00:00:00.000Z',
    'Freight': 113.15,
    'ShipName': 'Bon app',
    'ShipAddress': '12, rue des Bouchers',
    'ShipCity': 'Marseille',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10664,
    'CustomerID': 'FURIB',
    'OrderDate': '1997-09-10T00:00:00.000Z',
    'ShippedDate': '1997-09-19T00:00:00.000Z',
    'Freight': 1.27,
    'ShipName': 'Furia Bacalhau e Frutos do Mar',
    'ShipAddress': 'Jardim das rosas n. 32',
    'ShipCity': 'Lisboa',
    'ShipRegion': null,
    'ShipCountry': 'Portugal'
  },
  {
    'OrderID': 10665,
    'CustomerID': 'LONEP',
    'OrderDate': '1997-09-11T00:00:00.000Z',
    'ShippedDate': '1997-09-17T00:00:00.000Z',
    'Freight': 26.31,
    'ShipName': 'Lonesome Pine Restaurant',
    'ShipAddress': '89 Chiaroscuro Rd.',
    'ShipCity': 'Portland',
    'ShipRegion': 'OR',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10666,
    'CustomerID': 'RICSU',
    'OrderDate': '1997-09-12T00:00:00.000Z',
    'ShippedDate': '1997-09-22T00:00:00.000Z',
    'Freight': 232.42,
    'ShipName': 'Richter Supermarkt',
    'ShipAddress': 'Starenweg 5',
    'ShipCity': 'Genève',
    'ShipRegion': null,
    'ShipCountry': 'Switzerland'
  },
  {
    'OrderID': 10667,
    'CustomerID': 'ERNSH',
    'OrderDate': '1997-09-12T00:00:00.000Z',
    'ShippedDate': '1997-09-19T00:00:00.000Z',
    'Freight': 78.09,
    'ShipName': 'Ernst Handel',
    'ShipAddress': 'Kirchgasse 6',
    'ShipCity': 'Graz',
    'ShipRegion': null,
    'ShipCountry': 'Austria'
  },
  {

```

```

      'OrderID': 10668,
      'CustomerID': 'WANDK',
      'OrderDate': '1997-09-15T00:00:00.000Z',
      'ShippedDate': '1997-09-23T00:00:00.000Z',
      'Freight': 47.22,
      'ShipName': 'Die Wandernde Kuh',
      'ShipAddress': 'Adenauerallee 900',
      'ShipCity': 'Stuttgart',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10669,
      'CustomerID': 'SIMOB',
      'OrderDate': '1997-09-15T00:00:00.000Z',
      'ShippedDate': '1997-09-22T00:00:00.000Z',
      'Freight': 24.39,
      'ShipName': 'Simons bistro',
      'ShipAddress': 'Vinbæltet 34',
      'ShipCity': 'Kobenhavn',
      'ShipRegion': null,
      'ShipCountry': 'Denmark'
    },
    {
      'OrderID': 10670,
      'CustomerID': 'FRANK',
      'OrderDate': '1997-09-16T00:00:00.000Z',
      'ShippedDate': '1997-09-18T00:00:00.000Z',
      'Freight': 203.48,
      'ShipName': 'Frankenversand',
      'ShipAddress': 'Berliner Platz 43',
      'ShipCity': 'München',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10671,
      'CustomerID': 'FRANR',
      'OrderDate': '1997-09-17T00:00:00.000Z',
      'ShippedDate': '1997-09-24T00:00:00.000Z',
      'Freight': 30.34,
      'ShipName': 'France restauration',
      'ShipAddress': '54, rue Royale',
      'ShipCity': 'Nantes',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10672,
      'CustomerID': 'BERGS',
      'OrderDate': '1997-09-17T00:00:00.000Z',
      'ShippedDate': '1997-09-26T00:00:00.000Z',
      'Freight': 95.75,
      'ShipName': 'Berglunds snabbköp',
      'ShipAddress': 'Berguvsvägen 8',
      'ShipCity': 'Luleå',
      'ShipRegion': null,

```

```

    'ShipCountry': 'Sweden'
  },
  {
    'OrderID': 10673,
    'CustomerID': 'WILMK',
    'OrderDate': '1997-09-18T00:00:00.000Z',
    'ShippedDate': '1997-09-19T00:00:00.000Z',
    'Freight': 22.76,
    'ShipName': 'Wilman Kala',
    'ShipAddress': 'Keskuskatu 45',
    'ShipCity': 'Helsinki',
    'ShipRegion': null,
    'ShipCountry': 'Finland'
  },
  {
    'OrderID': 10674,
    'CustomerID': 'ISLAT',
    'OrderDate': '1997-09-18T00:00:00.000Z',
    'ShippedDate': '1997-09-30T00:00:00.000Z',
    'Freight': 0.9,
    'ShipName': 'Island Trading',
    'ShipAddress': 'Garden House Crowther Way',
    'ShipCity': 'Cowes',
    'ShipRegion': 'Isle of Wight',
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10675,
    'CustomerID': 'FRANK',
    'OrderDate': '1997-09-19T00:00:00.000Z',
    'ShippedDate': '1997-09-23T00:00:00.000Z',
    'Freight': 31.85,
    'ShipName': 'Frankenversand',
    'ShipAddress': 'Berliner Platz 43',
    'ShipCity': 'München',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10676,
    'CustomerID': 'TORTU',
    'OrderDate': '1997-09-22T00:00:00.000Z',
    'ShippedDate': '1997-09-29T00:00:00.000Z',
    'Freight': 2.01,
    'ShipName': 'Tortuga Restaurante',
    'ShipAddress': 'Avda. Azteca 123',
    'ShipCity': 'México D.F.',
    'ShipRegion': null,
    'ShipCountry': 'Mexico'
  },
  {
    'OrderID': 10677,
    'CustomerID': 'ANTON',
    'OrderDate': '1997-09-22T00:00:00.000Z',
    'ShippedDate': '1997-09-26T00:00:00.000Z',
    'Freight': 4.03,
    'ShipName': 'Antonio Moreno Taquería',

```

```

      'ShipAddress': 'Mataderos 2312',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10678,
      'CustomerID': 'SAVEA',
      'OrderDate': '1997-09-23T00:00:00.000Z',
      'ShippedDate': '1997-10-16T00:00:00.000Z',
      'Freight': 388.98,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10679,
      'CustomerID': 'BLONP',
      'OrderDate': '1997-09-23T00:00:00.000Z',
      'ShippedDate': '1997-09-30T00:00:00.000Z',
      'Freight': 27.94,
      'ShipName': 'Blondel père et fils',
      'ShipAddress': '24, place Kléber',
      'ShipCity': 'Strasbourg',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10680,
      'CustomerID': 'OLDWO',
      'OrderDate': '1997-09-24T00:00:00.000Z',
      'ShippedDate': '1997-09-26T00:00:00.000Z',
      'Freight': 26.61,
      'ShipName': 'Old World Delicatessen',
      'ShipAddress': '2743 Bering St.',
      'ShipCity': 'Anchorage',
      'ShipRegion': 'AK',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10681,
      'CustomerID': 'GREAL',
      'OrderDate': '1997-09-25T00:00:00.000Z',
      'ShippedDate': '1997-09-30T00:00:00.000Z',
      'Freight': 76.13,
      'ShipName': 'Great Lakes Food Market',
      'ShipAddress': '2732 Baker Blvd.',
      'ShipCity': 'Eugene',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10682,
      'CustomerID': 'ANTON',
      'OrderDate': '1997-09-25T00:00:00.000Z',

```

```

    'ShippedDate': '1997-10-01T00:00:00.000Z',
    'Freight': 36.13,
    'ShipName': 'Antonio Moreno Taquería',
    'ShipAddress': 'Mataderos 2312',
    'ShipCity': 'México D.F.',
    'ShipRegion': null,
    'ShipCountry': 'Mexico'
  },
  {
    'OrderID': 10683,
    'CustomerID': 'DUMON',
    'OrderDate': '1997-09-26T00:00:00.000Z',
    'ShippedDate': '1997-10-01T00:00:00.000Z',
    'Freight': 4.4,
    'ShipName': 'Du monde entier',
    'ShipAddress': '67, rue des Cinquante Otages',
    'ShipCity': 'Nantes',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10684,
    'CustomerID': 'OTTIK',
    'OrderDate': '1997-09-26T00:00:00.000Z',
    'ShippedDate': '1997-09-30T00:00:00.000Z',
    'Freight': 145.63,
    'ShipName': 'Ottilies Käseladen',
    'ShipAddress': 'Mehrheimerstr. 369',
    'ShipCity': 'Köln',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10685,
    'CustomerID': 'GOURL',
    'OrderDate': '1997-09-29T00:00:00.000Z',
    'ShippedDate': '1997-10-03T00:00:00.000Z',
    'Freight': 33.75,
    'ShipName': 'Gourmet Lanchonetes',
    'ShipAddress': 'Av. Brasil, 442',
    'ShipCity': 'Campinas',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10686,
    'CustomerID': 'PICCO',
    'OrderDate': '1997-09-30T00:00:00.000Z',
    'ShippedDate': '1997-10-08T00:00:00.000Z',
    'Freight': 96.5,
    'ShipName': 'Piccolo und mehr',
    'ShipAddress': 'Geislweg 14',
    'ShipCity': 'Salzburg',
    'ShipRegion': null,
    'ShipCountry': 'Austria'
  },
  {

```



```

    'OrderID': 10687,
    'CustomerID': 'HUNGO',
    'OrderDate': '1997-09-30T00:00:00.000Z',
    'ShippedDate': '1997-10-30T00:00:00.000Z',
    'Freight': 296.43,
    'ShipName': 'Hungry Owl All-Night Grocers',
    'ShipAddress': '8 Johnstown Road',
    'ShipCity': 'Cork',
    'ShipRegion': 'Co. Cork',
    'ShipCountry': 'Ireland'
  },
  {
    'OrderID': 10688,
    'CustomerID': 'VAFFE',
    'OrderDate': '1997-10-01T00:00:00.000Z',
    'ShippedDate': '1997-10-07T00:00:00.000Z',
    'Freight': 299.09,
    'ShipName': 'Vaffeljernet',
    'ShipAddress': 'Smagsloget 45',
    'ShipCity': 'Århus',
    'ShipRegion': null,
    'ShipCountry': 'Denmark'
  },
  {
    'OrderID': 10689,
    'CustomerID': 'BERGS',
    'OrderDate': '1997-10-01T00:00:00.000Z',
    'ShippedDate': '1997-10-07T00:00:00.000Z',
    'Freight': 13.42,
    'ShipName': 'Berglunds snabbköp',
    'ShipAddress': 'Berguvsvägen 8',
    'ShipCity': 'Luleå',
    'ShipRegion': null,
    'ShipCountry': 'Sweden'
  },
  {
    'OrderID': 10690,
    'CustomerID': 'HANAR',
    'OrderDate': '1997-10-02T00:00:00.000Z',
    'ShippedDate': '1997-10-03T00:00:00.000Z',
    'Freight': 15.8,
    'ShipName': 'Hanari Carnes',
    'ShipAddress': 'Rua do Paço, 67',
    'ShipCity': 'Rio de Janeiro',
    'ShipRegion': 'RJ',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10691,
    'CustomerID': 'QUICK',
    'OrderDate': '1997-10-03T00:00:00.000Z',
    'ShippedDate': '1997-10-22T00:00:00.000Z',
    'Freight': 810.05,
    'ShipName': 'QUICK-Stop',
    'ShipAddress': 'Taucherstraße 10',
    'ShipCity': 'Cunewalde',
    'ShipRegion': null,

```

```

    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10692,
    'CustomerID': 'ALFKI',
    'OrderDate': '1997-10-03T00:00:00.000Z',
    'ShippedDate': '1997-10-13T00:00:00.000Z',
    'Freight': 61.02,
    'ShipName': 'Alfred\ Futterkiste',
    'ShipAddress': 'Obere Str. 57',
    'ShipCity': 'Berlin',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10693,
    'CustomerID': 'WHITC',
    'OrderDate': '1997-10-06T00:00:00.000Z',
    'ShippedDate': '1997-10-10T00:00:00.000Z',
    'Freight': 139.34,
    'ShipName': 'White Clover Markets',
    'ShipAddress': '1029 - 12th Ave. S.',
    'ShipCity': 'Seattle',
    'ShipRegion': 'WA',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10694,
    'CustomerID': 'QUICK',
    'OrderDate': '1997-10-06T00:00:00.000Z',
    'ShippedDate': '1997-10-09T00:00:00.000Z',
    'Freight': 398.36,
    'ShipName': 'QUICK-Stop',
    'ShipAddress': 'Taucherstraße 10',
    'ShipCity': 'Cunewalde',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10695,
    'CustomerID': 'WILMK',
    'OrderDate': '1997-10-07T00:00:00.000Z',
    'ShippedDate': '1997-10-14T00:00:00.000Z',
    'Freight': 16.72,
    'ShipName': 'Wilman Kala',
    'ShipAddress': 'Keskuskatu 45',
    'ShipCity': 'Helsinki',
    'ShipRegion': null,
    'ShipCountry': 'Finland'
  },
  {
    'OrderID': 10696,
    'CustomerID': 'WHITC',
    'OrderDate': '1997-10-08T00:00:00.000Z',
    'ShippedDate': '1997-10-14T00:00:00.000Z',
    'Freight': 102.55,
    'ShipName': 'White Clover Markets',

```

```

      'ShipAddress': '1029 - 12th Ave. S.',
      'ShipCity': 'Seattle',
      'ShipRegion': 'WA',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10697,
      'CustomerID': 'LINOD',
      'OrderDate': '1997-10-08T00:00:00.000Z',
      'ShippedDate': '1997-10-14T00:00:00.000Z',
      'Freight': 45.52,
      'ShipName': 'LINO-Delicateses',
      'ShipAddress': 'Ave. 5 de Mayo Porlamar',
      'ShipCity': 'I. de Margarita',
      'ShipRegion': 'Nueva Esparta',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10698,
      'CustomerID': 'ERNSH',
      'OrderDate': '1997-10-09T00:00:00.000Z',
      'ShippedDate': '1997-10-17T00:00:00.000Z',
      'Freight': 272.47,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10699,
      'CustomerID': 'MORGK',
      'OrderDate': '1997-10-09T00:00:00.000Z',
      'ShippedDate': '1997-10-13T00:00:00.000Z',
      'Freight': 0.58,
      'ShipName': 'Morgenstern Gesundkost',
      'ShipAddress': 'Heerstr. 22',
      'ShipCity': 'Leipzig',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10700,
      'CustomerID': 'SAVEA',
      'OrderDate': '1997-10-10T00:00:00.000Z',
      'ShippedDate': '1997-10-16T00:00:00.000Z',
      'Freight': 65.1,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10701,
      'CustomerID': 'HUNGO',
      'OrderDate': '1997-10-13T00:00:00.000Z',

```

```

    'ShippedDate': '1997-10-15T00:00:00.000Z',
    'Freight': 220.31,
    'ShipName': 'Hungry Owl All-Night Grocers',
    'ShipAddress': '8 Johnstown Road',
    'ShipCity': 'Cork',
    'ShipRegion': 'Co. Cork',
    'ShipCountry': 'Ireland'
  },
  {
    'OrderID': 10702,
    'CustomerID': 'ALFKI',
    'OrderDate': '1997-10-13T00:00:00.000Z',
    'ShippedDate': '1997-10-21T00:00:00.000Z',
    'Freight': 23.94,
    'ShipName': 'Alfred Futterkiste',
    'ShipAddress': 'Obere Str. 57',
    'ShipCity': 'Berlin',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10703,
    'CustomerID': 'FOLKO',
    'OrderDate': '1997-10-14T00:00:00.000Z',
    'ShippedDate': '1997-10-20T00:00:00.000Z',
    'Freight': 152.3,
    'ShipName': 'Folk och fä HB',
    'ShipAddress': 'Åkergatan 24',
    'ShipCity': 'Bräcke',
    'ShipRegion': null,
    'ShipCountry': 'Sweden'
  },
  {
    'OrderID': 10704,
    'CustomerID': 'QUEEN',
    'OrderDate': '1997-10-14T00:00:00.000Z',
    'ShippedDate': '1997-11-07T00:00:00.000Z',
    'Freight': 4.78,
    'ShipName': 'Queen Cozinha',
    'ShipAddress': 'Alameda dos Canários, 891',
    'ShipCity': 'Sao Paulo',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10705,
    'CustomerID': 'HILAA',
    'OrderDate': '1997-10-15T00:00:00.000Z',
    'ShippedDate': '1997-11-18T00:00:00.000Z',
    'Freight': 3.52,
    'ShipName': 'HILARION-Abastos',
    'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
    'ShipCity': 'San Cristóbal',
    'ShipRegion': 'Táchira',
    'ShipCountry': 'Venezuela'
  }
]

```

```

    'OrderID': 10706,
    'CustomerID': 'OLDWO',
    'OrderDate': '1997-10-16T00:00:00.000Z',
    'ShippedDate': '1997-10-21T00:00:00.000Z',
    'Freight': 135.63,
    'ShipName': 'Old World Delicatessen',
    'ShipAddress': '2743 Bering St.',
    'ShipCity': 'Anchorage',
    'ShipRegion': 'AK',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10707,
    'CustomerID': 'AROUT',
    'OrderDate': '1997-10-16T00:00:00.000Z',
    'ShippedDate': '1997-10-23T00:00:00.000Z',
    'Freight': 21.74,
    'ShipName': 'Around the Horn',
    'ShipAddress': 'Brook Farm Stratford St. Mary',
    'ShipCity': 'Colchester',
    'ShipRegion': 'Essex',
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10708,
    'CustomerID': 'THEBI',
    'OrderDate': '1997-10-17T00:00:00.000Z',
    'ShippedDate': '1997-11-05T00:00:00.000Z',
    'Freight': 2.96,
    'ShipName': 'The Big Cheese',
    'ShipAddress': '89 Jefferson Way Suite 2',
    'ShipCity': 'Portland',
    'ShipRegion': 'OR',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10709,
    'CustomerID': 'GOURL',
    'OrderDate': '1997-10-17T00:00:00.000Z',
    'ShippedDate': '1997-11-20T00:00:00.000Z',
    'Freight': 210.8,
    'ShipName': 'Gourmet Lanchonetes',
    'ShipAddress': 'Av. Brasil, 442',
    'ShipCity': 'Campinas',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10710,
    'CustomerID': 'FRANS',
    'OrderDate': '1997-10-20T00:00:00.000Z',
    'ShippedDate': '1997-10-23T00:00:00.000Z',
    'Freight': 4.98,
    'ShipName': 'Franchi S.p.A.',
    'ShipAddress': 'Via Monte Bianco 34',
    'ShipCity': 'Torino',
    'ShipRegion': null,

```

```

    'ShipCountry': 'Italy'
  },
  {
    'OrderID': 10711,
    'CustomerID': 'SAVEA',
    'OrderDate': '1997-10-21T00:00:00.000Z',
    'ShippedDate': '1997-10-29T00:00:00.000Z',
    'Freight': 52.41,
    'ShipName': 'Save-a-lot Markets',
    'ShipAddress': '187 Suffolk Ln.',
    'ShipCity': 'Boise',
    'ShipRegion': 'ID',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10712,
    'CustomerID': 'HUNGO',
    'OrderDate': '1997-10-21T00:00:00.000Z',
    'ShippedDate': '1997-10-31T00:00:00.000Z',
    'Freight': 89.93,
    'ShipName': 'Hungry Owl All-Night Grocers',
    'ShipAddress': '8 Johnstown Road',
    'ShipCity': 'Cork',
    'ShipRegion': 'Co. Cork',
    'ShipCountry': 'Ireland'
  },
  {
    'OrderID': 10713,
    'CustomerID': 'SAVEA',
    'OrderDate': '1997-10-22T00:00:00.000Z',
    'ShippedDate': '1997-10-24T00:00:00.000Z',
    'Freight': 167.05,
    'ShipName': 'Save-a-lot Markets',
    'ShipAddress': '187 Suffolk Ln.',
    'ShipCity': 'Boise',
    'ShipRegion': 'ID',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10714,
    'CustomerID': 'SAVEA',
    'OrderDate': '1997-10-22T00:00:00.000Z',
    'ShippedDate': '1997-10-27T00:00:00.000Z',
    'Freight': 24.49,
    'ShipName': 'Save-a-lot Markets',
    'ShipAddress': '187 Suffolk Ln.',
    'ShipCity': 'Boise',
    'ShipRegion': 'ID',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10715,
    'CustomerID': 'BONAP',
    'OrderDate': '1997-10-23T00:00:00.000Z',
    'ShippedDate': '1997-10-29T00:00:00.000Z',
    'Freight': 63.2,
    'ShipName': 'Bon app',

```

```

    'ShipAddress': '12, rue des Bouchers',
    'ShipCity': 'Marseille',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10716,
    'CustomerID': 'RANCH',
    'OrderDate': '1997-10-24T00:00:00.000Z',
    'ShippedDate': '1997-10-27T00:00:00.000Z',
    'Freight': 22.57,
    'ShipName': 'Rancho grande',
    'ShipAddress': 'Av. del Libertador 900',
    'ShipCity': 'Buenos Aires',
    'ShipRegion': null,
    'ShipCountry': 'Argentina'
  },
  {
    'OrderID': 10717,
    'CustomerID': 'FRANK',
    'OrderDate': '1997-10-24T00:00:00.000Z',
    'ShippedDate': '1997-10-29T00:00:00.000Z',
    'Freight': 59.25,
    'ShipName': 'Frankenversand',
    'ShipAddress': 'Berliner Platz 43',
    'ShipCity': 'München',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10718,
    'CustomerID': 'KOENE',
    'OrderDate': '1997-10-27T00:00:00.000Z',
    'ShippedDate': '1997-10-29T00:00:00.000Z',
    'Freight': 170.88,
    'ShipName': 'Königlich Essen',
    'ShipAddress': 'Maubelstr. 90',
    'ShipCity': 'Brandenburg',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10719,
    'CustomerID': 'LETSS',
    'OrderDate': '1997-10-27T00:00:00.000Z',
    'ShippedDate': '1997-11-05T00:00:00.000Z',
    'Freight': 51.44,
    'ShipName': 'Let\' Stop N Shop',
    'ShipAddress': '87 Polk St. Suite 5',
    'ShipCity': 'San Francisco',
    'ShipRegion': 'CA',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10720,
    'CustomerID': 'QUEDE',
    'OrderDate': '1997-10-28T00:00:00.000Z',

```

```

      'ShippedDate': '1997-11-05T00:00:00.000Z',
      'Freight': 9.53,
      'ShipName': 'Que Delícia',
      'ShipAddress': 'Rua da Panificadora, 12',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10721,
      'CustomerID': 'QUICK',
      'OrderDate': '1997-10-29T00:00:00.000Z',
      'ShippedDate': '1997-10-31T00:00:00.000Z',
      'Freight': 48.92,
      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10722,
      'CustomerID': 'SAVEA',
      'OrderDate': '1997-10-29T00:00:00.000Z',
      'ShippedDate': '1997-11-04T00:00:00.000Z',
      'Freight': 74.58,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10723,
      'CustomerID': 'WHITC',
      'OrderDate': '1997-10-30T00:00:00.000Z',
      'ShippedDate': '1997-11-25T00:00:00.000Z',
      'Freight': 21.72,
      'ShipName': 'White Clover Markets',
      'ShipAddress': '1029 - 12th Ave. S.',
      'ShipCity': 'Seattle',
      'ShipRegion': 'WA',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10724,
      'CustomerID': 'MEREP',
      'OrderDate': '1997-10-30T00:00:00.000Z',
      'ShippedDate': '1997-11-05T00:00:00.000Z',
      'Freight': 57.75,
      'ShipName': 'Mère Paillarde',
      'ShipAddress': '43 rue St. Laurent',
      'ShipCity': 'Montréal',
      'ShipRegion': 'Québec',
      'ShipCountry': 'Canada'
    }
  ],
  {

```



```

    'OrderID': 10725,
    'CustomerID': 'FAMIA',
    'OrderDate': '1997-10-31T00:00:00.000Z',
    'ShippedDate': '1997-11-05T00:00:00.000Z',
    'Freight': 10.83,
    'ShipName': 'Familia Arquibaldo',
    'ShipAddress': 'Rua Orós, 92',
    'ShipCity': 'Sao Paulo',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10726,
    'CustomerID': 'EASTC',
    'OrderDate': '1997-11-03T00:00:00.000Z',
    'ShippedDate': '1997-12-05T00:00:00.000Z',
    'Freight': 16.56,
    'ShipName': 'Eastern Connection',
    'ShipAddress': '35 King George',
    'ShipCity': 'London',
    'ShipRegion': null,
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10727,
    'CustomerID': 'REGGC',
    'OrderDate': '1997-11-03T00:00:00.000Z',
    'ShippedDate': '1997-12-05T00:00:00.000Z',
    'Freight': 89.9,
    'ShipName': 'Reggiani Caseifici',
    'ShipAddress': 'Strada Provinciale 124',
    'ShipCity': 'Reggio Emilia',
    'ShipRegion': null,
    'ShipCountry': 'Italy'
  },
  {
    'OrderID': 10728,
    'CustomerID': 'QUEEN',
    'OrderDate': '1997-11-04T00:00:00.000Z',
    'ShippedDate': '1997-11-11T00:00:00.000Z',
    'Freight': 58.33,
    'ShipName': 'Queen Cozinha',
    'ShipAddress': 'Alameda dos Canários, 891',
    'ShipCity': 'Sao Paulo',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10729,
    'CustomerID': 'LINOD',
    'OrderDate': '1997-11-04T00:00:00.000Z',
    'ShippedDate': '1997-11-14T00:00:00.000Z',
    'Freight': 141.06,
    'ShipName': 'LINO-Delicateses',
    'ShipAddress': 'Ave. 5 de Mayo Porlamar',
    'ShipCity': 'I. de Margarita',
    'ShipRegion': 'Nueva Esparta',

```

```

    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10730,
    'CustomerID': 'BONAP',
    'OrderDate': '1997-11-05T00:00:00.000Z',
    'ShippedDate': '1997-11-14T00:00:00.000Z',
    'Freight': 20.12,
    'ShipName': 'Bon app',
    'ShipAddress': '12, rue des Bouchers',
    'ShipCity': 'Marseille',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10731,
    'CustomerID': 'CHOPS',
    'OrderDate': '1997-11-06T00:00:00.000Z',
    'ShippedDate': '1997-11-14T00:00:00.000Z',
    'Freight': 96.65,
    'ShipName': 'Chop-suey Chinese',
    'ShipAddress': 'Hauptstr. 31',
    'ShipCity': 'Bern',
    'ShipRegion': null,
    'ShipCountry': 'Switzerland'
  },
  {
    'OrderID': 10732,
    'CustomerID': 'BONAP',
    'OrderDate': '1997-11-06T00:00:00.000Z',
    'ShippedDate': '1997-11-07T00:00:00.000Z',
    'Freight': 16.97,
    'ShipName': 'Bon app',
    'ShipAddress': '12, rue des Bouchers',
    'ShipCity': 'Marseille',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10733,
    'CustomerID': 'BERGS',
    'OrderDate': '1997-11-07T00:00:00.000Z',
    'ShippedDate': '1997-11-10T00:00:00.000Z',
    'Freight': 110.11,
    'ShipName': 'Berglunds snabbköp',
    'ShipAddress': 'Berguvsvägen 8',
    'ShipCity': 'Luleå',
    'ShipRegion': null,
    'ShipCountry': 'Sweden'
  },
  {
    'OrderID': 10734,
    'CustomerID': 'GOURL',
    'OrderDate': '1997-11-07T00:00:00.000Z',
    'ShippedDate': '1997-11-12T00:00:00.000Z',
    'Freight': 1.63,
    'ShipName': 'Gourmet Lanchonetes',

```

```

      'ShipAddress': 'Av. Brasil, 442',
      'ShipCity': 'Campinas',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10735,
      'CustomerID': 'LETSS',
      'OrderDate': '1997-11-10T00:00:00.000Z',
      'ShippedDate': '1997-11-21T00:00:00.000Z',
      'Freight': 45.97,
      'ShipName': 'Let\ Stop N Shop',
      'ShipAddress': '87 Polk St. Suite 5',
      'ShipCity': 'San Francisco',
      'ShipRegion': 'CA',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10736,
      'CustomerID': 'HUNGO',
      'OrderDate': '1997-11-11T00:00:00.000Z',
      'ShippedDate': '1997-11-21T00:00:00.000Z',
      'Freight': 44.1,
      'ShipName': 'Hungry Owl All-Night Grocers',
      'ShipAddress': '8 Johnstown Road',
      'ShipCity': 'Cork',
      'ShipRegion': 'Co. Cork',
      'ShipCountry': 'Ireland'
    },
    {
      'OrderID': 10737,
      'CustomerID': 'VINET',
      'OrderDate': '1997-11-11T00:00:00.000Z',
      'ShippedDate': '1997-11-18T00:00:00.000Z',
      'Freight': 7.79,
      'ShipName': 'Vins et alcools Chevalier',
      'ShipAddress': '59 rue de l\ Abbaye',
      'ShipCity': 'Reims',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10738,
      'CustomerID': 'SPECB',
      'OrderDate': '1997-11-12T00:00:00.000Z',
      'ShippedDate': '1997-11-18T00:00:00.000Z',
      'Freight': 2.91,
      'ShipName': 'Spécialités du monde',
      'ShipAddress': '25, rue Lauriston',
      'ShipCity': 'Paris',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10739,
      'CustomerID': 'VINET',
      'OrderDate': '1997-11-12T00:00:00.000Z',

```

```

      'ShippedDate': '1997-11-17T00:00:00.000Z',
      'Freight': 11.08,
      'ShipName': 'Vins et alcools Chevalier',
      'ShipAddress': '59 rue de l'Abbaye',
      'ShipCity': 'Reims',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10740,
      'CustomerID': 'WHITC',
      'OrderDate': '1997-11-13T00:00:00.000Z',
      'ShippedDate': '1997-11-25T00:00:00.000Z',
      'Freight': 81.88,
      'ShipName': 'White Clover Markets',
      'ShipAddress': '1029 - 12th Ave. S.',
      'ShipCity': 'Seattle',
      'ShipRegion': 'WA',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10741,
      'CustomerID': 'AROUT',
      'OrderDate': '1997-11-14T00:00:00.000Z',
      'ShippedDate': '1997-11-18T00:00:00.000Z',
      'Freight': 10.96,
      'ShipName': 'Around the Horn',
      'ShipAddress': 'Brook Farm Stratford St. Mary',
      'ShipCity': 'Colchester',
      'ShipRegion': 'Essex',
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10742,
      'CustomerID': 'BOTTM',
      'OrderDate': '1997-11-14T00:00:00.000Z',
      'ShippedDate': '1997-11-18T00:00:00.000Z',
      'Freight': 243.73,
      'ShipName': 'Bottom-Dollar Markets',
      'ShipAddress': '23 Tsawassen Blvd.',
      'ShipCity': 'Tsawassen',
      'ShipRegion': 'BC',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10743,
      'CustomerID': 'AROUT',
      'OrderDate': '1997-11-17T00:00:00.000Z',
      'ShippedDate': '1997-11-21T00:00:00.000Z',
      'Freight': 23.72,
      'ShipName': 'Around the Horn',
      'ShipAddress': 'Brook Farm Stratford St. Mary',
      'ShipCity': 'Colchester',
      'ShipRegion': 'Essex',
      'ShipCountry': 'UK'
    }
  ],
  {

```

```

    'OrderID': 10744,
    'CustomerID': 'VAFFE',
    'OrderDate': '1997-11-17T00:00:00.000Z',
    'ShippedDate': '1997-11-24T00:00:00.000Z',
    'Freight': 69.19,
    'ShipName': 'Vaffeljernet',
    'ShipAddress': 'Smagsloget 45',
    'ShipCity': 'Århus',
    'ShipRegion': null,
    'ShipCountry': 'Denmark'
  },
  {
    'OrderID': 10745,
    'CustomerID': 'QUICK',
    'OrderDate': '1997-11-18T00:00:00.000Z',
    'ShippedDate': '1997-11-27T00:00:00.000Z',
    'Freight': 3.52,
    'ShipName': 'QUICK-Stop',
    'ShipAddress': 'Taucherstraße 10',
    'ShipCity': 'Cunewalde',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10746,
    'CustomerID': 'CHOPS',
    'OrderDate': '1997-11-19T00:00:00.000Z',
    'ShippedDate': '1997-11-21T00:00:00.000Z',
    'Freight': 31.43,
    'ShipName': 'Chop-suey Chinese',
    'ShipAddress': 'Hauptstr. 31',
    'ShipCity': 'Bern',
    'ShipRegion': null,
    'ShipCountry': 'Switzerland'
  },
  {
    'OrderID': 10747,
    'CustomerID': 'PICCO',
    'OrderDate': '1997-11-19T00:00:00.000Z',
    'ShippedDate': '1997-11-26T00:00:00.000Z',
    'Freight': 117.33,
    'ShipName': 'Piccolo und mehr',
    'ShipAddress': 'Geislweg 14',
    'ShipCity': 'Salzburg',
    'ShipRegion': null,
    'ShipCountry': 'Austria'
  },
  {
    'OrderID': 10748,
    'CustomerID': 'SAVEA',
    'OrderDate': '1997-11-20T00:00:00.000Z',
    'ShippedDate': '1997-11-28T00:00:00.000Z',
    'Freight': 232.55,
    'ShipName': 'Save-a-lot Markets',
    'ShipAddress': '187 Suffolk Ln.',
    'ShipCity': 'Boise',
    'ShipRegion': 'ID',

```

```

    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10749,
    'CustomerID': 'ISLAT',
    'OrderDate': '1997-11-20T00:00:00.000Z',
    'ShippedDate': '1997-12-19T00:00:00.000Z',
    'Freight': 61.53,
    'ShipName': 'Island Trading',
    'ShipAddress': 'Garden House Crowther Way',
    'ShipCity': 'Cowes',
    'ShipRegion': 'Isle of Wight',
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10750,
    'CustomerID': 'WARTH',
    'OrderDate': '1997-11-21T00:00:00.000Z',
    'ShippedDate': '1997-11-24T00:00:00.000Z',
    'Freight': 79.3,
    'ShipName': 'Wartian Herkku',
    'ShipAddress': 'Torikatu 38',
    'ShipCity': 'Oulu',
    'ShipRegion': null,
    'ShipCountry': 'Finland'
  },
  {
    'OrderID': 10751,
    'CustomerID': 'RICSU',
    'OrderDate': '1997-11-24T00:00:00.000Z',
    'ShippedDate': '1997-12-03T00:00:00.000Z',
    'Freight': 130.79,
    'ShipName': 'Richter Supermarkt',
    'ShipAddress': 'Starenweg 5',
    'ShipCity': 'Genève',
    'ShipRegion': null,
    'ShipCountry': 'Switzerland'
  },
  {
    'OrderID': 10752,
    'CustomerID': 'NORTS',
    'OrderDate': '1997-11-24T00:00:00.000Z',
    'ShippedDate': '1997-11-28T00:00:00.000Z',
    'Freight': 1.39,
    'ShipName': 'North/South',
    'ShipAddress': 'South House 300 Queensbridge',
    'ShipCity': 'London',
    'ShipRegion': null,
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10753,
    'CustomerID': 'FRANS',
    'OrderDate': '1997-11-25T00:00:00.000Z',
    'ShippedDate': '1997-11-27T00:00:00.000Z',
    'Freight': 7.7,
    'ShipName': 'Franchi S.p.A.',

```

```

      'ShipAddress': 'Via Monte Bianco 34',
      'ShipCity': 'Torino',
      'ShipRegion': null,
      'ShipCountry': 'Italy'
    },
    {
      'OrderID': 10754,
      'CustomerID': 'MAGAA',
      'OrderDate': '1997-11-25T00:00:00.000Z',
      'ShippedDate': '1997-11-27T00:00:00.000Z',
      'Freight': 2.38,
      'ShipName': 'Magazzini Alimentari Riuniti',
      'ShipAddress': 'Via Ludovico il Moro 22',
      'ShipCity': 'Bergamo',
      'ShipRegion': null,
      'ShipCountry': 'Italy'
    },
    {
      'OrderID': 10755,
      'CustomerID': 'BONAP',
      'OrderDate': '1997-11-26T00:00:00.000Z',
      'ShippedDate': '1997-11-28T00:00:00.000Z',
      'Freight': 16.71,
      'ShipName': 'Bon app',
      'ShipAddress': '12, rue des Bouchers',
      'ShipCity': 'Marseille',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10756,
      'CustomerID': 'SPLIR',
      'OrderDate': '1997-11-27T00:00:00.000Z',
      'ShippedDate': '1997-12-02T00:00:00.000Z',
      'Freight': 73.21,
      'ShipName': 'Split Rail Beer & Ale',
      'ShipAddress': 'P.O. Box 555',
      'ShipCity': 'Lander',
      'ShipRegion': 'WY',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10757,
      'CustomerID': 'SAVEA',
      'OrderDate': '1997-11-27T00:00:00.000Z',
      'ShippedDate': '1997-12-15T00:00:00.000Z',
      'Freight': 8.19,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10758,
      'CustomerID': 'RICSU',
      'OrderDate': '1997-11-28T00:00:00.000Z',

```

```

    'ShippedDate': '1997-12-04T00:00:00.000Z',
    'Freight': 138.17,
    'ShipName': 'Richter Supermarkt',
    'ShipAddress': 'Starenweg 5',
    'ShipCity': 'Genève',
    'ShipRegion': null,
    'ShipCountry': 'Switzerland'
  },
  {
    'OrderID': 10759,
    'CustomerID': 'ANATR',
    'OrderDate': '1997-11-28T00:00:00.000Z',
    'ShippedDate': '1997-12-12T00:00:00.000Z',
    'Freight': 11.99,
    'ShipName': 'Ana Trujillo Emparedados y helados',
    'ShipAddress': 'Avda. de la Constitución 2222',
    'ShipCity': 'México D.F.',
    'ShipRegion': null,
    'ShipCountry': 'Mexico'
  },
  {
    'OrderID': 10760,
    'CustomerID': 'MAISD',
    'OrderDate': '1997-12-01T00:00:00.000Z',
    'ShippedDate': '1997-12-10T00:00:00.000Z',
    'Freight': 155.64,
    'ShipName': 'Maison Dewey',
    'ShipAddress': 'Rue Joseph-Bens 532',
    'ShipCity': 'Bruxelles',
    'ShipRegion': null,
    'ShipCountry': 'Belgium'
  },
  {
    'OrderID': 10761,
    'CustomerID': 'RATTC',
    'OrderDate': '1997-12-02T00:00:00.000Z',
    'ShippedDate': '1997-12-08T00:00:00.000Z',
    'Freight': 18.66,
    'ShipName': 'Rattlesnake Canyon Grocery',
    'ShipAddress': '2817 Milton Dr.',
    'ShipCity': 'Albuquerque',
    'ShipRegion': 'NM',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10762,
    'CustomerID': 'FOLKO',
    'OrderDate': '1997-12-02T00:00:00.000Z',
    'ShippedDate': '1997-12-09T00:00:00.000Z',
    'Freight': 328.74,
    'ShipName': 'Folk och fä HB',
    'ShipAddress': 'Åkergatan 24',
    'ShipCity': 'Bräcke',
    'ShipRegion': null,
    'ShipCountry': 'Sweden'
  },
  {

```



```

      'OrderID': 10763,
      'CustomerID': 'FOLIG',
      'OrderDate': '1997-12-03T00:00:00.000Z',
      'ShippedDate': '1997-12-08T00:00:00.000Z',
      'Freight': 37.35,
      'ShipName': 'Folies gourmandes',
      'ShipAddress': '184, chaussée de Tournai',
      'ShipCity': 'Lille',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10764,
      'CustomerID': 'ERNSH',
      'OrderDate': '1997-12-03T00:00:00.000Z',
      'ShippedDate': '1997-12-08T00:00:00.000Z',
      'Freight': 145.45,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10765,
      'CustomerID': 'QUICK',
      'OrderDate': '1997-12-04T00:00:00.000Z',
      'ShippedDate': '1997-12-09T00:00:00.000Z',
      'Freight': 42.74,
      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10766,
      'CustomerID': 'OTTIK',
      'OrderDate': '1997-12-05T00:00:00.000Z',
      'ShippedDate': '1997-12-09T00:00:00.000Z',
      'Freight': 157.55,
      'ShipName': 'Ottilies Käseladen',
      'ShipAddress': 'Mehrheimerstr. 369',
      'ShipCity': 'Köln',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10767,
      'CustomerID': 'SUPRD',
      'OrderDate': '1997-12-05T00:00:00.000Z',
      'ShippedDate': '1997-12-15T00:00:00.000Z',
      'Freight': 1.59,
      'ShipName': 'Suprêmes délices',
      'ShipAddress': 'Boulevard Tirou, 255',
      'ShipCity': 'Charleroi',
      'ShipRegion': null,

```

```

    'ShipCountry': 'Belgium'
  },
  {
    'OrderID': 10768,
    'CustomerID': 'AROUT',
    'OrderDate': '1997-12-08T00:00:00.000Z',
    'ShippedDate': '1997-12-15T00:00:00.000Z',
    'Freight': 146.32,
    'ShipName': 'Around the Horn',
    'ShipAddress': 'Brook Farm Stratford St. Mary',
    'ShipCity': 'Colchester',
    'ShipRegion': 'Essex',
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10769,
    'CustomerID': 'VAFFE',
    'OrderDate': '1997-12-08T00:00:00.000Z',
    'ShippedDate': '1997-12-12T00:00:00.000Z',
    'Freight': 65.06,
    'ShipName': 'Vaffeljernet',
    'ShipAddress': 'Smagsloget 45',
    'ShipCity': 'Århus',
    'ShipRegion': null,
    'ShipCountry': 'Denmark'
  },
  {
    'OrderID': 10770,
    'CustomerID': 'HANAR',
    'OrderDate': '1997-12-09T00:00:00.000Z',
    'ShippedDate': '1997-12-17T00:00:00.000Z',
    'Freight': 5.32,
    'ShipName': 'Hanari Carnes',
    'ShipAddress': 'Rua do Paço, 67',
    'ShipCity': 'Rio de Janeiro',
    'ShipRegion': 'RJ',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10771,
    'CustomerID': 'ERNSH',
    'OrderDate': '1997-12-10T00:00:00.000Z',
    'ShippedDate': '1998-01-02T00:00:00.000Z',
    'Freight': 11.19,
    'ShipName': 'Ernst Handel',
    'ShipAddress': 'Kirchgasse 6',
    'ShipCity': 'Graz',
    'ShipRegion': null,
    'ShipCountry': 'Austria'
  },
  {
    'OrderID': 10772,
    'CustomerID': 'LEHMS',
    'OrderDate': '1997-12-10T00:00:00.000Z',
    'ShippedDate': '1997-12-19T00:00:00.000Z',
    'Freight': 91.28,
    'ShipName': 'Lehmanns Marktstand',

```

```

      'ShipAddress': 'Magazinweg 7',
      'ShipCity': 'Frankfurt a.M.',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10773,
      'CustomerID': 'ERNSH',
      'OrderDate': '1997-12-11T00:00:00.000Z',
      'ShippedDate': '1997-12-16T00:00:00.000Z',
      'Freight': 96.43,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10774,
      'CustomerID': 'FOLKO',
      'OrderDate': '1997-12-11T00:00:00.000Z',
      'ShippedDate': '1997-12-12T00:00:00.000Z',
      'Freight': 48.2,
      'ShipName': 'Folk och fä HB',
      'ShipAddress': 'Åkergatan 24',
      'ShipCity': 'Bräcke',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 10775,
      'CustomerID': 'THECR',
      'OrderDate': '1997-12-12T00:00:00.000Z',
      'ShippedDate': '1997-12-26T00:00:00.000Z',
      'Freight': 20.25,
      'ShipName': 'The Cracker Box',
      'ShipAddress': '55 Grizzly Peak Rd.',
      'ShipCity': 'Butte',
      'ShipRegion': 'MT',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10776,
      'CustomerID': 'ERNSH',
      'OrderDate': '1997-12-15T00:00:00.000Z',
      'ShippedDate': '1997-12-18T00:00:00.000Z',
      'Freight': 351.53,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10777,
      'CustomerID': 'GOURL',
      'OrderDate': '1997-12-15T00:00:00.000Z',

```

```

    'ShippedDate': '1998-01-21T00:00:00.000Z',
    'Freight': 3.01,
    'ShipName': 'Gourmet Lanchonetes',
    'ShipAddress': 'Av. Brasil, 442',
    'ShipCity': 'Campinas',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10778,
    'CustomerID': 'BERGS',
    'OrderDate': '1997-12-16T00:00:00.000Z',
    'ShippedDate': '1997-12-24T00:00:00.000Z',
    'Freight': 6.79,
    'ShipName': 'Berglunds snabbköp',
    'ShipAddress': 'Berguvsvägen 8',
    'ShipCity': 'Luleå',
    'ShipRegion': null,
    'ShipCountry': 'Sweden'
  },
  {
    'OrderID': 10779,
    'CustomerID': 'MORGK',
    'OrderDate': '1997-12-16T00:00:00.000Z',
    'ShippedDate': '1998-01-14T00:00:00.000Z',
    'Freight': 58.13,
    'ShipName': 'Morgenstern Gesundkost',
    'ShipAddress': 'Heerstr. 22',
    'ShipCity': 'Leipzig',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10780,
    'CustomerID': 'LILAS',
    'OrderDate': '1997-12-16T00:00:00.000Z',
    'ShippedDate': '1997-12-25T00:00:00.000Z',
    'Freight': 42.13,
    'ShipName': 'LILA-Supermercado',
    'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
    'ShipCity': 'Barquisimeto',
    'ShipRegion': 'Lara',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10781,
    'CustomerID': 'WARTH',
    'OrderDate': '1997-12-17T00:00:00.000Z',
    'ShippedDate': '1997-12-19T00:00:00.000Z',
    'Freight': 73.16,
    'ShipName': 'Wartian Herkku',
    'ShipAddress': 'Torikatu 38',
    'ShipCity': 'Oulu',
    'ShipRegion': null,
    'ShipCountry': 'Finland'
  }
]

```

```

      'OrderID': 10782,
      'CustomerID': 'CACTU',
      'OrderDate': '1997-12-17T00:00:00.000Z',
      'ShippedDate': '1997-12-22T00:00:00.000Z',
      'Freight': 1.1,
      'ShipName': 'Cactus Comidas para llevar',
      'ShipAddress': 'Cerrito 333',
      'ShipCity': 'Buenos Aires',
      'ShipRegion': null,
      'ShipCountry': 'Argentina'
    },
    {
      'OrderID': 10783,
      'CustomerID': 'HANAR',
      'OrderDate': '1997-12-18T00:00:00.000Z',
      'ShippedDate': '1997-12-19T00:00:00.000Z',
      'Freight': 124.98,
      'ShipName': 'Hanari Carnes',
      'ShipAddress': 'Rua do Paço, 67',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10784,
      'CustomerID': 'MAGAA',
      'OrderDate': '1997-12-18T00:00:00.000Z',
      'ShippedDate': '1997-12-22T00:00:00.000Z',
      'Freight': 70.09,
      'ShipName': 'Magazzini Alimentari Riuniti',
      'ShipAddress': 'Via Ludovico il Moro 22',
      'ShipCity': 'Bergamo',
      'ShipRegion': null,
      'ShipCountry': 'Italy'
    },
    {
      'OrderID': 10785,
      'CustomerID': 'GROSR',
      'OrderDate': '1997-12-18T00:00:00.000Z',
      'ShippedDate': '1997-12-24T00:00:00.000Z',
      'Freight': 1.51,
      'ShipName': 'GROSELLA-Restaurante',
      'ShipAddress': '5ª Ave. Los Palos Grandes',
      'ShipCity': 'Caracas',
      'ShipRegion': 'DF',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10786,
      'CustomerID': 'QUEEN',
      'OrderDate': '1997-12-19T00:00:00.000Z',
      'ShippedDate': '1997-12-23T00:00:00.000Z',
      'Freight': 110.87,
      'ShipName': 'Queen Cozinha',
      'ShipAddress': 'Alameda dos Canários, 891',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',

```

```

    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10787,
    'CustomerID': 'LAMAI',
    'OrderDate': '1997-12-19T00:00:00.000Z',
    'ShippedDate': '1997-12-26T00:00:00.000Z',
    'Freight': 249.93,
    'ShipName': 'La maison d\'Asie',
    'ShipAddress': '1 rue Alsace-Lorraine',
    'ShipCity': 'Toulouse',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10788,
    'CustomerID': 'QUICK',
    'OrderDate': '1997-12-22T00:00:00.000Z',
    'ShippedDate': '1998-01-19T00:00:00.000Z',
    'Freight': 42.7,
    'ShipName': 'QUICK-Stop',
    'ShipAddress': 'Taucherstraße 10',
    'ShipCity': 'Cunewalde',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10789,
    'CustomerID': 'FOLIG',
    'OrderDate': '1997-12-22T00:00:00.000Z',
    'ShippedDate': '1997-12-31T00:00:00.000Z',
    'Freight': 100.6,
    'ShipName': 'Folies gourmandes',
    'ShipAddress': '184, chaussée de Tournai',
    'ShipCity': 'Lille',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10790,
    'CustomerID': 'GOURL',
    'OrderDate': '1997-12-22T00:00:00.000Z',
    'ShippedDate': '1997-12-26T00:00:00.000Z',
    'Freight': 28.23,
    'ShipName': 'Gourmet Lanchonetes',
    'ShipAddress': 'Av. Brasil, 442',
    'ShipCity': 'Campinas',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10791,
    'CustomerID': 'FRANK',
    'OrderDate': '1997-12-23T00:00:00.000Z',
    'ShippedDate': '1998-01-01T00:00:00.000Z',
    'Freight': 16.85,
    'ShipName': 'Frankenversand',

```

```

      'ShipAddress': 'Berliner Platz 43',
      'ShipCity': 'München',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10792,
      'CustomerID': 'WOLZA',
      'OrderDate': '1997-12-23T00:00:00.000Z',
      'ShippedDate': '1997-12-31T00:00:00.000Z',
      'Freight': 23.79,
      'ShipName': 'Wolski Zajazd',
      'ShipAddress': 'ul. Filtrowa 68',
      'ShipCity': 'Warszawa',
      'ShipRegion': null,
      'ShipCountry': 'Poland'
    },
    {
      'OrderID': 10793,
      'CustomerID': 'AROUT',
      'OrderDate': '1997-12-24T00:00:00.000Z',
      'ShippedDate': '1998-01-08T00:00:00.000Z',
      'Freight': 4.52,
      'ShipName': 'Around the Horn',
      'ShipAddress': 'Brook Farm Stratford St. Mary',
      'ShipCity': 'Colchester',
      'ShipRegion': 'Essex',
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10794,
      'CustomerID': 'QUEDE',
      'OrderDate': '1997-12-24T00:00:00.000Z',
      'ShippedDate': '1998-01-02T00:00:00.000Z',
      'Freight': 21.49,
      'ShipName': 'Que Delícia',
      'ShipAddress': 'Rua da Panificadora, 12',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10795,
      'CustomerID': 'ERNSH',
      'OrderDate': '1997-12-24T00:00:00.000Z',
      'ShippedDate': '1998-01-20T00:00:00.000Z',
      'Freight': 126.66,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10796,
      'CustomerID': 'HILAA',
      'OrderDate': '1997-12-25T00:00:00.000Z',

```

```

    'ShippedDate': '1998-01-14T00:00:00.000Z',
    'Freight': 26.52,
    'ShipName': 'HILARION-Abastos',
    'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
    'ShipCity': 'San Cristóbal',
    'ShipRegion': 'Táchira',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10797,
    'CustomerID': 'DRACD',
    'OrderDate': '1997-12-25T00:00:00.000Z',
    'ShippedDate': '1998-01-05T00:00:00.000Z',
    'Freight': 33.35,
    'ShipName': 'Drachenblut Delikatessen',
    'ShipAddress': 'Walserweg 21',
    'ShipCity': 'Aachen',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10798,
    'CustomerID': 'ISLAT',
    'OrderDate': '1997-12-26T00:00:00.000Z',
    'ShippedDate': '1998-01-05T00:00:00.000Z',
    'Freight': 2.33,
    'ShipName': 'Island Trading',
    'ShipAddress': 'Garden House Crowther Way',
    'ShipCity': 'Cowes',
    'ShipRegion': 'Isle of Wight',
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10799,
    'CustomerID': 'KOENE',
    'OrderDate': '1997-12-26T00:00:00.000Z',
    'ShippedDate': '1998-01-05T00:00:00.000Z',
    'Freight': 30.76,
    'ShipName': 'Königlich Essen',
    'ShipAddress': 'Maubelstr. 90',
    'ShipCity': 'Brandenburg',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10800,
    'CustomerID': 'SEVES',
    'OrderDate': '1997-12-26T00:00:00.000Z',
    'ShippedDate': '1998-01-05T00:00:00.000Z',
    'Freight': 137.44,
    'ShipName': 'Seven Seas Imports',
    'ShipAddress': '90 Wadhurst Rd.',
    'ShipCity': 'London',
    'ShipRegion': null,
    'ShipCountry': 'UK'
  },
  {

```



```

      'OrderID': 10801,
      'CustomerID': 'BOLID',
      'OrderDate': '1997-12-29T00:00:00.000Z',
      'ShippedDate': '1997-12-31T00:00:00.000Z',
      'Freight': 97.09,
      'ShipName': 'Bólido Comidas preparadas',
      'ShipAddress': 'C/ Araquil, 67',
      'ShipCity': 'Madrid',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    },
    {
      'OrderID': 10802,
      'CustomerID': 'SIMOB',
      'OrderDate': '1997-12-29T00:00:00.000Z',
      'ShippedDate': '1998-01-02T00:00:00.000Z',
      'Freight': 257.26,
      'ShipName': 'Simons bistro',
      'ShipAddress': 'Vinbæltet 34',
      'ShipCity': 'Kobenhavn',
      'ShipRegion': null,
      'ShipCountry': 'Denmark'
    },
    {
      'OrderID': 10803,
      'CustomerID': 'WELLI',
      'OrderDate': '1997-12-30T00:00:00.000Z',
      'ShippedDate': '1998-01-06T00:00:00.000Z',
      'Freight': 55.23,
      'ShipName': 'Wellington Importadora',
      'ShipAddress': 'Rua do Mercado, 12',
      'ShipCity': 'Resende',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10804,
      'CustomerID': 'SEVES',
      'OrderDate': '1997-12-30T00:00:00.000Z',
      'ShippedDate': '1998-01-07T00:00:00.000Z',
      'Freight': 27.33,
      'ShipName': 'Seven Seas Imports',
      'ShipAddress': '90 Wadhurst Rd.',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10805,
      'CustomerID': 'THEBI',
      'OrderDate': '1997-12-30T00:00:00.000Z',
      'ShippedDate': '1998-01-09T00:00:00.000Z',
      'Freight': 237.34,
      'ShipName': 'The Big Cheese',
      'ShipAddress': '89 Jefferson Way Suite 2',
      'ShipCity': 'Portland',
      'ShipRegion': 'OR',

```

```

    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10806,
    'CustomerID': 'VICTE',
    'OrderDate': '1997-12-31T00:00:00.000Z',
    'ShippedDate': '1998-01-05T00:00:00.000Z',
    'Freight': 22.11,
    'ShipName': 'Victuailles en stock',
    'ShipAddress': '2, rue du Commerce',
    'ShipCity': 'Lyon',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10807,
    'CustomerID': 'FRANS',
    'OrderDate': '1997-12-31T00:00:00.000Z',
    'ShippedDate': '1998-01-30T00:00:00.000Z',
    'Freight': 1.36,
    'ShipName': 'Franchi S.p.A.',
    'ShipAddress': 'Via Monte Bianco 34',
    'ShipCity': 'Torino',
    'ShipRegion': null,
    'ShipCountry': 'Italy'
  },
  {
    'OrderID': 10808,
    'CustomerID': 'OLDWO',
    'OrderDate': '1998-01-01T00:00:00.000Z',
    'ShippedDate': '1998-01-09T00:00:00.000Z',
    'Freight': 45.53,
    'ShipName': 'Old World Delicatessen',
    'ShipAddress': '2743 Bering St.',
    'ShipCity': 'Anchorage',
    'ShipRegion': 'AK',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10809,
    'CustomerID': 'WELLI',
    'OrderDate': '1998-01-01T00:00:00.000Z',
    'ShippedDate': '1998-01-07T00:00:00.000Z',
    'Freight': 4.87,
    'ShipName': 'Wellington Importadora',
    'ShipAddress': 'Rua do Mercado, 12',
    'ShipCity': 'Resende',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10810,
    'CustomerID': 'LAUGB',
    'OrderDate': '1998-01-01T00:00:00.000Z',
    'ShippedDate': '1998-01-07T00:00:00.000Z',
    'Freight': 4.33,
    'ShipName': 'Laughing Bacchus Wine Cellars',

```

```

      'ShipAddress': '2319 Elm St.',
      'ShipCity': 'Vancouver',
      'ShipRegion': 'BC',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10811,
      'CustomerID': 'LINOD',
      'OrderDate': '1998-01-02T00:00:00.000Z',
      'ShippedDate': '1998-01-08T00:00:00.000Z',
      'Freight': 31.22,
      'ShipName': 'LINO-Delicateses',
      'ShipAddress': 'Ave. 5 de Mayo Porlamar',
      'ShipCity': 'I. de Margarita',
      'ShipRegion': 'Nueva Esparta',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10812,
      'CustomerID': 'REGGC',
      'OrderDate': '1998-01-02T00:00:00.000Z',
      'ShippedDate': '1998-01-12T00:00:00.000Z',
      'Freight': 59.78,
      'ShipName': 'Reggiani Caseifici',
      'ShipAddress': 'Strada Provinciale 124',
      'ShipCity': 'Reggio Emilia',
      'ShipRegion': null,
      'ShipCountry': 'Italy'
    },
    {
      'OrderID': 10813,
      'CustomerID': 'RICAR',
      'OrderDate': '1998-01-05T00:00:00.000Z',
      'ShippedDate': '1998-01-09T00:00:00.000Z',
      'Freight': 47.38,
      'ShipName': 'Ricardo Adocicados',
      'ShipAddress': 'Av. Copacabana, 267',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10814,
      'CustomerID': 'VICTE',
      'OrderDate': '1998-01-05T00:00:00.000Z',
      'ShippedDate': '1998-01-14T00:00:00.000Z',
      'Freight': 130.94,
      'ShipName': 'Victuailles en stock',
      'ShipAddress': '2, rue du Commerce',
      'ShipCity': 'Lyon',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10815,
      'CustomerID': 'SAVEA',
      'OrderDate': '1998-01-05T00:00:00.000Z',

```

```

        'ShippedDate': '1998-01-14T00:00:00.000Z',
        'Freight': 14.62,
        'ShipName': 'Save-a-lot Markets',
        'ShipAddress': '187 Suffolk Ln.',
        'ShipCity': 'Boise',
        'ShipRegion': 'ID',
        'ShipCountry': 'USA'
    },
    {
        'OrderID': 10816,
        'CustomerID': 'GREAL',
        'OrderDate': '1998-01-06T00:00:00.000Z',
        'ShippedDate': '1998-02-04T00:00:00.000Z',
        'Freight': 719.78,
        'ShipName': 'Great Lakes Food Market',
        'ShipAddress': '2732 Baker Blvd.',
        'ShipCity': 'Eugene',
        'ShipRegion': 'OR',
        'ShipCountry': 'USA'
    },
    {
        'OrderID': 10817,
        'CustomerID': 'KOENE',
        'OrderDate': '1998-01-06T00:00:00.000Z',
        'ShippedDate': '1998-01-13T00:00:00.000Z',
        'Freight': 306.07,
        'ShipName': 'Königlich Essen',
        'ShipAddress': 'Maubelstr. 90',
        'ShipCity': 'Brandenburg',
        'ShipRegion': null,
        'ShipCountry': 'Germany'
    },
    {
        'OrderID': 10818,
        'CustomerID': 'MAGAA',
        'OrderDate': '1998-01-07T00:00:00.000Z',
        'ShippedDate': '1998-01-12T00:00:00.000Z',
        'Freight': 65.48,
        'ShipName': 'Magazzini Alimentari Riuniti',
        'ShipAddress': 'Via Ludovico il Moro 22',
        'ShipCity': 'Bergamo',
        'ShipRegion': null,
        'ShipCountry': 'Italy'
    },
    {
        'OrderID': 10819,
        'CustomerID': 'CACTU',
        'OrderDate': '1998-01-07T00:00:00.000Z',
        'ShippedDate': '1998-01-16T00:00:00.000Z',
        'Freight': 19.76,
        'ShipName': 'Cactus Comidas para llevar',
        'ShipAddress': 'Cerrito 333',
        'ShipCity': 'Buenos Aires',
        'ShipRegion': null,
        'ShipCountry': 'Argentina'
    },
    {

```

```

      'OrderID': 10820,
      'CustomerID': 'RATTC',
      'OrderDate': '1998-01-07T00:00:00.000Z',
      'ShippedDate': '1998-01-13T00:00:00.000Z',
      'Freight': 37.52,
      'ShipName': 'Rattlesnake Canyon Grocery',
      'ShipAddress': '2817 Milton Dr.',
      'ShipCity': 'Albuquerque',
      'ShipRegion': 'NM',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10821,
      'CustomerID': 'SPLIR',
      'OrderDate': '1998-01-08T00:00:00.000Z',
      'ShippedDate': '1998-01-15T00:00:00.000Z',
      'Freight': 36.68,
      'ShipName': 'Split Rail Beer & Ale',
      'ShipAddress': 'P.O. Box 555',
      'ShipCity': 'Lander',
      'ShipRegion': 'WY',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10822,
      'CustomerID': 'TRAIH',
      'OrderDate': '1998-01-08T00:00:00.000Z',
      'ShippedDate': '1998-01-16T00:00:00.000Z',
      'Freight': 7,
      'ShipName': 'Trail\' Head Gourmet Provisioners',
      'ShipAddress': '722 DaVinci Blvd.',
      'ShipCity': 'Kirkland',
      'ShipRegion': 'WA',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10823,
      'CustomerID': 'LILAS',
      'OrderDate': '1998-01-09T00:00:00.000Z',
      'ShippedDate': '1998-01-13T00:00:00.000Z',
      'Freight': 163.97,
      'ShipName': 'LILA-Supermercado',
      'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
      'ShipCity': 'Barquisimeto',
      'ShipRegion': 'Lara',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10824,
      'CustomerID': 'FOLKO',
      'OrderDate': '1998-01-09T00:00:00.000Z',
      'ShippedDate': '1998-01-30T00:00:00.000Z',
      'Freight': 1.23,
      'ShipName': 'Folk och fä HB',
      'ShipAddress': 'Åkergatan 24',
      'ShipCity': 'Bräcke',
      'ShipRegion': null,

```

```

    'ShipCountry': 'Sweden'
  },
  {
    'OrderID': 10825,
    'CustomerID': 'DRACD',
    'OrderDate': '1998-01-09T00:00:00.000Z',
    'ShippedDate': '1998-01-14T00:00:00.000Z',
    'Freight': 79.25,
    'ShipName': 'Drachenblut Delikatessen',
    'ShipAddress': 'Walserweg 21',
    'ShipCity': 'Aachen',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10826,
    'CustomerID': 'BLONP',
    'OrderDate': '1998-01-12T00:00:00.000Z',
    'ShippedDate': '1998-02-06T00:00:00.000Z',
    'Freight': 7.09,
    'ShipName': 'Blondel père et fils',
    'ShipAddress': '24, place Kléber',
    'ShipCity': 'Strasbourg',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10827,
    'CustomerID': 'BONAP',
    'OrderDate': '1998-01-12T00:00:00.000Z',
    'ShippedDate': '1998-02-06T00:00:00.000Z',
    'Freight': 63.54,
    'ShipName': 'Bon app',
    'ShipAddress': '12, rue des Bouchers',
    'ShipCity': 'Marseille',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10828,
    'CustomerID': 'RANCH',
    'OrderDate': '1998-01-13T00:00:00.000Z',
    'ShippedDate': '1998-02-04T00:00:00.000Z',
    'Freight': 90.85,
    'ShipName': 'Rancho grande',
    'ShipAddress': 'Av. del Libertador 900',
    'ShipCity': 'Buenos Aires',
    'ShipRegion': null,
    'ShipCountry': 'Argentina'
  },
  {
    'OrderID': 10829,
    'CustomerID': 'ISLAT',
    'OrderDate': '1998-01-13T00:00:00.000Z',
    'ShippedDate': '1998-01-23T00:00:00.000Z',
    'Freight': 154.72,
    'ShipName': 'Island Trading',

```

```

      'ShipAddress': 'Garden House Crowther Way',
      'ShipCity': 'Cowes',
      'ShipRegion': 'Isle of Wight',
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10830,
      'CustomerID': 'TRADH',
      'OrderDate': '1998-01-13T00:00:00.000Z',
      'ShippedDate': '1998-01-21T00:00:00.000Z',
      'Freight': 81.83,
      'ShipName': 'Tradição Hipermercados',
      'ShipAddress': 'Av. Inês de Castro, 414',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10831,
      'CustomerID': 'SANTG',
      'OrderDate': '1998-01-14T00:00:00.000Z',
      'ShippedDate': '1998-01-23T00:00:00.000Z',
      'Freight': 72.19,
      'ShipName': 'Santé Gourmet',
      'ShipAddress': 'Erling Skakkess gate 78',
      'ShipCity': 'Stavern',
      'ShipRegion': null,
      'ShipCountry': 'Norway'
    },
    {
      'OrderID': 10832,
      'CustomerID': 'LAMAI',
      'OrderDate': '1998-01-14T00:00:00.000Z',
      'ShippedDate': '1998-01-19T00:00:00.000Z',
      'Freight': 43.26,
      'ShipName': 'La maison d\'Asie',
      'ShipAddress': '1 rue Alsace-Lorraine',
      'ShipCity': 'Toulouse',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10833,
      'CustomerID': 'OTTIK',
      'OrderDate': '1998-01-15T00:00:00.000Z',
      'ShippedDate': '1998-01-23T00:00:00.000Z',
      'Freight': 71.49,
      'ShipName': 'Ottilies Käseladen',
      'ShipAddress': 'Mehrheimerstr. 369',
      'ShipCity': 'Köln',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10834,
      'CustomerID': 'TRADH',
      'OrderDate': '1998-01-15T00:00:00.000Z',

```

```

    'ShippedDate': '1998-01-19T00:00:00.000Z',
    'Freight': 29.78,
    'ShipName': 'Tradiçao Hipermercados',
    'ShipAddress': 'Av. Inês de Castro, 414',
    'ShipCity': 'Sao Paulo',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10835,
    'CustomerID': 'ALFKI',
    'OrderDate': '1998-01-15T00:00:00.000Z',
    'ShippedDate': '1998-01-21T00:00:00.000Z',
    'Freight': 69.53,
    'ShipName': 'Alfred\' Futterkiste',
    'ShipAddress': 'Obere Str. 57',
    'ShipCity': 'Berlin',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10836,
    'CustomerID': 'ERNSH',
    'OrderDate': '1998-01-16T00:00:00.000Z',
    'ShippedDate': '1998-01-21T00:00:00.000Z',
    'Freight': 411.88,
    'ShipName': 'Ernst Handel',
    'ShipAddress': 'Kirchgasse 6',
    'ShipCity': 'Graz',
    'ShipRegion': null,
    'ShipCountry': 'Austria'
  },
  {
    'OrderID': 10837,
    'CustomerID': 'BERGS',
    'OrderDate': '1998-01-16T00:00:00.000Z',
    'ShippedDate': '1998-01-23T00:00:00.000Z',
    'Freight': 13.32,
    'ShipName': 'Berglunds snabbköp',
    'ShipAddress': 'Berguvsvägen 8',
    'ShipCity': 'Luleå',
    'ShipRegion': null,
    'ShipCountry': 'Sweden'
  },
  {
    'OrderID': 10838,
    'CustomerID': 'LINOD',
    'OrderDate': '1998-01-19T00:00:00.000Z',
    'ShippedDate': '1998-01-23T00:00:00.000Z',
    'Freight': 59.28,
    'ShipName': 'LINO-Delicateses',
    'ShipAddress': 'Ave. 5 de Mayo Porlamar',
    'ShipCity': 'I. de Margarita',
    'ShipRegion': 'Nueva Esparta',
    'ShipCountry': 'Venezuela'
  },
  {

```



```

      'OrderID': 10839,
      'CustomerID': 'TRADH',
      'OrderDate': '1998-01-19T00:00:00.000Z',
      'ShippedDate': '1998-01-22T00:00:00.000Z',
      'Freight': 35.43,
      'ShipName': 'Tradiçao Hipermercados',
      'ShipAddress': 'Av. Inês de Castro, 414',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10840,
      'CustomerID': 'LINOD',
      'OrderDate': '1998-01-19T00:00:00.000Z',
      'ShippedDate': '1998-02-16T00:00:00.000Z',
      'Freight': 2.71,
      'ShipName': 'LINO-Delicateses',
      'ShipAddress': 'Ave. 5 de Mayo Porlamar',
      'ShipCity': 'I. de Margarita',
      'ShipRegion': 'Nueva Esparta',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10841,
      'CustomerID': 'SUPRD',
      'OrderDate': '1998-01-20T00:00:00.000Z',
      'ShippedDate': '1998-01-29T00:00:00.000Z',
      'Freight': 424.3,
      'ShipName': 'Suprêmes délices',
      'ShipAddress': 'Boulevard Tirou, 255',
      'ShipCity': 'Charleroi',
      'ShipRegion': null,
      'ShipCountry': 'Belgium'
    },
    {
      'OrderID': 10842,
      'CustomerID': 'TORTU',
      'OrderDate': '1998-01-20T00:00:00.000Z',
      'ShippedDate': '1998-01-29T00:00:00.000Z',
      'Freight': 54.42,
      'ShipName': 'Tortuga Restaurante',
      'ShipAddress': 'Avda. Azteca 123',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10843,
      'CustomerID': 'VICTE',
      'OrderDate': '1998-01-21T00:00:00.000Z',
      'ShippedDate': '1998-01-26T00:00:00.000Z',
      'Freight': 9.26,
      'ShipName': 'Victuailles en stock',
      'ShipAddress': '2, rue du Commerce',
      'ShipCity': 'Lyon',
      'ShipRegion': null,

```

```

    'ShipCountry': 'France'
  },
  {
    'OrderID': 10844,
    'CustomerID': 'PICCO',
    'OrderDate': '1998-01-21T00:00:00.000Z',
    'ShippedDate': '1998-01-26T00:00:00.000Z',
    'Freight': 25.22,
    'ShipName': 'Piccolo und mehr',
    'ShipAddress': 'Geislweg 14',
    'ShipCity': 'Salzburg',
    'ShipRegion': null,
    'ShipCountry': 'Austria'
  },
  {
    'OrderID': 10845,
    'CustomerID': 'QUICK',
    'OrderDate': '1998-01-21T00:00:00.000Z',
    'ShippedDate': '1998-01-30T00:00:00.000Z',
    'Freight': 212.98,
    'ShipName': 'QUICK-Stop',
    'ShipAddress': 'Taucherstraße 10',
    'ShipCity': 'Cunewalde',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10846,
    'CustomerID': 'SUPRD',
    'OrderDate': '1998-01-22T00:00:00.000Z',
    'ShippedDate': '1998-01-23T00:00:00.000Z',
    'Freight': 56.46,
    'ShipName': 'Suprêmes délices',
    'ShipAddress': 'Boulevard Tirou, 255',
    'ShipCity': 'Charleroi',
    'ShipRegion': null,
    'ShipCountry': 'Belgium'
  },
  {
    'OrderID': 10847,
    'CustomerID': 'SAVEA',
    'OrderDate': '1998-01-22T00:00:00.000Z',
    'ShippedDate': '1998-02-10T00:00:00.000Z',
    'Freight': 487.57,
    'ShipName': 'Save-a-lot Markets',
    'ShipAddress': '187 Suffolk Ln.',
    'ShipCity': 'Boise',
    'ShipRegion': 'ID',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10848,
    'CustomerID': 'CONSH',
    'OrderDate': '1998-01-23T00:00:00.000Z',
    'ShippedDate': '1998-01-29T00:00:00.000Z',
    'Freight': 38.24,
    'ShipName': 'Consolidated Holdings',

```

```

      'ShipAddress': 'Berkeley Gardens 12 Brewery',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10849,
      'CustomerID': 'KOENE',
      'OrderDate': '1998-01-23T00:00:00.000Z',
      'ShippedDate': '1998-01-30T00:00:00.000Z',
      'Freight': 0.56,
      'ShipName': 'Königlich Essen',
      'ShipAddress': 'Maubelstr. 90',
      'ShipCity': 'Brandenburg',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10850,
      'CustomerID': 'VICTE',
      'OrderDate': '1998-01-23T00:00:00.000Z',
      'ShippedDate': '1998-01-30T00:00:00.000Z',
      'Freight': 49.19,
      'ShipName': 'Victuailles en stock',
      'ShipAddress': '2, rue du Commerce',
      'ShipCity': 'Lyon',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10851,
      'CustomerID': 'RICAR',
      'OrderDate': '1998-01-26T00:00:00.000Z',
      'ShippedDate': '1998-02-02T00:00:00.000Z',
      'Freight': 160.55,
      'ShipName': 'Ricardo Adocicados',
      'ShipAddress': 'Av. Copacabana, 267',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10852,
      'CustomerID': 'RATTC',
      'OrderDate': '1998-01-26T00:00:00.000Z',
      'ShippedDate': '1998-01-30T00:00:00.000Z',
      'Freight': 174.05,
      'ShipName': 'Rattlesnake Canyon Grocery',
      'ShipAddress': '2817 Milton Dr.',
      'ShipCity': 'Albuquerque',
      'ShipRegion': 'NM',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10853,
      'CustomerID': 'BLAUS',
      'OrderDate': '1998-01-27T00:00:00.000Z',

```

```

    'ShippedDate': '1998-02-03T00:00:00.000Z',
    'Freight': 53.83,
    'ShipName': 'Blauer See Delikatessen',
    'ShipAddress': 'Forsterstr. 57',
    'ShipCity': 'Mannheim',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10854,
    'CustomerID': 'ERNSH',
    'OrderDate': '1998-01-27T00:00:00.000Z',
    'ShippedDate': '1998-02-05T00:00:00.000Z',
    'Freight': 100.22,
    'ShipName': 'Ernst Handel',
    'ShipAddress': 'Kirchgasse 6',
    'ShipCity': 'Graz',
    'ShipRegion': null,
    'ShipCountry': 'Austria'
  },
  {
    'OrderID': 10855,
    'CustomerID': 'OLDWO',
    'OrderDate': '1998-01-27T00:00:00.000Z',
    'ShippedDate': '1998-02-04T00:00:00.000Z',
    'Freight': 170.97,
    'ShipName': 'Old World Delicatessen',
    'ShipAddress': '2743 Bering St.',
    'ShipCity': 'Anchorage',
    'ShipRegion': 'AK',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10856,
    'CustomerID': 'ANTON',
    'OrderDate': '1998-01-28T00:00:00.000Z',
    'ShippedDate': '1998-02-10T00:00:00.000Z',
    'Freight': 58.43,
    'ShipName': 'Antonio Moreno Taquería',
    'ShipAddress': 'Mataderos 2312',
    'ShipCity': 'México D.F.',
    'ShipRegion': null,
    'ShipCountry': 'Mexico'
  },
  {
    'OrderID': 10857,
    'CustomerID': 'BERGS',
    'OrderDate': '1998-01-28T00:00:00.000Z',
    'ShippedDate': '1998-02-06T00:00:00.000Z',
    'Freight': 188.85,
    'ShipName': 'Berglunds snabbköp',
    'ShipAddress': 'Berguvsvägen 8',
    'ShipCity': 'Luleå',
    'ShipRegion': null,
    'ShipCountry': 'Sweden'
  },
  {

```

```

      'OrderID': 10858,
      'CustomerID': 'LACOR',
      'OrderDate': '1998-01-29T00:00:00.000Z',
      'ShippedDate': '1998-02-03T00:00:00.000Z',
      'Freight': 52.51,
      'ShipName': 'La corne d\'abondance',
      'ShipAddress': '67, avenue de l\'Europe',
      'ShipCity': 'Versailles',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10859,
      'CustomerID': 'FRANK',
      'OrderDate': '1998-01-29T00:00:00.000Z',
      'ShippedDate': '1998-02-02T00:00:00.000Z',
      'Freight': 76.1,
      'ShipName': 'Frankenversand',
      'ShipAddress': 'Berliner Platz 43',
      'ShipCity': 'München',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10860,
      'CustomerID': 'FRANR',
      'OrderDate': '1998-01-29T00:00:00.000Z',
      'ShippedDate': '1998-02-04T00:00:00.000Z',
      'Freight': 19.26,
      'ShipName': 'France restauration',
      'ShipAddress': '54, rue Royale',
      'ShipCity': 'Nantes',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10861,
      'CustomerID': 'WHITC',
      'OrderDate': '1998-01-30T00:00:00.000Z',
      'ShippedDate': '1998-02-17T00:00:00.000Z',
      'Freight': 14.93,
      'ShipName': 'White Clover Markets',
      'ShipAddress': '1029 - 12th Ave. S.',
      'ShipCity': 'Seattle',
      'ShipRegion': 'WA',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10862,
      'CustomerID': 'LEHMS',
      'OrderDate': '1998-01-30T00:00:00.000Z',
      'ShippedDate': '1998-02-02T00:00:00.000Z',
      'Freight': 53.23,
      'ShipName': 'Lehmanns Marktstand',
      'ShipAddress': 'Magazinweg 7',
      'ShipCity': 'Frankfurt a.M.',
      'ShipRegion': null,

```

```

    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10863,
    'CustomerID': 'HILAA',
    'OrderDate': '1998-02-02T00:00:00.000Z',
    'ShippedDate': '1998-02-17T00:00:00.000Z',
    'Freight': 30.26,
    'ShipName': 'HILARION-Abastos',
    'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
    'ShipCity': 'San Cristóbal',
    'ShipRegion': 'Táchira',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10864,
    'CustomerID': 'AROUT',
    'OrderDate': '1998-02-02T00:00:00.000Z',
    'ShippedDate': '1998-02-09T00:00:00.000Z',
    'Freight': 3.04,
    'ShipName': 'Around the Horn',
    'ShipAddress': 'Brook Farm Stratford St. Mary',
    'ShipCity': 'Colchester',
    'ShipRegion': 'Essex',
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10865,
    'CustomerID': 'QUICK',
    'OrderDate': '1998-02-02T00:00:00.000Z',
    'ShippedDate': '1998-02-12T00:00:00.000Z',
    'Freight': 348.14,
    'ShipName': 'QUICK-Stop',
    'ShipAddress': 'Taucherstraße 10',
    'ShipCity': 'Cunewalde',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10866,
    'CustomerID': 'BERGS',
    'OrderDate': '1998-02-03T00:00:00.000Z',
    'ShippedDate': '1998-02-12T00:00:00.000Z',
    'Freight': 109.11,
    'ShipName': 'Berglunds snabbköp',
    'ShipAddress': 'Berguvsvägen 8',
    'ShipCity': 'Luleå',
    'ShipRegion': null,
    'ShipCountry': 'Sweden'
  },
  {
    'OrderID': 10867,
    'CustomerID': 'LONEP',
    'OrderDate': '1998-02-03T00:00:00.000Z',
    'ShippedDate': '1998-02-11T00:00:00.000Z',
    'Freight': 1.93,
    'ShipName': 'Lonesome Pine Restaurant',

```

```

      'ShipAddress': '89 Chiaroscuro Rd.',
      'ShipCity': 'Portland',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10868,
      'CustomerID': 'QUEEN',
      'OrderDate': '1998-02-04T00:00:00.000Z',
      'ShippedDate': '1998-02-23T00:00:00.000Z',
      'Freight': 191.27,
      'ShipName': 'Queen Cozinha',
      'ShipAddress': 'Alameda dos Canários, 891',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10869,
      'CustomerID': 'SEVES',
      'OrderDate': '1998-02-04T00:00:00.000Z',
      'ShippedDate': '1998-02-09T00:00:00.000Z',
      'Freight': 143.28,
      'ShipName': 'Seven Seas Imports',
      'ShipAddress': '90 Wadhurst Rd.',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10870,
      'CustomerID': 'WOLZA',
      'OrderDate': '1998-02-04T00:00:00.000Z',
      'ShippedDate': '1998-02-13T00:00:00.000Z',
      'Freight': 12.04,
      'ShipName': 'Wolski Zajazd',
      'ShipAddress': 'ul. Filtrowa 68',
      'ShipCity': 'Warszawa',
      'ShipRegion': null,
      'ShipCountry': 'Poland'
    },
    {
      'OrderID': 10871,
      'CustomerID': 'BONAP',
      'OrderDate': '1998-02-05T00:00:00.000Z',
      'ShippedDate': '1998-02-10T00:00:00.000Z',
      'Freight': 112.27,
      'ShipName': 'Bon app',
      'ShipAddress': '12, rue des Bouchers',
      'ShipCity': 'Marseille',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10872,
      'CustomerID': 'GODOS',
      'OrderDate': '1998-02-05T00:00:00.000Z',

```

```

      'ShippedDate': '1998-02-09T00:00:00.000Z',
      'Freight': 175.32,
      'ShipName': 'Godos Cocina Típica',
      'ShipAddress': 'C/ Romero, 33',
      'ShipCity': 'Sevilla',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    },
    {
      'OrderID': 10873,
      'CustomerID': 'WILMK',
      'OrderDate': '1998-02-06T00:00:00.000Z',
      'ShippedDate': '1998-02-09T00:00:00.000Z',
      'Freight': 0.82,
      'ShipName': 'Wilman Kala',
      'ShipAddress': 'Keskuskatu 45',
      'ShipCity': 'Helsinki',
      'ShipRegion': null,
      'ShipCountry': 'Finland'
    },
    {
      'OrderID': 10874,
      'CustomerID': 'GODOS',
      'OrderDate': '1998-02-06T00:00:00.000Z',
      'ShippedDate': '1998-02-11T00:00:00.000Z',
      'Freight': 19.58,
      'ShipName': 'Godos Cocina Típica',
      'ShipAddress': 'C/ Romero, 33',
      'ShipCity': 'Sevilla',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    },
    {
      'OrderID': 10875,
      'CustomerID': 'BERGS',
      'OrderDate': '1998-02-06T00:00:00.000Z',
      'ShippedDate': '1998-03-03T00:00:00.000Z',
      'Freight': 32.37,
      'ShipName': 'Berglunds snabbköp',
      'ShipAddress': 'Berguvsvägen 8',
      'ShipCity': 'Luleå',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 10876,
      'CustomerID': 'BONAP',
      'OrderDate': '1998-02-09T00:00:00.000Z',
      'ShippedDate': '1998-02-12T00:00:00.000Z',
      'Freight': 60.42,
      'ShipName': 'Bon app',
      'ShipAddress': '12, rue des Bouchers',
      'ShipCity': 'Marseille',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {

```



```

    'OrderID': 10877,
    'CustomerID': 'RICAR',
    'OrderDate': '1998-02-09T00:00:00.000Z',
    'ShippedDate': '1998-02-19T00:00:00.000Z',
    'Freight': 38.06,
    'ShipName': 'Ricardo Adocicados',
    'ShipAddress': 'Av. Copacabana, 267',
    'ShipCity': 'Rio de Janeiro',
    'ShipRegion': 'RJ',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10878,
    'CustomerID': 'QUICK',
    'OrderDate': '1998-02-10T00:00:00.000Z',
    'ShippedDate': '1998-02-12T00:00:00.000Z',
    'Freight': 46.69,
    'ShipName': 'QUICK-Stop',
    'ShipAddress': 'Taucherstraße 10',
    'ShipCity': 'Cunewalde',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10879,
    'CustomerID': 'WILMK',
    'OrderDate': '1998-02-10T00:00:00.000Z',
    'ShippedDate': '1998-02-12T00:00:00.000Z',
    'Freight': 8.5,
    'ShipName': 'Wilman Kala',
    'ShipAddress': 'Keskuskatu 45',
    'ShipCity': 'Helsinki',
    'ShipRegion': null,
    'ShipCountry': 'Finland'
  },
  {
    'OrderID': 10880,
    'CustomerID': 'FOLKO',
    'OrderDate': '1998-02-10T00:00:00.000Z',
    'ShippedDate': '1998-02-18T00:00:00.000Z',
    'Freight': 88.01,
    'ShipName': 'Folk och fä HB',
    'ShipAddress': 'Åkergatan 24',
    'ShipCity': 'Bräcke',
    'ShipRegion': null,
    'ShipCountry': 'Sweden'
  },
  {
    'OrderID': 10881,
    'CustomerID': 'CACTU',
    'OrderDate': '1998-02-11T00:00:00.000Z',
    'ShippedDate': '1998-02-18T00:00:00.000Z',
    'Freight': 2.84,
    'ShipName': 'Cactus Comidas para llevar',
    'ShipAddress': 'Cerrito 333',
    'ShipCity': 'Buenos Aires',
    'ShipRegion': null,

```

```

    'ShipCountry': 'Argentina'
  },
  {
    'OrderID': 10882,
    'CustomerID': 'SAVEA',
    'OrderDate': '1998-02-11T00:00:00.000Z',
    'ShippedDate': '1998-02-20T00:00:00.000Z',
    'Freight': 23.1,
    'ShipName': 'Save-a-lot Markets',
    'ShipAddress': '187 Suffolk Ln.',
    'ShipCity': 'Boise',
    'ShipRegion': 'ID',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10883,
    'CustomerID': 'LONEP',
    'OrderDate': '1998-02-12T00:00:00.000Z',
    'ShippedDate': '1998-02-20T00:00:00.000Z',
    'Freight': 0.53,
    'ShipName': 'Lonesome Pine Restaurant',
    'ShipAddress': '89 Chiaroscuro Rd.',
    'ShipCity': 'Portland',
    'ShipRegion': 'OR',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10884,
    'CustomerID': 'LETSS',
    'OrderDate': '1998-02-12T00:00:00.000Z',
    'ShippedDate': '1998-02-13T00:00:00.000Z',
    'Freight': 90.97,
    'ShipName': 'Let\' Stop N Shop',
    'ShipAddress': '87 Polk St. Suite 5',
    'ShipCity': 'San Francisco',
    'ShipRegion': 'CA',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10885,
    'CustomerID': 'SUPRD',
    'OrderDate': '1998-02-12T00:00:00.000Z',
    'ShippedDate': '1998-02-18T00:00:00.000Z',
    'Freight': 5.64,
    'ShipName': 'Suprêmes délices',
    'ShipAddress': 'Boulevard Tirou, 255',
    'ShipCity': 'Charleroi',
    'ShipRegion': null,
    'ShipCountry': 'Belgium'
  },
  {
    'OrderID': 10886,
    'CustomerID': 'HANAR',
    'OrderDate': '1998-02-13T00:00:00.000Z',
    'ShippedDate': '1998-03-02T00:00:00.000Z',
    'Freight': 4.99,
    'ShipName': 'Hanari Carnes',

```

```

      'ShipAddress': 'Rua do Paço, 67',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10887,
      'CustomerID': 'GALED',
      'OrderDate': '1998-02-13T00:00:00.000Z',
      'ShippedDate': '1998-02-16T00:00:00.000Z',
      'Freight': 1.25,
      'ShipName': 'Galería del gastronómo',
      'ShipAddress': 'Rambla de Cataluña, 23',
      'ShipCity': 'Barcelona',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    },
    {
      'OrderID': 10888,
      'CustomerID': 'GODOS',
      'OrderDate': '1998-02-16T00:00:00.000Z',
      'ShippedDate': '1998-02-23T00:00:00.000Z',
      'Freight': 51.87,
      'ShipName': 'Godos Cocina Típica',
      'ShipAddress': 'C/ Romero, 33',
      'ShipCity': 'Sevilla',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    },
    {
      'OrderID': 10889,
      'CustomerID': 'RATTC',
      'OrderDate': '1998-02-16T00:00:00.000Z',
      'ShippedDate': '1998-02-23T00:00:00.000Z',
      'Freight': 280.61,
      'ShipName': 'Rattlesnake Canyon Grocery',
      'ShipAddress': '2817 Milton Dr.',
      'ShipCity': 'Albuquerque',
      'ShipRegion': 'NM',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10890,
      'CustomerID': 'DUMON',
      'OrderDate': '1998-02-16T00:00:00.000Z',
      'ShippedDate': '1998-02-18T00:00:00.000Z',
      'Freight': 32.76,
      'ShipName': 'Du monde entier',
      'ShipAddress': '67, rue des Cinquante Otages',
      'ShipCity': 'Nantes',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10891,
      'CustomerID': 'LEHMS',
      'OrderDate': '1998-02-17T00:00:00.000Z',

```

```

      'ShippedDate': '1998-02-19T00:00:00.000Z',
      'Freight': 20.37,
      'ShipName': 'Lehmanns Marktstand',
      'ShipAddress': 'Magazinweg 7',
      'ShipCity': 'Frankfurt a.M.',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10892,
      'CustomerID': 'MAISD',
      'OrderDate': '1998-02-17T00:00:00.000Z',
      'ShippedDate': '1998-02-19T00:00:00.000Z',
      'Freight': 120.27,
      'ShipName': 'Maison Dewey',
      'ShipAddress': 'Rue Joseph-Bens 532',
      'ShipCity': 'Bruxelles',
      'ShipRegion': null,
      'ShipCountry': 'Belgium'
    },
    {
      'OrderID': 10893,
      'CustomerID': 'KOENE',
      'OrderDate': '1998-02-18T00:00:00.000Z',
      'ShippedDate': '1998-02-20T00:00:00.000Z',
      'Freight': 77.78,
      'ShipName': 'Königlich Essen',
      'ShipAddress': 'Maubelstr. 90',
      'ShipCity': 'Brandenburg',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10894,
      'CustomerID': 'SAVEA',
      'OrderDate': '1998-02-18T00:00:00.000Z',
      'ShippedDate': '1998-02-20T00:00:00.000Z',
      'Freight': 116.13,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10895,
      'CustomerID': 'ERNSH',
      'OrderDate': '1998-02-18T00:00:00.000Z',
      'ShippedDate': '1998-02-23T00:00:00.000Z',
      'Freight': 162.75,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    }
  ],
  {

```

```

    'OrderID': 10896,
    'CustomerID': 'MAISD',
    'OrderDate': '1998-02-19T00:00:00.000Z',
    'ShippedDate': '1998-02-27T00:00:00.000Z',
    'Freight': 32.45,
    'ShipName': 'Maison Dewey',
    'ShipAddress': 'Rue Joseph-Bens 532',
    'ShipCity': 'Bruxelles',
    'ShipRegion': null,
    'ShipCountry': 'Belgium'
  },
  {
    'OrderID': 10897,
    'CustomerID': 'HUNGO',
    'OrderDate': '1998-02-19T00:00:00.000Z',
    'ShippedDate': '1998-02-25T00:00:00.000Z',
    'Freight': 603.54,
    'ShipName': 'Hungry Owl All-Night Grocers',
    'ShipAddress': '8 Johnstown Road',
    'ShipCity': 'Cork',
    'ShipRegion': 'Co. Cork',
    'ShipCountry': 'Ireland'
  },
  {
    'OrderID': 10898,
    'CustomerID': 'OCEAN',
    'OrderDate': '1998-02-20T00:00:00.000Z',
    'ShippedDate': '1998-03-06T00:00:00.000Z',
    'Freight': 1.27,
    'ShipName': 'Océano Atlántico Ltda.',
    'ShipAddress': 'Ing. Gustavo Moncada 8585 Piso 20-A',
    'ShipCity': 'Buenos Aires',
    'ShipRegion': null,
    'ShipCountry': 'Argentina'
  },
  {
    'OrderID': 10899,
    'CustomerID': 'LILAS',
    'OrderDate': '1998-02-20T00:00:00.000Z',
    'ShippedDate': '1998-02-26T00:00:00.000Z',
    'Freight': 1.21,
    'ShipName': 'LILA-Supermercado',
    'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
    'ShipCity': 'Barquisimeto',
    'ShipRegion': 'Lara',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10900,
    'CustomerID': 'WELLI',
    'OrderDate': '1998-02-20T00:00:00.000Z',
    'ShippedDate': '1998-03-04T00:00:00.000Z',
    'Freight': 1.66,
    'ShipName': 'Wellington Importadora',
    'ShipAddress': 'Rua do Mercado, 12',
    'ShipCity': 'Resende',
    'ShipRegion': 'SP',
  }

```

```

    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10901,
    'CustomerID': 'HILAA',
    'OrderDate': '1998-02-23T00:00:00.000Z',
    'ShippedDate': '1998-02-26T00:00:00.000Z',
    'Freight': 62.09,
    'ShipName': 'HILARION-Abastos',
    'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
    'ShipCity': 'San Cristóbal',
    'ShipRegion': 'Táchira',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10902,
    'CustomerID': 'FOLKO',
    'OrderDate': '1998-02-23T00:00:00.000Z',
    'ShippedDate': '1998-03-03T00:00:00.000Z',
    'Freight': 44.15,
    'ShipName': 'Folk och fä HB',
    'ShipAddress': 'Åkergatan 24',
    'ShipCity': 'Bräcke',
    'ShipRegion': null,
    'ShipCountry': 'Sweden'
  },
  {
    'OrderID': 10903,
    'CustomerID': 'HANAR',
    'OrderDate': '1998-02-24T00:00:00.000Z',
    'ShippedDate': '1998-03-04T00:00:00.000Z',
    'Freight': 36.71,
    'ShipName': 'Hanari Carnes',
    'ShipAddress': 'Rua do Paço, 67',
    'ShipCity': 'Rio de Janeiro',
    'ShipRegion': 'RJ',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10904,
    'CustomerID': 'WHITC',
    'OrderDate': '1998-02-24T00:00:00.000Z',
    'ShippedDate': '1998-02-27T00:00:00.000Z',
    'Freight': 162.95,
    'ShipName': 'White Clover Markets',
    'ShipAddress': '1029 - 12th Ave. S.',
    'ShipCity': 'Seattle',
    'ShipRegion': 'WA',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10905,
    'CustomerID': 'WELLI',
    'OrderDate': '1998-02-24T00:00:00.000Z',
    'ShippedDate': '1998-03-06T00:00:00.000Z',
    'Freight': 13.72,
    'ShipName': 'Wellington Importadora',

```

```

      'ShipAddress': 'Rua do Mercado, 12',
      'ShipCity': 'Resende',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10906,
      'CustomerID': 'WOLZA',
      'OrderDate': '1998-02-25T00:00:00.000Z',
      'ShippedDate': '1998-03-03T00:00:00.000Z',
      'Freight': 26.29,
      'ShipName': 'Wolski Zajazd',
      'ShipAddress': 'ul. Filtrowa 68',
      'ShipCity': 'Warszawa',
      'ShipRegion': null,
      'ShipCountry': 'Poland'
    },
    {
      'OrderID': 10907,
      'CustomerID': 'SPECB',
      'OrderDate': '1998-02-25T00:00:00.000Z',
      'ShippedDate': '1998-02-27T00:00:00.000Z',
      'Freight': 9.19,
      'ShipName': 'Spécialités du monde',
      'ShipAddress': '25, rue Lauriston',
      'ShipCity': 'Paris',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10908,
      'CustomerID': 'REGGC',
      'OrderDate': '1998-02-26T00:00:00.000Z',
      'ShippedDate': '1998-03-06T00:00:00.000Z',
      'Freight': 32.96,
      'ShipName': 'Reggiani Caseifici',
      'ShipAddress': 'Strada Provinciale 124',
      'ShipCity': 'Reggio Emilia',
      'ShipRegion': null,
      'ShipCountry': 'Italy'
    },
    {
      'OrderID': 10909,
      'CustomerID': 'SANTG',
      'OrderDate': '1998-02-26T00:00:00.000Z',
      'ShippedDate': '1998-03-10T00:00:00.000Z',
      'Freight': 53.05,
      'ShipName': 'Santé Gourmet',
      'ShipAddress': 'Erling Skakkes gate 78',
      'ShipCity': 'Stavern',
      'ShipRegion': null,
      'ShipCountry': 'Norway'
    },
    {
      'OrderID': 10910,
      'CustomerID': 'WILMK',
      'OrderDate': '1998-02-26T00:00:00.000Z',

```

```

    'ShippedDate': '1998-03-04T00:00:00.000Z',
    'Freight': 38.11,
    'ShipName': 'Wilman Kala',
    'ShipAddress': 'Keskuskatu 45',
    'ShipCity': 'Helsinki',
    'ShipRegion': null,
    'ShipCountry': 'Finland'
  },
  {
    'OrderID': 10911,
    'CustomerID': 'GODOS',
    'OrderDate': '1998-02-26T00:00:00.000Z',
    'ShippedDate': '1998-03-05T00:00:00.000Z',
    'Freight': 38.19,
    'ShipName': 'Godos Cocina Típica',
    'ShipAddress': 'C/ Romero, 33',
    'ShipCity': 'Sevilla',
    'ShipRegion': null,
    'ShipCountry': 'Spain'
  },
  {
    'OrderID': 10912,
    'CustomerID': 'HUNGO',
    'OrderDate': '1998-02-26T00:00:00.000Z',
    'ShippedDate': '1998-03-18T00:00:00.000Z',
    'Freight': 580.91,
    'ShipName': 'Hungry Owl All-Night Grocers',
    'ShipAddress': '8 Johnstown Road',
    'ShipCity': 'Cork',
    'ShipRegion': 'Co. Cork',
    'ShipCountry': 'Ireland'
  },
  {
    'OrderID': 10913,
    'CustomerID': 'QUEEN',
    'OrderDate': '1998-02-26T00:00:00.000Z',
    'ShippedDate': '1998-03-04T00:00:00.000Z',
    'Freight': 33.05,
    'ShipName': 'Queen Cozinha',
    'ShipAddress': 'Alameda dos Canários, 891',
    'ShipCity': 'Sao Paulo',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10914,
    'CustomerID': 'QUEEN',
    'OrderDate': '1998-02-27T00:00:00.000Z',
    'ShippedDate': '1998-03-02T00:00:00.000Z',
    'Freight': 21.19,
    'ShipName': 'Queen Cozinha',
    'ShipAddress': 'Alameda dos Canários, 891',
    'ShipCity': 'Sao Paulo',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  }
]

```



```

      'OrderID': 10915,
      'CustomerID': 'TORTU',
      'OrderDate': '1998-02-27T00:00:00.000Z',
      'ShippedDate': '1998-03-02T00:00:00.000Z',
      'Freight': 3.51,
      'ShipName': 'Tortuga Restaurante',
      'ShipAddress': 'Avda. Azteca 123',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10916,
      'CustomerID': 'RANCH',
      'OrderDate': '1998-02-27T00:00:00.000Z',
      'ShippedDate': '1998-03-09T00:00:00.000Z',
      'Freight': 63.77,
      'ShipName': 'Rancho grande',
      'ShipAddress': 'Av. del Libertador 900',
      'ShipCity': 'Buenos Aires',
      'ShipRegion': null,
      'ShipCountry': 'Argentina'
    },
    {
      'OrderID': 10917,
      'CustomerID': 'ROMEY',
      'OrderDate': '1998-03-02T00:00:00.000Z',
      'ShippedDate': '1998-03-11T00:00:00.000Z',
      'Freight': 8.29,
      'ShipName': 'Romero y tomillo',
      'ShipAddress': 'Gran Vía, 1',
      'ShipCity': 'Madrid',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    },
    {
      'OrderID': 10918,
      'CustomerID': 'BOTTM',
      'OrderDate': '1998-03-02T00:00:00.000Z',
      'ShippedDate': '1998-03-11T00:00:00.000Z',
      'Freight': 48.83,
      'ShipName': 'Bottom-Dollar Markets',
      'ShipAddress': '23 Tsawassen Blvd.',
      'ShipCity': 'Tsawassen',
      'ShipRegion': 'BC',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10919,
      'CustomerID': 'LINOD',
      'OrderDate': '1998-03-02T00:00:00.000Z',
      'ShippedDate': '1998-03-04T00:00:00.000Z',
      'Freight': 19.8,
      'ShipName': 'LINO-Delicateses',
      'ShipAddress': 'Ave. 5 de Mayo Porlamar',
      'ShipCity': 'I. de Margarita',
      'ShipRegion': 'Nueva Esparta',

```

```

    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10920,
    'CustomerID': 'AROUT',
    'OrderDate': '1998-03-03T00:00:00.000Z',
    'ShippedDate': '1998-03-09T00:00:00.000Z',
    'Freight': 29.61,
    'ShipName': 'Around the Horn',
    'ShipAddress': 'Brook Farm Stratford St. Mary',
    'ShipCity': 'Colchester',
    'ShipRegion': 'Essex',
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10921,
    'CustomerID': 'VAFFE',
    'OrderDate': '1998-03-03T00:00:00.000Z',
    'ShippedDate': '1998-03-09T00:00:00.000Z',
    'Freight': 176.48,
    'ShipName': 'Vaffeljernet',
    'ShipAddress': 'Smagsloget 45',
    'ShipCity': 'Århus',
    'ShipRegion': null,
    'ShipCountry': 'Denmark'
  },
  {
    'OrderID': 10922,
    'CustomerID': 'HANAR',
    'OrderDate': '1998-03-03T00:00:00.000Z',
    'ShippedDate': '1998-03-05T00:00:00.000Z',
    'Freight': 62.74,
    'ShipName': 'Hanari Carnes',
    'ShipAddress': 'Rua do Paço, 67',
    'ShipCity': 'Rio de Janeiro',
    'ShipRegion': 'RJ',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10923,
    'CustomerID': 'LAMAI',
    'OrderDate': '1998-03-03T00:00:00.000Z',
    'ShippedDate': '1998-03-13T00:00:00.000Z',
    'Freight': 68.26,
    'ShipName': 'La maison d\'Asie',
    'ShipAddress': '1 rue Alsace-Lorraine',
    'ShipCity': 'Toulouse',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10924,
    'CustomerID': 'BERGS',
    'OrderDate': '1998-03-04T00:00:00.000Z',
    'ShippedDate': '1998-04-08T00:00:00.000Z',
    'Freight': 151.52,
    'ShipName': 'Berglunds snabbköp',

```

```

      'ShipAddress': 'Berguvsvägen 8',
      'ShipCity': 'Luleå',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 10925,
      'CustomerID': 'HANAR',
      'OrderDate': '1998-03-04T00:00:00.000Z',
      'ShippedDate': '1998-03-13T00:00:00.000Z',
      'Freight': 2.27,
      'ShipName': 'Hanari Carnes',
      'ShipAddress': 'Rua do Paço, 67',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10926,
      'CustomerID': 'ANATR',
      'OrderDate': '1998-03-04T00:00:00.000Z',
      'ShippedDate': '1998-03-11T00:00:00.000Z',
      'Freight': 39.92,
      'ShipName': 'Ana Trujillo Emparedados y helados',
      'ShipAddress': 'Avda. de la Constitución 2222',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10927,
      'CustomerID': 'LACOR',
      'OrderDate': '1998-03-05T00:00:00.000Z',
      'ShippedDate': '1998-04-08T00:00:00.000Z',
      'Freight': 19.79,
      'ShipName': 'La corne d\'abondance',
      'ShipAddress': '67, avenue de l\'Europe',
      'ShipCity': 'Versailles',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10928,
      'CustomerID': 'GALED',
      'OrderDate': '1998-03-05T00:00:00.000Z',
      'ShippedDate': '1998-03-18T00:00:00.000Z',
      'Freight': 1.36,
      'ShipName': 'Galería del gastronómo',
      'ShipAddress': 'Rambla de Cataluña, 23',
      'ShipCity': 'Barcelona',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    },
    {
      'OrderID': 10929,
      'CustomerID': 'FRANK',
      'OrderDate': '1998-03-05T00:00:00.000Z',

```

```

    'ShippedDate': '1998-03-12T00:00:00.000Z',
    'Freight': 33.93,
    'ShipName': 'Frankenversand',
    'ShipAddress': 'Berliner Platz 43',
    'ShipCity': 'München',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10930,
    'CustomerID': 'SUPRD',
    'OrderDate': '1998-03-06T00:00:00.000Z',
    'ShippedDate': '1998-03-18T00:00:00.000Z',
    'Freight': 15.55,
    'ShipName': 'Suprêmes délices',
    'ShipAddress': 'Boulevard Tirou, 255',
    'ShipCity': 'Charleroi',
    'ShipRegion': null,
    'ShipCountry': 'Belgium'
  },
  {
    'OrderID': 10931,
    'CustomerID': 'RICSU',
    'OrderDate': '1998-03-06T00:00:00.000Z',
    'ShippedDate': '1998-03-19T00:00:00.000Z',
    'Freight': 13.6,
    'ShipName': 'Richter Supermarkt',
    'ShipAddress': 'Starenweg 5',
    'ShipCity': 'Genève',
    'ShipRegion': null,
    'ShipCountry': 'Switzerland'
  },
  {
    'OrderID': 10932,
    'CustomerID': 'BONAP',
    'OrderDate': '1998-03-06T00:00:00.000Z',
    'ShippedDate': '1998-03-24T00:00:00.000Z',
    'Freight': 134.64,
    'ShipName': 'Bon app',
    'ShipAddress': '12, rue des Bouchers',
    'ShipCity': 'Marseille',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10933,
    'CustomerID': 'ISLAT',
    'OrderDate': '1998-03-06T00:00:00.000Z',
    'ShippedDate': '1998-03-16T00:00:00.000Z',
    'Freight': 54.15,
    'ShipName': 'Island Trading',
    'ShipAddress': 'Garden House Crowther Way',
    'ShipCity': 'Cowes',
    'ShipRegion': 'Isle of Wight',
    'ShipCountry': 'UK'
  },
  {

```

```

      'OrderID': 10934,
      'CustomerID': 'LEHMS',
      'OrderDate': '1998-03-09T00:00:00.000Z',
      'ShippedDate': '1998-03-12T00:00:00.000Z',
      'Freight': 32.01,
      'ShipName': 'Lehmanns Marktstand',
      'ShipAddress': 'Magazinweg 7',
      'ShipCity': 'Frankfurt a.M.',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10935,
      'CustomerID': 'WELLI',
      'OrderDate': '1998-03-09T00:00:00.000Z',
      'ShippedDate': '1998-03-18T00:00:00.000Z',
      'Freight': 47.59,
      'ShipName': 'Wellington Importadora',
      'ShipAddress': 'Rua do Mercado, 12',
      'ShipCity': 'Resende',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10936,
      'CustomerID': 'GREAL',
      'OrderDate': '1998-03-09T00:00:00.000Z',
      'ShippedDate': '1998-03-18T00:00:00.000Z',
      'Freight': 33.68,
      'ShipName': 'Great Lakes Food Market',
      'ShipAddress': '2732 Baker Blvd.',
      'ShipCity': 'Eugene',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10937,
      'CustomerID': 'CACTU',
      'OrderDate': '1998-03-10T00:00:00.000Z',
      'ShippedDate': '1998-03-13T00:00:00.000Z',
      'Freight': 31.51,
      'ShipName': 'Cactus Comidas para llevar',
      'ShipAddress': 'Cerrito 333',
      'ShipCity': 'Buenos Aires',
      'ShipRegion': null,
      'ShipCountry': 'Argentina'
    },
    {
      'OrderID': 10938,
      'CustomerID': 'QUICK',
      'OrderDate': '1998-03-10T00:00:00.000Z',
      'ShippedDate': '1998-03-16T00:00:00.000Z',
      'Freight': 31.89,
      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,

```

```

    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10939,
    'CustomerID': 'MAGAA',
    'OrderDate': '1998-03-10T00:00:00.000Z',
    'ShippedDate': '1998-03-13T00:00:00.000Z',
    'Freight': 76.33,
    'ShipName': 'Magazzini Alimentari Riuniti',
    'ShipAddress': 'Via Ludovico il Moro 22',
    'ShipCity': 'Bergamo',
    'ShipRegion': null,
    'ShipCountry': 'Italy'
  },
  {
    'OrderID': 10940,
    'CustomerID': 'BONAP',
    'OrderDate': '1998-03-11T00:00:00.000Z',
    'ShippedDate': '1998-03-23T00:00:00.000Z',
    'Freight': 19.77,
    'ShipName': 'Bon app',
    'ShipAddress': '12, rue des Bouchers',
    'ShipCity': 'Marseille',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10941,
    'CustomerID': 'SAVEA',
    'OrderDate': '1998-03-11T00:00:00.000Z',
    'ShippedDate': '1998-03-20T00:00:00.000Z',
    'Freight': 400.81,
    'ShipName': 'Save-a-lot Markets',
    'ShipAddress': '187 Suffolk Ln.',
    'ShipCity': 'Boise',
    'ShipRegion': 'ID',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10942,
    'CustomerID': 'REGGC',
    'OrderDate': '1998-03-11T00:00:00.000Z',
    'ShippedDate': '1998-03-18T00:00:00.000Z',
    'Freight': 17.95,
    'ShipName': 'Reggiani Caseifici',
    'ShipAddress': 'Strada Provinciale 124',
    'ShipCity': 'Reggio Emilia',
    'ShipRegion': null,
    'ShipCountry': 'Italy'
  },
  {
    'OrderID': 10943,
    'CustomerID': 'BSBEV',
    'OrderDate': '1998-03-11T00:00:00.000Z',
    'ShippedDate': '1998-03-19T00:00:00.000Z',
    'Freight': 2.17,
    'ShipName': 'B\ Beverages',

```

```

      'ShipAddress': 'Fauntleroy Circus',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10944,
      'CustomerID': 'BOTTM',
      'OrderDate': '1998-03-12T00:00:00.000Z',
      'ShippedDate': '1998-03-13T00:00:00.000Z',
      'Freight': 52.92,
      'ShipName': 'Bottom-Dollar Markets',
      'ShipAddress': '23 Tsawassen Blvd.',
      'ShipCity': 'Tsawassen',
      'ShipRegion': 'BC',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10945,
      'CustomerID': 'MORGK',
      'OrderDate': '1998-03-12T00:00:00.000Z',
      'ShippedDate': '1998-03-18T00:00:00.000Z',
      'Freight': 10.22,
      'ShipName': 'Morgenstern Gesundkost',
      'ShipAddress': 'Heerstr. 22',
      'ShipCity': 'Leipzig',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10946,
      'CustomerID': 'VAFFE',
      'OrderDate': '1998-03-12T00:00:00.000Z',
      'ShippedDate': '1998-03-19T00:00:00.000Z',
      'Freight': 27.2,
      'ShipName': 'Vaffeljernet',
      'ShipAddress': 'Smagsloget 45',
      'ShipCity': 'Århus',
      'ShipRegion': null,
      'ShipCountry': 'Denmark'
    },
    {
      'OrderID': 10947,
      'CustomerID': 'BSBEV',
      'OrderDate': '1998-03-13T00:00:00.000Z',
      'ShippedDate': '1998-03-16T00:00:00.000Z',
      'Freight': 3.26,
      'ShipName': 'B\ Beverages',
      'ShipAddress': 'Fauntleroy Circus',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10948,
      'CustomerID': 'GODOS',
      'OrderDate': '1998-03-13T00:00:00.000Z',

```

```

    'ShippedDate': '1998-03-19T00:00:00.000Z',
    'Freight': 23.39,
    'ShipName': 'Godos Cocina Típica',
    'ShipAddress': 'C/ Romero, 33',
    'ShipCity': 'Sevilla',
    'ShipRegion': null,
    'ShipCountry': 'Spain'
  },
  {
    'OrderID': 10949,
    'CustomerID': 'BOTTM',
    'OrderDate': '1998-03-13T00:00:00.000Z',
    'ShippedDate': '1998-03-17T00:00:00.000Z',
    'Freight': 74.44,
    'ShipName': 'Bottom-Dollar Markets',
    'ShipAddress': '23 Tsawassen Blvd.',
    'ShipCity': 'Tsawassen',
    'ShipRegion': 'BC',
    'ShipCountry': 'Canada'
  },
  {
    'OrderID': 10950,
    'CustomerID': 'MAGAA',
    'OrderDate': '1998-03-16T00:00:00.000Z',
    'ShippedDate': '1998-03-23T00:00:00.000Z',
    'Freight': 2.5,
    'ShipName': 'Magazzini Alimentari Riuniti',
    'ShipAddress': 'Via Ludovico il Moro 22',
    'ShipCity': 'Bergamo',
    'ShipRegion': null,
    'ShipCountry': 'Italy'
  },
  {
    'OrderID': 10951,
    'CustomerID': 'RICSU',
    'OrderDate': '1998-03-16T00:00:00.000Z',
    'ShippedDate': '1998-04-07T00:00:00.000Z',
    'Freight': 30.85,
    'ShipName': 'Richter Supermarkt',
    'ShipAddress': 'Starenweg 5',
    'ShipCity': 'Genève',
    'ShipRegion': null,
    'ShipCountry': 'Switzerland'
  },
  {
    'OrderID': 10952,
    'CustomerID': 'ALFKI',
    'OrderDate': '1998-03-16T00:00:00.000Z',
    'ShippedDate': '1998-03-24T00:00:00.000Z',
    'Freight': 40.42,
    'ShipName': 'Alfred Futterkiste',
    'ShipAddress': 'Obere Str. 57',
    'ShipCity': 'Berlin',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  }
]

```



```

    'OrderID': 10953,
    'CustomerID': 'AROUT',
    'OrderDate': '1998-03-16T00:00:00.000Z',
    'ShippedDate': '1998-03-25T00:00:00.000Z',
    'Freight': 23.72,
    'ShipName': 'Around the Horn',
    'ShipAddress': 'Brook Farm Stratford St. Mary',
    'ShipCity': 'Colchester',
    'ShipRegion': 'Essex',
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10954,
    'CustomerID': 'LINOD',
    'OrderDate': '1998-03-17T00:00:00.000Z',
    'ShippedDate': '1998-03-20T00:00:00.000Z',
    'Freight': 27.91,
    'ShipName': 'LINO-Delicateses',
    'ShipAddress': 'Ave. 5 de Mayo Porlamar',
    'ShipCity': 'I. de Margarita',
    'ShipRegion': 'Nueva Esparta',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10955,
    'CustomerID': 'FOLKO',
    'OrderDate': '1998-03-17T00:00:00.000Z',
    'ShippedDate': '1998-03-20T00:00:00.000Z',
    'Freight': 3.26,
    'ShipName': 'Folk och fä HB',
    'ShipAddress': 'Åkergatan 24',
    'ShipCity': 'Bräcke',
    'ShipRegion': null,
    'ShipCountry': 'Sweden'
  },
  {
    'OrderID': 10956,
    'CustomerID': 'BLAUS',
    'OrderDate': '1998-03-17T00:00:00.000Z',
    'ShippedDate': '1998-03-20T00:00:00.000Z',
    'Freight': 44.65,
    'ShipName': 'Blauer See Delikatessen',
    'ShipAddress': 'Forsterstr. 57',
    'ShipCity': 'Mannheim',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10957,
    'CustomerID': 'HILAA',
    'OrderDate': '1998-03-18T00:00:00.000Z',
    'ShippedDate': '1998-03-27T00:00:00.000Z',
    'Freight': 105.36,
    'ShipName': 'HILARION-Abastos',
    'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
    'ShipCity': 'San Cristóbal',
    'ShipRegion': 'Táchira',

```

```

    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10958,
    'CustomerID': 'OCEAN',
    'OrderDate': '1998-03-18T00:00:00.000Z',
    'ShippedDate': '1998-03-27T00:00:00.000Z',
    'Freight': 49.56,
    'ShipName': 'Océano Atlántico Ltda.',
    'ShipAddress': 'Ing. Gustavo Moncada 8585 Piso 20-A',
    'ShipCity': 'Buenos Aires',
    'ShipRegion': null,
    'ShipCountry': 'Argentina'
  },
  {
    'OrderID': 10959,
    'CustomerID': 'GOURL',
    'OrderDate': '1998-03-18T00:00:00.000Z',
    'ShippedDate': '1998-03-23T00:00:00.000Z',
    'Freight': 4.98,
    'ShipName': 'Gourmet Lanchonetes',
    'ShipAddress': 'Av. Brasil, 442',
    'ShipCity': 'Campinas',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10960,
    'CustomerID': 'HILAA',
    'OrderDate': '1998-03-19T00:00:00.000Z',
    'ShippedDate': '1998-04-08T00:00:00.000Z',
    'Freight': 2.08,
    'ShipName': 'HILARION-Abastos',
    'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
    'ShipCity': 'San Cristóbal',
    'ShipRegion': 'Táchira',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10961,
    'CustomerID': 'QUEEN',
    'OrderDate': '1998-03-19T00:00:00.000Z',
    'ShippedDate': '1998-03-30T00:00:00.000Z',
    'Freight': 104.47,
    'ShipName': 'Queen Cozinha',
    'ShipAddress': 'Alameda dos Canários, 891',
    'ShipCity': 'Sao Paulo',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10962,
    'CustomerID': 'QUICK',
    'OrderDate': '1998-03-19T00:00:00.000Z',
    'ShippedDate': '1998-03-23T00:00:00.000Z',
    'Freight': 275.79,
    'ShipName': 'QUICK-Stop',

```

```

    'ShipAddress': 'Taucherstraße 10',
    'ShipCity': 'Cunewalde',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10963,
    'CustomerID': 'FURIB',
    'OrderDate': '1998-03-19T00:00:00.000Z',
    'ShippedDate': '1998-03-26T00:00:00.000Z',
    'Freight': 2.7,
    'ShipName': 'Furia Bacalhau e Frutos do Mar',
    'ShipAddress': 'Jardim das rosas n. 32',
    'ShipCity': 'Lisboa',
    'ShipRegion': null,
    'ShipCountry': 'Portugal'
  },
  {
    'OrderID': 10964,
    'CustomerID': 'SPECB',
    'OrderDate': '1998-03-20T00:00:00.000Z',
    'ShippedDate': '1998-03-24T00:00:00.000Z',
    'Freight': 87.38,
    'ShipName': 'Spécialités du monde',
    'ShipAddress': '25, rue Lauriston',
    'ShipCity': 'Paris',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10965,
    'CustomerID': 'OLDWO',
    'OrderDate': '1998-03-20T00:00:00.000Z',
    'ShippedDate': '1998-03-30T00:00:00.000Z',
    'Freight': 144.38,
    'ShipName': 'Old World Delicatessen',
    'ShipAddress': '2743 Bering St.',
    'ShipCity': 'Anchorage',
    'ShipRegion': 'AK',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10966,
    'CustomerID': 'CHOPS',
    'OrderDate': '1998-03-20T00:00:00.000Z',
    'ShippedDate': '1998-04-08T00:00:00.000Z',
    'Freight': 27.19,
    'ShipName': 'Chop-suey Chinese',
    'ShipAddress': 'Hauptstr. 31',
    'ShipCity': 'Bern',
    'ShipRegion': null,
    'ShipCountry': 'Switzerland'
  },
  {
    'OrderID': 10967,
    'CustomerID': 'TOMSP',
    'OrderDate': '1998-03-23T00:00:00.000Z',

```

```

      'ShippedDate': '1998-04-02T00:00:00.000Z',
      'Freight': 62.22,
      'ShipName': 'Toms Spezialitäten',
      'ShipAddress': 'Luisenstr. 48',
      'ShipCity': 'Münster',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10968,
      'CustomerID': 'ERNSH',
      'OrderDate': '1998-03-23T00:00:00.000Z',
      'ShippedDate': '1998-04-01T00:00:00.000Z',
      'Freight': 74.6,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10969,
      'CustomerID': 'COMMI',
      'OrderDate': '1998-03-23T00:00:00.000Z',
      'ShippedDate': '1998-03-30T00:00:00.000Z',
      'Freight': 0.21,
      'ShipName': 'Comércio Mineiro',
      'ShipAddress': 'Av. dos Lusíadas, 23',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10970,
      'CustomerID': 'BOLID',
      'OrderDate': '1998-03-24T00:00:00.000Z',
      'ShippedDate': '1998-04-24T00:00:00.000Z',
      'Freight': 16.16,
      'ShipName': 'Bólido Comidas preparadas',
      'ShipAddress': 'C/ Araquil, 67',
      'ShipCity': 'Madrid',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    },
    {
      'OrderID': 10971,
      'CustomerID': 'FRANR',
      'OrderDate': '1998-03-24T00:00:00.000Z',
      'ShippedDate': '1998-04-02T00:00:00.000Z',
      'Freight': 121.82,
      'ShipName': 'France restauration',
      'ShipAddress': '54, rue Royale',
      'ShipCity': 'Nantes',
      'ShipRegion': null,
      'ShipCountry': 'France'
    }
  ],
  {

```

```

    'OrderID': 10972,
    'CustomerID': 'LACOR',
    'OrderDate': '1998-03-24T00:00:00.000Z',
    'ShippedDate': '1998-03-26T00:00:00.000Z',
    'Freight': 0.02,
    'ShipName': 'La corne d\'abondance',
    'ShipAddress': '67, avenue de l\'Europe',
    'ShipCity': 'Versailles',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10973,
    'CustomerID': 'LACOR',
    'OrderDate': '1998-03-24T00:00:00.000Z',
    'ShippedDate': '1998-03-27T00:00:00.000Z',
    'Freight': 15.17,
    'ShipName': 'La corne d\'abondance',
    'ShipAddress': '67, avenue de l\'Europe',
    'ShipCity': 'Versailles',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10974,
    'CustomerID': 'SPLIR',
    'OrderDate': '1998-03-25T00:00:00.000Z',
    'ShippedDate': '1998-04-03T00:00:00.000Z',
    'Freight': 12.96,
    'ShipName': 'Split Rail Beer & Ale',
    'ShipAddress': 'P.O. Box 555',
    'ShipCity': 'Lander',
    'ShipRegion': 'WY',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10975,
    'CustomerID': 'BOTTM',
    'OrderDate': '1998-03-25T00:00:00.000Z',
    'ShippedDate': '1998-03-27T00:00:00.000Z',
    'Freight': 32.27,
    'ShipName': 'Bottom-Dollar Markets',
    'ShipAddress': '23 Tsawassen Blvd.',
    'ShipCity': 'Tsawassen',
    'ShipRegion': 'BC',
    'ShipCountry': 'Canada'
  },
  {
    'OrderID': 10976,
    'CustomerID': 'HILAA',
    'OrderDate': '1998-03-25T00:00:00.000Z',
    'ShippedDate': '1998-04-03T00:00:00.000Z',
    'Freight': 37.97,
    'ShipName': 'HILARION-Abastos',
    'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
    'ShipCity': 'San Cristóbal',
    'ShipRegion': 'Táchira',

```

```

      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10977,
      'CustomerID': 'FOLKO',
      'OrderDate': '1998-03-26T00:00:00.000Z',
      'ShippedDate': '1998-04-10T00:00:00.000Z',
      'Freight': 208.5,
      'ShipName': 'Folk och fä HB',
      'ShipAddress': 'Åkergatan 24',
      'ShipCity': 'Bräcke',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 10978,
      'CustomerID': 'MAISD',
      'OrderDate': '1998-03-26T00:00:00.000Z',
      'ShippedDate': '1998-04-23T00:00:00.000Z',
      'Freight': 32.82,
      'ShipName': 'Maison Dewey',
      'ShipAddress': 'Rue Joseph-Bens 532',
      'ShipCity': 'Bruxelles',
      'ShipRegion': null,
      'ShipCountry': 'Belgium'
    },
    {
      'OrderID': 10979,
      'CustomerID': 'ERNSH',
      'OrderDate': '1998-03-26T00:00:00.000Z',
      'ShippedDate': '1998-03-31T00:00:00.000Z',
      'Freight': 353.07,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10980,
      'CustomerID': 'FOLKO',
      'OrderDate': '1998-03-27T00:00:00.000Z',
      'ShippedDate': '1998-04-17T00:00:00.000Z',
      'Freight': 1.26,
      'ShipName': 'Folk och fä HB',
      'ShipAddress': 'Åkergatan 24',
      'ShipCity': 'Bräcke',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 10981,
      'CustomerID': 'HANAR',
      'OrderDate': '1998-03-27T00:00:00.000Z',
      'ShippedDate': '1998-04-02T00:00:00.000Z',
      'Freight': 193.37,
      'ShipName': 'Hanari Carnes',

```

```

      'ShipAddress': 'Rua do Paço, 67',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10982,
      'CustomerID': 'BOTTM',
      'OrderDate': '1998-03-27T00:00:00.000Z',
      'ShippedDate': '1998-04-08T00:00:00.000Z',
      'Freight': 14.01,
      'ShipName': 'Bottom-Dollar Markets',
      'ShipAddress': '23 Tsawassen Blvd.',
      'ShipCity': 'Tsawassen',
      'ShipRegion': 'BC',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10983,
      'CustomerID': 'SAVEA',
      'OrderDate': '1998-03-27T00:00:00.000Z',
      'ShippedDate': '1998-04-06T00:00:00.000Z',
      'Freight': 657.54,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10984,
      'CustomerID': 'SAVEA',
      'OrderDate': '1998-03-30T00:00:00.000Z',
      'ShippedDate': '1998-04-03T00:00:00.000Z',
      'Freight': 211.22,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10985,
      'CustomerID': 'HUNGO',
      'OrderDate': '1998-03-30T00:00:00.000Z',
      'ShippedDate': '1998-04-02T00:00:00.000Z',
      'Freight': 91.51,
      'ShipName': 'Hungry Owl All-Night Grocers',
      'ShipAddress': '8 Johnstown Road',
      'ShipCity': 'Cork',
      'ShipRegion': 'Co. Cork',
      'ShipCountry': 'Ireland'
    },
    {
      'OrderID': 10986,
      'CustomerID': 'OCEAN',
      'OrderDate': '1998-03-30T00:00:00.000Z',

```

```

      'ShippedDate': '1998-04-21T00:00:00.000Z',
      'Freight': 217.86,
      'ShipName': 'Océano Atlántico Ltda.',
      'ShipAddress': 'Ing. Gustavo Moncada 8585 Piso 20-A',
      'ShipCity': 'Buenos Aires',
      'ShipRegion': null,
      'ShipCountry': 'Argentina'
    },
    {
      'OrderID': 10987,
      'CustomerID': 'EASTC',
      'OrderDate': '1998-03-31T00:00:00.000Z',
      'ShippedDate': '1998-04-06T00:00:00.000Z',
      'Freight': 185.48,
      'ShipName': 'Eastern Connection',
      'ShipAddress': '35 King George',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10988,
      'CustomerID': 'RATTC',
      'OrderDate': '1998-03-31T00:00:00.000Z',
      'ShippedDate': '1998-04-10T00:00:00.000Z',
      'Freight': 61.14,
      'ShipName': 'Rattlesnake Canyon Grocery',
      'ShipAddress': '2817 Milton Dr.',
      'ShipCity': 'Albuquerque',
      'ShipRegion': 'NM',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10989,
      'CustomerID': 'QUEDE',
      'OrderDate': '1998-03-31T00:00:00.000Z',
      'ShippedDate': '1998-04-02T00:00:00.000Z',
      'Freight': 34.76,
      'ShipName': 'Que Delícia',
      'ShipAddress': 'Rua da Panificadora, 12',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10990,
      'CustomerID': 'ERNSH',
      'OrderDate': '1998-04-01T00:00:00.000Z',
      'ShippedDate': '1998-04-07T00:00:00.000Z',
      'Freight': 117.61,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    }
  ],
  {

```



```

      'OrderID': 10991,
      'CustomerID': 'QUICK',
      'OrderDate': '1998-04-01T00:00:00.000Z',
      'ShippedDate': '1998-04-07T00:00:00.000Z',
      'Freight': 38.51,
      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10992,
      'CustomerID': 'THEBI',
      'OrderDate': '1998-04-01T00:00:00.000Z',
      'ShippedDate': '1998-04-03T00:00:00.000Z',
      'Freight': 4.27,
      'ShipName': 'The Big Cheese',
      'ShipAddress': '89 Jefferson Way Suite 2',
      'ShipCity': 'Portland',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10993,
      'CustomerID': 'FOLKO',
      'OrderDate': '1998-04-01T00:00:00.000Z',
      'ShippedDate': '1998-04-10T00:00:00.000Z',
      'Freight': 8.81,
      'ShipName': 'Folk och fä HB',
      'ShipAddress': 'Åkergatan 24',
      'ShipCity': 'Bräcke',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 10994,
      'CustomerID': 'VAFFE',
      'OrderDate': '1998-04-02T00:00:00.000Z',
      'ShippedDate': '1998-04-09T00:00:00.000Z',
      'Freight': 65.53,
      'ShipName': 'Vaffeljernet',
      'ShipAddress': 'Smagsloget 45',
      'ShipCity': 'Århus',
      'ShipRegion': null,
      'ShipCountry': 'Denmark'
    },
    {
      'OrderID': 10995,
      'CustomerID': 'PERIC',
      'OrderDate': '1998-04-02T00:00:00.000Z',
      'ShippedDate': '1998-04-06T00:00:00.000Z',
      'Freight': 46,
      'ShipName': 'Pericles Comidas clásicas',
      'ShipAddress': 'Calle Dr. Jorge Cash 321',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,

```

```

    'ShipCountry': 'Mexico'
  },
  {
    'OrderID': 10996,
    'CustomerID': 'QUICK',
    'OrderDate': '1998-04-02T00:00:00.000Z',
    'ShippedDate': '1998-04-10T00:00:00.000Z',
    'Freight': 1.12,
    'ShipName': 'QUICK-Stop',
    'ShipAddress': 'Taucherstraße 10',
    'ShipCity': 'Cunewalde',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10997,
    'CustomerID': 'LILAS',
    'OrderDate': '1998-04-03T00:00:00.000Z',
    'ShippedDate': '1998-04-13T00:00:00.000Z',
    'Freight': 73.91,
    'ShipName': 'LILA-Supermercado',
    'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
    'ShipCity': 'Barquisimeto',
    'ShipRegion': 'Lara',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10998,
    'CustomerID': 'WOLZA',
    'OrderDate': '1998-04-03T00:00:00.000Z',
    'ShippedDate': '1998-04-17T00:00:00.000Z',
    'Freight': 20.31,
    'ShipName': 'Wolski Zajazd',
    'ShipAddress': 'ul. Filtrowa 68',
    'ShipCity': 'Warszawa',
    'ShipRegion': null,
    'ShipCountry': 'Poland'
  },
  {
    'OrderID': 10999,
    'CustomerID': 'OTTIK',
    'OrderDate': '1998-04-03T00:00:00.000Z',
    'ShippedDate': '1998-04-10T00:00:00.000Z',
    'Freight': 96.35,
    'ShipName': 'Ottilies Käseladen',
    'ShipAddress': 'Mehrheimerstr. 369',
    'ShipCity': 'Köln',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 11000,
    'CustomerID': 'RATTC',
    'OrderDate': '1998-04-06T00:00:00.000Z',
    'ShippedDate': '1998-04-14T00:00:00.000Z',
    'Freight': 55.12,
    'ShipName': 'Rattlesnake Canyon Grocery',

```

```

      'ShipAddress': '2817 Milton Dr.',
      'ShipCity': 'Albuquerque',
      'ShipRegion': 'NM',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 11001,
      'CustomerID': 'FOLKO',
      'OrderDate': '1998-04-06T00:00:00.000Z',
      'ShippedDate': '1998-04-14T00:00:00.000Z',
      'Freight': 197.3,
      'ShipName': 'Folk och fä HB',
      'ShipAddress': 'Åkergatan 24',
      'ShipCity': 'Bräcke',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 11002,
      'CustomerID': 'SAVEA',
      'OrderDate': '1998-04-06T00:00:00.000Z',
      'ShippedDate': '1998-04-16T00:00:00.000Z',
      'Freight': 141.16,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 11003,
      'CustomerID': 'THECR',
      'OrderDate': '1998-04-06T00:00:00.000Z',
      'ShippedDate': '1998-04-08T00:00:00.000Z',
      'Freight': 14.91,
      'ShipName': 'The Cracker Box',
      'ShipAddress': '55 Grizzly Peak Rd.',
      'ShipCity': 'Butte',
      'ShipRegion': 'MT',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 11004,
      'CustomerID': 'MAISD',
      'OrderDate': '1998-04-07T00:00:00.000Z',
      'ShippedDate': '1998-04-20T00:00:00.000Z',
      'Freight': 44.84,
      'ShipName': 'Maison Dewey',
      'ShipAddress': 'Rue Joseph-Bens 532',
      'ShipCity': 'Bruxelles',
      'ShipRegion': null,
      'ShipCountry': 'Belgium'
    },
    {
      'OrderID': 11005,
      'CustomerID': 'WILMK',
      'OrderDate': '1998-04-07T00:00:00.000Z',

```

```

      'ShippedDate': '1998-04-10T00:00:00.000Z',
      'Freight': 0.75,
      'ShipName': 'Wilman Kala',
      'ShipAddress': 'Keskuskatu 45',
      'ShipCity': 'Helsinki',
      'ShipRegion': null,
      'ShipCountry': 'Finland'
    },
    {
      'OrderID': 11006,
      'CustomerID': 'GREAL',
      'OrderDate': '1998-04-07T00:00:00.000Z',
      'ShippedDate': '1998-04-15T00:00:00.000Z',
      'Freight': 25.19,
      'ShipName': 'Great Lakes Food Market',
      'ShipAddress': '2732 Baker Blvd.',
      'ShipCity': 'Eugene',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 11007,
      'CustomerID': 'PRINI',
      'OrderDate': '1998-04-08T00:00:00.000Z',
      'ShippedDate': '1998-04-13T00:00:00.000Z',
      'Freight': 202.24,
      'ShipName': 'Princesa Isabel Vinhos',
      'ShipAddress': 'Estrada da saúde n. 58',
      'ShipCity': 'Lisboa',
      'ShipRegion': null,
      'ShipCountry': 'Portugal'
    },
    {
      'OrderID': 11008,
      'CustomerID': 'ERNSH',
      'OrderDate': '1998-04-08T00:00:00.000Z',
      'ShippedDate': null,
      'Freight': 79.46,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 11009,
      'CustomerID': 'GODOS',
      'OrderDate': '1998-04-08T00:00:00.000Z',
      'ShippedDate': '1998-04-10T00:00:00.000Z',
      'Freight': 59.11,
      'ShipName': 'Godos Cocina Típica',
      'ShipAddress': 'C/ Romero, 33',
      'ShipCity': 'Sevilla',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    }
  ],
  {

```

```

      'OrderID': 11010,
      'CustomerID': 'REGGC',
      'OrderDate': '1998-04-09T00:00:00.000Z',
      'ShippedDate': '1998-04-21T00:00:00.000Z',
      'Freight': 28.71,
      'ShipName': 'Reggiani Caseifici',
      'ShipAddress': 'Strada Provinciale 124',
      'ShipCity': 'Reggio Emilia',
      'ShipRegion': null,
      'ShipCountry': 'Italy'
    },
    {
      'OrderID': 11011,
      'CustomerID': 'ALFKI',
      'OrderDate': '1998-04-09T00:00:00.000Z',
      'ShippedDate': '1998-04-13T00:00:00.000Z',
      'Freight': 1.21,
      'ShipName': 'Alfred\' Futterkiste',
      'ShipAddress': 'Obere Str. 57',
      'ShipCity': 'Berlin',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 11012,
      'CustomerID': 'FRANK',
      'OrderDate': '1998-04-09T00:00:00.000Z',
      'ShippedDate': '1998-04-17T00:00:00.000Z',
      'Freight': 242.95,
      'ShipName': 'Frankenversand',
      'ShipAddress': 'Berliner Platz 43',
      'ShipCity': 'München',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 11013,
      'CustomerID': 'ROMEY',
      'OrderDate': '1998-04-09T00:00:00.000Z',
      'ShippedDate': '1998-04-10T00:00:00.000Z',
      'Freight': 32.99,
      'ShipName': 'Romero y tomillo',
      'ShipAddress': 'Gran Vía, 1',
      'ShipCity': 'Madrid',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    },
    {
      'OrderID': 11014,
      'CustomerID': 'LINOD',
      'OrderDate': '1998-04-10T00:00:00.000Z',
      'ShippedDate': '1998-04-15T00:00:00.000Z',
      'Freight': 23.6,
      'ShipName': 'LINO-Delicateses',
      'ShipAddress': 'Ave. 5 de Mayo Porlamar',
      'ShipCity': 'I. de Margarita',
      'ShipRegion': 'Nueva Esparta',

```

```

      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 11015,
      'CustomerID': 'SANTG',
      'OrderDate': '1998-04-10T00:00:00.000Z',
      'ShippedDate': '1998-04-20T00:00:00.000Z',
      'Freight': 4.62,
      'ShipName': 'Santé Gourmet',
      'ShipAddress': 'Erling Skakkes gate 78',
      'ShipCity': 'Stavern',
      'ShipRegion': null,
      'ShipCountry': 'Norway'
    },
    {
      'OrderID': 11016,
      'CustomerID': 'AROUT',
      'OrderDate': '1998-04-10T00:00:00.000Z',
      'ShippedDate': '1998-04-13T00:00:00.000Z',
      'Freight': 33.8,
      'ShipName': 'Around the Horn',
      'ShipAddress': 'Brook Farm Stratford St. Mary',
      'ShipCity': 'Colchester',
      'ShipRegion': 'Essex',
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 11017,
      'CustomerID': 'ERNSH',
      'OrderDate': '1998-04-13T00:00:00.000Z',
      'ShippedDate': '1998-04-20T00:00:00.000Z',
      'Freight': 754.26,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 11018,
      'CustomerID': 'LONEP',
      'OrderDate': '1998-04-13T00:00:00.000Z',
      'ShippedDate': '1998-04-16T00:00:00.000Z',
      'Freight': 11.65,
      'ShipName': 'Lonesome Pine Restaurant',
      'ShipAddress': '89 Chiaroscuro Rd.',
      'ShipCity': 'Portland',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 11019,
      'CustomerID': 'RANCH',
      'OrderDate': '1998-04-13T00:00:00.000Z',
      'ShippedDate': null,
      'Freight': 3.17,
      'ShipName': 'Rancho grande',

```

```

    'ShipAddress': 'Av. del Libertador 900',
    'ShipCity': 'Buenos Aires',
    'ShipRegion': null,
    'ShipCountry': 'Argentina'
  },
  {
    'OrderID': 11020,
    'CustomerID': 'OTTIK',
    'OrderDate': '1998-04-14T00:00:00.000Z',
    'ShippedDate': '1998-04-16T00:00:00.000Z',
    'Freight': 43.3,
    'ShipName': 'Ottilies Käseladen',
    'ShipAddress': 'Mehrheimerstr. 369',
    'ShipCity': 'Köln',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 11021,
    'CustomerID': 'QUICK',
    'OrderDate': '1998-04-14T00:00:00.000Z',
    'ShippedDate': '1998-04-21T00:00:00.000Z',
    'Freight': 297.18,
    'ShipName': 'QUICK-Stop',
    'ShipAddress': 'Taucherstraße 10',
    'ShipCity': 'Cunewalde',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 11022,
    'CustomerID': 'HANAR',
    'OrderDate': '1998-04-14T00:00:00.000Z',
    'ShippedDate': '1998-05-04T00:00:00.000Z',
    'Freight': 6.27,
    'ShipName': 'Hanari Carnes',
    'ShipAddress': 'Rua do Paço, 67',
    'ShipCity': 'Rio de Janeiro',
    'ShipRegion': 'RJ',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 11023,
    'CustomerID': 'BSBEV',
    'OrderDate': '1998-04-14T00:00:00.000Z',
    'ShippedDate': '1998-04-24T00:00:00.000Z',
    'Freight': 123.83,
    'ShipName': 'B\ Beverages',
    'ShipAddress': 'Fauntleroy Circus',
    'ShipCity': 'London',
    'ShipRegion': null,
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 11024,
    'CustomerID': 'EASTC',
    'OrderDate': '1998-04-15T00:00:00.000Z',

```

```

      'ShippedDate': '1998-04-20T00:00:00.000Z',
      'Freight': 74.36,
      'ShipName': 'Eastern Connection',
      'ShipAddress': '35 King George',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 11025,
      'CustomerID': 'WARTH',
      'OrderDate': '1998-04-15T00:00:00.000Z',
      'ShippedDate': '1998-04-24T00:00:00.000Z',
      'Freight': 29.17,
      'ShipName': 'Wartian Herkku',
      'ShipAddress': 'Torikatu 38',
      'ShipCity': 'Oulu',
      'ShipRegion': null,
      'ShipCountry': 'Finland'
    },
    {
      'OrderID': 11026,
      'CustomerID': 'FRANS',
      'OrderDate': '1998-04-15T00:00:00.000Z',
      'ShippedDate': '1998-04-28T00:00:00.000Z',
      'Freight': 47.09,
      'ShipName': 'Franchi S.p.A.',
      'ShipAddress': 'Via Monte Bianco 34',
      'ShipCity': 'Torino',
      'ShipRegion': null,
      'ShipCountry': 'Italy'
    },
    {
      'OrderID': 11027,
      'CustomerID': 'BOTTM',
      'OrderDate': '1998-04-16T00:00:00.000Z',
      'ShippedDate': '1998-04-20T00:00:00.000Z',
      'Freight': 52.52,
      'ShipName': 'Bottom-Dollar Markets',
      'ShipAddress': '23 Tsawassen Blvd.',
      'ShipCity': 'Tsawassen',
      'ShipRegion': 'BC',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 11028,
      'CustomerID': 'KOENE',
      'OrderDate': '1998-04-16T00:00:00.000Z',
      'ShippedDate': '1998-04-22T00:00:00.000Z',
      'Freight': 29.59,
      'ShipName': 'Königlich Essen',
      'ShipAddress': 'Maubelstr. 90',
      'ShipCity': 'Brandenburg',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    }
  ],
  {

```



```

      'OrderID': 11029,
      'CustomerID': 'CHOPS',
      'OrderDate': '1998-04-16T00:00:00.000Z',
      'ShippedDate': '1998-04-27T00:00:00.000Z',
      'Freight': 47.84,
      'ShipName': 'Chop-suey Chinese',
      'ShipAddress': 'Hauptstr. 31',
      'ShipCity': 'Bern',
      'ShipRegion': null,
      'ShipCountry': 'Switzerland'
    },
    {
      'OrderID': 11030,
      'CustomerID': 'SAVEA',
      'OrderDate': '1998-04-17T00:00:00.000Z',
      'ShippedDate': '1998-04-27T00:00:00.000Z',
      'Freight': 830.75,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 11031,
      'CustomerID': 'SAVEA',
      'OrderDate': '1998-04-17T00:00:00.000Z',
      'ShippedDate': '1998-04-24T00:00:00.000Z',
      'Freight': 227.22,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 11032,
      'CustomerID': 'WHITC',
      'OrderDate': '1998-04-17T00:00:00.000Z',
      'ShippedDate': '1998-04-23T00:00:00.000Z',
      'Freight': 606.19,
      'ShipName': 'White Clover Markets',
      'ShipAddress': '1029 - 12th Ave. S.',
      'ShipCity': 'Seattle',
      'ShipRegion': 'WA',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 11033,
      'CustomerID': 'RICSU',
      'OrderDate': '1998-04-17T00:00:00.000Z',
      'ShippedDate': '1998-04-23T00:00:00.000Z',
      'Freight': 84.74,
      'ShipName': 'Richter Supermarkt',
      'ShipAddress': 'Starenweg 5',
      'ShipCity': 'Genève',
      'ShipRegion': null,

```

```

    'ShipCountry': 'Switzerland'
  },
  {
    'OrderID': 11034,
    'CustomerID': 'OLDWO',
    'OrderDate': '1998-04-20T00:00:00.000Z',
    'ShippedDate': '1998-04-27T00:00:00.000Z',
    'Freight': 40.32,
    'ShipName': 'Old World Delicatessen',
    'ShipAddress': '2743 Bering St.',
    'ShipCity': 'Anchorage',
    'ShipRegion': 'AK',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 11035,
    'CustomerID': 'SUPRD',
    'OrderDate': '1998-04-20T00:00:00.000Z',
    'ShippedDate': '1998-04-24T00:00:00.000Z',
    'Freight': 0.17,
    'ShipName': 'Suprêmes délices',
    'ShipAddress': 'Boulevard Tirou, 255',
    'ShipCity': 'Charleroi',
    'ShipRegion': null,
    'ShipCountry': 'Belgium'
  },
  {
    'OrderID': 11036,
    'CustomerID': 'DRACD',
    'OrderDate': '1998-04-20T00:00:00.000Z',
    'ShippedDate': '1998-04-22T00:00:00.000Z',
    'Freight': 149.47,
    'ShipName': 'Drachenblut Delikatessen',
    'ShipAddress': 'Walserweg 21',
    'ShipCity': 'Aachen',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 11037,
    'CustomerID': 'GODOS',
    'OrderDate': '1998-04-21T00:00:00.000Z',
    'ShippedDate': '1998-04-27T00:00:00.000Z',
    'Freight': 3.2,
    'ShipName': 'Godos Cocina Típica',
    'ShipAddress': 'C/ Romero, 33',
    'ShipCity': 'Sevilla',
    'ShipRegion': null,
    'ShipCountry': 'Spain'
  },
  {
    'OrderID': 11038,
    'CustomerID': 'SUPRD',
    'OrderDate': '1998-04-21T00:00:00.000Z',
    'ShippedDate': '1998-04-30T00:00:00.000Z',
    'Freight': 29.59,
    'ShipName': 'Suprêmes délices',

```

```

      'ShipAddress': 'Boulevard Tirou, 255',
      'ShipCity': 'Charleroi',
      'ShipRegion': null,
      'ShipCountry': 'Belgium'
    },
    {
      'OrderID': 11039,
      'CustomerID': 'LINOD',
      'OrderDate': '1998-04-21T00:00:00.000Z',
      'ShippedDate': null,
      'Freight': 65,
      'ShipName': 'LINO-Delicatesses',
      'ShipAddress': 'Ave. 5 de Mayo Porlamar',
      'ShipCity': 'I. de Margarita',
      'ShipRegion': 'Nueva Esparta',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 11040,
      'CustomerID': 'GREAL',
      'OrderDate': '1998-04-22T00:00:00.000Z',
      'ShippedDate': null,
      'Freight': 18.84,
      'ShipName': 'Great Lakes Food Market',
      'ShipAddress': '2732 Baker Blvd.',
      'ShipCity': 'Eugene',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 11041,
      'CustomerID': 'CHOPS',
      'OrderDate': '1998-04-22T00:00:00.000Z',
      'ShippedDate': '1998-04-28T00:00:00.000Z',
      'Freight': 48.22,
      'ShipName': 'Chop-suey Chinese',
      'ShipAddress': 'Hauptstr. 31',
      'ShipCity': 'Bern',
      'ShipRegion': null,
      'ShipCountry': 'Switzerland'
    },
    {
      'OrderID': 11042,
      'CustomerID': 'COMMI',
      'OrderDate': '1998-04-22T00:00:00.000Z',
      'ShippedDate': '1998-05-01T00:00:00.000Z',
      'Freight': 29.99,
      'ShipName': 'Comércio Mineiro',
      'ShipAddress': 'Av. dos Lusíadas, 23',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 11043,
      'CustomerID': 'SPECB',
      'OrderDate': '1998-04-22T00:00:00.000Z',

```

```

    'ShippedDate': '1998-04-29T00:00:00.000Z',
    'Freight': 8.8,
    'ShipName': 'Spécialités du monde',
    'ShipAddress': '25, rue Lauriston',
    'ShipCity': 'Paris',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 11044,
    'CustomerID': 'WOLZA',
    'OrderDate': '1998-04-23T00:00:00.000Z',
    'ShippedDate': '1998-05-01T00:00:00.000Z',
    'Freight': 8.72,
    'ShipName': 'Wolski Zajazd',
    'ShipAddress': 'ul. Filtrowa 68',
    'ShipCity': 'Warszawa',
    'ShipRegion': null,
    'ShipCountry': 'Poland'
  },
  {
    'OrderID': 11045,
    'CustomerID': 'BOTTM',
    'OrderDate': '1998-04-23T00:00:00.000Z',
    'ShippedDate': null,
    'Freight': 70.58,
    'ShipName': 'Bottom-Dollar Markets',
    'ShipAddress': '23 Tsawassen Blvd.',
    'ShipCity': 'Tsawassen',
    'ShipRegion': 'BC',
    'ShipCountry': 'Canada'
  },
  {
    'OrderID': 11046,
    'CustomerID': 'WANDK',
    'OrderDate': '1998-04-23T00:00:00.000Z',
    'ShippedDate': '1998-04-24T00:00:00.000Z',
    'Freight': 71.64,
    'ShipName': 'Die Wandernde Kuh',
    'ShipAddress': 'Adenauerallee 900',
    'ShipCity': 'Stuttgart',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 11047,
    'CustomerID': 'EASTC',
    'OrderDate': '1998-04-24T00:00:00.000Z',
    'ShippedDate': '1998-05-01T00:00:00.000Z',
    'Freight': 46.62,
    'ShipName': 'Eastern Connection',
    'ShipAddress': '35 King George',
    'ShipCity': 'London',
    'ShipRegion': null,
    'ShipCountry': 'UK'
  },
  {

```

```

      'OrderID': 11048,
      'CustomerID': 'BOTTM',
      'OrderDate': '1998-04-24T00:00:00.000Z',
      'ShippedDate': '1998-04-30T00:00:00.000Z',
      'Freight': 24.12,
      'ShipName': 'Bottom-Dollar Markets',
      'ShipAddress': '23 Tsawassen Blvd.',
      'ShipCity': 'Tsawassen',
      'ShipRegion': 'BC',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 11049,
      'CustomerID': 'GOURL',
      'OrderDate': '1998-04-24T00:00:00.000Z',
      'ShippedDate': '1998-05-04T00:00:00.000Z',
      'Freight': 8.34,
      'ShipName': 'Gourmet Lanchonetes',
      'ShipAddress': 'Av. Brasil, 442',
      'ShipCity': 'Campinas',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 11050,
      'CustomerID': 'FOLKO',
      'OrderDate': '1998-04-27T00:00:00.000Z',
      'ShippedDate': '1998-05-05T00:00:00.000Z',
      'Freight': 59.41,
      'ShipName': 'Folk och fä HB',
      'ShipAddress': 'Åkergatan 24',
      'ShipCity': 'Bräcke',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 11051,
      'CustomerID': 'LAMAI',
      'OrderDate': '1998-04-27T00:00:00.000Z',
      'ShippedDate': null,
      'Freight': 2.79,
      'ShipName': 'La maison d\'Asie',
      'ShipAddress': '1 rue Alsace-Lorraine',
      'ShipCity': 'Toulouse',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 11052,
      'CustomerID': 'HANAR',
      'OrderDate': '1998-04-27T00:00:00.000Z',
      'ShippedDate': '1998-05-01T00:00:00.000Z',
      'Freight': 67.26,
      'ShipName': 'Hanari Carnes',
      'ShipAddress': 'Rua do Paço, 67',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',

```

```

    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 11053,
    'CustomerID': 'PICCO',
    'OrderDate': '1998-04-27T00:00:00.000Z',
    'ShippedDate': '1998-04-29T00:00:00.000Z',
    'Freight': 53.05,
    'ShipName': 'Piccolo und mehr',
    'ShipAddress': 'Geislweg 14',
    'ShipCity': 'Salzburg',
    'ShipRegion': null,
    'ShipCountry': 'Austria'
  },
  {
    'OrderID': 11054,
    'CustomerID': 'CACTU',
    'OrderDate': '1998-04-28T00:00:00.000Z',
    'ShippedDate': null,
    'Freight': 0.33,
    'ShipName': 'Cactus Comidas para llevar',
    'ShipAddress': 'Cerrito 333',
    'ShipCity': 'Buenos Aires',
    'ShipRegion': null,
    'ShipCountry': 'Argentina'
  },
  {
    'OrderID': 11055,
    'CustomerID': 'HILAA',
    'OrderDate': '1998-04-28T00:00:00.000Z',
    'ShippedDate': '1998-05-05T00:00:00.000Z',
    'Freight': 120.92,
    'ShipName': 'HILARION-Abastos',
    'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
    'ShipCity': 'San Cristóbal',
    'ShipRegion': 'Táchira',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 11056,
    'CustomerID': 'EASTC',
    'OrderDate': '1998-04-28T00:00:00.000Z',
    'ShippedDate': '1998-05-01T00:00:00.000Z',
    'Freight': 278.96,
    'ShipName': 'Eastern Connection',
    'ShipAddress': '35 King George',
    'ShipCity': 'London',
    'ShipRegion': null,
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 11057,
    'CustomerID': 'NORTS',
    'OrderDate': '1998-04-29T00:00:00.000Z',
    'ShippedDate': '1998-05-01T00:00:00.000Z',
    'Freight': 4.13,
    'ShipName': 'North/South',

```

```

      'ShipAddress': 'South House 300 Queensbridge',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 11058,
      'CustomerID': 'BLAUS',
      'OrderDate': '1998-04-29T00:00:00.000Z',
      'ShippedDate': null,
      'Freight': 31.14,
      'ShipName': 'Blauer See Delikatessen',
      'ShipAddress': 'Forsterstr. 57',
      'ShipCity': 'Mannheim',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 11059,
      'CustomerID': 'RICAR',
      'OrderDate': '1998-04-29T00:00:00.000Z',
      'ShippedDate': null,
      'Freight': 85.8,
      'ShipName': 'Ricardo Adocicados',
      'ShipAddress': 'Av. Copacabana, 267',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 11060,
      'CustomerID': 'FRANS',
      'OrderDate': '1998-04-30T00:00:00.000Z',
      'ShippedDate': '1998-05-04T00:00:00.000Z',
      'Freight': 10.98,
      'ShipName': 'Franchi S.p.A.',
      'ShipAddress': 'Via Monte Bianco 34',
      'ShipCity': 'Torino',
      'ShipRegion': null,
      'ShipCountry': 'Italy'
    },
    {
      'OrderID': 11061,
      'CustomerID': 'GREAL',
      'OrderDate': '1998-04-30T00:00:00.000Z',
      'ShippedDate': null,
      'Freight': 14.01,
      'ShipName': 'Great Lakes Food Market',
      'ShipAddress': '2732 Baker Blvd.',
      'ShipCity': 'Eugene',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 11062,
      'CustomerID': 'REGGC',
      'OrderDate': '1998-04-30T00:00:00.000Z',

```

```

        'ShippedDate': null,
        'Freight': 29.93,
        'ShipName': 'Reggiani Caseifici',
        'ShipAddress': 'Strada Provinciale 124',
        'ShipCity': 'Reggio Emilia',
        'ShipRegion': null,
        'ShipCountry': 'Italy'
    },
    {
        'OrderID': 11063,
        'CustomerID': 'HUNGO',
        'OrderDate': '1998-04-30T00:00:00.000Z',
        'ShippedDate': '1998-05-06T00:00:00.000Z',
        'Freight': 81.73,
        'ShipName': 'Hungry Owl All-Night Grocers',
        'ShipAddress': '8 Johnstown Road',
        'ShipCity': 'Cork',
        'ShipRegion': 'Co. Cork',
        'ShipCountry': 'Ireland'
    },
    {
        'OrderID': 11064,
        'CustomerID': 'SAVEA',
        'OrderDate': '1998-05-01T00:00:00.000Z',
        'ShippedDate': '1998-05-04T00:00:00.000Z',
        'Freight': 30.09,
        'ShipName': 'Save-a-lot Markets',
        'ShipAddress': '187 Suffolk Ln.',
        'ShipCity': 'Boise',
        'ShipRegion': 'ID',
        'ShipCountry': 'USA'
    },
    {
        'OrderID': 11065,
        'CustomerID': 'LILAS',
        'OrderDate': '1998-05-01T00:00:00.000Z',
        'ShippedDate': null,
        'Freight': 12.91,
        'ShipName': 'LILA-Supermercado',
        'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
        'ShipCity': 'Barquisimeto',
        'ShipRegion': 'Lara',
        'ShipCountry': 'Venezuela'
    },
    {
        'OrderID': 11066,
        'CustomerID': 'WHITC',
        'OrderDate': '1998-05-01T00:00:00.000Z',
        'ShippedDate': '1998-05-04T00:00:00.000Z',
        'Freight': 44.72,
        'ShipName': 'White Clover Markets',
        'ShipAddress': '1029 - 12th Ave. S.',
        'ShipCity': 'Seattle',
        'ShipRegion': 'WA',
        'ShipCountry': 'USA'
    },
    {

```



```

      'OrderID': 11067,
      'CustomerID': 'DRACD',
      'OrderDate': '1998-05-04T00:00:00.000Z',
      'ShippedDate': '1998-05-06T00:00:00.000Z',
      'Freight': 7.98,
      'ShipName': 'Drachenblut Delikatessen',
      'ShipAddress': 'Walserweg 21',
      'ShipCity': 'Aachen',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 11068,
      'CustomerID': 'QUEEN',
      'OrderDate': '1998-05-04T00:00:00.000Z',
      'ShippedDate': null,
      'Freight': 81.75,
      'ShipName': 'Queen Cozinha',
      'ShipAddress': 'Alameda dos Canários, 891',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 11069,
      'CustomerID': 'TORTU',
      'OrderDate': '1998-05-04T00:00:00.000Z',
      'ShippedDate': '1998-05-06T00:00:00.000Z',
      'Freight': 15.67,
      'ShipName': 'Tortuga Restaurante',
      'ShipAddress': 'Avda. Azteca 123',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 11070,
      'CustomerID': 'LEHMS',
      'OrderDate': '1998-05-05T00:00:00.000Z',
      'ShippedDate': null,
      'Freight': 136,
      'ShipName': 'Lehmanns Marktstand',
      'ShipAddress': 'Magazinweg 7',
      'ShipCity': 'Frankfurt a.M.',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 11071,
      'CustomerID': 'LILAS',
      'OrderDate': '1998-05-05T00:00:00.000Z',
      'ShippedDate': null,
      'Freight': 0.93,
      'ShipName': 'LILA-Supermercado',
      'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
      'ShipCity': 'Barquisimeto',
      'ShipRegion': 'Lara',

```

```

    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 11072,
    'CustomerID': 'ERNSH',
    'OrderDate': '1998-05-05T00:00:00.000Z',
    'ShippedDate': null,
    'Freight': 258.64,
    'ShipName': 'Ernst Handel',
    'ShipAddress': 'Kirchgasse 6',
    'ShipCity': 'Graz',
    'ShipRegion': null,
    'ShipCountry': 'Austria'
  },
  {
    'OrderID': 11073,
    'CustomerID': 'PERIC',
    'OrderDate': '1998-05-05T00:00:00.000Z',
    'ShippedDate': null,
    'Freight': 24.95,
    'ShipName': 'Pericles Comidas clásicas',
    'ShipAddress': 'Calle Dr. Jorge Cash 321',
    'ShipCity': 'México D.F.',
    'ShipRegion': null,
    'ShipCountry': 'Mexico'
  },
  {
    'OrderID': 11074,
    'CustomerID': 'SIMOB',
    'OrderDate': '1998-05-06T00:00:00.000Z',
    'ShippedDate': null,
    'Freight': 18.44,
    'ShipName': 'Simons bistro',
    'ShipAddress': 'Vinbæltet 34',
    'ShipCity': 'Kobenhavn',
    'ShipRegion': null,
    'ShipCountry': 'Denmark'
  },
  {
    'OrderID': 11075,
    'CustomerID': 'RICSU',
    'OrderDate': '1998-05-06T00:00:00.000Z',
    'ShippedDate': null,
    'Freight': 6.19,
    'ShipName': 'Richter Supermarkt',
    'ShipAddress': 'Starenweg 5',
    'ShipCity': 'Genève',
    'ShipRegion': null,
    'ShipCountry': 'Switzerland'
  },
  {
    'OrderID': 11076,
    'CustomerID': 'BONAP',
    'OrderDate': '1998-05-06T00:00:00.000Z',
    'ShippedDate': null,
    'Freight': 38.28,
    'ShipName': 'Bon app',

```

```

        'ShipAddress': '12, rue des Bouchers',
        'ShipCity': 'Marseille',
        'ShipRegion': null,
        'ShipCountry': 'France'
    },
    {
        'OrderID': 11077,
        'CustomerID': 'RATTC',
        'OrderDate': '1998-05-06T00:00:00.000Z',
        'ShippedDate': null,
        'Freight': 8.53,
        'ShipName': 'Rattlesnake Canyon Grocery',
        'ShipAddress': '2817 Milton Dr.',
        'ShipCity': 'Albuquerque',
        'ShipRegion': 'NM',
        'ShipCountry': 'USA'
    }
]);
export const orderData = JSON.parse(stringData, (field, value) => {
    let dupValue = value;
    if (typeof value === 'string' && /^(\\d{4}\\-\\d\\d\\-\\d\\d([tT][\\d:\\.]*){1})([zZ]|([+\\-])(\\d\\d):?(\\d\\d):?$)/.test(value)) {
        let arr = dupValue.split(/[0-9]/);
        let arg = parseInt(arr[4], 10);
        let arg1 = parseInt(arr[5], 10);
        value = new Date(parseInt(arr[0], 10), parseInt(arr[1], 10) - 1,
            parseInt(arr[2], 10), parseInt(arr[3], 10), arg, arg1);
    }
    return value;
});
export const categoryData = [
    {
        'CategoryName': 'Beverages',
        'ProductName': 'Chai',
        'QuantityPerUnit': '10 boxes x 20 bags',
        'UnitsInStock': 39,
        'Discontinued': true
    },
    {
        'CategoryName': 'Beverages',
        'ProductName': 'Chang',
        'QuantityPerUnit': '24 - 12 oz bottles',
        'UnitsInStock': 17,
        'Discontinued': true
    },
    {
        'CategoryName': 'Beverages',
        'ProductName': 'Chartreuse verte',
        'QuantityPerUnit': '750 cc per bottle',
        'UnitsInStock': 69,
        'Discontinued': true
    },
    {
        'CategoryName': 'Beverages',
        'ProductName': 'C\\u00f4te de Blaye',
        'QuantityPerUnit': '12 - 75 cl bottles',
        'UnitsInStock': 17,
    }
];

```

```

    'Discontinued': false
  },
  {
    'CategoryName': 'Beverages',
    'ProductName': 'Ipoh Coffee',
    'QuantityPerUnit': '16 - 500 g tins',
    'UnitsInStock': 17,
    'Discontinued': true
  },
  {
    'CategoryName': 'Beverages',
    'ProductName': 'Lakkalik\u00f6\u00f6ri',
    'QuantityPerUnit': '500 ml',
    'UnitsInStock': 57,
    'Discontinued': true
  },
  {
    'CategoryName': 'Beverages',
    'ProductName': 'Laughing Lumberjack Lager',
    'QuantityPerUnit': '24 - 12 oz bottles',
    'UnitsInStock': 52,
    'Discontinued': false
  },
  {
    'CategoryName': 'Beverages',
    'ProductName': 'Outback Lager',
    'QuantityPerUnit': '24 - 355 ml bottles',
    'UnitsInStock': 15,
    'Discontinued': false
  },
  {
    'CategoryName': 'Beverages',
    'ProductName': 'Rh\u00f6nbr\u00e4u Klosterbier',
    'QuantityPerUnit': '24 - 0.5 l bottles',
    'UnitsInStock': 125,
    'Discontinued': false
  },
  {
    'CategoryName': 'Beverages',
    'ProductName': 'Sasquatch Ale',
    'QuantityPerUnit': '24 - 12 oz bottles',
    'UnitsInStock': 111,
    'Discontinued': true
  },
  {
    'CategoryName': 'Beverages',
    'ProductName': 'Steeleye Stout',
    'QuantityPerUnit': '24 - 12 oz bottles',
    'UnitsInStock': 20,
    'Discontinued': true
  },
  {
    'CategoryName': 'Condiments',
    'ProductName': 'Aniseed Syrup',
    'QuantityPerUnit': '12 - 550 ml bottles',
    'UnitsInStock': 13,
    'Discontinued': true
  }

```

```

    },
    {
      'CategoryName': 'Condiments',
      'ProductName': 'Chef Anton\'s Cajun Seasoning',
      'QuantityPerUnit': '48 - 6 oz jars',
      'UnitsInStock': 53,
      'Discontinued': false
    },
    {
      'CategoryName': 'Condiments',
      'ProductName': 'Genen Shouyu',
      'QuantityPerUnit': '24 - 250 ml bottles',
      'UnitsInStock': 39,
      'Discontinued': false
    },
    {
      'CategoryName': 'Condiments',
      'ProductName': 'Grandma\'s Boysenberry Spread',
      'QuantityPerUnit': '12 - 8 oz jars',
      'UnitsInStock': 120,
      'Discontinued': false
    },
    {
      'CategoryName': 'Condiments',
      'ProductName': 'Gula Malacca',
      'QuantityPerUnit': '20 - 2 kg bags',
      'UnitsInStock': 27,
      'Discontinued': true
    },
    {
      'CategoryName': 'Condiments',
      'ProductName': 'Louisiana Fiery Hot Pepper Sauce',
      'QuantityPerUnit': '32 - 8 oz bottles',
      'UnitsInStock': 76,
      'Discontinued': false
    },
    {
      'CategoryName': 'Condiments',
      'ProductName': 'Louisiana Hot Spiced Okra',
      'QuantityPerUnit': '24 - 8 oz jars',
      'UnitsInStock': 4,
      'Discontinued': false
    },
    {
      'CategoryName': 'Condiments',
      'ProductName': 'Northwoods Cranberry Sauce',
      'QuantityPerUnit': '12 - 12 oz jars',
      'UnitsInStock': 6,
      'Discontinued': true
    },
    {
      'CategoryName': 'Condiments',
      'ProductName': 'Original Frankfurter gr\u00f6\u00dfe',
      'QuantityPerUnit': '12 boxes',
      'UnitsInStock': 32,
      'Discontinued': false
    },
  ],

```

```

{
  'CategoryName': 'Condiments',
  'ProductName': 'Sirop d'\u00e9rable',
  'QuantityPerUnit': '24 - 500 ml bottles',
  'UnitsInStock': 113,
  'Discontinued': false
},
{
  'CategoryName': 'Condiments',
  'ProductName': 'Vegie-spread',
  'QuantityPerUnit': '15 - 625 g jars',
  'UnitsInStock': 24,
  'Discontinued': false
},
{
  'CategoryName': 'Confections',
  'ProductName': 'Chocolade',
  'QuantityPerUnit': '10 pkgs.',
  'UnitsInStock': 15,
  'Discontinued': true
},
{
  'CategoryName': 'Confections',
  'ProductName': 'Gumb\u00e4r Gummib\u00e4rchen',
  'QuantityPerUnit': '100 - 250 g bags',
  'UnitsInStock': 15,
  'Discontinued': false
},
{
  'CategoryName': 'Confections',
  'ProductName': 'Maxilaku',
  'QuantityPerUnit': '24 - 50 g pkgs.',
  'UnitsInStock': 10,
  'Discontinued': false
},
{
  'CategoryName': 'Confections',
  'ProductName': 'NuNuCa Nu\u00df-Nougat-Creme',
  'QuantityPerUnit': '20 - 450 g glasses',
  'UnitsInStock': 76,
  'Discontinued': true
},
{
  'CategoryName': 'Confections',
  'ProductName': 'Pavlova',
  'QuantityPerUnit': '32 - 500 g boxes',
  'UnitsInStock': 29,
  'Discontinued': true
},
{
  'CategoryName': 'Confections',
  'ProductName': 'Schoggi Schokolade',
  'QuantityPerUnit': '100 - 100 g pieces',
  'UnitsInStock': 49,
  'Discontinued': true
},
{

```

```

    'CategoryName': 'Confections',
    'ProductName': 'Scottish Longbreads',
    'QuantityPerUnit': '10 boxes x 8 pieces',
    'UnitsInStock': 6,
    'Discontinued': true
  },
  {
    'CategoryName': 'Confections',
    'ProductName': 'Sir Rodney\'s Marmalade',
    'QuantityPerUnit': '30 gift boxes',
    'UnitsInStock': 40,
    'Discontinued': true
  },
  {
    'CategoryName': 'Confections',
    'ProductName': 'Sir Rodney\'s Scones',
    'QuantityPerUnit': '24 pkgs. x 4 pieces',
    'UnitsInStock': 3,
    'Discontinued': true
  },
  {
    'CategoryName': 'Confections',
    'ProductName': 'Tarte au sucre',
    'QuantityPerUnit': '48 pies',
    'UnitsInStock': 17,
    'Discontinued': false
  },
  {
    'CategoryName': 'Confections',
    'ProductName': 'Teatime Chocolate Biscuits',
    'QuantityPerUnit': '10 boxes x 12 pieces',
    'UnitsInStock': 25,
    'Discontinued': false
  },
  {
    'CategoryName': 'Confections',
    'ProductName': 'Valkoinen suklaa',
    'QuantityPerUnit': '12 - 100 g bars',
    'UnitsInStock': 65,
    'Discontinued': false
  },
  {
    'CategoryName': 'Confections',
    'ProductName': 'Zaanse koeken',
    'QuantityPerUnit': '10 - 4 oz boxes',
    'UnitsInStock': 36,
    'Discontinued': true
  },
  {
    'CategoryName': 'Dairy Products',
    'ProductName': 'Camembert Pierrot',
    'QuantityPerUnit': '15 - 300 g rounds',
    'UnitsInStock': 19,
    'Discontinued': false
  },
  {
    'CategoryName': 'Dairy Products',

```

```

    'ProductName': 'Flotemysost',
    'QuantityPerUnit': '10 - 500 g pkgs.',
    'UnitsInStock': 26,
    'Discontinued': false
  },
  {
    'CategoryName': 'Dairy Products',
    'ProductName': 'Geitost',
    'QuantityPerUnit': '500 g',
    'UnitsInStock': 112,
    'Discontinued': false
  },
  {
    'CategoryName': 'Dairy Products',
    'ProductName': 'Gorgonzola Telino',
    'QuantityPerUnit': '12 - 100 g pkgs',
    'UnitsInStock': 0,
    'Discontinued': true
  },
  {
    'CategoryName': 'Dairy Products',
    'ProductName': 'Gudbrandsdalsost',
    'QuantityPerUnit': '10 kg pkg.',
    'UnitsInStock': 26,
    'Discontinued': true
  },
  {
    'CategoryName': 'Dairy Products',
    'ProductName': 'Mascarpone Fabioli',
    'QuantityPerUnit': '24 - 200 g pkgs.',
    'UnitsInStock': 9,
    'Discontinued': true
  },
  {
    'CategoryName': 'Dairy Products',
    'ProductName': 'Mozzarella di Giovanni',
    'QuantityPerUnit': '24 - 200 g pkgs.',
    'UnitsInStock': 14,
    'Discontinued': true
  },
  {
    'CategoryName': 'Dairy Products',
    'ProductName': 'Queso Cabrales',
    'QuantityPerUnit': '1 kg pkg.',
    'UnitsInStock': 22,
    'Discontinued': true
  },
  {
    'CategoryName': 'Dairy Products',
    'ProductName': 'Queso Manchego La Pastora',
    'QuantityPerUnit': '10 - 500 g pkgs.',
    'UnitsInStock': 86,
    'Discontinued': true
  },
  {
    'CategoryName': 'Dairy Products',
    'ProductName': 'Raclette Courdavault',

```



```

    'QuantityPerUnit': '5 kg pkg.',
    'UnitsInStock': 79,
    'Discontinued': false
  },
  {
    'CategoryName': 'Grains/Cereals',
    'ProductName': 'Filo Mix',
    'QuantityPerUnit': '16 - 2 kg boxes',
    'UnitsInStock': 38,
    'Discontinued': false
  },
  {
    'CategoryName': 'Grains/Cereals',
    'ProductName': 'Gnocchi di nonna Alice',
    'QuantityPerUnit': '24 - 250 g pkgs.',
    'UnitsInStock': 21,
    'Discontinued': false
  },
  {
    'CategoryName': 'Grains/Cereals',
    'ProductName': 'Gustaf\'s Kn\u00e4ckebr\u00f6d',
    'QuantityPerUnit': '24 - 500 g pkgs.',
    'UnitsInStock': 104,
    'Discontinued': true
  },
  {
    'CategoryName': 'Grains/Cereals',
    'ProductName': 'Ravioli Angelo',
    'QuantityPerUnit': '24 - 250 g pkgs.',
    'UnitsInStock': 36,
    'Discontinued': true
  },
  {
    'CategoryName': 'Grains/Cereals',
    'ProductName': 'Tunnbr\u00f6d',
    'QuantityPerUnit': '12 - 250 g pkgs.',
    'UnitsInStock': 61,
    'Discontinued': true
  },
  {
    'CategoryName': 'Grains/Cereals',
    'ProductName': 'Wimmers gute Semmelkn\u00f6del',
    'QuantityPerUnit': '20 bags x 4 pieces',
    'UnitsInStock': 22,
    'Discontinued': true
  },
  {
    'CategoryName': 'Meat/Poultry',
    'ProductName': 'P\u00e2t\u00e9 chinois',
    'QuantityPerUnit': '24 boxes x 2 pies',
    'UnitsInStock': 115,
    'Discontinued': false
  },
  {
    'CategoryName': 'Meat/Poultry',
    'ProductName': 'Tourti\u00e8re',
    'QuantityPerUnit': '16 pies',

```

```

      'UnitsInStock': 21,
      'Discontinued': false
    },
    {
      'CategoryName': 'Produce',
      'ProductName': 'Longlife Tofu',
      'QuantityPerUnit': '5 kg pkg.',
      'UnitsInStock': 4,
      'Discontinued': false
    },
    {
      'CategoryName': 'Produce',
      'ProductName': 'Manjimup Dried Apples',
      'QuantityPerUnit': '50 - 300 g pkgs.',
      'UnitsInStock': 20,
      'Discontinued': false
    },
    {
      'CategoryName': 'Produce',
      'ProductName': 'Tofu',
      'QuantityPerUnit': '40 - 100 g pkgs.',
      'UnitsInStock': 35,
      'Discontinued': true
    },
    {
      'CategoryName': 'Produce',
      'ProductName': 'Uncle Bob\'s Organic Dried Pears',
      'QuantityPerUnit': '12 - 1 lb pkgs.',
      'UnitsInStock': 15,
      'Discontinued': true
    },
    {
      'CategoryName': 'Seafood',
      'ProductName': 'Boston Crab Meat',
      'QuantityPerUnit': '24 - 4 oz tins',
      'UnitsInStock': 123,
      'Discontinued': true
    },
    {
      'CategoryName': 'Seafood',
      'ProductName': 'Carnarvon Tigers',
      'QuantityPerUnit': '16 kg pkg.',
      'UnitsInStock': 42,
      'Discontinued': true
    },
    {
      'CategoryName': 'Seafood',
      'ProductName': 'Escargots de Bourgogne',
      'QuantityPerUnit': '24 pieces',
      'UnitsInStock': 62,
      'Discontinued': true
    },
    {
      'CategoryName': 'Seafood',
      'ProductName': 'Gravad lax',
      'QuantityPerUnit': '12 - 500 g pkgs.',
      'UnitsInStock': 11,

```

```

    'Discontinued': true
  },
  {
    'CategoryName': 'Seafood',
    'ProductName': 'Ikura',
    'QuantityPerUnit': '12 - 200 ml jars',
    'UnitsInStock': 31,
    'Discontinued': false
  },
  {
    'CategoryName': 'Seafood',
    'ProductName': 'Inlagd Sill',
    'QuantityPerUnit': '24 - 250 g jars',
    'UnitsInStock': 112,
    'Discontinued': false
  },
  {
    'CategoryName': 'Seafood',
    'ProductName': 'Jack\'s New England Clam Chowder',
    'QuantityPerUnit': '12 - 12 oz cans',
    'UnitsInStock': 85,
    'Discontinued': false
  },
  {
    'CategoryName': 'Seafood',
    'ProductName': 'Konbu',
    'QuantityPerUnit': '2 kg box',
    'UnitsInStock': 24,
    'Discontinued': false
  },
  {
    'CategoryName': 'Seafood',
    'ProductName': 'Nord-Ost Matjeshering',
    'QuantityPerUnit': '10 - 200 g glasses',
    'UnitsInStock': 10,
    'Discontinued': false
  },
  {
    'CategoryName': 'Seafood',
    'ProductName': 'R\u00f6d Kaviar',
    'QuantityPerUnit': '24 - 150 g jars',
    'UnitsInStock': 101,
    'Discontinued': false
  },
  {
    'CategoryName': 'Seafood',
    'ProductName': 'Rogede sild',
    'QuantityPerUnit': '1k pkg.',
    'UnitsInStock': 5,
    'Discontinued': false
  },
  {
    'CategoryName': 'Seafood',
    'ProductName': 'Spegesild',
    'QuantityPerUnit': '4 - 450 g glasses',
    'UnitsInStock': 95,
    'Discontinued': true
  }

```

```

    }
  ];
  export const customerData = [
    {
      'CustomerID': 'ALFKI',
      'ContactName': 'Maria ',
      'CompanyName': 'Alfreds Futterkiste',
      'Address': 'Obere Str. 57',
      'Country': 'Germany'
    },
    {
      'CustomerID': 'ANATR',
      'ContactName': 'Ana Trujillo',
      'CompanyName': 'Ana Trujillo Emparedados y helados',
      'Address': 'Avda. de la Constitución 2222',
      'Country': 'Mexico'
    },
    {
      'CustomerID': 'ANTON',
      'ContactName': 'Antonio Moreno',
      'CompanyName': 'Antonio Moreno Taquería',
      'Address': 'Mataderos 2312',
      'Country': 'Mexico'
    },
    {
      'CustomerID': 'AROUT',
      'ContactName': 'Thomas Hardy',
      'CompanyName': 'Around the Horn',
      'Address': '120 Hanover Sq.',
      'Country': 'UK'
    },
    {
      'CustomerID': 'BERGS',
      'ContactName': 'Christina Berglund',
      'CompanyName': 'Berglunds snabbköp',
      'Address': 'Berguvsvägen 8',
      'Country': 'Sweden'
    },
    {
      'CustomerID': 'BLAUS',
      'ContactName': 'Hanna Moos',
      'CompanyName': 'Blauer See Delikatessen',
      'Address': 'Forsterstr. 57',
      'Country': 'Germany'
    },
    {
      'CustomerID': 'BLONP',
      'ContactName': 'Frédérique Citeaux',
      'CompanyName': 'Blondesddsl père et fils',
      'Address': '24, place Kléber',
      'Country': 'France'
    },
    {
      'CustomerID': 'BOLID',
      'ContactName': 'Martín Sommer',
      'CompanyName': 'Bólido Comidas preparadas',
      'Address': 'C/ Araquil, 67',
    }
  ]

```

```

    'Country': 'Spain'
  },
  {
    'CustomerID': 'BONAP',
    'ContactName': 'Laurence Lebihan',
    'CompanyName': 'Bon app',
    'Address': '12, rue des Bouchers',
    'Country': 'France'
  },
  {
    'CustomerID': 'BOTTM',
    'ContactName': 'Elizabeth Lincoln',
    'CompanyName': 'Bottom-Dollar Markets',
    'Address': '23 Tsawassen Blvd.',
    'Country': 'Canada'
  },
  {
    'CustomerID': 'BSBEV',
    'ContactName': 'Victoria Ashworth',
    'CompanyName': 'B\'s Beverages',
    'Address': 'Fauntleroy Circus',
    'Country': 'UK'
  },
  {
    'CustomerID': 'CACTU',
    'ContactName': 'Patricio Simpson',
    'CompanyName': 'Cactus Comidas para llevar',
    'Address': 'Cerrito 333',
    'Country': 'Argentina'
  },
  {
    'CustomerID': 'CENTC',
    'ContactName': 'Francisco Chang',
    'CompanyName': 'Centro comercial Moctezuma',
    'Address': 'Sierras de Granada 9993',
    'Country': 'Mexico'
  },
  {
    'CustomerID': 'CHOPS',
    'ContactName': 'Yang Wang',
    'CompanyName': 'Chop-suey Chinese',
    'Address': 'Hauptstr. 29',
    'Country': 'Switzerland'
  },
  {
    'CustomerID': 'COMMI',
    'ContactName': 'Pedro Afonso',
    'CompanyName': 'Comércio Mineiro',
    'Address': 'Av. dos Lusíadas, 23',
    'Country': 'Brazil'
  },
  {
    'CustomerID': 'CONSH',
    'ContactName': 'Elizabeth Brown',
    'CompanyName': 'Consolidated Holdings',
    'Address': 'Berkeley Gardens 12 Brewery',
    'Country': 'UK'
  }

```

```

    },
    {
      'CustomerID': 'DRACD',
      'ContactName': 'Sven Ottlieb',
      'CompanyName': 'Drachenblut Delikatessen',
      'Address': 'Walserweg 21',
      'Country': 'Germany'
    },
    {
      'CustomerID': 'DUMON',
      'ContactName': 'Janine Labrune',
      'CompanyName': 'Du monde entier',
      'Address': '67, rue des Cinquante Otages',
      'Country': 'France'
    },
    {
      'CustomerID': 'EASTC',
      'ContactName': 'Ann Devon',
      'CompanyName': 'Eastern Connection',
      'Address': '35 King George',
      'Country': 'UK'
    },
    {
      'CustomerID': 'ERNSH',
      'ContactName': 'Roland Mendel',
      'CompanyName': 'Ernst Handel',
      'Address': 'Kirchgasse 6',
      'Country': 'Austria'
    },
    {
      'CustomerID': 'FAMIA',
      'ContactName': 'Aria Cruz',
      'CompanyName': 'Familia Arquibaldo',
      'Address': 'Rua Orós, 92',
      'Country': 'Brazil'
    },
    {
      'CustomerID': 'FISSA',
      'ContactName': 'Diego Roel',
      'CompanyName': 'FISSA Fabrica Inter. Salchichas S.A.',
      'Address': 'C/ Moralzarzal, 86',
      'Country': 'Spain'
    },
    {
      'CustomerID': 'FOLIG',
      'ContactName': 'Martine Rancé',
      'CompanyName': 'Folies gourmandes',
      'Address': '184, chaussée de Tournai',
      'Country': 'France'
    },
    {
      'CustomerID': 'FOLKO',
      'ContactName': 'Maria Larsson',
      'CompanyName': 'Folk och fä HB',
      'Address': 'Åkergatan 24',
      'Country': 'Sweden'
    },
  ],

```

```

{
  'CustomerID': 'FRANK',
  'ContactName': 'Peter Franken',
  'CompanyName': 'Frankenversand',
  'Address': 'Berliner Platz 43',
  'Country': 'Germany'
},
{
  'CustomerID': 'FRANR',
  'ContactName': 'Carine Schmitt',
  'CompanyName': 'France restauration',
  'Address': '54, rue Royale',
  'Country': 'France'
},
{
  'CustomerID': 'FRANS',
  'ContactName': 'Paolo Accorti',
  'CompanyName': 'Franchi S.p.A.',
  'Address': 'Via Monte Bianco 34',
  'Country': 'Italy'
},
{
  'CustomerID': 'FURIB',
  'ContactName': 'Lino Rodriguez',
  'CompanyName': 'Furia Bacalhau e Frutos do Mar',
  'Address': 'Jardim das rosas n. 32',
  'Country': 'Portugal'
},
{
  'CustomerID': 'GALED',
  'ContactName': 'Eduardo Saavedra',
  'CompanyName': 'Galería del gastrónomo',
  'Address': 'Rambla de Cataluña, 23',
  'Country': 'Spain'
},
{
  'CustomerID': 'GODOS',
  'ContactName': 'José Pedro Freyre',
  'CompanyName': 'Godos Cocina Típica',
  'Address': 'C/ Romero, 33',
  'Country': 'Spain'
},
{
  'CustomerID': 'GOURL',
  'ContactName': 'André Fonseca',
  'CompanyName': 'Gourmet Lanchonetes',
  'Address': 'Av. Brasil, 442',
  'Country': 'Brazil'
},
{
  'CustomerID': 'GREAL',
  'ContactName': 'Howard Snyder',
  'CompanyName': 'Great Lakes Food Market',
  'Address': '2732 Baker Blvd.',
  'Country': 'USA'
},
{

```

```

    'CustomerID': 'GROSR',
    'ContactName': 'Manuel Pereira',
    'CompanyName': 'GROSELLA-Restaurante',
    'Address': '5ª Ave. Los Palos Grandes',
    'Country': 'Venezuela'
  },
  {
    'CustomerID': 'HANAR',
    'ContactName': 'Mario Pontes',
    'CompanyName': 'Hanari Carnes',
    'Address': 'Rua do Paço, 67',
    'Country': 'Brazil'
  },
  {
    'CustomerID': 'HILAA',
    'ContactName': 'Carlos Hernández',
    'CompanyName': 'HILARION-Abastos',
    'Address': 'Carrera 22 con Ave. Carlos Soublette #8-35',
    'Country': 'Venezuela'
  },
  {
    'CustomerID': 'HUNGC',
    'ContactName': 'Yoshi Latimer',
    'CompanyName': 'Hungry Coyote Import Store',
    'Address': 'City Center Plaza 516 Main St.',
    'Country': 'USA'
  },
  {
    'CustomerID': 'HUNGO',
    'ContactName': 'Patricia McKenna',
    'CompanyName': 'Hungry Owl All-Night Grocers',
    'Address': '8 Johnstown Road',
    'Country': 'Ireland'
  },
  {
    'CustomerID': 'ISLAT',
    'ContactName': 'Helen Bennett',
    'CompanyName': 'Island Trading',
    'Address': 'Garden House Crowther Way',
    'Country': 'UK'
  },
  {
    'CustomerID': 'KOENE',
    'ContactName': 'Philip Cramer',
    'CompanyName': 'Königlich Essen',
    'Address': 'Maubelstr. 90',
    'Country': 'Germany'
  },
  {
    'CustomerID': 'LACOR',
    'ContactName': 'Daniel Tonini',
    'CompanyName': 'La corne d\'abondance',
    'Address': '67, avenue de l\'Europe',
    'Country': 'France'
  },
  {
    'CustomerID': 'LAMAI',

```



```

    'ContactName': 'Annette Roulet',
    'CompanyName': 'La maison d\'Asie',
    'Address': '1 rue Alsace-Lorraine',
    'Country': 'France'
  },
  {
    'CustomerID': 'LAUGB',
    'ContactName': 'Yoshi Tannamuri',
    'CompanyName': 'Laughing Bacchus Wine Cellars',
    'Address': '1900 Oak St.',
    'Country': 'Canada'
  },
  {
    'CustomerID': 'LAZYK',
    'ContactName': 'John Steel',
    'CompanyName': 'Lazy K Kountry Store',
    'Address': '12 Orchestra Terrace',
    'Country': 'USA'
  },
  {
    'CustomerID': 'LEHMS',
    'ContactName': 'Renate Messner',
    'CompanyName': 'Lehmanns Marktstand',
    'Address': 'Magazinweg 7',
    'Country': 'Germany'
  },
  {
    'CustomerID': 'LETSS',
    'ContactName': 'Jaime Yorres',
    'CompanyName': 'Let\'s Stop N Shop',
    'Address': '87 Polk St. Suite 5',
    'Country': 'USA'
  },
  {
    'CustomerID': 'LILAS',
    'ContactName': 'Carlos González',
    'CompanyName': 'LILA-Supermercado',
    'Address': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
    'Country': 'Venezuela'
  },
  {
    'CustomerID': 'LINOD',
    'ContactName': 'Felipe Izquierdo',
    'CompanyName': 'LINO-Delicateses',
    'Address': 'Ave. 5 de Mayo Porlamar',
    'Country': 'Venezuela'
  },
  {
    'CustomerID': 'LONEP',
    'ContactName': 'Fran Wilson',
    'CompanyName': 'Lonesome Pine Restaurant',
    'Address': '89 Chiaroscuro Rd.',
    'Country': 'USA'
  },
  {
    'CustomerID': 'MAGAA',
    'ContactName': 'Giovanni Rovelli',

```

```

    'CompanyName': 'Magazzini Alimentari Riuniti',
    'Address': 'Via Ludovico il Moro 22',
    'Country': 'Italy'
  },
  {
    'CustomerID': 'MAISD',
    'ContactName': 'Catherine Dewey',
    'CompanyName': 'Maison Dewey',
    'Address': 'Rue Joseph-Bens 532',
    'Country': 'Belgium'
  },
  {
    'CustomerID': 'MEREP',
    'ContactName': 'Jean Fresnière',
    'CompanyName': 'Mère Paillarde',
    'Address': '43 rue St. Laurent',
    'Country': 'Canada'
  },
  {
    'CustomerID': 'MORGK',
    'ContactName': 'Alexander Feuer',
    'CompanyName': 'Morgenstern Gesundkost',
    'Address': 'Heerstr. 22',
    'Country': 'Germany'
  },
  {
    'CustomerID': 'NORTS',
    'ContactName': 'Simon Crowther',
    'CompanyName': 'North/South',
    'Address': 'South House 300 Queensbridge',
    'Country': 'UK'
  },
  {
    'CustomerID': 'OCEAN',
    'ContactName': 'Yvonne Moncada',
    'CompanyName': 'Océano Atlántico Ltda.',
    'Address': 'Ing. Gustavo Moncada 8585 Piso 20-A',
    'Country': 'Argentina'
  },
  {
    'CustomerID': 'OLDWO',
    'ContactName': 'Rene Phillips',
    'CompanyName': 'Old World Delicatessen',
    'Address': '2743 Bering St.',
    'Country': 'USA'
  },
  {
    'CustomerID': 'OTTIK',
    'ContactName': 'Henriette Pfalzheim',
    'CompanyName': 'Ottilies Käseladen',
    'Address': 'Mehrheimerstr. 369',
    'Country': 'Germany'
  },
  {
    'CustomerID': 'PARIS',
    'ContactName': 'Marie Bertrand',
    'CompanyName': 'Paris spécialités',

```

```

    'Address': '265, boulevard Charonne',
    'Country': 'France'
  },
  {
    'CustomerID': 'PERIC',
    'ContactName': 'Guillermo Fernández',
    'CompanyName': 'Pericles Comidas clásicas',
    'Address': 'Calle Dr. Jorge Cash 321',
    'Country': 'Mexico'
  },
  {
    'CustomerID': 'PICCO',
    'ContactName': 'Georg Pipps',
    'CompanyName': 'Piccolo und mehr',
    'Address': 'Geislweg 14',
    'Country': 'Austria'
  },
  {
    'CustomerID': 'PRINI',
    'ContactName': 'Isabel de Castro',
    'CompanyName': 'Princesa Isabel Vinhos',
    'Address': 'Estrada da saúde n. 58',
    'Country': 'Portugal'
  },
  {
    'CustomerID': 'QUEDE',
    'ContactName': 'Bernardo Batista',
    'CompanyName': 'Que Delícia',
    'Address': 'Rua da Panificadora, 12',
    'Country': 'Brazil'
  },
  {
    'CustomerID': 'QUEEN',
    'ContactName': 'Lúcia Carvalho',
    'CompanyName': 'Queen Cozinha',
    'Address': 'Alameda dos Canários, 891',
    'Country': 'Brazil'
  },
  {
    'CustomerID': 'QUICK',
    'ContactName': 'Horst Kloss',
    'CompanyName': 'QUICK-Stop',
    'Address': 'Taucherstraße 10',
    'Country': 'Germany'
  },
  {
    'CustomerID': 'RANCH',
    'ContactName': 'Sergio Gutiérrez',
    'CompanyName': 'Rancho grande',
    'Address': 'Av. del Libertador 900',
    'Country': 'Argentina'
  },
  {
    'CustomerID': 'RATTC',
    'ContactName': 'Paula Wilson',
    'CompanyName': 'Rattlesnake Canyon Grocery',
    'Address': '2817 Milton Dr.',

```

```

    'Country': 'USA'
  },
  {
    'CustomerID': 'REGGC',
    'ContactName': 'Maurizio Moroni',
    'CompanyName': 'Reggiani Caseifici',
    'Address': 'Strada Provinciale 124',
    'Country': 'Italy'
  },
  {
    'CustomerID': 'RICAR',
    'ContactName': 'Janete Limeira',
    'CompanyName': 'Ricardo Adocicados',
    'Address': 'Av. Copacabana, 267',
    'Country': 'Brazil'
  },
  {
    'CustomerID': 'RICSU',
    'ContactName': 'Michael Holz',
    'CompanyName': 'Richter Supermarkt',
    'Address': 'Grenzacherweg 237',
    'Country': 'Switzerland'
  },
  {
    'CustomerID': 'ROMEY',
    'ContactName': 'Alejandra Camino',
    'CompanyName': 'Romero y tomillo',
    'Address': 'Gran Vía, 1',
    'Country': 'Spain'
  },
  {
    'CustomerID': 'SANTG',
    'ContactName': 'Jonas Bergulfsen',
    'CompanyName': 'Santé Gourmet',
    'Address': 'Erling Skakkes gate 78',
    'Country': 'Norway'
  },
  {
    'CustomerID': 'SAVEA',
    'ContactName': 'Jose Pavarotti',
    'CompanyName': 'Save-a-lot Markets',
    'Address': '187 Suffolk Ln.',
    'Country': 'USA'
  },
  {
    'CustomerID': 'SEVES',
    'ContactName': 'Hari Kumar',
    'CompanyName': 'Seven Seas Imports',
    'Address': '90 Wadhurst Rd.',
    'Country': 'UK'
  },
  {
    'CustomerID': 'SIMOB',
    'ContactName': 'Jytte Petersen',
    'CompanyName': 'Simons bistro',
    'Address': 'Vinbæltet 34',
    'Country': 'Denmark'
  }

```

```

    },
    {
      'CustomerID': 'SPEC'D',
      'ContactName': 'Dominique Perrier',
      'CompanyName': 'Spécialités du monde',
      'Address': '25, rue Lauriston',
      'Country': 'France'
    },
    {
      'CustomerID': 'SPLIR',
      'ContactName': 'Art Braunschweiger',
      'CompanyName': 'Split Rail Beer & Ale',
      'Address': 'P.O. Box 555',
      'Country': 'USA'
    },
    {
      'CustomerID': 'SUPRD',
      'ContactName': 'Pascale Cartrain',
      'CompanyName': 'Suprêmes délices',
      'Address': 'Boulevard Tirou, 255',
      'Country': 'Belgium'
    },
    {
      'CustomerID': 'THEBI',
      'ContactName': 'Liz Nixon',
      'CompanyName': 'The Big Cheese',
      'Address': '89 Jefferson Way Suite 2',
      'Country': 'USA'
    },
    {
      'CustomerID': 'THECR',
      'ContactName': 'Liu Wong',
      'CompanyName': 'The Cracker Box',
      'Address': '55 Grizzly Peak Rd.',
      'Country': 'USA'
    },
    {
      'CustomerID': 'TOMSP',
      'ContactName': 'Karin Josephs',
      'CompanyName': 'Toms Spezialitäten',
      'Address': 'Luisenstr. 48',
      'Country': 'Germany'
    },
    {
      'CustomerID': 'TORTU',
      'ContactName': 'Miguel Angel Paolino',
      'CompanyName': 'Tortuga Restaurante',
      'Address': 'Avda. Azteca 123',
      'Country': 'Mexico'
    },
    {
      'CustomerID': 'TRADH',
      'ContactName': 'Anabela Domingues',
      'CompanyName': 'Tradição Hipermercados',
      'Address': 'Av. Inês de Castro, 414',
      'Country': 'Brazil'
    },
  ],

```

```

{
  'CustomerID': 'TRAIH',
  'ContactName': 'Helvetius Nagy',
  'CompanyName': 'Trail\'s Head Gourmet Provisioners',
  'Address': '722 DaVinci Blvd.',
  'Country': 'USA'
},
{
  'CustomerID': 'VAFFE',
  'ContactName': 'Palle Ibsen',
  'CompanyName': 'Vaffeljernet',
  'Address': 'Smagsloget 45',
  'Country': 'Denmark'
},
{
  'CustomerID': 'VICTE',
  'ContactName': 'Mary Saveley',
  'CompanyName': 'Victuailles en stock',
  'Address': '2, rue du Commerce',
  'Country': 'France'
},
{
  'CustomerID': 'VINET',
  'ContactName': 'Paul Henriot',
  'CompanyName': 'Vins et alcools Chevalier',
  'Address': '59 rue de l\'Abbaye',
  'Country': 'France'
},
{
  'CustomerID': 'WANDK',
  'ContactName': 'Rita Müller',
  'CompanyName': 'Die Wandernde Kuh',
  'Address': 'Adenauerallee 900',
  'Country': 'Germany'
},
{
  'CustomerID': 'WARTH',
  'ContactName': 'Pirkko Koskitalo',
  'CompanyName': 'Wartian Herkku',
  'Address': 'Torikatu 38',
  'Country': 'Finland'
},
{
  'CustomerID': 'WELLI',
  'ContactName': 'Paula Parente',
  'CompanyName': 'Wellington Importadora',
  'Address': 'Rua do Mercado, 12',
  'Country': 'Brazil'
},
{
  'CustomerID': 'WHITC',
  'ContactName': 'Karl Jablonski',
  'CompanyName': 'White Clover Markets',
  'Address': '305 - 14th Ave. S. Suite 3B',
  'Country': 'USA'
},
{

```

```

        'CustomerID': 'WILMK',
        'ContactName': 'Matti Karttunen',
        'CompanyName': 'Wilman Kala',
        'Address': 'Keskuskatu 45',
        'Country': 'Finland'
    },
    {
        'CustomerID': 'WOLZA',
        'ContactName': 'Zbyszek Piestrzeniewicz',
        'CompanyName': 'Wolski Zajazd',
        'Address': 'ul. Filtrowa 68',
        'Country': 'Poland'
    }
];
export const data = orderData.map((item) => {
    let name = (<cType />)[0] > customerData;
}).filter((cItem) => {
    return cItem.CustomerID === item.CustomerID;
})[0];
item.CustomerName = (name || <cType>).ContactName;
return item;
});
export const inventoryData: Object[] = [
    { 'Inventor': 'Kia Silverbrook',
      'NumberofPatentFamilies': 4737,
      'Country': 'Australia',
      'Number of INPADOC patents': 9839,
      'Active': '1994-2016',
      'Mainfieldsofinvention': 'Printing, Digital paper, Internet,
Electronics, Lab-on-a-chip, MEMS, Mechanical, VLSI',
    },
    { 'Inventor': 'Shunpei Yamazaki',
      'NumberofPatentFamilies': 4677,
      'Country': 'Japan',
      'Number of INPADOC patents': '10000+',
      'Active': '1976-2016',
      'Mainfieldsofinvention': 'Thin film transistors, Liquid crystal
displays, Solar cells, Flash memory, OLED',
    },
    { 'Inventor': 'Lowell L. Wood, Jr.',
      'NumberofPatentFamilies': 1419,
      'Country': 'USA',
      'Number of INPADOC patents': 1332,
      'Active': '1977-2016',
      'Mainfieldsofinvention': 'Mosquito laser, Nuclear weapons',
    },
    { 'Inventor': 'Paul Lapstun',
      'NumberofPatentFamilies': 1281,
      'Country': 'Australia',
      'Number of INPADOC patents': 3099,
      'Active': '2000-2016',
      'Mainfieldsofinvention': 'Printing, Digital paper, Internet,
Electronics, CGI, VLSI',
    },
    { 'Inventor': 'Gurtej Sandhu',
      'NumberofPatentFamilies': 1255,
      'Country': 'India',

```

```

      'Number of INPADOC patents': 2038,
      'Active': '1991-2016',
      'Mainfieldsofinvention': 'Thin film processes and materials, VLSI,
Semiconductor device fabrication',
    },
    {'Inventor': 'Jun Koyama',
      'NumberofPatentFamilies': 1240,
      'Country': 'Japan',
      'Number of INPADOC patents': 4126,
      'Active': '1991-2016',
      'Mainfieldsofinvention': 'Thin film transistors, Liquid crystal
displays, OLED',
    },
    {'Inventor': 'Roderick A. Hyde',
      'NumberofPatentFamilies': 1240,
      'Country': 'USA',
      'Number of INPADOC patents': 3360,
      'Active': '2001-2016',
      'Mainfieldsofinvention': 'Various',
    },
    {'Inventor': 'Leonard Forbes',
      'NumberofPatentFamilies': 1093,
      'Country': 'Canada',
      'Number of INPADOC patents': 1398,
      'Active': '1991-2016',
      'Mainfieldsofinvention': 'Semiconductor Memories, CCDs, Thin film
processes and materials, VLSI',
    },
    {'Inventor': 'Thomas Edison',
      'NumberofPatentFamilies': 1084,
      'Country': 'USA',
      'Number of INPADOC patents': 2332,
      'Active': '1847(b)-1931(d)',
      'Mainfieldsofinvention': 'Electric power, Lighting, Batteries,
Phonograph, Cement, Telegraphy, Mining',
    },
    {'Inventor': 'Donald E. Weder',
      'NumberofPatentFamilies': 999,
      'Country': 'USA',
      'Number of INPADOC patents': 1993,
      'Active': '1976-2015',
      'Mainfieldsofinvention': 'Florist supplies',
    },
    {'Inventor': 'George Albert Lyon',
      'NumberofPatentFamilies': 993,
      'Country': 'Canada',
      'Number of INPADOC patents': 'NA',
      'Active': '1882(b)-1961(d)',
      'Mainfieldsofinvention': 'Automotive, Stainless steel products',
    },
    {'Inventor': 'John F. O\'Connor',
      'NumberofPatentFamilies': 949,
      'Country': 'USA',
      'Number of INPADOC patents': 'NA',
      'Active': '1864(b)-1938(d)',
      'Mainfieldsofinvention': 'Railway draft gearing',
    },
  ],

```



```

    {'Inventor': 'Melvin De Groote',
      'NumberofPatentFamilies': 925,
      'Country': 'USA',
      'Number of INPADOC patents': 'NA',
      'Active': '1895(b)-1963(d)',
      'Mainfieldsofinvention': 'Chemical de-emulsifiers',
    },
    {'Inventor': 'Jay S. Walker',
      'NumberofPatentFamilies': 918,
      'Country': 'USA',
      'Number of INPADOC patents': 2206,
      'Active': '1998-2016',
      'Mainfieldsofinvention': 'Gaming machines',
    },
    {'Inventor': 'Edward K. Y. Jung',
      'NumberofPatentFamilies': 911,
      'Country': 'USA',
      'Number of INPADOC patents': 2254,
      'Active': '1996-2016',
      'Mainfieldsofinvention': 'Various',
    },
    {'Inventor': 'Francis H. Richards',
      'NumberofPatentFamilies': 894,
      'Country': 'USA',
      'Number of INPADOC patents': 'NA',
      'Active': '1850(b)-19??(d)',
      'Mainfieldsofinvention': 'Mechanical, automation',
    },
    {'Inventor': 'Kangguo Cheng',
      'NumberofPatentFamilies': 884,
      'Country': 'USA',
      'Number of INPADOC patents': 1314,
      'Active': '2004-2016',
      'Mainfieldsofinvention': 'Semiconductor device fabrication,
Semiconductor memory, Semiconductor device',
    },
    {'Inventor': 'Clarence T. Tegreene',
      'NumberofPatentFamilies': 872,
      'Country': 'USA',
      'Number of INPADOC patents': 2255,
      'Active': '2000-2016',
      'Mainfieldsofinvention': 'Various',
    },
    {'Inventor': 'Ahmadreza Rofougaran',
      'NumberofPatentFamilies': 808,
      'Country': 'USA',
      'Number of INPADOC patents': 1396,
      'Active': '2002-2016',
      'Mainfieldsofinvention': 'Radio Frequency Integrated Circuits',
    },
    {'Inventor': 'Shou-Shan Fan',
      'NumberofPatentFamilies': 805,
      'Country': 'China',
      'Number of INPADOC patents': 2120,
      'Active': '2006-2016',
      'Mainfieldsofinvention': 'Carbon nanotubes and applications of
carbon nanotubes',
  }

```

```

    },
    { 'Inventor': 'Michael J. Sullivan',
      'NumberofPatentFamilies': 788,
      'Country': 'USA',
      'Number of INPADOC patents': 1560,
      'Active': '1977-2016',
      'Mainfieldsofinvention': 'Golf balls',
    },
    { 'Inventor': 'Rick Allen Hamilton II',
      'NumberofPatentFamilies': 773,
      'Country': 'USA',
      'Number of INPADOC patents': 1064,
      'Active': '1999-2016',
      'Mainfieldsofinvention': 'Various',
    },
    { 'Inventor': 'Warren Farnworth',
      'NumberofPatentFamilies': 770,
      'Country': 'USA',
      'Number of INPADOC patents': 931,
      'Active': '1990-2016',
      'Mainfieldsofinvention': 'Semiconductor packaging',
    },
    { 'Inventor': 'Carleton Ellis',
      'NumberofPatentFamilies': 753,
      'Country': 'USA',
      'Number of INPADOC patents': 'NA',
      'Active': '1876(b)-1941(d)',
      'Mainfieldsofinvention': 'Margarine, Polyester, Anti-knock gasoline,
Paint stripper',
    },
    { 'Inventor': 'William H. Eby',
      'NumberofPatentFamilies': 733,
      'Country': 'USA',
      'Number of INPADOC patents': 758,
      'Active': '1994-2016',
      'Mainfieldsofinvention': 'Transgenic soybeans',
    },
    { 'Inventor': 'Hideo Ando',
      'NumberofPatentFamilies': 728,
      'Country': 'Japan',
      'Number of INPADOC patents': 2588,
      'Active': '1983-2016',
      'Mainfieldsofinvention': 'Optical recording',
    },
    { 'Inventor': 'Salman Akram',
      'NumberofPatentFamilies': 728,
      'Country': 'USA',
      'Number of INPADOC patents': 915,
      'Active': '1995-2016',
      'Mainfieldsofinvention': 'Semiconductor packaging',
    },
    { 'Inventor': 'George Spector',
      'NumberofPatentFamilies': 722,
      'Country': 'USA',
      'Number of INPADOC patents': 747,
      'Active': '1976-1998',
      'Mainfieldsofinvention': 'Gadgets, Toys',
    },

```

```

    },
    { 'Inventor': 'Jeyhan Karaoguz',
      'NumberofPatentFamilies': 721,
      'Country': 'USA',
      'Number of INPADOC patents': 1530,
      'Active': '1996-2016',
      'Mainfieldsofinvention': 'Wireless communications, Computer
networks',
    },
    { 'Inventor': 'Elihu Thomson',
      'NumberofPatentFamilies': 696,
      'Country': 'UK',
      'Number of INPADOC patents': 'NA',
      'Active': '1853(b)-1937(d)',
      'Mainfieldsofinvention': 'Electric power, Arc lamp, Electric motors,
Lightning arrester, Arc welder',
    },
    { 'Inventor': 'Austin L. Gurney',
      'NumberofPatentFamilies': 695,
      'Country': 'USA',
      'Number of INPADOC patents': 3909,
      'Active': '1999-2016',
      'Mainfieldsofinvention': 'Proteins, Antibodies',
    },
    { 'Inventor': 'Tetsujiro Kondo',
      'NumberofPatentFamilies': 684,
      'Country': 'Japan',
      'Number of INPADOC patents': 4158,
      'Active': '1987-2015',
      'Mainfieldsofinvention': 'Signal processing, Image processing',
    },
    { 'Inventor': 'Nathan Myhrvold',
      'NumberofPatentFamilies': 661,
      'Country': 'USA',
      'Number of INPADOC patents': 1690,
      'Active': '1994-2016',
      'Mainfieldsofinvention': 'Various',
    },
    { 'Inventor': 'William I. Wood',
      'NumberofPatentFamilies': 653,
      'Country': 'USA',
      'Number of INPADOC patents': 3560,
      'Active': '1981-2016',
      'Mainfieldsofinvention': 'Proteins, Antibodies',
    },
    { 'Inventor': 'Simon R. Walmsley',
      'NumberofPatentFamilies': 651,
      'Country': 'Australia',
      'Number of INPADOC patents': 1249,
      'Active': '1995-2015',
      'Mainfieldsofinvention': 'Printing, Electronics, VLSI,
Cryptography',
    },
    { 'Inventor': 'Mark Malamud',
      'NumberofPatentFamilies': 632,
      'Country': 'USA',
      'Number of INPADOC patents': 1759,

```

```

      'Active': '1997-2016',
      'Mainfieldsofinvention': 'Various',
    },
    { 'Inventor': 'Royce A. Levien',
      'NumberofPatentFamilies': 630,
      'Country': 'USA',
      'Number of INPADOC patents': 1799,
      'Active': '1997-2016',
      'Mainfieldsofinvention': 'Various',
    },
    { 'Inventor': 'Audrey D. Goddard',
      'NumberofPatentFamilies': 622,
      'Country': 'USA',
      'Number of INPADOC patents': 3416,
      'Active': '1997-2014',
      'Mainfieldsofinvention': 'Proteins, Antibodies',
    },
    { 'Inventor': 'Muriel Y. Ishikawa',
      'NumberofPatentFamilies': 619,
      'Country': 'USA',
      'Number of INPADOC patents': 1660,
      'Active': '2002-2016',
      'Mainfieldsofinvention': 'Various',
    },
    { 'Inventor': 'Robert W. Lord',
      'NumberofPatentFamilies': 618,
      'Country': 'USA',
      'Number of INPADOC patents': 1708,
      'Active': '2003-2016',
      'Mainfieldsofinvention': 'Various',
    },
    { 'Inventor': 'Jerome Lemelson',
      'NumberofPatentFamilies': 606,
      'Country': 'USA',
      'Number of INPADOC patents': 'NA',
      'Active': '1923(b)-1997(d)',
      'Mainfieldsofinvention': 'Toys, Industrial robots, Cordless
telephones, Fax machines, Videocassette recorders',
    },
    { 'Inventor': 'Béla Barényi',
      'NumberofPatentFamilies': 595,
      'Country': 'Austria',
      'Number of INPADOC patents': 1244,
      'Active': '1907(b)-1997(d)',
      'Mainfieldsofinvention': 'Passive safety in automobiles',
    },
    { 'Inventor': 'Kie Y Ahn',
      'NumberofPatentFamilies': 593,
      'Country': 'USA',
      'Number of INPADOC patents': 709,
      'Active': '1976-2016',
      'Mainfieldsofinvention': 'Thin film processes and materials, VLSI,
Semiconductor device fabrication',
    },
    { 'Inventor': 'Tadahiro Ohmi',
      'NumberofPatentFamilies': 592,
      'Country': 'Japan',

```

```

      'Number of INPADOC patents': 2691,
      'Active': '1981-2016',
      'Mainfieldsofinvention': 'Thin film processes and materials,
Semiconductor device fabrication',
    },
    { 'Inventor': 'Jordin T. Kare',
      'NumberofPatentFamilies': 585,
      'Country': 'USA',
      'Number of INPADOC patents': 1559,
      'Active': '1992-2016',
      'Mainfieldsofinvention': 'Various',
    },
    { 'Inventor': 'Paul J. Godowski',
      'NumberofPatentFamilies': 579,
      'Country': 'USA',
      'Number of INPADOC patents': 2605,
      'Active': '1994-2014',
      'Mainfieldsofinvention': 'Proteins, Antibodies',
    },
    { 'Inventor': 'Artur Fischer',
      'NumberofPatentFamilies': 570,
      'Country': 'Germany',
      'Number of INPADOC patents': 3097,
      'Active': '1976-2002',
      'Mainfieldsofinvention': 'Fasteners, Construction toys',
    },
    { 'Inventor': 'Edward J. Nowak',
      'NumberofPatentFamilies': 564,
      'Country': 'USA',
      'Number of INPADOC patents': 1145,
      'Active': '1979-2016',
      'Mainfieldsofinvention': 'Semiconductor device fabrication,
Semiconductor memory, Semiconductor device',
    },
    { 'Inventor': 'Louis L. Hsu',
      'NumberofPatentFamilies': 551,
      'Country': 'USA',
      'Number of INPADOC patents': 914,
      'Active': '1988-2016',
      'Mainfieldsofinvention': 'Various',
    },
    { 'Inventor': 'Edwin H. Land',
      'NumberofPatentFamilies': 535,
      'Country': 'USA',
      'Number of INPADOC patents': 1236,
      'Active': '1909(b)-1991(d)',
      'Mainfieldsofinvention': 'Instant photography, Polarizing film',
    },
    { 'Inventor': 'Henri Dreyfus',
      'NumberofPatentFamilies': 524,
      'Country': 'Switzerland',
      'Number of INPADOC patents': 2117,
      'Active': '1882(b)-1944(d)',
      'Mainfieldsofinvention': 'Polymers, Synthetic fibers, Dyes',
    },
    { 'Inventor': 'Bruce B. Doris',
      'NumberofPatentFamilies': 522,

```

```

    'Country': 'USA',
    'Number of INPADOC patents': 867,
    'Active': '1995-2016',
    'Mainfieldsofinvention': 'Integrated Circuits, CMOS, DRAM,
Semiconductor device fabrication',
  },
  {'Inventor': 'Clyde C. Farmer',
    'NumberofPatentFamilies': 513,
    'Country': 'USA',
    'Number of INPADOC patents': 830,
    'Active': '18??(b)-19??(d)',
    'Mainfieldsofinvention': 'Railway air brakes',
  },
  {'Inventor': 'Heinz Focke',
    'NumberofPatentFamilies': 512,
    'Country': 'Germany',
    'Number of INPADOC patents': 2896,
    'Active': '1976-2013',
    'Mainfieldsofinvention': 'Cigarette packaging',
  },
  {'Inventor': 'Mark I. Gardner',
    'NumberofPatentFamilies': 511,
    'Country': 'USA',
    'Number of INPADOC patents': 587,
    'Active': '1994-2010',
    'Mainfieldsofinvention': 'Consumer electronics, Energy, Computers,
Semiconductors, Physics',
  },
  {'Inventor': 'Ravi K. Arimilli',
    'NumberofPatentFamilies': 506,
    'Country': 'India',
    'Number of INPADOC patents': 767,
    'Active': '1992-2016',
    'Mainfieldsofinvention': 'Computer architecture, Semiconductor
memory, Cache coherence, Symmetric multiprocessing',
  },
  {'Inventor': 'Louis H. Morin',
    'NumberofPatentFamilies': 503,
    'Country': 'USA',
    'Number of INPADOC patents': 720,
    'Active': '18??(b)-19??(d)',
    'Mainfieldsofinvention': 'Fasteners, Locks, Bobbins',
  },
  {'Inventor': 'Tobin A. King',
    'NumberofPatentFamilies': 497,
    'Country': 'Australia',
    'Number of INPADOC patents': 1218,
    'Active': '2000-2015',
    'Mainfieldsofinvention': 'Printing, Digital paper, Mechanical',
  },
  {'Inventor': 'Eric C. Leuthardt',
    'NumberofPatentFamilies': 495,
    'Country': 'USA',
    'Number of INPADOC patents': 1274,
    'Active': '2006-2016',
    'Mainfieldsofinvention': 'Medical devices',
  },
},

```

```

    {'Inventor': 'Ali Khakifirooz',
      'NumberofPatentFamilies': 489,
      'Country': 'USA',
      'Number of INPADOC patents': 737,
      'Active': '2011-2016',
      'Mainfieldsofinvention': 'Integrated Circuits, CMOS, Semiconductor
device fabrication',
    },
    {'Inventor': 'Jack A. Mandelman',
      'NumberofPatentFamilies': 481,
      'Country': 'USA',
      'Number of INPADOC patents': 889,
      'Active': '1987-2014',
      'Mainfieldsofinvention': 'Various',
    },
    {'Inventor': 'Jeffrey P. Gambino',
      'NumberofPatentFamilies': 479,
      'Country': 'USA',
      'Number of INPADOC patents': 798,
      'Active': '1992-2016',
      'Mainfieldsofinvention': 'MEMS, CMOS, BiCMOS, DRAM, Image Sensors,
RF, Biosensors, 3D Integrated Circuits',
    },
    {'Inventor': 'John M. Santosuosso',
      'NumberofPatentFamilies': 473,
      'Country': 'USA',
      'Number of INPADOC patents': 683,
      'Active': '2001-2016',
      'Mainfieldsofinvention': 'Various',
    },
    {'Inventor': 'James M. Hart',
      'NumberofPatentFamilies': 464,
      'Country': 'USA',
      'Number of INPADOC patents': 1145,
      'Active': '1988-2016',
      'Mainfieldsofinvention': 'Motor vehicle transmission',
    },
    {'Inventor': 'Eberhard Ammermann',
      'NumberofPatentFamilies': 451,
      'Country': 'Germany',
      'Number of INPADOC patents': 5178,
      'Active': '1979-2015',
      'Mainfieldsofinvention': 'Fungicides',
    },
    {'Inventor': 'Thomas E. Murray',
      'NumberofPatentFamilies': 449,
      'Country': 'USA',
      'Number of INPADOC patents': 462,
      'Active': '1860(b)-1929(d)',
      'Mainfieldsofinvention': 'Electrical, HVAC, Wheels, Metal working,
Light dimmer',
    },
    {'Inventor': 'Akira Nakazawa',
      'NumberofPatentFamilies': 445,
      'Country': 'Australia',
      'Number of INPADOC patents': 1340,
      'Active': '1980-2016',

```

```

    'Mainfieldsofinvention': 'Printing, Mechanical',
  },
  { 'Inventor': 'Hongyong Zhang',
    'NumberofPatentFamilies': 440,
    'Country': 'Japan',
    'Number of INPADOC patents': 858,
    'Active': '1993-2016',
    'Mainfieldsofinvention': 'Thin film transistors, Liquid crystal
displays',
  },
  { 'Inventor': 'Ronald S. Cok',
    'NumberofPatentFamilies': 436,
    'Country': 'USA',
    'Number of INPADOC patents': 747,
    'Active': '1986-2016',
    'Mainfieldsofinvention': 'OLED displays; image processing',
  },
  { 'Inventor': 'fe',
    'NumberofPatentFamilies': 430,
    'Country': 'USA',
    'Number of INPADOC patents': 1759,
    'Active': '1983-2016',
    'Mainfieldsofinvention': 'Biotechnology, Drug delivery, Tissue
engineering',
  },
  { 'Inventor': 'Scott H. Wittkopp',
    'NumberofPatentFamilies': 429,
    'Country': 'USA',
    'Number of INPADOC patents': 1010,
    'Active': '2001-2016',
    'Mainfieldsofinvention': 'Motor vehicle transmission',
  },
  { 'Inventor': 'John Hays Hammond, Jr.',
    'NumberofPatentFamilies': 417,
    'Country': 'USA',
    'Number of INPADOC patents': 460,
    'Active': '1888(b)-1965(d)',
    'Mainfieldsofinvention': 'Radio control, Radio communications,
Torpedoes',
  },
  { 'Inventor': 'Wilhelm Brandes',
    'NumberofPatentFamilies': 411,
    'Country': 'Germany',
    'Number of INPADOC patents': 2923,
    'Active': '1976-2010',
    'Mainfieldsofinvention': 'Fungicides',
  },
  { 'Inventor': 'Anthony K. Stamper',
    'NumberofPatentFamilies': 411,
    'Country': 'USA',
    'Number of INPADOC patents': 726,
    'Active': '1998-2016',
    'Mainfieldsofinvention': 'MEMS, CMOS, BiCMOS, Silicon-germanium',
  },
  { 'Inventor': 'Hossein Eslambolchi',
    'NumberofPatentFamilies': 410,
    'Country': 'USA',

```



```

      'Number of INPADOC patents': 631,
      'Active': '1993-2016',
      'Mainfieldsofinvention': 'Telecommunications, Network intelligence,
information Technology, communications technology',
    },
    { 'Inventor': 'Stanford R. Ovshinsky',
      'NumberofPatentFamilies': 400,
      'Country': 'USA',
      'Number of INPADOC patents': 1649,
      'Active': '1922 (b)-2012 (d)',
      'Mainfieldsofinvention': 'Batteries, Solar cells, Liquid crystal
displays, Hydrogen fuel cells, Computer data storage',
    },
    { 'Inventor': 'Victoria Y. H. Wood',
      'NumberofPatentFamilies': 400,
      'Country': 'USA',
      'Number of INPADOC patents': 1045,
      'Active': '2009-2016',
      'Mainfieldsofinvention': 'Various',
    },
    { 'Inventor': 'Josef Theurer',
      'NumberofPatentFamilies': 388,
      'Country': 'Austria',
      'Number of INPADOC patents': 5085,
      'Active': '1976-2016',
      'Mainfieldsofinvention': 'Railroad maintenance machines',
    },
    { 'Inventor': 'Cary L. Bates',
      'NumberofPatentFamilies': 384,
      'Country': 'USA',
      'Number of INPADOC patents': 570,
      'Active': '1994-2016',
      'Mainfieldsofinvention': 'Programming tools, DBX, Memory debuggers',
    },
    { 'Inventor': 'David V. Horak',
      'NumberofPatentFamilies': 380,
      'Country': 'USA',
      'Number of INPADOC patents': 616,
      'Active': '1992-2016',
      'Mainfieldsofinvention': 'Various',
    },
    { 'Inventor': 'Kai-Li Jiang',
      'NumberofPatentFamilies': 379,
      'Country': 'China',
      'Number of INPADOC patents': 829,
      'Active': '2006-2016',
      'Mainfieldsofinvention': 'Carbon nanotubes and applications of
carbon nanotubes',
    },
    { 'Inventor': 'Hans-Joachim Santel',
      'NumberofPatentFamilies': 377,
      'Country': 'Germany',
      'Number of INPADOC patents': 2623,
      'Active': '1986-2013',
      'Mainfieldsofinvention': 'Herbicides, Pesticides, Organic
chemistry',
    },
  ],

```

```

    {'Inventor': 'Xuemin (Sherman) Chen',
      'NumberofPatentFamilies': 377,
      'Country': 'USA',
      'Number of INPADOC patents': 1151,
      'Active': '1997-2017',
      'Mainfieldsofinvention': 'Computer networks, Integrated Circuits,
Signal Processing',
    },
    {'Inventor': 'George P. Liang',
      'NumberofPatentFamilies': 375,
      'Country': 'China',
      'Number of INPADOC patents': 508,
      'Active': '1983-2016',
      'Mainfieldsofinvention': 'Gas turbine cooling',
    },
    {'Inventor': 'Gisela Lorenz',
      'NumberofPatentFamilies': 374,
      'Country': 'Germany',
      'Number of INPADOC patents': 4155,
      'Active': '1990-2015',
      'Mainfieldsofinvention': 'Fungicides, Organic chemistry',
    },
    {'Inventor': 'Garry R. Jackson',
      'NumberofPatentFamilies': 367,
      'Country': 'Australia',
      'Number of INPADOC patents': 656,
      'Active': '2001-2016',
      'Mainfieldsofinvention': 'Printing, Mechanical',
    },
    {'Inventor': 'Paul W. Dent',
      'NumberofPatentFamilies': 362,
      'Country': 'USA',
      'Number of INPADOC patents': 2252,
      'Active': '1984-2015',
      'Mainfieldsofinvention': 'Wireless communications',
    },
    {'Inventor': 'George Westinghouse',
      'NumberofPatentFamilies': 361,
      'Country': 'USA',
      'Number of INPADOC patents': 'NA',
      'Active': '1846(b)-1914(d)',
      'Mainfieldsofinvention': 'Electric power, Electricity meter, Railway
air brake, Steam engines',
    },
    {'Inventor': 'Wael W. Diab',
      'NumberofPatentFamilies': 358,
      'Country': 'USA',
      'Number of INPADOC patents': 774,
      'Active': '2003-2016',
      'Mainfieldsofinvention': 'Computer networks',
    },
    {'Inventor': 'Devendra K. Sadana',
      'NumberofPatentFamilies': 356,
      'Country': 'India',
      'Number of INPADOC patents': 794,
      'Active': '1983-2016',
    },

```

```

    'Mainfieldsofinvention': 'Solar cells, OLED, Integrated Circuits,
CMOS, DRAM, LEDs',
  },
  { 'Inventor': 'Vincent J. Zimmer',
    'NumberofPatentFamilies': 354,
    'Country': 'USA',
    'Number of INPADOC patents': 972,
    'Active': '1999-2016',
    'Mainfieldsofinvention': 'Computer software and firmware',
  },
  { 'Inventor': 'Robert R. Schmidt',
    'NumberofPatentFamilies': 350,
    'Country': 'Germany',
    'Number of INPADOC patents': 2467,
    'Active': '1971-2005',
    'Mainfieldsofinvention': 'Herbicides, Fungicides, Organic
chemistry',
  },
  { 'Inventor': 'Norman M. Berry',
    'NumberofPatentFamilies': 347,
    'Country': 'Australia',
    'Number of INPADOC patents': 516,
    'Active': '2006-2016',
    'Mainfieldsofinvention': 'Printing, Mechanical',
  },
  { 'Inventor': 'Chih-Chao Yang',
    'NumberofPatentFamilies': 345,
    'Country': 'USA',
    'Number of INPADOC patents': 690,
    'Active': '2003-2016',
    'Mainfieldsofinvention': 'Integrated Circuits',
  },
  { 'Inventor': 'Gregory J. Boss',
    'NumberofPatentFamilies': 345,
    'Country': 'USA',
    'Number of INPADOC patents': 588,
    'Active': '2008-2016',
    'Mainfieldsofinvention': 'Various'
  },
  { 'Inventor': 'Mark W. Kroll',
    'NumberofPatentFamilies': 343,
    'Country': 'USA',
    'Number of INPADOC patents': 460,
    'Active': '1987-2016',
    'Mainfieldsofinvention': 'Implantable medical devices',
  },
  { 'Inventor': 'Brian M. O\'Connell',
    'NumberofPatentFamilies': 331,
    'Country': 'USA',
    'Number of INPADOC patents': 592,
    'Active': '2009-2016',
    'Mainfieldsofinvention': 'Various',
  },
  { 'Inventor': 'William Daniel Hillis',
    'NumberofPatentFamilies': 328,
    'Country': 'USA',
    'Number of INPADOC patents': 229,
  },

```

```

      'Active': '1986-2016',
      'Mainfieldsofinvention': 'Various',
    },
    { 'Inventor': 'Brent A. Anderson',
      'NumberofPatentFamilies': 323,
      'Country': 'USA',
      'Number of INPADOC patents': 454,
      'Active': '2001-2016',
      'Mainfieldsofinvention': 'Semiconductor device fabrication,
Semiconductor memory, Semiconductor device',
    },
    { 'Inventor': 'Jeffrey E. Stahmann',
      'NumberofPatentFamilies': 321,
      'Country': 'USA',
      'Number of INPADOC patents': 640,
      'Active': '1994-2016',
      'Mainfieldsofinvention': 'Medical devices',
    },
    { 'Inventor': 'Carl J. Radens',
      'NumberofPatentFamilies': 317,
      'Country': 'USA',
      'Number of INPADOC patents': 636,
      'Active': '1994-2016',
      'Mainfieldsofinvention': 'Integrated Circuits, CMOS, DRAM,
Semiconductor device fabrication',
    },
    { 'Inventor': 'Clifford A. Pickover',
      'NumberofPatentFamilies': 317,
      'Country': 'USA',
      'Number of INPADOC patents': 653,
      'Active': '1992-2016',
      'Mainfieldsofinvention': 'Various',
    },
    { 'Inventor': 'Liang Liu',
      'NumberofPatentFamilies': 310,
      'Country': 'China',
      'Number of INPADOC patents': 777,
      'Active': '2005-2016',
      'Mainfieldsofinvention': 'Carbon nanotubes and applications of
carbon nanotubes',
    },
    { 'Inventor': 'Steven L. Teig',
      'NumberofPatentFamilies': 307,
      'Country': 'USA',
      'Number of INPADOC patents': 366,
      'Active': '1995-2016',
      'Mainfieldsofinvention': 'Integrated Circuits',
    },
    { 'Inventor': 'Victoria Smith',
      'NumberofPatentFamilies': 305,
      'Country': 'USA',
      'Number of INPADOC patents': 2040,
      'Active': '2006-2016',
      'Mainfieldsofinvention': 'Proteins, Antibodies',
    },
    { 'Inventor': 'Robert G. LeTourneau',
      'NumberofPatentFamilies': 299,

```

```

    'Country': 'USA',
    'Number of INPADOC patents': 'NA',
    'Active': '1888(b)-1969(d)',
    'Mainfieldsofinvention': 'Earthworks (engineering), Heavy Equipment,
Industrial Machinery',
  },
  {'Inventor': 'William R. Tonti',
    'NumberofPatentFamilies': 291,
    'Country': 'USA',
    'Number of INPADOC patents': 441,
    'Active': '1994-2016',
    'Mainfieldsofinvention': 'Integrated Circuits, CMOS, DRAM,
Semiconductor device fabrication',
  },
  {'Inventor': 'Keith R. Walker',
    'NumberofPatentFamilies': 282,
    'Country': 'Saudi Arabia',
    'Number of INPADOC patents': 318,
    'Active': '2003-2016',
    'Mainfieldsofinvention': 'Various',
  },
  {'Inventor': 'Victor S. Moore',
    'NumberofPatentFamilies': 280,
    'Country': 'USA',
    'Number of INPADOC patents': 428,
    'Active': '1982-2016',
    'Mainfieldsofinvention': 'Various',
  },
  {'Inventor': 'Hanson S. Gifford III',
    'NumberofPatentFamilies': 276,
    'Country': 'USA',
    'Number of INPADOC patents': 795,
    'Active': '1987-2016',
    'Mainfieldsofinvention': 'Medical Devices',
  },
  {'Inventor': 'Daniel J. Winarski',
    'NumberofPatentFamilies': 275,
    'Country': 'USA',
    'Number of INPADOC patents': 506,
    'Active': '1982-2016',
    'Mainfieldsofinvention': 'Various',
  },
  {'Inventor': 'Adam Heller',
    'NumberofPatentFamilies': 272,
    'Country': 'Romania',
    'Number of INPADOC patents': 711,
    'Active': '1968-2016',
    'Mainfieldsofinvention': 'Solar cells, Glucose meters, Lasers',
  },
  {'Inventor': 'Lisa Seacat DeLuca',
    'NumberofPatentFamilies': 271,
    'Country': 'USA',
    'Number of INPADOC patents': 385,
    'Active': '2009-2016',
    'Mainfieldsofinvention': 'Various',
  },
  {'Inventor': 'Brent Keeth',

```

```

      'NumberofPatentFamilies': 270,
      'Country': 'USA',
      'Number of INPADOC patents': 470,
      'Active': '1994-2016',
      'Mainfieldsofinvention': 'Integrated Circuits, CMOS, DRAM',
    },
    {'Inventor': 'Hartley Owen',
      'NumberofPatentFamilies': 267,
      'Country': 'USA',
      'Number of INPADOC patents': 751,
      'Active': '1976-2010',
      'Mainfieldsofinvention': 'Fluid catalytic cracking',
    },
    {'Inventor': 'Michael A. Rothman',
      'NumberofPatentFamilies': 256,
      'Country': 'USA',
      'Number of INPADOC patents': 687,
      'Active': '2001-2017',
      'Mainfieldsofinvention': 'Computer software and firmware',
    },
    {'Inventor': 'Yoshihiro Kikuchi',
      'NumberofPatentFamilies': 255,
      'Country': 'Japan',
      'Number of INPADOC patents': 1120,
      'Active': '1994-2015',
      'Mainfieldsofinvention': 'Video processing',
    },
    {'Inventor': 'Kulvir S. Bhogal',
      'NumberofPatentFamilies': 252,
      'Country': 'USA',
      'Number of INPADOC patents': 486,
      'Active': '2003-2016',
      'Mainfieldsofinvention': 'Various',
    },
    {'Inventor': 'Bengt Lindoff',
      'NumberofPatentFamilies': 248,
      'Country': 'Sweden',
      'Number of INPADOC patents': 1658,
      'Active': '2000-2017',
      'Mainfieldsofinvention': 'Wireless communications',
    },
    {'Inventor': 'Nobuyuki Taniguchi',
      'NumberofPatentFamilies': 245,
      'Country': 'Japan',
      'Number of INPADOC patents': 967,
      'Active': '1979-2015',
      'Mainfieldsofinvention': 'Cameras',
    },
    {'Inventor': 'Dean L. Kamen',
      'NumberofPatentFamilies': 243,
      'Country': 'USA',
      'Number of INPADOC patents': 1186,
      'Active': '1979-2016',
      'Mainfieldsofinvention': 'Battery-powered electric vehicles, Medical
devices, Stirling engines, Water purification, Wheelchairs',
    },
    {'Inventor': 'Philip S. Yu',

```

```

      'NumberofPatentFamilies': 236,
      'Country': 'USA',
      'Number of INPADOC patents': 158,
      'Active': '1982-2016',
      'Mainfieldsofinvention': 'Various',
    },
    {'Inventor': 'Rajiv V. Joshi',
      'NumberofPatentFamilies': 235,
      'Country': 'USA',
      'Number of INPADOC patents': 354,
      'Active': '1986-2016',
      'Mainfieldsofinvention': 'Electronics, analytics',
    },
    {'Inventor': 'Lawrence A. Clevenger',
      'NumberofPatentFamilies': 235,
      'Country': 'USA',
      'Number of INPADOC patents': 526,
      'Active': '1996-2017',
      'Mainfieldsofinvention': 'Semiconductor, Cognitive, Memory,
Security, Analytics',
    },
    {'Inventor': 'Johnny M. Shieh',
      'NumberofPatentFamilies': 231,
      'Country': 'USA',
      'Number of INPADOC patents': 444,
      'Active': '1996-2016',
      'Mainfieldsofinvention': 'Various',
    },
    {'Inventor': 'Takeshi Chujoh',
      'NumberofPatentFamilies': 229,
      'Country': 'Japan',
      'Number of INPADOC patents': 1065,
      'Active': '1995-2016',
      'Mainfieldsofinvention': 'Video processing',
    },
    {'Inventor': 'Bran Ferren',
      'NumberofPatentFamilies': 225,
      'Country': 'USA',
      'Number of INPADOC patents': 589,
      'Active': '1986-2017',
      'Mainfieldsofinvention': 'Computers, Consumer Electronics, Optical
Systems, Medical, User Interfaces, Automotive',
    },
    {'Inventor': 'Paul Ian Mackey',
      'NumberofPatentFamilies': 220,
      'Country': 'Australia',
      'Number of INPADOC patents': 246,
      'Active': '2008-2016',
      'Mainfieldsofinvention': 'Printing, Mechanical',
    },
    {'Inventor': 'Louis Rosenberg',
      'NumberofPatentFamilies': 218,
      'Country': 'USA',
      'Number of INPADOC patents': 444,
      'Active': '1995-2016',
      'Mainfieldsofinvention': 'Augmented Reality, Virtual Reality, A.I.,
HCI',

```

```

    },
    { 'Inventor': 'Thomas J. Kennedy III',
      'NumberofPatentFamilies': 218,
      'Country': 'USA',
      'Number of INPADOC patents': 513,
      'Active': '1992-2016',
      'Mainfieldsofinvention': 'Sporting Goods, Wind Turbines',
    },
    { 'Inventor': 'Gerald F. McBrearty',
      'NumberofPatentFamilies': 213,
      'Country': 'USA',
      'Number of INPADOC patents': 387,
      'Active': '1997-2016',
      'Mainfieldsofinvention': 'Various',
    },
    { 'Inventor': 'Esmael H. Dinan',
      'NumberofPatentFamilies': 208,
      'Country': 'USA',
      'Number of INPADOC patents': 344,
      'Active': '2000-2017',
      'Mainfieldsofinvention': 'Communication Networks',
    },
    { 'Inventor': 'Imad Libbus',
      'NumberofPatentFamilies': 207,
      'Country': 'USA',
      'Number of INPADOC patents': 472,
      'Active': '2007-2017',
      'Mainfieldsofinvention': 'Medical devices',
    },
    { 'Inventor': 'Hiroshi (You) Yoshioka',
      'NumberofPatentFamilies': 205,
      'Country': 'Japan',
      'Number of INPADOC patents': 181,
      'Active': '1997-2015',
      'Mainfieldsofinvention': 'Cameras',
    },
    { 'Inventor': 'Patrick B. Usoro',
      'NumberofPatentFamilies': 205,
      'Country': 'USA',
      'Number of INPADOC patents': 343,
      'Active': '1999-2016',
      'Mainfieldsofinvention': 'Transmissions, Hybrid Powertrains, Vehicle Thermal Management',
    },
    { 'Inventor': 'Gregory McAvoy',
      'NumberofPatentFamilies': 205,
      'Country': 'Australia',
      'Number of INPADOC patents': 433,
      'Active': '2003-2014',
      'Mainfieldsofinvention': 'Printing, MEMS',
    },
    { 'Inventor': 'Sebastian T Ventrone',
      'NumberofPatentFamilies': 204,
      'Country': 'USA',
      'Number of INPADOC patents': 283,
      'Active': '1989-2017',
      'Mainfieldsofinvention': 'Semiconductor, Logic, Architecture',
    },

```



```

    },
    {'Inventor': 'Dorin Comaniciu',
     'NumberofPatentFamilies': 200,
     'Country': 'USA',
     'Number of INPADOC patents': 452,
     'Active': '2003-2017',
     'Mainfieldsofinvention': 'Machine Intelligence, Medical Imaging,
Image-Guided Surgery, Computer Vision',
    }
  ];
export interface ColumnSpanDataType {EmployeeID}: number;
  EmployeeName: string;
  '9:00': string;
  '9:30': string;
  '10:00': string;
  '10:30': string;
  '11:00': string;
  '11:30': string;
  '12:00': string;
  '12:30': string;
  '1:00': string;
  '1:30': string;
  '2:00': string;
  '2:30': string;
  '3:00': string;
  '3:30': string;
  '4:00': string;
  '4:30': string;
  '5:00': string;
}
export let columnSpanData: ColumnSpanDataType[] = [
  {EmployeeID}: 10001,
    EmployeeName: 'Davolio',
    '9:00': 'Analysis Tasks',
    '9:30': 'Analysis Tasks',
    '10:00': 'Team Meeting',
    '10:30': 'Testing',
    '11:00': 'Development',
    '11:30': 'Development',
    '12:00': 'Development',
    '12:30': 'Support',
    '1:00': 'Lunch Break',
    '1:30': 'Lunch Break',
    '2:00': 'Lunch Break',
    '2:30': 'Testing',
    '3:00': 'Testing',
    '3:30': 'Development',
    '4:00': 'Conference',
    '4:30': 'Team Meeting',
    '5:00': 'Team Meeting'
  },
  {EmployeeID}: 10002,
    EmployeeName: 'Buchanan',
    '9:00': 'Task Assign',
    '9:30': 'Support',
    '10:00': 'Support',
    '10:30': 'Support',

```

```

        '11:00': 'Testing',
        '11:30': 'Testing',
        '12:00': 'Testing',
        '12:30': 'Testing',
        '1:00': 'Lunch Break',
        '1:30': 'Lunch Break',
        '2:00': 'Lunch Break',
        '2:30': 'Development',
        '3:00': 'Development',
        '3:30': 'Check Mail',
        '4:00': 'Check Mail',
        '4:30': 'Team Meeting',
        '5:00': 'Team Meeting'
    },
    {EmployeeID: 10003,
      EmployeeName: 'Fuller',
      '9:00': 'Check Mail',
      '9:30': 'Check Mail',
      '10:00': 'Check Mail',
      '10:30': 'Analysis Tasks',
      '11:00': 'Analysis Tasks',
      '11:30': 'Support',
      '12:00': 'Support',
      '12:30': 'Support',
      '1:00': 'Lunch Break',
      '1:30': 'Lunch Break',
      '2:00': 'Lunch Break',
      '2:30': 'Development',
      '3:00': 'Development',
      '3:30': 'Team Meeting',
      '4:00': 'Team Meeting',
      '4:30': 'Development',
      '5:00': 'Development'
    },
    {EmployeeID: 10004,
      EmployeeName: 'Leverling',
      '9:00': 'Testing',
      '9:30': 'Check Mail',
      '10:00': 'Check Mail',
      '10:30': 'Support',
      '11:00': 'Testing',
      '11:30': 'Testing',
      '12:00': 'Testing',
      '12:30': 'Testing',
      '1:00': 'Lunch Break',
      '1:30': 'Lunch Break',
      '2:00': 'Lunch Break',
      '2:30': 'Development',
      '3:00': 'Development',
      '3:30': 'Check Mail',
      '4:00': 'Conference',
      '4:30': 'Conference',
      '5:00': 'Team Meeting'
    },
    {EmployeeID: 10005,
      EmployeeName: 'Peacock',
      '9:00': 'Task Assign',

```

```

        '9:30': 'Task Assign',
        '10:00': 'Task Assign',
        '10:30': 'Task Assign',
        '11:00': 'Check Mail',
        '11:30': 'Support',
        '12:00': 'Support',
        '12:30': 'Support',
        '1:00': 'Lunch Break',
        '1:30': 'Lunch Break',
        '2:00': 'Lunch Break',
        '2:30': 'Development',
        '3:00': 'Development',
        '3:30': 'Team Meeting',
        '4:00': 'Team Meeting',
        '4:30': 'Testing',
        '5:00': 'Testing'
    },
    {EmployeeID: 10006,
      EmployeeName: 'Janet',
      '9:00': 'Testing',
      '9:30': 'Testing',
      '10:00': 'Support',
      '10:30': 'Support',
      '11:00': 'Support',
      '11:30': 'Team Meeting',
      '12:00': 'Team Meeting',
      '12:30': 'Team Meeting',
      '1:00': 'Lunch Break',
      '1:30': 'Lunch Break',
      '2:00': 'Lunch Break',
      '2:30': 'Development',
      '3:00': 'Development',
      '3:30': 'Team Meeting',
      '4:00': 'Team Meeting',
      '4:30': 'Development',
      '5:00': 'Development'
    },
    {EmployeeID: 10007,
      EmployeeName: 'Suyama',
      '9:00': 'Analysis Tasks',
      '9:30': 'Analysis Tasks',
      '10:00': 'Testing',
      '10:30': 'Development',
      '11:00': 'Development',
      '11:30': 'Testing',
      '12:00': 'Testing',
      '12:30': 'Testing',
      '1:00': 'Lunch Break',
      '1:30': 'Lunch Break',
      '2:00': 'Lunch Break',
      '2:30': 'Support',
      '3:00': 'Build',
      '3:30': 'Build',
      '4:00': 'Check Mail',
      '4:30': 'Check Mail',
      '5:00': 'Check Mail'
    },
  ],

```

```

{EmployeeID}: 10008,
  EmployeeName: 'Robert',
  '9:00': 'Task Assign',
  '9:30': 'Task Assign',
  '10:00': 'Task Assign',
  '10:30': 'Development',
  '11:00': 'Development',
  '11:30': 'Development',
  '12:00': 'Testing',
  '12:30': 'Support',
  '1:00': 'Lunch Break',
  '1:30': 'Lunch Break',
  '2:00': 'Lunch Break',
  '2:30': 'Check Mail',
  '3:00': 'Check Mail',
  '3:30': 'Check Mail',
  '4:00': 'Team Meeting',
  '4:30': 'Team Meeting',
  '5:00': 'Build'
},
{EmployeeID}: 10009,
  EmployeeName: 'Andrew',
  '9:00': 'Check Mail',
  '9:30': 'Team Meeting',
  '10:00': 'Team Meeting',
  '10:30': 'Support',
  '11:00': 'Testing',
  '11:30': 'Development',
  '12:00': 'Development',
  '12:30': 'Development',
  '1:00': 'Lunch Break',
  '1:30': 'Lunch Break',
  '2:00': 'Lunch Break',
  '2:30': 'Check Mail',
  '3:00': 'Check Mail',
  '3:30': 'Check Mail',
  '4:00': 'Team Meeting',
  '4:30': 'Development',
  '5:00': 'Development'
},
{EmployeeID}: 10010,
  EmployeeName: 'Michael',
  '9:00': 'Task Assign',
  '9:30': 'Task Assign',
  '10:00': 'Task Assign',
  '10:30': 'Analysis Tasks',
  '11:00': 'Analysis Tasks',
  '11:30': 'Development',
  '12:00': 'Development',
  '12:30': 'Development',
  '1:00': 'Lunch Break',
  '1:30': 'Lunch Break',
  '2:00': 'Lunch Break',
  '2:30': 'Testing',
  '3:00': 'Testing',
  '3:30': 'Testing',
  '4:00': 'Build',

```

```

        '4:30': 'Build',
        '5:00': 'Build'
    }
];</>;

```

DATASOURCE.TS

```

export let productData: Object[] = [
    {
        'ProductID': 1,
        'ProductName': 'Chai',
        'SupplierID': 1,
        'QuantityPerUnit': '10 boxes x 20 bags',
        'UnitPrice': 18.00,
        'UnitsInStock': 39,
        'Discontinued': true
    },
    {
        'ProductID': 2,
        'ProductName': 'Chang',
        'SupplierID': 1,
        'QuantityPerUnit': '24 - 12 oz bottles',
        'UnitPrice': 19.00,
        'UnitsInStock': 17,
        'Discontinued': true
    },
    {
        'ProductID': 3,
        'ProductName': 'Aniseed Syrup',
        'SupplierID': 1,
        'QuantityPerUnit': '12 - 550 ml bottles',
        'UnitPrice': 10.00,
        'UnitsInStock': 13,
        'Discontinued': true
    },
    {
        'ProductID': 4,
        'ProductName': 'Chef Anton\'s Cajun Seasoning',
        'SupplierID': 2,
        'QuantityPerUnit': '48 - 6 oz jars',
        'UnitPrice': 22.00,
        'UnitsInStock': 53,
        'Discontinued': true
    },
    {
        'ProductID': 5,
        'ProductName': 'Chef Anton\'s Gumbo Mix',
        'SupplierID': 2,
        'QuantityPerUnit': '36 boxes',
        'UnitPrice': 21.35,
        'UnitsInStock': 0,
        'Discontinued': true
    },
    {
        'ProductID': 6,
        'ProductName': 'Grandma\'s Boysenberry Spread',

```

```

    'SupplierID': 3,
    'QuantityPerUnit': '12 - 8 oz jars',
    'UnitPrice': 25.00,
    'UnitsInStock': 120,
    'Discontinued': false
  },
  {
    'ProductID': 7,
    'ProductName': 'Uncle Bob\'s Organic Dried Pears',
    'SupplierID': 3,
    'QuantityPerUnit': '12 - 1 lb pkgs.',
    'UnitPrice': 30.00,
    'UnitsInStock': 15,
    'Discontinued': false
  },
  {
    'ProductID': 8,
    'ProductName': 'Northwoods Cranberry Sauce',
    'SupplierID': 3,
    'QuantityPerUnit': '12 - 12 oz jars',
    'UnitPrice': 40.00,
    'UnitsInStock': 6,
    'Discontinued': false
  },
  {
    'ProductID': 9,
    'ProductName': 'Mishi Kobe Niku',
    'SupplierID': 4,
    'QuantityPerUnit': '18 - 500 g pkgs.',
    'UnitPrice': 97.00,
    'UnitsInStock': 29,
    'Discontinued': true
  },
  {
    'ProductID': 10,
    'ProductName': 'Ikura',
    'SupplierID': 4,
    'QuantityPerUnit': '12 - 200 ml jars',
    'UnitPrice': 31.00,
    'UnitsInStock': 31,
    'Discontinued': false
  },
  {
    'ProductID': 11,
    'ProductName': 'Queso Cabrales',
    'SupplierID': 5,
    'QuantityPerUnit': '1 kg pkg.',
    'UnitPrice': 21.00,
    'UnitsInStock': 22,
    'Discontinued': false
  },
  {
    'ProductID': 12,
    'ProductName': 'Queso Manchego La Pastora',
    'SupplierID': 5,
    'QuantityPerUnit': '10 - 500 g pkgs.',
    'UnitPrice': 38.00,

```

```

      'UnitsInStock': 86,
      'Discontinued': false
    },
    {
      'ProductID': 13,
      'ProductName': 'Konbu',
      'SupplierID': 6,
      'QuantityPerUnit': '2 kg box',
      'UnitPrice': 6.00,
      'UnitsInStock': 24,
      'Discontinued': true
    },
    {
      'ProductID': 14,
      'ProductName': 'Tofu',
      'SupplierID': 6,
      'QuantityPerUnit': '40 - 100 g pkgs.',
      'UnitPrice': 23.25,
      'UnitsInStock': 35,
      'Discontinued': true
    },
    {
      'ProductID': 15,
      'ProductName': 'Genen Shouyu',
      'SupplierID': 6,
      'QuantityPerUnit': '24 - 250 ml bottles',
      'UnitPrice': 15.50,
      'UnitsInStock': 39,
      'Discontinued': true
    },
    {
      'ProductID': 16,
      'ProductName': 'Pavlova',
      'SupplierID': 7,
      'QuantityPerUnit': '32 - 500 g boxes',
      'UnitPrice': 17.45,
      'UnitsInStock': 29,
      'Discontinued': true
    },
    {
      'ProductID': 17,
      'ProductName': 'Alice Mutton',
      'SupplierID': 7,
      'QuantityPerUnit': '20 - 1 kg tins',
      'UnitPrice': 39.00,
      'UnitsInStock': 0,
      'Discontinued': true
    },
    {
      'ProductID': 18,
      'ProductName': 'Carnarvon Tigers',
      'SupplierID': 7,
      'QuantityPerUnit': '16 kg pkg.',
      'UnitPrice': 62.50,
      'UnitsInStock': 42,
      'Discontinued': false
    }
  ],

```

```

{
  'ProductID': 19,
  'ProductName': 'Teatime Chocolate Biscuits',
  'SupplierID': 8,
  'QuantityPerUnit': '10 boxes x 12 pieces',
  'UnitPrice': 9.20,
  'UnitsInStock': 25,
  'Discontinued': false
},
{
  'ProductID': 20,
  'ProductName': 'Sir Rodney\'s Marmalade',
  'SupplierID': 8,
  'QuantityPerUnit': '30 gift boxes',
  'UnitPrice': 81.00,
  'UnitsInStock': 40,
  'Discontinued': false
},
{
  'ProductID': 21,
  'ProductName': 'Sir Rodney\'s Scones',
  'SupplierID': 8,
  'QuantityPerUnit': '24 pkgs. x 4 pieces',
  'UnitPrice': 10.00,
  'UnitsInStock': 3,
  'Discontinued': false
},
{
  'ProductID': 22,
  'ProductName': 'Gustaf\'s Knäckebröd',
  'SupplierID': 9,
  'QuantityPerUnit': '24 - 500 g pkgs.',
  'UnitPrice': 21.00,
  'UnitsInStock': 104,
  'Discontinued': false
},
{
  'ProductID': 23,
  'ProductName': 'Tunnbröd',
  'SupplierID': 9,
  'QuantityPerUnit': '12 - 250 g pkgs.',
  'UnitPrice': 9.00,
  'UnitsInStock': 61,
  'Discontinued': false
},
{
  'ProductID': 24,
  'ProductName': 'Guaraná Fantástica',
  'SupplierID': 10,
  'QuantityPerUnit': '12 - 355 ml cans',
  'UnitPrice': 4.50,
  'UnitsInStock': 20,
  'Discontinued': true
},
{
  'ProductID': 25,
  'ProductName': 'NuNuCa Nuß-Nougat-Creme',

```



```

    'SupplierID': 11,
    'QuantityPerUnit': '20 - 450 g glasses',
    'UnitPrice': 14.00,
    'UnitsInStock': 76,
    'Discontinued': false
  },
  {
    'ProductID': 26,
    'ProductName': 'Gumbär Gummibärchen',
    'SupplierID': 11,
    'QuantityPerUnit': '100 - 250 g bags',
    'UnitPrice': 31.23,
    'UnitsInStock': 15,
    'Discontinued': true
  },
  {
    'ProductID': 27,
    'ProductName': 'Schoggi Schokolade',
    'SupplierID': 11,
    'QuantityPerUnit': '100 - 100 g pieces',
    'UnitPrice': 43.90,
    'UnitsInStock': 49,
    'Discontinued': true
  },
  {
    'ProductID': 28,
    'ProductName': 'Rössle Sauerkraut',
    'SupplierID': 12,
    'QuantityPerUnit': '25 - 825 g cans',
    'UnitPrice': 45.60,
    'UnitsInStock': 26,
    'Discontinued': true
  },
  {
    'ProductID': 29,
    'ProductName': 'Thüringer Rostbratwurst',
    'SupplierID': 12,
    'QuantityPerUnit': '50 bags x 30 sausgs.',
    'UnitPrice': 123.79,
    'UnitsInStock': 0,
    'Discontinued': true
  },
  {
    'ProductID': 30,
    'ProductName': 'Nord-Ost Matjeshering',
    'SupplierID': 13,
    'QuantityPerUnit': '10 - 200 g glasses',
    'UnitPrice': 25.89,
    'UnitsInStock': 10,
    'Discontinued': true
  },
  {
    'ProductID': 31,
    'ProductName': 'Gorgonzola Telino',
    'SupplierID': 14,
    'QuantityPerUnit': '12 - 100 g pkgs',
    'UnitPrice': 12.50,

```

```
    'UnitsInStock': 0,  
    'Discontinued': true  
  },  
  {  
    'ProductID': 32,  
    'ProductName': 'Mascarpone Fabioli',  
    'SupplierID': 14,  
    'QuantityPerUnit': '24 - 200 g pkgs.',  
    'UnitPrice': 32.00,  
    'UnitsInStock': 9,  
    'Discontinued': false  
  },  
  {  
    'ProductID': 33,  
    'ProductName': 'Geitost',  
    'SupplierID': 15,  
    'QuantityPerUnit': '500 g',  
    'UnitPrice': 2.50,  
    'UnitsInStock': 112,  
    'Discontinued': false  
  },  
  {  
    'ProductID': 34,  
    'ProductName': 'Sasquatch Ale',  
    'SupplierID': 16,  
    'QuantityPerUnit': '24 - 12 oz bottles',  
    'UnitPrice': 14.00,  
    'UnitsInStock': 111,  
    'Discontinued': false  
  },  
  {  
    'ProductID': 35,  
    'ProductName': 'Steeleye Stout',  
    'SupplierID': 16,  
    'QuantityPerUnit': '24 - 12 oz bottles',  
    'UnitPrice': 18.00,  
    'UnitsInStock': 20,  
    'Discontinued': false  
  },  
  {  
    'ProductID': 36,  
    'ProductName': 'Inlagd Sill',  
    'SupplierID': 17,  
    'QuantityPerUnit': '24 - 250 g jars',  
    'UnitPrice': 19.00,  
    'UnitsInStock': 112,  
    'Discontinued': false  
  },  
  {  
    'ProductID': 37,  
    'ProductName': 'Gravad lax',  
    'SupplierID': 17,  
    'QuantityPerUnit': '12 - 500 g pkgs.',  
    'UnitPrice': 26.00,  
    'UnitsInStock': 11,  
    'Discontinued': false  
  },  
}
```

```

{
  'ProductID': 38,
  'ProductName': 'Côte de Blaye',
  'SupplierID': 18,
  'QuantityPerUnit': '12 - 75 cl bottles',
  'UnitPrice': 263.50,
  'UnitsInStock': 17,
  'Discontinued': false
},
{
  'ProductID': 39,
  'ProductName': 'Chartreuse verte',
  'SupplierID': 18,
  'QuantityPerUnit': '750 cc per bottle',
  'UnitPrice': 18.00,
  'UnitsInStock': 69,
  'Discontinued': true
},
{
  'ProductID': 40,
  'ProductName': 'Boston Crab Meat',
  'SupplierID': 19,
  'QuantityPerUnit': '24 - 4 oz tins',
  'UnitPrice': 18.40,
  'UnitsInStock': 123,
  'Discontinued': true
},
{
  'ProductID': 41,
  'ProductName': 'Jack\'s New England Clam Chowder',
  'SupplierID': 19,
  'QuantityPerUnit': '12 - 12 oz cans',
  'UnitPrice': 9.65,
  'UnitsInStock': 85,
  'Discontinued': false
},
{
  'ProductID': 42,
  'ProductName': 'Singaporean Hokkien Fried Mee',
  'SupplierID': 20,
  'QuantityPerUnit': '32 - 1 kg pkgs.',
  'UnitPrice': 14.00,
  'UnitsInStock': 26,
  'Discontinued': true
},
{
  'ProductID': 43,
  'ProductName': 'Ipoh Coffee',
  'SupplierID': 20,
  'QuantityPerUnit': '16 - 500 g tins',
  'UnitPrice': 46.00,
  'UnitsInStock': 17,
  'Discontinued': false
},
{
  'ProductID': 44,
  'ProductName': 'Gula Malacca',

```

```

    'SupplierID': 20,
    'QuantityPerUnit': '20 - 2 kg bags',
    'UnitPrice': 19.45,
    'UnitsInStock': 27,
    'Discontinued': false
  },
  {
    'ProductID': 45,
    'ProductName': 'Rogede sild',
    'SupplierID': 21,
    'QuantityPerUnit': '1k pkg.',
    'UnitPrice': 9.50,
    'UnitsInStock': 5,
    'Discontinued': true
  },
  {
    'ProductID': 46,
    'ProductName': 'Spegesild',
    'SupplierID': 21,
    'QuantityPerUnit': '4 - 450 g glasses',
    'UnitPrice': 12.00,
    'UnitsInStock': 95,
    'Discontinued': true
  },
  {
    'ProductID': 47,
    'ProductName': 'Zaanse koeken',
    'SupplierID': 22,
    'QuantityPerUnit': '10 - 4 oz boxes',
    'UnitPrice': 9.50,
    'UnitsInStock': 36,
    'Discontinued': true
  },
  {
    'ProductID': 48,
    'ProductName': 'Chocolade',
    'SupplierID': 22,
    'QuantityPerUnit': '10 pkgs.',
    'UnitPrice': 12.75,
    'UnitsInStock': 15,
    'Discontinued': true
  },
  {
    'ProductID': 49,
    'ProductName': 'Maxilaku',
    'SupplierID': 23,
    'QuantityPerUnit': '24 - 50 g pkgs.',
    'UnitPrice': 20.00,
    'UnitsInStock': 10,
    'Discontinued': false
  },
  {
    'ProductID': 50,
    'ProductName': 'Valkoinen suklaa',
    'SupplierID': 23,
    'QuantityPerUnit': '12 - 100 g bars',
    'UnitPrice': 16.25,

```

```

      'UnitsInStock': 65,
      'Discontinued': false
    },
    {
      'ProductID': 51,
      'ProductName': 'Manjimup Dried Apples',
      'SupplierID': 24,
      'QuantityPerUnit': '50 - 300 g pkgs.',
      'UnitPrice': 53.00,
      'UnitsInStock': 20,
      'Discontinued': false
    },
    {
      'ProductID': 52,
      'ProductName': 'Filo Mix',
      'SupplierID': 24,
      'QuantityPerUnit': '16 - 2 kg boxes',
      'UnitPrice': 7.00,
      'UnitsInStock': 38,
      'Discontinued': true
    },
    {
      'ProductID': 53,
      'ProductName': 'Perth Pasties',
      'SupplierID': 24,
      'QuantityPerUnit': '48 pieces',
      'UnitPrice': 32.80,
      'UnitsInStock': 0,
      'Discontinued': true
    },
    {
      'ProductID': 54,
      'ProductName': 'Tourtière',
      'SupplierID': 25,
      'QuantityPerUnit': '16 pies',
      'UnitPrice': 7.45,
      'UnitsInStock': 21,
      'Discontinued': true
    },
    {
      'ProductID': 55,
      'ProductName': 'Pâté chinois',
      'SupplierID': 25,
      'QuantityPerUnit': '24 boxes x 2 pies',
      'UnitPrice': 24.00,
      'UnitsInStock': 115,
      'Discontinued': true
    },
    {
      'ProductID': 56,
      'ProductName': 'Gnocchi di nonna Alice',
      'SupplierID': 26,
      'QuantityPerUnit': '24 - 250 g pkgs.',
      'UnitPrice': 38.00,
      'UnitsInStock': 21,
      'Discontinued': false
    },
  ],

```

```

{
  'ProductID': 57,
  'ProductName': 'Ravioli Angelo',
  'SupplierID': 26,
  'QuantityPerUnit': '24 - 250 g pkgs.',
  'UnitPrice': 19.50,
  'UnitsInStock': 36,
  'Discontinued': false
},
{
  'ProductID': 58,
  'ProductName': 'Escargots de Bourgogne',
  'SupplierID': 27,
  'QuantityPerUnit': '24 pieces',
  'UnitPrice': 13.25,
  'UnitsInStock': 62,
  'Discontinued': false
},
{
  'ProductID': 59,
  'ProductName': 'Raclette Courdavault',
  'SupplierID': 28,
  'QuantityPerUnit': '5 kg pkg.',
  'UnitPrice': 55.00,
  'UnitsInStock': 79,
  'Discontinued': false
},
{
  'ProductID': 60,
  'ProductName': 'Camembert Pierrot',
  'SupplierID': 28,
  'QuantityPerUnit': '15 - 300 g rounds',
  'UnitPrice': 34.00,
  'UnitsInStock': 19,
  'Discontinued': false
},
{
  'ProductID': 61,
  'ProductName': 'Sirop d\'érable',
  'SupplierID': 29,
  'QuantityPerUnit': '24 - 500 ml bottles',
  'UnitPrice': 28.50,
  'UnitsInStock': 113,
  'Discontinued': false
},
{
  'ProductID': 62,
  'ProductName': 'Tarte au sucre',
  'SupplierID': 29,
  'QuantityPerUnit': '48 pies',
  'UnitPrice': 49.30,
  'UnitsInStock': 17,
  'Discontinued': false
},
{
  'ProductID': 63,
  'ProductName': 'Veggie-spread',

```

```

    'SupplierID': 7,
    'QuantityPerUnit': '15 - 625 g jars',
    'UnitPrice': 43.90,
    'UnitsInStock': 24,
    'Discontinued': true
  },
  {
    'ProductID': 64,
    'ProductName': 'Wimmers gute Semmelknödel',
    'SupplierID': 12,
    'QuantityPerUnit': '20 bags x 4 pieces',
    'UnitPrice': 33.25,
    'UnitsInStock': 22,
    'Discontinued': true
  },
  {
    'ProductID': 65,
    'ProductName': 'Louisiana Fiery Hot Pepper Sauce',
    'SupplierID': 2,
    'QuantityPerUnit': '32 - 8 oz bottles',
    'UnitPrice': 21.05,
    'UnitsInStock': 76,
    'Discontinued': true
  },
  {
    'ProductID': 66,
    'ProductName': 'Louisiana Hot Spiced Okra',
    'SupplierID': 2,
    'QuantityPerUnit': '24 - 8 oz jars',
    'UnitPrice': 17.00,
    'UnitsInStock': 4,
    'Discontinued': false
  },
  {
    'ProductID': 67,
    'ProductName': 'Laughing Lumberjack Lager',
    'SupplierID': 16,
    'QuantityPerUnit': '24 - 12 oz bottles',
    'UnitPrice': 14.00,
    'UnitsInStock': 52,
    'Discontinued': false
  },
  {
    'ProductID': 68,
    'ProductName': 'Scottish Longbreads',
    'SupplierID': 8,
    'QuantityPerUnit': '10 boxes x 8 pieces',
    'UnitPrice': 12.50,
    'UnitsInStock': 6,
    'Discontinued': false
  },
  {
    'ProductID': 69,
    'ProductName': 'Gudbrandsdalsost',
    'SupplierID': 15,
    'QuantityPerUnit': '10 kg pkg.',
    'UnitPrice': 36.00,

```

```

      'UnitsInStock': 26,
      'Discontinued': false
    },
    {
      'ProductID': 70,
      'ProductName': 'Outback Lager',
      'SupplierID': 7,
      'QuantityPerUnit': '24 - 355 ml bottles',
      'UnitPrice': 15.00,
      'UnitsInStock': 15,
      'Discontinued': false
    },
    {
      'ProductID': 71,
      'ProductName': 'Flotemysost',
      'SupplierID': 15,
      'QuantityPerUnit': '10 - 500 g pkgs.',
      'UnitPrice': 21.50,
      'UnitsInStock': 26,
      'Discontinued': true
    },
    {
      'ProductID': 72,
      'ProductName': 'Mozzarella di Giovanni',
      'SupplierID': 14,
      'QuantityPerUnit': '24 - 200 g pkgs.',
      'UnitPrice': 34.80,
      'UnitsInStock': 14,
      'Discontinued': true
    },
    {
      'ProductID': 73,
      'ProductName': 'Röd Kaviar',
      'SupplierID': 17,
      'QuantityPerUnit': '24 - 150 g jars',
      'UnitPrice': 15.00,
      'UnitsInStock': 101,
      'Discontinued': true
    },
    {
      'ProductID': 74,
      'ProductName': 'Longlife Tofu',
      'SupplierID': 4,
      'QuantityPerUnit': '5 kg pkg.',
      'UnitPrice': 10.00,
      'UnitsInStock': 4,
      'Discontinued': true
    },
    {
      'ProductID': 75,
      'ProductName': 'Rhönbräu Klosterbier',
      'SupplierID': 12,
      'QuantityPerUnit': '24 - 0.5 l bottles',
      'UnitPrice': 7.75,
      'UnitsInStock': 125,
      'Discontinued': true
    }
  ],

```



```

{
  'ProductID': 76,
  'ProductName': 'Lakkalikööri',
  'SupplierID': 23,
  'QuantityPerUnit': '500 ml',
  'UnitPrice': 18.00,
  'UnitsInStock': 57,
  'Discontinued': true
},
{
  'ProductID': 77,
  'ProductName': 'Original Frankfurter grüne Soße',
  'SupplierID': 12,
  'QuantityPerUnit': '12 boxes',
  'UnitPrice': 13.00,
  'UnitsInStock': 32,
  'Discontinued': false
}
]];
export let employeeData: Object[] = [{
  'EmployeeID': 1,
  'LastName': 'Davolio',
  'FirstName': 'Nancy',
  'Title': 'Sales Representative',
  'TitleOfCourtesy': 'Ms.',
  'BirthDate': new Date(-664743600000),
  'HireDate': new Date(704692800000),
  'Address': '507 - 20th Ave. E.\r\nApt. 2A',
  'City': 'Seattle',
  'Region': 'WA',
  'PostalCode': '98122',
  'Country': 'USA',
  'HomePhone': '(206) 555-9857',
  'Extension': '5467',
  'Photo': { 'Length': 21626 },
  'Notes': 'Education includes a BA in psychology from Colorado State University in 1970. She also completed\'
  \'The Art of the Cold Call.\' Nancy is a member of Toastmasters International.',
  'ReportsTo': 2,
  'PhotoPath': 'http://accweb/emmployees/davolio.bmp'
},
{
  'EmployeeID': 2,
  'LastName': 'Fuller',
  'FirstName': 'Andrew',
  'Title': 'Vice President, Sales',
  'TitleOfCourtesy': 'Dr.',
  'BirthDate': new Date(-563828400000),
  'HireDate': new Date(713764800000),
  'Address': '908 W. Capital Way',
  'City': 'Tacoma',
  'Region': 'WA',
  'PostalCode': '98401',
  'Country': 'USA',
  'HomePhone': '(206) 555-9482',
  'Extension': '3457',
  'Photo': { 'Length': 21626 },

```

```

      'Notes': '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.',
      'ReportsTo': 0,
      'PhotoPath': 'http://accweb/emmployees/fuller.bmp'
    },
    {
      'EmployeeID': 3,
      'LastName': 'Leverling',
      'FirstName': 'Janet',
      'Title': 'Sales Representative',
      'TitleOfCourtesy': 'Ms.',
      'BirthDate': new Date(-200088000000),
      'HireDate': new Date(702104400000),
      'Address': '722 Moss Bay Blvd.',
      'City': 'Kirkland',
      'Region': 'WA',
      'PostalCode': '98033',
      'Country': 'USA',
      'HomePhone': '(206) 555-3412',
      'Extension': '3355',
      'Photo': { 'Length': 21722 },
      'Notes': '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.',
      'ReportsTo': 2,
      'PhotoPath': 'http://accweb/emmployees/leverling.bmp'
    },
    {
      'EmployeeID': 4,
      'LastName': 'Peacock',
      'FirstName': 'Margaret',
      'Title': 'Sales Representative',
      'TitleOfCourtesy': 'Mrs.',
      'BirthDate': new Date(-1018814400000),
      'HireDate': new Date(736401600000),
      'Address': '4110 Old Redmond Rd.',
      'City': 'Redmond',
      'Region': 'WA',
      'PostalCode': '98052',
      'Country': 'USA',
      'HomePhone': '(206) 555-8122',
      'Extension': '5176',
      'Photo': { 'Length': 21626 },
      'Notes': '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.',
      'ReportsTo': 2,

```

```

    'PhotoPath': 'http://accweb/emmployees/peacock.bmp'
  },
  {
    'EmployeeID': 5,
    'LastName': 'Buchanan',
    'FirstName': 'Steven',
    'Title': 'Sales Manager',
    'TitleOfCourtesy': 'Mr.',
    'BirthDate': new Date(-468010800000),
    'HireDate': new Date(750830400000),
    'Address': '14 Garrett Hill',
    'City': 'London',
    'Region': null,
    'PostalCode':
    'SW1 8JR',
    'Country': 'UK',
    'HomePhone': '(71) 555-4848',
    'Extension': '3453',
    'Photo': { 'Length': 21626 },
    'Notes': 'Steven Buchanan graduated from St. Andrews University,
    Scotland, with a BSC degree in 1976. Upon joining the company as \
    a sales representative in 1992, he spent 6 months in an orientation
    program at the Seattle office and then returned to his permanent \
    post in London. He was promoted to sales manager in March 1993. Mr.
    Buchanan has completed the courses \'Successful \
    Telemarketing\' and \'International Sales Management.\' He is fluent in
    French.',
    'ReportsTo': 2,
    'PhotoPath': 'http://accweb/emmployees/buchanan.bmp'
  },
  {
    'EmployeeID': 6,
    'LastName': 'Suyama',
    'FirstName': 'Michael',
    'Title': 'Sales Representative',
    'TitleOfCourtesy': 'Mr.',
    'BirthDate': new Date(-205185600000),
    'HireDate': new Date(750830400000),
    'Address': 'Coventry House\r\nMiner Rd.',
    'City': 'London',
    'Region': null,
    'PostalCode': 'EC2 7JR',
    'Country': 'UK',
    'HomePhone': '(71) 555-7773',
    'Extension': '428',
    'Photo': { 'Length': 21626 },
    'Notes': 'Michael is a graduate of Sussex University (MA, economics,
    1983) and the University of California at Los Angeles \
    (MBA, marketing, 1986). He has also taken the courses \'Multi-Cultural
    Selling\' and \'Time Management for the Sales Professional.\' \
    He is fluent in Japanese and can read and write French, Portuguese, and
    Spanish.',
    'ReportsTo': 5,
    'PhotoPath': 'http://accweb/emmployees/davolio.bmp'
  },
  {
    'EmployeeID': 7,

```

```

      'LastName': 'King',
      'FirstName': 'Robert',
      'Title': 'Sales Representative',
      'TitleOfCourtesy': 'Mr.',
      'BirthDate': new Date(-302731200000),
      'HireDate': new Date(757486800000),
      'Address': 'Edgeham Hollow\r\nWinchester Way',
      'City': 'London',
      'Region': null,
      'PostalCode': 'RG1 9SP',
      'Country': 'UK',
      'HomePhone': '(71) 555-5598',
      'Extension': '465',
      'Photo': { 'Length': 21626 },
      'Notes': 'Robert King served in the Peace Corps and traveled extensively
before completing his degree in English at the \
University of Michigan in 1992, the year he joined the company. After
completing a course entitled \'Selling in Europe,\' \
he was transferred to the London office in March 1993.',
      'ReportsTo': 5,
      'PhotoPath': 'http://accweb/emmployees/davolio.bmp'
    },
    {
      'EmployeeID': 8,
      'LastName': 'Callahan',
      'FirstName': 'Laura',
      'Title': 'Inside Sales Coordinator',
      'TitleOfCourtesy': 'Ms.',
      'BirthDate': new Date(-377982000000),
      'HireDate': new Date(762843600000),
      'Address': '4726 - 11th Ave. N.E.',
      'City': 'Seattle',
      'Region': 'WA',
      'PostalCode': '98105',
      'Country': 'USA',
      'HomePhone': '(206) 555-1189',
      'Extension': '2344',
      'Photo': { 'Length': 21626 },
      'Notes': 'Laura received a BA in psychology from the University of
Washington. She has also completed a course in business \
French. She reads and writes French.',
      'ReportsTo': 2,
      'PhotoPath': 'http://accweb/emmployees/davolio.bmp'
    },
    {
      'EmployeeID': 9,
      'LastName': 'Dodsworth',
      'FirstName': 'Anne',
      'Title': 'Sales Representative',
      'TitleOfCourtesy': 'Ms.',
      'BirthDate': new Date(-123966000000),
      'HireDate': new Date(784875600000),
      'Address': '7 Houndstooth Rd.',
      'City': 'London',
      'Region': null,
      'PostalCode': 'WG2 7LT',
      'Country': 'UK',

```

```

    'HomePhone': '(71) 555-4444',
    'Extension': '452',
    'Photo': { 'Length': 21626 },
    'Notes': 'Anne has a BA degree in English from St. Lawrence College.
She is fluent in French and German.',
    'ReportsTo': 5,
    'PhotoPath': 'http://accweb/emmployees/davolio.bmp'
  }];
let stringData: string = JSON.stringify([
  {
    'OrderID': 10248,
    'CustomerID': 'VINET',
    'OrderDate': '1996-07-04T00:00:00.000Z',
    'ShippedDate': '1996-07-16T00:00:00.000Z',
    'Freight': 32.38,
    'ShipName': 'Vins et alcools Chevalier',
    'ShipAddress': '59 rue de l\'Abbaye',
    'ShipCity': 'Reims',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10249,
    'CustomerID': 'TOMSP',
    'OrderDate': '1996-07-05T00:00:00.000Z',
    'ShippedDate': '1996-07-10T00:00:00.000Z',
    'Freight': 11.61,
    'ShipName': 'Toms Spezialitäten',
    'ShipAddress': 'Luisenstr. 48',
    'ShipCity': 'Münster',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10250,
    'CustomerID': 'HANAR',
    'OrderDate': '1996-07-08T00:00:00.000Z',
    'ShippedDate': '1996-07-12T00:00:00.000Z',
    'Freight': 65.83,
    'ShipName': 'Hanari Carnes',
    'ShipAddress': 'Rua do Paço, 67',
    'ShipCity': 'Rio de Janeiro',
    'ShipRegion': 'RJ',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10251,
    'CustomerID': 'VICTE',
    'OrderDate': '1996-07-08T00:00:00.000Z',
    'ShippedDate': '1996-07-15T00:00:00.000Z',
    'Freight': 41.34,
    'ShipName': 'Victuailles en stock',
    'ShipAddress': '2, rue du Commerce',
    'ShipCity': 'Lyon',
    'ShipRegion': null,
    'ShipCountry': 'France'
  }
],

```

```

{
  'OrderID': 10252,
  'CustomerID': 'SUPRD',
  'OrderDate': '1996-07-09T00:00:00.000Z',
  'ShippedDate': '1996-07-11T00:00:00.000Z',
  'Freight': 51.3,
  'ShipName': 'Suprêmes délices',
  'ShipAddress': 'Boulevard Tirou, 255',
  'ShipCity': 'Charleroi',
  'ShipRegion': null,
  'ShipCountry': 'Belgium'
},
{
  'OrderID': 10253,
  'CustomerID': 'HANAR',
  'OrderDate': '1996-07-10T00:00:00.000Z',
  'ShippedDate': '1996-07-16T00:00:00.000Z',
  'Freight': 58.17,
  'ShipName': 'Hanari Carnes',
  'ShipAddress': 'Rua do Paço, 67',
  'ShipCity': 'Rio de Janeiro',
  'ShipRegion': 'RJ',
  'ShipCountry': 'Brazil'
},
{
  'OrderID': 10254,
  'CustomerID': 'CHOPS',
  'OrderDate': '1996-07-11T00:00:00.000Z',
  'ShippedDate': '1996-07-23T00:00:00.000Z',
  'Freight': 22.98,
  'ShipName': 'Chop-suey Chinese',
  'ShipAddress': 'Hauptstr. 31',
  'ShipCity': 'Bern',
  'ShipRegion': null,
  'ShipCountry': 'Switzerland'
},
{
  'OrderID': 10255,
  'CustomerID': 'RICSU',
  'OrderDate': '1996-07-12T00:00:00.000Z',
  'ShippedDate': '1996-07-15T00:00:00.000Z',
  'Freight': 148.33,
  'ShipName': 'Richter Supermarkt',
  'ShipAddress': 'Starenweg 5',
  'ShipCity': 'Genève',
  'ShipRegion': null,
  'ShipCountry': 'Switzerland'
},
{
  'OrderID': 10256,
  'CustomerID': 'WELLI',
  'OrderDate': '1996-07-15T00:00:00.000Z',
  'ShippedDate': '1996-07-17T00:00:00.000Z',
  'Freight': 13.97,
  'ShipName': 'Wellington Importadora',
  'ShipAddress': 'Rua do Mercado, 12',
  'ShipCity': 'Resende',

```

```

      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10257,
      'CustomerID': 'HILAA',
      'OrderDate': '1996-07-16T00:00:00.000Z',
      'ShippedDate': '1996-07-22T00:00:00.000Z',
      'Freight': 81.91,
      'ShipName': 'HILARION-Abastos',
      'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
      'ShipCity': 'San Cristóbal',
      'ShipRegion': 'Táchira',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10258,
      'CustomerID': 'ERNSH',
      'OrderDate': '1996-07-17T00:00:00.000Z',
      'ShippedDate': '1996-07-23T00:00:00.000Z',
      'Freight': 140.51,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10259,
      'CustomerID': 'CENTC',
      'OrderDate': '1996-07-18T00:00:00.000Z',
      'ShippedDate': '1996-07-25T00:00:00.000Z',
      'Freight': 3.25,
      'ShipName': 'Centro comercial Moctezuma',
      'ShipAddress': 'Sierras de Granada 9993',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10260,
      'CustomerID': 'OTTIK',
      'OrderDate': '1996-07-19T00:00:00.000Z',
      'ShippedDate': '1996-07-29T00:00:00.000Z',
      'Freight': 55.09,
      'ShipName': 'Ottilies Käseladen',
      'ShipAddress': 'Mehrheimerstr. 369',
      'ShipCity': 'Köln',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10261,
      'CustomerID': 'QUEDE',
      'OrderDate': '1996-07-19T00:00:00.000Z',
      'ShippedDate': '1996-07-30T00:00:00.000Z',
      'Freight': 3.05,

```

```

    'ShipName': 'Que Delícia',
    'ShipAddress': 'Rua da Panificadora, 12',
    'ShipCity': 'Rio de Janeiro',
    'ShipRegion': 'RJ',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10262,
    'CustomerID': 'RATTC',
    'OrderDate': '1996-07-22T00:00:00.000Z',
    'ShippedDate': '1996-07-25T00:00:00.000Z',
    'Freight': 48.29,
    'ShipName': 'Rattlesnake Canyon Grocery',
    'ShipAddress': '2817 Milton Dr.',
    'ShipCity': 'Albuquerque',
    'ShipRegion': 'NM',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10263,
    'CustomerID': 'ERNSH',
    'OrderDate': '1996-07-23T00:00:00.000Z',
    'ShippedDate': '1996-07-31T00:00:00.000Z',
    'Freight': 146.06,
    'ShipName': 'Ernst Handel',
    'ShipAddress': 'Kirchgasse 6',
    'ShipCity': 'Graz',
    'ShipRegion': null,
    'ShipCountry': 'Austria'
  },
  {
    'OrderID': 10264,
    'CustomerID': 'FOLKO',
    'OrderDate': '1996-07-24T00:00:00.000Z',
    'ShippedDate': '1996-08-23T00:00:00.000Z',
    'Freight': 3.67,
    'ShipName': 'Folk och fä HB',
    'ShipAddress': 'Åkergatan 24',
    'ShipCity': 'Bräcke',
    'ShipRegion': null,
    'ShipCountry': 'Sweden'
  },
  {
    'OrderID': 10265,
    'CustomerID': 'BLONP',
    'OrderDate': '1996-07-25T00:00:00.000Z',
    'ShippedDate': '1996-08-12T00:00:00.000Z',
    'Freight': 55.28,
    'ShipName': 'Blondel père et fils',
    'ShipAddress': '24, place Kléber',
    'ShipCity': 'Strasbourg',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10266,
    'CustomerID': 'WARTH',

```



```

      'OrderDate': '1996-07-26T00:00:00.000Z',
      'ShippedDate': '1996-07-31T00:00:00.000Z',
      'Freight': 25.73,
      'ShipName': 'Wartian Herkku',
      'ShipAddress': 'Torikatu 38',
      'ShipCity': 'Oulu',
      'ShipRegion': null,
      'ShipCountry': 'Finland'
    },
    {
      'OrderID': 10267,
      'CustomerID': 'FRANK',
      'OrderDate': '1996-07-29T00:00:00.000Z',
      'ShippedDate': '1996-08-06T00:00:00.000Z',
      'Freight': 208.58,
      'ShipName': 'Frankenversand',
      'ShipAddress': 'Berliner Platz 43',
      'ShipCity': 'München',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10268,
      'CustomerID': 'GROSR',
      'OrderDate': '1996-07-30T00:00:00.000Z',
      'ShippedDate': '1996-08-02T00:00:00.000Z',
      'Freight': 66.29,
      'ShipName': 'GROSELLA-Restaurante',
      'ShipAddress': '5a Ave. Los Palos Grandes',
      'ShipCity': 'Caracas',
      'ShipRegion': 'DF',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10269,
      'CustomerID': 'WHITC',
      'OrderDate': '1996-07-31T00:00:00.000Z',
      'ShippedDate': '1996-08-09T00:00:00.000Z',
      'Freight': 4.56,
      'ShipName': 'White Clover Markets',
      'ShipAddress': '1029 - 12th Ave. S.',
      'ShipCity': 'Seattle',
      'ShipRegion': 'WA',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10270,
      'CustomerID': 'WARTH',
      'OrderDate': '1996-08-01T00:00:00.000Z',
      'ShippedDate': '1996-08-02T00:00:00.000Z',
      'Freight': 136.54,
      'ShipName': 'Wartian Herkku',
      'ShipAddress': 'Torikatu 38',
      'ShipCity': 'Oulu',
      'ShipRegion': null,
      'ShipCountry': 'Finland'
    }
  ],

```

```

{
  'OrderID': 10271,
  'CustomerID': 'SPLIR',
  'OrderDate': '1996-08-01T00:00:00.000Z',
  'ShippedDate': '1996-08-30T00:00:00.000Z',
  'Freight': 4.54,
  'ShipName': 'Split Rail Beer & Ale',
  'ShipAddress': 'P.O. Box 555',
  'ShipCity': 'Lander',
  'ShipRegion': 'WY',
  'ShipCountry': 'USA'
},
{
  'OrderID': 10272,
  'CustomerID': 'RATTC',
  'OrderDate': '1996-08-02T00:00:00.000Z',
  'ShippedDate': '1996-08-06T00:00:00.000Z',
  'Freight': 98.03,
  'ShipName': 'Rattlesnake Canyon Grocery',
  'ShipAddress': '2817 Milton Dr.',
  'ShipCity': 'Albuquerque',
  'ShipRegion': 'NM',
  'ShipCountry': 'USA'
},
{
  'OrderID': 10273,
  'CustomerID': 'QUICK',
  'OrderDate': '1996-08-05T00:00:00.000Z',
  'ShippedDate': '1996-08-12T00:00:00.000Z',
  'Freight': 76.07,
  'ShipName': 'QUICK-Stop',
  'ShipAddress': 'Taucherstraße 10',
  'ShipCity': 'Cunewalde',
  'ShipRegion': null,
  'ShipCountry': 'Germany'
},
{
  'OrderID': 10274,
  'CustomerID': 'VINET',
  'OrderDate': '1996-08-06T00:00:00.000Z',
  'ShippedDate': '1996-08-16T00:00:00.000Z',
  'Freight': 6.01,
  'ShipName': 'Vins et alcools Chevalier',
  'ShipAddress': '59 rue de l\'Abbaye',
  'ShipCity': 'Reims',
  'ShipRegion': null,
  'ShipCountry': 'France'
},
{
  'OrderID': 10275,
  'CustomerID': 'MAGAA',
  'OrderDate': '1996-08-07T00:00:00.000Z',
  'ShippedDate': '1996-08-09T00:00:00.000Z',
  'Freight': 26.93,
  'ShipName': 'Magazzini Alimentari Riuniti',
  'ShipAddress': 'Via Ludovico il Moro 22',
  'ShipCity': 'Bergamo',

```

```

    'ShipRegion': null,
    'ShipCountry': 'Italy'
  },
  {
    'OrderID': 10276,
    'CustomerID': 'TORTU',
    'OrderDate': '1996-08-08T00:00:00.000Z',
    'ShippedDate': '1996-08-14T00:00:00.000Z',
    'Freight': 13.84,
    'ShipName': 'Tortuga Restaurante',
    'ShipAddress': 'Avda. Azteca 123',
    'ShipCity': 'México D.F.',
    'ShipRegion': null,
    'ShipCountry': 'Mexico'
  },
  {
    'OrderID': 10277,
    'CustomerID': 'MORGK',
    'OrderDate': '1996-08-09T00:00:00.000Z',
    'ShippedDate': '1996-08-13T00:00:00.000Z',
    'Freight': 125.77,
    'ShipName': 'Morgenstern Gesundkost',
    'ShipAddress': 'Heerstr. 22',
    'ShipCity': 'Leipzig',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10278,
    'CustomerID': 'BERGS',
    'OrderDate': '1996-08-12T00:00:00.000Z',
    'ShippedDate': '1996-08-16T00:00:00.000Z',
    'Freight': 92.69,
    'ShipName': 'Berglunds snabbköp',
    'ShipAddress': 'Berguvsvägen 8',
    'ShipCity': 'Luleå',
    'ShipRegion': null,
    'ShipCountry': 'Sweden'
  },
  {
    'OrderID': 10279,
    'CustomerID': 'LEHMS',
    'OrderDate': '1996-08-13T00:00:00.000Z',
    'ShippedDate': '1996-08-16T00:00:00.000Z',
    'Freight': 25.83,
    'ShipName': 'Lehmanns Marktstand',
    'ShipAddress': 'Magazinweg 7',
    'ShipCity': 'Frankfurt a.M.',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10280,
    'CustomerID': 'BERGS',
    'OrderDate': '1996-08-14T00:00:00.000Z',
    'ShippedDate': '1996-09-12T00:00:00.000Z',
    'Freight': 8.98,

```

```

    'ShipName': 'Berglunds snabbköp',
    'ShipAddress': 'Berguvsvägen 8',
    'ShipCity': 'Luleå',
    'ShipRegion': null,
    'ShipCountry': 'Sweden'
  },
  {
    'OrderID': 10281,
    'CustomerID': 'ROMEY',
    'OrderDate': '1996-08-14T00:00:00.000Z',
    'ShippedDate': '1996-08-21T00:00:00.000Z',
    'Freight': 2.94,
    'ShipName': 'Romero y tomillo',
    'ShipAddress': 'Gran Vía, 1',
    'ShipCity': 'Madrid',
    'ShipRegion': null,
    'ShipCountry': 'Spain'
  },
  {
    'OrderID': 10282,
    'CustomerID': 'ROMEY',
    'OrderDate': '1996-08-15T00:00:00.000Z',
    'ShippedDate': '1996-08-21T00:00:00.000Z',
    'Freight': 12.69,
    'ShipName': 'Romero y tomillo',
    'ShipAddress': 'Gran Vía, 1',
    'ShipCity': 'Madrid',
    'ShipRegion': null,
    'ShipCountry': 'Spain'
  },
  {
    'OrderID': 10283,
    'CustomerID': 'LILAS',
    'OrderDate': '1996-08-16T00:00:00.000Z',
    'ShippedDate': '1996-08-23T00:00:00.000Z',
    'Freight': 84.81,
    'ShipName': 'LILA-Supermercado',
    'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
    'ShipCity': 'Barquisimeto',
    'ShipRegion': 'Lara',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10284,
    'CustomerID': 'LEHMS',
    'OrderDate': '1996-08-19T00:00:00.000Z',
    'ShippedDate': '1996-08-27T00:00:00.000Z',
    'Freight': 76.56,
    'ShipName': 'Lehmanns Marktstand',
    'ShipAddress': 'Magazinweg 7',
    'ShipCity': 'Frankfurt a.M.',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10285,
    'CustomerID': 'QUICK',

```

```

      'OrderDate': '1996-08-20T00:00:00.000Z',
      'ShippedDate': '1996-08-26T00:00:00.000Z',
      'Freight': 76.83,
      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10286,
      'CustomerID': 'QUICK',
      'OrderDate': '1996-08-21T00:00:00.000Z',
      'ShippedDate': '1996-08-30T00:00:00.000Z',
      'Freight': 229.24,
      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10287,
      'CustomerID': 'RICAR',
      'OrderDate': '1996-08-22T00:00:00.000Z',
      'ShippedDate': '1996-08-28T00:00:00.000Z',
      'Freight': 12.76,
      'ShipName': 'Ricardo Adocicados',
      'ShipAddress': 'Av. Copacabana, 267',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10288,
      'CustomerID': 'REGGC',
      'OrderDate': '1996-08-23T00:00:00.000Z',
      'ShippedDate': '1996-09-03T00:00:00.000Z',
      'Freight': 7.45,
      'ShipName': 'Reggiani Caseifici',
      'ShipAddress': 'Strada Provinciale 124',
      'ShipCity': 'Reggio Emilia',
      'ShipRegion': null,
      'ShipCountry': 'Italy'
    },
    {
      'OrderID': 10289,
      'CustomerID': 'BSBEV',
      'OrderDate': '1996-08-26T00:00:00.000Z',
      'ShippedDate': '1996-08-28T00:00:00.000Z',
      'Freight': 22.77,
      'ShipName': 'B\ Beverages',
      'ShipAddress': 'Fauntleroy Circus',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    }
  ],

```

```

{
  'OrderID': 10290,
  'CustomerID': 'COMMI',
  'OrderDate': '1996-08-27T00:00:00.000Z',
  'ShippedDate': '1996-09-03T00:00:00.000Z',
  'Freight': 79.7,
  'ShipName': 'Comércio Mineiro',
  'ShipAddress': 'Av. dos Lusíadas, 23',
  'ShipCity': 'Sao Paulo',
  'ShipRegion': 'SP',
  'ShipCountry': 'Brazil'
},
{
  'OrderID': 10291,
  'CustomerID': 'QUEDE',
  'OrderDate': '1996-08-27T00:00:00.000Z',
  'ShippedDate': '1996-09-04T00:00:00.000Z',
  'Freight': 6.4,
  'ShipName': 'Que Delícia',
  'ShipAddress': 'Rua da Panificadora, 12',
  'ShipCity': 'Rio de Janeiro',
  'ShipRegion': 'RJ',
  'ShipCountry': 'Brazil'
},
{
  'OrderID': 10292,
  'CustomerID': 'TRADH',
  'OrderDate': '1996-08-28T00:00:00.000Z',
  'ShippedDate': '1996-09-02T00:00:00.000Z',
  'Freight': 1.35,
  'ShipName': 'Tradição Hipermercados',
  'ShipAddress': 'Av. Inês de Castro, 414',
  'ShipCity': 'Sao Paulo',
  'ShipRegion': 'SP',
  'ShipCountry': 'Brazil'
},
{
  'OrderID': 10293,
  'CustomerID': 'TORTU',
  'OrderDate': '1996-08-29T00:00:00.000Z',
  'ShippedDate': '1996-09-11T00:00:00.000Z',
  'Freight': 21.18,
  'ShipName': 'Tortuga Restaurante',
  'ShipAddress': 'Avda. Azteca 123',
  'ShipCity': 'México D.F.',
  'ShipRegion': null,
  'ShipCountry': 'Mexico'
},
{
  'OrderID': 10294,
  'CustomerID': 'RATTC',
  'OrderDate': '1996-08-30T00:00:00.000Z',
  'ShippedDate': '1996-09-05T00:00:00.000Z',
  'Freight': 147.26,
  'ShipName': 'Rattlesnake Canyon Grocery',
  'ShipAddress': '2817 Milton Dr.',
  'ShipCity': 'Albuquerque',

```

```

    'ShipRegion': 'NM',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10295,
    'CustomerID': 'VINET',
    'OrderDate': '1996-09-02T00:00:00.000Z',
    'ShippedDate': '1996-09-10T00:00:00.000Z',
    'Freight': 1.15,
    'ShipName': 'Vins et alcools Chevalier',
    'ShipAddress': '59 rue de l\'Abbaye',
    'ShipCity': 'Reims',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10296,
    'CustomerID': 'LILAS',
    'OrderDate': '1996-09-03T00:00:00.000Z',
    'ShippedDate': '1996-09-11T00:00:00.000Z',
    'Freight': 0.12,
    'ShipName': 'LILA-Supermercado',
    'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
    'ShipCity': 'Barquisimeto',
    'ShipRegion': 'Lara',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10297,
    'CustomerID': 'BLONP',
    'OrderDate': '1996-09-04T00:00:00.000Z',
    'ShippedDate': '1996-09-10T00:00:00.000Z',
    'Freight': 5.74,
    'ShipName': 'Blondel père et fils',
    'ShipAddress': '24, place Kléber',
    'ShipCity': 'Strasbourg',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10298,
    'CustomerID': 'HUNGO',
    'OrderDate': '1996-09-05T00:00:00.000Z',
    'ShippedDate': '1996-09-11T00:00:00.000Z',
    'Freight': 168.22,
    'ShipName': 'Hungry Owl All-Night Grocers',
    'ShipAddress': '8 Johnstown Road',
    'ShipCity': 'Cork',
    'ShipRegion': 'Co. Cork',
    'ShipCountry': 'Ireland'
  },
  {
    'OrderID': 10299,
    'CustomerID': 'RICAR',
    'OrderDate': '1996-09-06T00:00:00.000Z',
    'ShippedDate': '1996-09-13T00:00:00.000Z',
    'Freight': 29.76,

```

```

      'ShipName': 'Ricardo Adocicados',
      'ShipAddress': 'Av. Copacabana, 267',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10300,
      'CustomerID': 'MAGAA',
      'OrderDate': '1996-09-09T00:00:00.000Z',
      'ShippedDate': '1996-09-18T00:00:00.000Z',
      'Freight': 17.68,
      'ShipName': 'Magazzini Alimentari Riuniti',
      'ShipAddress': 'Via Ludovico il Moro 22',
      'ShipCity': 'Bergamo',
      'ShipRegion': null,
      'ShipCountry': 'Italy'
    },
    {
      'OrderID': 10301,
      'CustomerID': 'WANDK',
      'OrderDate': '1996-09-09T00:00:00.000Z',
      'ShippedDate': '1996-09-17T00:00:00.000Z',
      'Freight': 45.08,
      'ShipName': 'Die Wandernde Kuh',
      'ShipAddress': 'Adenauerallee 900',
      'ShipCity': 'Stuttgart',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10302,
      'CustomerID': 'SUPRD',
      'OrderDate': '1996-09-10T00:00:00.000Z',
      'ShippedDate': '1996-10-09T00:00:00.000Z',
      'Freight': 6.27,
      'ShipName': 'Suprêmes délices',
      'ShipAddress': 'Boulevard Tirou, 255',
      'ShipCity': 'Charleroi',
      'ShipRegion': null,
      'ShipCountry': 'Belgium'
    },
    {
      'OrderID': 10303,
      'CustomerID': 'GODOS',
      'OrderDate': '1996-09-11T00:00:00.000Z',
      'ShippedDate': '1996-09-18T00:00:00.000Z',
      'Freight': 107.83,
      'ShipName': 'Godos Cocina Típica',
      'ShipAddress': 'C/ Romero, 33',
      'ShipCity': 'Sevilla',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    },
    {
      'OrderID': 10304,
      'CustomerID': 'TORTU',

```



```

      'OrderDate': '1996-09-12T00:00:00.000Z',
      'ShippedDate': '1996-09-17T00:00:00.000Z',
      'Freight': 63.79,
      'ShipName': 'Tortuga Restaurante',
      'ShipAddress': 'Avda. Azteca 123',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10305,
      'CustomerID': 'OLDWO',
      'OrderDate': '1996-09-13T00:00:00.000Z',
      'ShippedDate': '1996-10-09T00:00:00.000Z',
      'Freight': 257.62,
      'ShipName': 'Old World Delicatessen',
      'ShipAddress': '2743 Bering St.',
      'ShipCity': 'Anchorage',
      'ShipRegion': 'AK',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10306,
      'CustomerID': 'ROMEY',
      'OrderDate': '1996-09-16T00:00:00.000Z',
      'ShippedDate': '1996-09-23T00:00:00.000Z',
      'Freight': 7.56,
      'ShipName': 'Romero y tomillo',
      'ShipAddress': 'Gran Vía, 1',
      'ShipCity': 'Madrid',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    },
    {
      'OrderID': 10307,
      'CustomerID': 'LONEP',
      'OrderDate': '1996-09-17T00:00:00.000Z',
      'ShippedDate': '1996-09-25T00:00:00.000Z',
      'Freight': 0.56,
      'ShipName': 'Lonesome Pine Restaurant',
      'ShipAddress': '89 Chiaroscuro Rd.',
      'ShipCity': 'Portland',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10308,
      'CustomerID': 'ANATR',
      'OrderDate': '1996-09-18T00:00:00.000Z',
      'ShippedDate': '1996-09-24T00:00:00.000Z',
      'Freight': 1.61,
      'ShipName': 'Ana Trujillo Emparedados y helados',
      'ShipAddress': 'Avda. de la Constitución 2222',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    }
  ],

```

```

{
  'OrderID': 10309,
  'CustomerID': 'HUNGO',
  'OrderDate': '1996-09-19T00:00:00.000Z',
  'ShippedDate': '1996-10-23T00:00:00.000Z',
  'Freight': 47.3,
  'ShipName': 'Hungry Owl All-Night Grocers',
  'ShipAddress': '8 Johnstown Road',
  'ShipCity': 'Cork',
  'ShipRegion': 'Co. Cork',
  'ShipCountry': 'Ireland'
},
{
  'OrderID': 10310,
  'CustomerID': 'THEBI',
  'OrderDate': '1996-09-20T00:00:00.000Z',
  'ShippedDate': '1996-09-27T00:00:00.000Z',
  'Freight': 17.52,
  'ShipName': 'The Big Cheese',
  'ShipAddress': '89 Jefferson Way Suite 2',
  'ShipCity': 'Portland',
  'ShipRegion': 'OR',
  'ShipCountry': 'USA'
},
{
  'OrderID': 10311,
  'CustomerID': 'DUMON',
  'OrderDate': '1996-09-20T00:00:00.000Z',
  'ShippedDate': '1996-09-26T00:00:00.000Z',
  'Freight': 24.69,
  'ShipName': 'Du monde entier',
  'ShipAddress': '67, rue des Cinquante Otages',
  'ShipCity': 'Nantes',
  'ShipRegion': null,
  'ShipCountry': 'France'
},
{
  'OrderID': 10312,
  'CustomerID': 'WANDK',
  'OrderDate': '1996-09-23T00:00:00.000Z',
  'ShippedDate': '1996-10-03T00:00:00.000Z',
  'Freight': 40.26,
  'ShipName': 'Die Wandernde Kuh',
  'ShipAddress': 'Adenauerallee 900',
  'ShipCity': 'Stuttgart',
  'ShipRegion': null,
  'ShipCountry': 'Germany'
},
{
  'OrderID': 10313,
  'CustomerID': 'QUICK',
  'OrderDate': '1996-09-24T00:00:00.000Z',
  'ShippedDate': '1996-10-04T00:00:00.000Z',
  'Freight': 1.96,
  'ShipName': 'QUICK-Stop',
  'ShipAddress': 'Taucherstraße 10',
  'ShipCity': 'Cunewalde',

```

```

    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10314,
    'CustomerID': 'RATTC',
    'OrderDate': '1996-09-25T00:00:00.000Z',
    'ShippedDate': '1996-10-04T00:00:00.000Z',
    'Freight': 74.16,
    'ShipName': 'Rattlesnake Canyon Grocery',
    'ShipAddress': '2817 Milton Dr.',
    'ShipCity': 'Albuquerque',
    'ShipRegion': 'NM',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10315,
    'CustomerID': 'ISLAT',
    'OrderDate': '1996-09-26T00:00:00.000Z',
    'ShippedDate': '1996-10-03T00:00:00.000Z',
    'Freight': 41.76,
    'ShipName': 'Island Trading',
    'ShipAddress': 'Garden House Crowther Way',
    'ShipCity': 'Cowes',
    'ShipRegion': 'Isle of Wight',
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10316,
    'CustomerID': 'RATTC',
    'OrderDate': '1996-09-27T00:00:00.000Z',
    'ShippedDate': '1996-10-08T00:00:00.000Z',
    'Freight': 150.15,
    'ShipName': 'Rattlesnake Canyon Grocery',
    'ShipAddress': '2817 Milton Dr.',
    'ShipCity': 'Albuquerque',
    'ShipRegion': 'NM',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10317,
    'CustomerID': 'LONEP',
    'OrderDate': '1996-09-30T00:00:00.000Z',
    'ShippedDate': '1996-10-10T00:00:00.000Z',
    'Freight': 12.69,
    'ShipName': 'Lonesome Pine Restaurant',
    'ShipAddress': '89 Chiaroscuro Rd.',
    'ShipCity': 'Portland',
    'ShipRegion': 'OR',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10318,
    'CustomerID': 'ISLAT',
    'OrderDate': '1996-10-01T00:00:00.000Z',
    'ShippedDate': '1996-10-04T00:00:00.000Z',
    'Freight': 4.73,

```

```

      'ShipName': 'Island Trading',
      'ShipAddress': 'Garden House Crowther Way',
      'ShipCity': 'Cowes',
      'ShipRegion': 'Isle of Wight',
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10319,
      'CustomerID': 'TORTU',
      'OrderDate': '1996-10-02T00:00:00.000Z',
      'ShippedDate': '1996-10-11T00:00:00.000Z',
      'Freight': 64.5,
      'ShipName': 'Tortuga Restaurante',
      'ShipAddress': 'Avda. Azteca 123',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10320,
      'CustomerID': 'WARTH',
      'OrderDate': '1996-10-03T00:00:00.000Z',
      'ShippedDate': '1996-10-18T00:00:00.000Z',
      'Freight': 34.57,
      'ShipName': 'Wartian Herkku',
      'ShipAddress': 'Torikatu 38',
      'ShipCity': 'Oulu',
      'ShipRegion': null,
      'ShipCountry': 'Finland'
    },
    {
      'OrderID': 10321,
      'CustomerID': 'ISLAT',
      'OrderDate': '1996-10-03T00:00:00.000Z',
      'ShippedDate': '1996-10-11T00:00:00.000Z',
      'Freight': 3.43,
      'ShipName': 'Island Trading',
      'ShipAddress': 'Garden House Crowther Way',
      'ShipCity': 'Cowes',
      'ShipRegion': 'Isle of Wight',
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10322,
      'CustomerID': 'PERIC',
      'OrderDate': '1996-10-04T00:00:00.000Z',
      'ShippedDate': '1996-10-23T00:00:00.000Z',
      'Freight': 0.4,
      'ShipName': 'Pericles Comidas clásicas',
      'ShipAddress': 'Calle Dr. Jorge Cash 321',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10323,
      'CustomerID': 'KOENE',

```

```

      'OrderDate': '1996-10-07T00:00:00.000Z',
      'ShippedDate': '1996-10-14T00:00:00.000Z',
      'Freight': 4.88,
      'ShipName': 'Königlich Essen',
      'ShipAddress': 'Maubelstr. 90',
      'ShipCity': 'Brandenburg',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10324,
      'CustomerID': 'SAVEA',
      'OrderDate': '1996-10-08T00:00:00.000Z',
      'ShippedDate': '1996-10-10T00:00:00.000Z',
      'Freight': 214.27,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10325,
      'CustomerID': 'KOENE',
      'OrderDate': '1996-10-09T00:00:00.000Z',
      'ShippedDate': '1996-10-14T00:00:00.000Z',
      'Freight': 64.86,
      'ShipName': 'Königlich Essen',
      'ShipAddress': 'Maubelstr. 90',
      'ShipCity': 'Brandenburg',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10326,
      'CustomerID': 'BOLID',
      'OrderDate': '1996-10-10T00:00:00.000Z',
      'ShippedDate': '1996-10-14T00:00:00.000Z',
      'Freight': 77.92,
      'ShipName': 'Bólido Comidas preparadas',
      'ShipAddress': 'C/ Araquil, 67',
      'ShipCity': 'Madrid',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    },
    {
      'OrderID': 10327,
      'CustomerID': 'FOLKO',
      'OrderDate': '1996-10-11T00:00:00.000Z',
      'ShippedDate': '1996-10-14T00:00:00.000Z',
      'Freight': 63.36,
      'ShipName': 'Folk och fä HB',
      'ShipAddress': 'Åkergatan 24',
      'ShipCity': 'Bräcke',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    }
  ],

```

```

{
  'OrderID': 10328,
  'CustomerID': 'FURIB',
  'OrderDate': '1996-10-14T00:00:00.000Z',
  'ShippedDate': '1996-10-17T00:00:00.000Z',
  'Freight': 87.03,
  'ShipName': 'Furia Bacalhau e Frutos do Mar',
  'ShipAddress': 'Jardim das rosas n. 32',
  'ShipCity': 'Lisboa',
  'ShipRegion': null,
  'ShipCountry': 'Portugal'
},
{
  'OrderID': 10329,
  'CustomerID': 'SPLIR',
  'OrderDate': '1996-10-15T00:00:00.000Z',
  'ShippedDate': '1996-10-23T00:00:00.000Z',
  'Freight': 191.67,
  'ShipName': 'Split Rail Beer & Ale',
  'ShipAddress': 'P.O. Box 555',
  'ShipCity': 'Lander',
  'ShipRegion': 'WY',
  'ShipCountry': 'USA'
},
{
  'OrderID': 10330,
  'CustomerID': 'LILAS',
  'OrderDate': '1996-10-16T00:00:00.000Z',
  'ShippedDate': '1996-10-28T00:00:00.000Z',
  'Freight': 12.75,
  'ShipName': 'LILA-Supermercado',
  'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
  'ShipCity': 'Barquisimeto',
  'ShipRegion': 'Lara',
  'ShipCountry': 'Venezuela'
},
{
  'OrderID': 10331,
  'CustomerID': 'BONAP',
  'OrderDate': '1996-10-16T00:00:00.000Z',
  'ShippedDate': '1996-10-21T00:00:00.000Z',
  'Freight': 10.19,
  'ShipName': 'Bon app',
  'ShipAddress': '12, rue des Bouchers',
  'ShipCity': 'Marseille',
  'ShipRegion': null,
  'ShipCountry': 'France'
},
{
  'OrderID': 10332,
  'CustomerID': 'MEREP',
  'OrderDate': '1996-10-17T00:00:00.000Z',
  'ShippedDate': '1996-10-21T00:00:00.000Z',
  'Freight': 52.84,
  'ShipName': 'Mère Paillarde',
  'ShipAddress': '43 rue St. Laurent',
  'ShipCity': 'Montréal',

```

```

      'ShipRegion': 'Québec',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10333,
      'CustomerID': 'WARTH',
      'OrderDate': '1996-10-18T00:00:00.000Z',
      'ShippedDate': '1996-10-25T00:00:00.000Z',
      'Freight': 0.59,
      'ShipName': 'Wartian Herkku',
      'ShipAddress': 'Torikatu 38',
      'ShipCity': 'Oulu',
      'ShipRegion': null,
      'ShipCountry': 'Finland'
    },
    {
      'OrderID': 10334,
      'CustomerID': 'VICTE',
      'OrderDate': '1996-10-21T00:00:00.000Z',
      'ShippedDate': '1996-10-28T00:00:00.000Z',
      'Freight': 8.56,
      'ShipName': 'Victuailles en stock',
      'ShipAddress': '2, rue du Commerce',
      'ShipCity': 'Lyon',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10335,
      'CustomerID': 'HUNGO',
      'OrderDate': '1996-10-22T00:00:00.000Z',
      'ShippedDate': '1996-10-24T00:00:00.000Z',
      'Freight': 42.11,
      'ShipName': 'Hungry Owl All-Night Grocers',
      'ShipAddress': '8 Johnstown Road',
      'ShipCity': 'Cork',
      'ShipRegion': 'Co. Cork',
      'ShipCountry': 'Ireland'
    },
    {
      'OrderID': 10336,
      'CustomerID': 'PRINI',
      'OrderDate': '1996-10-23T00:00:00.000Z',
      'ShippedDate': '1996-10-25T00:00:00.000Z',
      'Freight': 15.51,
      'ShipName': 'Princesa Isabel Vinhos',
      'ShipAddress': 'Estrada da saúde n. 58',
      'ShipCity': 'Lisboa',
      'ShipRegion': null,
      'ShipCountry': 'Portugal'
    },
    {
      'OrderID': 10337,
      'CustomerID': 'FRANK',
      'OrderDate': '1996-10-24T00:00:00.000Z',
      'ShippedDate': '1996-10-29T00:00:00.000Z',
      'Freight': 108.26,

```

```

    'ShipName': 'Frankenversand',
    'ShipAddress': 'Berliner Platz 43',
    'ShipCity': 'München',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10338,
    'CustomerID': 'OLDWO',
    'OrderDate': '1996-10-25T00:00:00.000Z',
    'ShippedDate': '1996-10-29T00:00:00.000Z',
    'Freight': 84.21,
    'ShipName': 'Old World Delicatessen',
    'ShipAddress': '2743 Bering St.',
    'ShipCity': 'Anchorage',
    'ShipRegion': 'AK',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10339,
    'CustomerID': 'MEREP',
    'OrderDate': '1996-10-28T00:00:00.000Z',
    'ShippedDate': '1996-11-04T00:00:00.000Z',
    'Freight': 15.66,
    'ShipName': 'Mère Paillarde',
    'ShipAddress': '43 rue St. Laurent',
    'ShipCity': 'Montréal',
    'ShipRegion': 'Québec',
    'ShipCountry': 'Canada'
  },
  {
    'OrderID': 10340,
    'CustomerID': 'BONAP',
    'OrderDate': '1996-10-29T00:00:00.000Z',
    'ShippedDate': '1996-11-08T00:00:00.000Z',
    'Freight': 166.31,
    'ShipName': 'Bon app',
    'ShipAddress': '12, rue des Bouchers',
    'ShipCity': 'Marseille',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10341,
    'CustomerID': 'SIMOB',
    'OrderDate': '1996-10-29T00:00:00.000Z',
    'ShippedDate': '1996-11-05T00:00:00.000Z',
    'Freight': 26.78,
    'ShipName': 'Simons bistro',
    'ShipAddress': 'Vinbæltet 34',
    'ShipCity': 'Kobenhavn',
    'ShipRegion': null,
    'ShipCountry': 'Denmark'
  },
  {
    'OrderID': 10342,
    'CustomerID': 'FRANK',

```



```

      'OrderDate': '1996-10-30T00:00:00.000Z',
      'ShippedDate': '1996-11-04T00:00:00.000Z',
      'Freight': 54.83,
      'ShipName': 'Frankenversand',
      'ShipAddress': 'Berliner Platz 43',
      'ShipCity': 'München',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10343,
      'CustomerID': 'LEHMS',
      'OrderDate': '1996-10-31T00:00:00.000Z',
      'ShippedDate': '1996-11-06T00:00:00.000Z',
      'Freight': 110.37,
      'ShipName': 'Lehmanns Marktstand',
      'ShipAddress': 'Magazinweg 7',
      'ShipCity': 'Frankfurt a.M.',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10344,
      'CustomerID': 'WHITC',
      'OrderDate': '1996-11-01T00:00:00.000Z',
      'ShippedDate': '1996-11-05T00:00:00.000Z',
      'Freight': 23.29,
      'ShipName': 'White Clover Markets',
      'ShipAddress': '1029 - 12th Ave. S.',
      'ShipCity': 'Seattle',
      'ShipRegion': 'WA',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10345,
      'CustomerID': 'QUICK',
      'OrderDate': '1996-11-04T00:00:00.000Z',
      'ShippedDate': '1996-11-11T00:00:00.000Z',
      'Freight': 249.06,
      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10346,
      'CustomerID': 'RATTC',
      'OrderDate': '1996-11-05T00:00:00.000Z',
      'ShippedDate': '1996-11-08T00:00:00.000Z',
      'Freight': 142.08,
      'ShipName': 'Rattlesnake Canyon Grocery',
      'ShipAddress': '2817 Milton Dr.',
      'ShipCity': 'Albuquerque',
      'ShipRegion': 'NM',
      'ShipCountry': 'USA'
    }
  ],

```

```

{
  'OrderID': 10347,
  'CustomerID': 'FAMIA',
  'OrderDate': '1996-11-06T00:00:00.000Z',
  'ShippedDate': '1996-11-08T00:00:00.000Z',
  'Freight': 3.1,
  'ShipName': 'Familia Arquibaldo',
  'ShipAddress': 'Rua Orós, 92',
  'ShipCity': 'Sao Paulo',
  'ShipRegion': 'SP',
  'ShipCountry': 'Brazil'
},
{
  'OrderID': 10348,
  'CustomerID': 'WANDK',
  'OrderDate': '1996-11-07T00:00:00.000Z',
  'ShippedDate': '1996-11-15T00:00:00.000Z',
  'Freight': 0.78,
  'ShipName': 'Die Wandernde Kuh',
  'ShipAddress': 'Adenauerallee 900',
  'ShipCity': 'Stuttgart',
  'ShipRegion': null,
  'ShipCountry': 'Germany'
},
{
  'OrderID': 10349,
  'CustomerID': 'SPLIR',
  'OrderDate': '1996-11-08T00:00:00.000Z',
  'ShippedDate': '1996-11-15T00:00:00.000Z',
  'Freight': 8.63,
  'ShipName': 'Split Rail Beer & Ale',
  'ShipAddress': 'P.O. Box 555',
  'ShipCity': 'Lander',
  'ShipRegion': 'WY',
  'ShipCountry': 'USA'
},
{
  'OrderID': 10350,
  'CustomerID': 'LAMAI',
  'OrderDate': '1996-11-11T00:00:00.000Z',
  'ShippedDate': '1996-12-03T00:00:00.000Z',
  'Freight': 64.19,
  'ShipName': 'La maison d\'Asie',
  'ShipAddress': '1 rue Alsace-Lorraine',
  'ShipCity': 'Toulouse',
  'ShipRegion': null,
  'ShipCountry': 'France'
},
{
  'OrderID': 10351,
  'CustomerID': 'ERNSH',
  'OrderDate': '1996-11-11T00:00:00.000Z',
  'ShippedDate': '1996-11-20T00:00:00.000Z',
  'Freight': 162.33,
  'ShipName': 'Ernst Handel',
  'ShipAddress': 'Kirchgasse 6',
  'ShipCity': 'Graz',

```

```

      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10352,
      'CustomerID': 'FURIB',
      'OrderDate': '1996-11-12T00:00:00.000Z',
      'ShippedDate': '1996-11-18T00:00:00.000Z',
      'Freight': 1.3,
      'ShipName': 'Furia Bacalhau e Frutos do Mar',
      'ShipAddress': 'Jardim das rosas n. 32',
      'ShipCity': 'Lisboa',
      'ShipRegion': null,
      'ShipCountry': 'Portugal'
    },
    {
      'OrderID': 10353,
      'CustomerID': 'PICCO',
      'OrderDate': '1996-11-13T00:00:00.000Z',
      'ShippedDate': '1996-11-25T00:00:00.000Z',
      'Freight': 360.63,
      'ShipName': 'Piccolo und mehr',
      'ShipAddress': 'Geislweg 14',
      'ShipCity': 'Salzburg',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10354,
      'CustomerID': 'PERIC',
      'OrderDate': '1996-11-14T00:00:00.000Z',
      'ShippedDate': '1996-11-20T00:00:00.000Z',
      'Freight': 53.8,
      'ShipName': 'Pericles Comidas clásicas',
      'ShipAddress': 'Calle Dr. Jorge Cash 321',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10355,
      'CustomerID': 'AROUT',
      'OrderDate': '1996-11-15T00:00:00.000Z',
      'ShippedDate': '1996-11-20T00:00:00.000Z',
      'Freight': 41.95,
      'ShipName': 'Around the Horn',
      'ShipAddress': 'Brook Farm Stratford St. Mary',
      'ShipCity': 'Colchester',
      'ShipRegion': 'Essex',
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10356,
      'CustomerID': 'WANDK',
      'OrderDate': '1996-11-18T00:00:00.000Z',
      'ShippedDate': '1996-11-27T00:00:00.000Z',
      'Freight': 36.71,

```

```

      'ShipName': 'Die Wandernde Kuh',
      'ShipAddress': 'Adenauerallee 900',
      'ShipCity': 'Stuttgart',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10357,
      'CustomerID': 'LILAS',
      'OrderDate': '1996-11-19T00:00:00.000Z',
      'ShippedDate': '1996-12-02T00:00:00.000Z',
      'Freight': 34.88,
      'ShipName': 'LILA-Supermercado',
      'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
      'ShipCity': 'Barquisimeto',
      'ShipRegion': 'Lara',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10358,
      'CustomerID': 'LAMAI',
      'OrderDate': '1996-11-20T00:00:00.000Z',
      'ShippedDate': '1996-11-27T00:00:00.000Z',
      'Freight': 19.64,
      'ShipName': 'La maison d\'Asie',
      'ShipAddress': '1 rue Alsace-Lorraine',
      'ShipCity': 'Toulouse',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10359,
      'CustomerID': 'SEVES',
      'OrderDate': '1996-11-21T00:00:00.000Z',
      'ShippedDate': '1996-11-26T00:00:00.000Z',
      'Freight': 288.43,
      'ShipName': 'Seven Seas Imports',
      'ShipAddress': '90 Wadhurst Rd.',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10360,
      'CustomerID': 'BLONP',
      'OrderDate': '1996-11-22T00:00:00.000Z',
      'ShippedDate': '1996-12-02T00:00:00.000Z',
      'Freight': 131.7,
      'ShipName': 'Blondel père et fils',
      'ShipAddress': '24, place Kléber',
      'ShipCity': 'Strasbourg',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10361,
      'CustomerID': 'QUICK',

```

```

      'OrderDate': '1996-11-22T00:00:00.000Z',
      'ShippedDate': '1996-12-03T00:00:00.000Z',
      'Freight': 183.17,
      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10362,
      'CustomerID': 'BONAP',
      'OrderDate': '1996-11-25T00:00:00.000Z',
      'ShippedDate': '1996-11-28T00:00:00.000Z',
      'Freight': 96.04,
      'ShipName': 'Bon app',
      'ShipAddress': '12, rue des Bouchers',
      'ShipCity': 'Marseille',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10363,
      'CustomerID': 'DRACD',
      'OrderDate': '1996-11-26T00:00:00.000Z',
      'ShippedDate': '1996-12-04T00:00:00.000Z',
      'Freight': 30.54,
      'ShipName': 'Drachenblut Delikatessen',
      'ShipAddress': 'Walserweg 21',
      'ShipCity': 'Aachen',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10364,
      'CustomerID': 'EASTC',
      'OrderDate': '1996-11-26T00:00:00.000Z',
      'ShippedDate': '1996-12-04T00:00:00.000Z',
      'Freight': 71.97,
      'ShipName': 'Eastern Connection',
      'ShipAddress': '35 King George',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10365,
      'CustomerID': 'ANTON',
      'OrderDate': '1996-11-27T00:00:00.000Z',
      'ShippedDate': '1996-12-02T00:00:00.000Z',
      'Freight': 22,
      'ShipName': 'Antonio Moreno Taquería',
      'ShipAddress': 'Mataderos 2312',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    }
  ],

```

```

{
  'OrderID': 10366,
  'CustomerID': 'GALED',
  'OrderDate': '1996-11-28T00:00:00.000Z',
  'ShippedDate': '1996-12-30T00:00:00.000Z',
  'Freight': 10.14,
  'ShipName': 'Galería del gastronómo',
  'ShipAddress': 'Rambla de Cataluña, 23',
  'ShipCity': 'Barcelona',
  'ShipRegion': null,
  'ShipCountry': 'Spain'
},
{
  'OrderID': 10367,
  'CustomerID': 'VAFFE',
  'OrderDate': '1996-11-28T00:00:00.000Z',
  'ShippedDate': '1996-12-02T00:00:00.000Z',
  'Freight': 13.55,
  'ShipName': 'Vaffeljernet',
  'ShipAddress': 'Smagsloget 45',
  'ShipCity': 'Århus',
  'ShipRegion': null,
  'ShipCountry': 'Denmark'
},
{
  'OrderID': 10368,
  'CustomerID': 'ERNSH',
  'OrderDate': '1996-11-29T00:00:00.000Z',
  'ShippedDate': '1996-12-02T00:00:00.000Z',
  'Freight': 101.95,
  'ShipName': 'Ernst Handel',
  'ShipAddress': 'Kirchgasse 6',
  'ShipCity': 'Graz',
  'ShipRegion': null,
  'ShipCountry': 'Austria'
},
{
  'OrderID': 10369,
  'CustomerID': 'SPLIR',
  'OrderDate': '1996-12-02T00:00:00.000Z',
  'ShippedDate': '1996-12-09T00:00:00.000Z',
  'Freight': 195.68,
  'ShipName': 'Split Rail Beer & Ale',
  'ShipAddress': 'P.O. Box 555',
  'ShipCity': 'Lander',
  'ShipRegion': 'WY',
  'ShipCountry': 'USA'
},
{
  'OrderID': 10370,
  'CustomerID': 'CHOPS',
  'OrderDate': '1996-12-03T00:00:00.000Z',
  'ShippedDate': '1996-12-27T00:00:00.000Z',
  'Freight': 1.17,
  'ShipName': 'Chop-suey Chinese',
  'ShipAddress': 'Hauptstr. 31',
  'ShipCity': 'Bern',

```

```

    'ShipRegion': null,
    'ShipCountry': 'Switzerland'
  },
  {
    'OrderID': 10371,
    'CustomerID': 'LAMAI',
    'OrderDate': '1996-12-03T00:00:00.000Z',
    'ShippedDate': '1996-12-24T00:00:00.000Z',
    'Freight': 0.45,
    'ShipName': 'La maison d\'Asie',
    'ShipAddress': '1 rue Alsace-Lorraine',
    'ShipCity': 'Toulouse',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10372,
    'CustomerID': 'QUEEN',
    'OrderDate': '1996-12-04T00:00:00.000Z',
    'ShippedDate': '1996-12-09T00:00:00.000Z',
    'Freight': 890.78,
    'ShipName': 'Queen Cozinha',
    'ShipAddress': 'Alameda dos Canários, 891',
    'ShipCity': 'Sao Paulo',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10373,
    'CustomerID': 'HUNGO',
    'OrderDate': '1996-12-05T00:00:00.000Z',
    'ShippedDate': '1996-12-11T00:00:00.000Z',
    'Freight': 124.12,
    'ShipName': 'Hungry Owl All-Night Grocers',
    'ShipAddress': '8 Johnstown Road',
    'ShipCity': 'Cork',
    'ShipRegion': 'Co. Cork',
    'ShipCountry': 'Ireland'
  },
  {
    'OrderID': 10374,
    'CustomerID': 'WOLZA',
    'OrderDate': '1996-12-05T00:00:00.000Z',
    'ShippedDate': '1996-12-09T00:00:00.000Z',
    'Freight': 3.94,
    'ShipName': 'Wolski Zajazd',
    'ShipAddress': 'ul. Filtrowa 68',
    'ShipCity': 'Warszawa',
    'ShipRegion': null,
    'ShipCountry': 'Poland'
  },
  {
    'OrderID': 10375,
    'CustomerID': 'HUNGC',
    'OrderDate': '1996-12-06T00:00:00.000Z',
    'ShippedDate': '1996-12-09T00:00:00.000Z',
    'Freight': 20.12,

```

```

    'ShipName': 'Hungry Coyote Import Store',
    'ShipAddress': 'City Center Plaza 516 Main St.',
    'ShipCity': 'Elgin',
    'ShipRegion': 'OR',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10376,
    'CustomerID': 'MEREP',
    'OrderDate': '1996-12-09T00:00:00.000Z',
    'ShippedDate': '1996-12-13T00:00:00.000Z',
    'Freight': 20.39,
    'ShipName': 'Mère Paillarde',
    'ShipAddress': '43 rue St. Laurent',
    'ShipCity': 'Montréal',
    'ShipRegion': 'Québec',
    'ShipCountry': 'Canada'
  },
  {
    'OrderID': 10377,
    'CustomerID': 'SEVES',
    'OrderDate': '1996-12-09T00:00:00.000Z',
    'ShippedDate': '1996-12-13T00:00:00.000Z',
    'Freight': 22.21,
    'ShipName': 'Seven Seas Imports',
    'ShipAddress': '90 Wadhurst Rd.',
    'ShipCity': 'London',
    'ShipRegion': null,
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10378,
    'CustomerID': 'FOLKO',
    'OrderDate': '1996-12-10T00:00:00.000Z',
    'ShippedDate': '1996-12-19T00:00:00.000Z',
    'Freight': 5.44,
    'ShipName': 'Folk och fä HB',
    'ShipAddress': 'Åkergatan 24',
    'ShipCity': 'Bräcke',
    'ShipRegion': null,
    'ShipCountry': 'Sweden'
  },
  {
    'OrderID': 10379,
    'CustomerID': 'QUEDE',
    'OrderDate': '1996-12-11T00:00:00.000Z',
    'ShippedDate': '1996-12-13T00:00:00.000Z',
    'Freight': 45.03,
    'ShipName': 'Que Delícia',
    'ShipAddress': 'Rua da Panificadora, 12',
    'ShipCity': 'Rio de Janeiro',
    'ShipRegion': 'RJ',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10380,
    'CustomerID': 'HUNGO',

```



```

      'OrderDate': '1996-12-12T00:00:00.000Z',
      'ShippedDate': '1997-01-16T00:00:00.000Z',
      'Freight': 35.03,
      'ShipName': 'Hungry Owl All-Night Grocers',
      'ShipAddress': '8 Johnstown Road',
      'ShipCity': 'Cork',
      'ShipRegion': 'Co. Cork',
      'ShipCountry': 'Ireland'
    },
    {
      'OrderID': 10381,
      'CustomerID': 'LILAS',
      'OrderDate': '1996-12-12T00:00:00.000Z',
      'ShippedDate': '1996-12-13T00:00:00.000Z',
      'Freight': 7.99,
      'ShipName': 'LILA-Supermercado',
      'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
      'ShipCity': 'Barquisimeto',
      'ShipRegion': 'Lara',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10382,
      'CustomerID': 'ERNSH',
      'OrderDate': '1996-12-13T00:00:00.000Z',
      'ShippedDate': '1996-12-16T00:00:00.000Z',
      'Freight': 94.77,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10383,
      'CustomerID': 'AROUT',
      'OrderDate': '1996-12-16T00:00:00.000Z',
      'ShippedDate': '1996-12-18T00:00:00.000Z',
      'Freight': 34.24,
      'ShipName': 'Around the Horn',
      'ShipAddress': 'Brook Farm Stratford St. Mary',
      'ShipCity': 'Colchester',
      'ShipRegion': 'Essex',
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10384,
      'CustomerID': 'BERGS',
      'OrderDate': '1996-12-16T00:00:00.000Z',
      'ShippedDate': '1996-12-20T00:00:00.000Z',
      'Freight': 168.64,
      'ShipName': 'Berglunds snabbköp',
      'ShipAddress': 'Berguvsvägen 8',
      'ShipCity': 'Luleå',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    }
  ],

```

```

{
  'OrderID': 10385,
  'CustomerID': 'SPLIR',
  'OrderDate': '1996-12-17T00:00:00.000Z',
  'ShippedDate': '1996-12-23T00:00:00.000Z',
  'Freight': 30.96,
  'ShipName': 'Split Rail Beer & Ale',
  'ShipAddress': 'P.O. Box 555',
  'ShipCity': 'Lander',
  'ShipRegion': 'WY',
  'ShipCountry': 'USA'
},
{
  'OrderID': 10386,
  'CustomerID': 'FAMIA',
  'OrderDate': '1996-12-18T00:00:00.000Z',
  'ShippedDate': '1996-12-25T00:00:00.000Z',
  'Freight': 13.99,
  'ShipName': 'Familia Arquibaldo',
  'ShipAddress': 'Rua Orós, 92',
  'ShipCity': 'Sao Paulo',
  'ShipRegion': 'SP',
  'ShipCountry': 'Brazil'
},
{
  'OrderID': 10387,
  'CustomerID': 'SANTG',
  'OrderDate': '1996-12-18T00:00:00.000Z',
  'ShippedDate': '1996-12-20T00:00:00.000Z',
  'Freight': 93.63,
  'ShipName': 'Santé Gourmet',
  'ShipAddress': 'Erling Skakkes gate 78',
  'ShipCity': 'Stavern',
  'ShipRegion': null,
  'ShipCountry': 'Norway'
},
{
  'OrderID': 10388,
  'CustomerID': 'SEVES',
  'OrderDate': '1996-12-19T00:00:00.000Z',
  'ShippedDate': '1996-12-20T00:00:00.000Z',
  'Freight': 34.86,
  'ShipName': 'Seven Seas Imports',
  'ShipAddress': '90 Wadhurst Rd.',
  'ShipCity': 'London',
  'ShipRegion': null,
  'ShipCountry': 'UK'
},
{
  'OrderID': 10389,
  'CustomerID': 'BOTTM',
  'OrderDate': '1996-12-20T00:00:00.000Z',
  'ShippedDate': '1996-12-24T00:00:00.000Z',
  'Freight': 47.42,
  'ShipName': 'Bottom-Dollar Markets',
  'ShipAddress': '23 Tsawassen Blvd.',
  'ShipCity': 'Tsawassen',

```

```

      'ShipRegion': 'BC',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10390,
      'CustomerID': 'ERNSH',
      'OrderDate': '1996-12-23T00:00:00.000Z',
      'ShippedDate': '1996-12-26T00:00:00.000Z',
      'Freight': 126.38,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10391,
      'CustomerID': 'DRACD',
      'OrderDate': '1996-12-23T00:00:00.000Z',
      'ShippedDate': '1996-12-31T00:00:00.000Z',
      'Freight': 5.45,
      'ShipName': 'Drachenblut Delikatessen',
      'ShipAddress': 'Walserweg 21',
      'ShipCity': 'Aachen',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10392,
      'CustomerID': 'PICCO',
      'OrderDate': '1996-12-24T00:00:00.000Z',
      'ShippedDate': '1997-01-01T00:00:00.000Z',
      'Freight': 122.46,
      'ShipName': 'Piccolo und mehr',
      'ShipAddress': 'Geislweg 14',
      'ShipCity': 'Salzburg',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10393,
      'CustomerID': 'SAVEA',
      'OrderDate': '1996-12-25T00:00:00.000Z',
      'ShippedDate': '1997-01-03T00:00:00.000Z',
      'Freight': 126.56,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10394,
      'CustomerID': 'HUNGC',
      'OrderDate': '1996-12-25T00:00:00.000Z',
      'ShippedDate': '1997-01-03T00:00:00.000Z',
      'Freight': 30.34,

```

```

    'ShipName': 'Hungry Coyote Import Store',
    'ShipAddress': 'City Center Plaza 516 Main St.',
    'ShipCity': 'Elgin',
    'ShipRegion': 'OR',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10395,
    'CustomerID': 'HILAA',
    'OrderDate': '1996-12-26T00:00:00.000Z',
    'ShippedDate': '1997-01-03T00:00:00.000Z',
    'Freight': 184.41,
    'ShipName': 'HILARION-Abastos',
    'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
    'ShipCity': 'San Cristóbal',
    'ShipRegion': 'Táchira',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10396,
    'CustomerID': 'FRANK',
    'OrderDate': '1996-12-27T00:00:00.000Z',
    'ShippedDate': '1997-01-06T00:00:00.000Z',
    'Freight': 135.35,
    'ShipName': 'Frankenversand',
    'ShipAddress': 'Berliner Platz 43',
    'ShipCity': 'München',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10397,
    'CustomerID': 'PRINI',
    'OrderDate': '1996-12-27T00:00:00.000Z',
    'ShippedDate': '1997-01-02T00:00:00.000Z',
    'Freight': 60.26,
    'ShipName': 'Princesa Isabel Vinhos',
    'ShipAddress': 'Estrada da saúde n. 58',
    'ShipCity': 'Lisboa',
    'ShipRegion': null,
    'ShipCountry': 'Portugal'
  },
  {
    'OrderID': 10398,
    'CustomerID': 'SAVEA',
    'OrderDate': '1996-12-30T00:00:00.000Z',
    'ShippedDate': '1997-01-09T00:00:00.000Z',
    'Freight': 89.16,
    'ShipName': 'Save-a-lot Markets',
    'ShipAddress': '187 Suffolk Ln.',
    'ShipCity': 'Boise',
    'ShipRegion': 'ID',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10399,
    'CustomerID': 'VAFFE',

```

```

      'OrderDate': '1996-12-31T00:00:00.000Z',
      'ShippedDate': '1997-01-08T00:00:00.000Z',
      'Freight': 27.36,
      'ShipName': 'Vaffeljernet',
      'ShipAddress': 'Smagsloget 45',
      'ShipCity': 'Århus',
      'ShipRegion': null,
      'ShipCountry': 'Denmark'
    },
    {
      'OrderID': 10400,
      'CustomerID': 'EASTC',
      'OrderDate': '1997-01-01T00:00:00.000Z',
      'ShippedDate': '1997-01-16T00:00:00.000Z',
      'Freight': 83.93,
      'ShipName': 'Eastern Connection',
      'ShipAddress': '35 King George',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10401,
      'CustomerID': 'RATTC',
      'OrderDate': '1997-01-01T00:00:00.000Z',
      'ShippedDate': '1997-01-10T00:00:00.000Z',
      'Freight': 12.51,
      'ShipName': 'Rattlesnake Canyon Grocery',
      'ShipAddress': '2817 Milton Dr.',
      'ShipCity': 'Albuquerque',
      'ShipRegion': 'NM',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10402,
      'CustomerID': 'ERNSH',
      'OrderDate': '1997-01-02T00:00:00.000Z',
      'ShippedDate': '1997-01-10T00:00:00.000Z',
      'Freight': 67.88,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10403,
      'CustomerID': 'ERNSH',
      'OrderDate': '1997-01-03T00:00:00.000Z',
      'ShippedDate': '1997-01-09T00:00:00.000Z',
      'Freight': 73.79,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    }
  ],

```

```

{
  'OrderID': 10404,
  'CustomerID': 'MAGAA',
  'OrderDate': '1997-01-03T00:00:00.000Z',
  'ShippedDate': '1997-01-08T00:00:00.000Z',
  'Freight': 155.97,
  'ShipName': 'Magazzini Alimentari Riuniti',
  'ShipAddress': 'Via Ludovico il Moro 22',
  'ShipCity': 'Bergamo',
  'ShipRegion': null,
  'ShipCountry': 'Italy'
},
{
  'OrderID': 10405,
  'CustomerID': 'LINOD',
  'OrderDate': '1997-01-06T00:00:00.000Z',
  'ShippedDate': '1997-01-22T00:00:00.000Z',
  'Freight': 34.82,
  'ShipName': 'LINO-Delicateses',
  'ShipAddress': 'Ave. 5 de Mayo Porlamar',
  'ShipCity': 'I. de Margarita',
  'ShipRegion': 'Nueva Esparta',
  'ShipCountry': 'Venezuela'
},
{
  'OrderID': 10406,
  'CustomerID': 'QUEEN',
  'OrderDate': '1997-01-07T00:00:00.000Z',
  'ShippedDate': '1997-01-13T00:00:00.000Z',
  'Freight': 108.04,
  'ShipName': 'Queen Cozinha',
  'ShipAddress': 'Alameda dos Canários, 891',
  'ShipCity': 'Sao Paulo',
  'ShipRegion': 'SP',
  'ShipCountry': 'Brazil'
},
{
  'OrderID': 10407,
  'CustomerID': 'OTTIK',
  'OrderDate': '1997-01-07T00:00:00.000Z',
  'ShippedDate': '1997-01-30T00:00:00.000Z',
  'Freight': 91.48,
  'ShipName': 'Ottilies Käseladen',
  'ShipAddress': 'Mehrheimerstr. 369',
  'ShipCity': 'Köln',
  'ShipRegion': null,
  'ShipCountry': 'Germany'
},
{
  'OrderID': 10408,
  'CustomerID': 'FOLIG',
  'OrderDate': '1997-01-08T00:00:00.000Z',
  'ShippedDate': '1997-01-14T00:00:00.000Z',
  'Freight': 11.26,
  'ShipName': 'Folies gourmandes',
  'ShipAddress': '184, chaussée de Tournai',
  'ShipCity': 'Lille',

```

```

    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10409,
    'CustomerID': 'OCEAN',
    'OrderDate': '1997-01-09T00:00:00.000Z',
    'ShippedDate': '1997-01-14T00:00:00.000Z',
    'Freight': 29.83,
    'ShipName': 'Océano Atlántico Ltda.',
    'ShipAddress': 'Ing. Gustavo Moncada 8585 Piso 20-A',
    'ShipCity': 'Buenos Aires',
    'ShipRegion': null,
    'ShipCountry': 'Argentina'
  },
  {
    'OrderID': 10410,
    'CustomerID': 'BOTTM',
    'OrderDate': '1997-01-10T00:00:00.000Z',
    'ShippedDate': '1997-01-15T00:00:00.000Z',
    'Freight': 2.4,
    'ShipName': 'Bottom-Dollar Markets',
    'ShipAddress': '23 Tsawassen Blvd.',
    'ShipCity': 'Tsawassen',
    'ShipRegion': 'BC',
    'ShipCountry': 'Canada'
  },
  {
    'OrderID': 10411,
    'CustomerID': 'BOTTM',
    'OrderDate': '1997-01-10T00:00:00.000Z',
    'ShippedDate': '1997-01-21T00:00:00.000Z',
    'Freight': 23.65,
    'ShipName': 'Bottom-Dollar Markets',
    'ShipAddress': '23 Tsawassen Blvd.',
    'ShipCity': 'Tsawassen',
    'ShipRegion': 'BC',
    'ShipCountry': 'Canada'
  },
  {
    'OrderID': 10412,
    'CustomerID': 'WARTH',
    'OrderDate': '1997-01-13T00:00:00.000Z',
    'ShippedDate': '1997-01-15T00:00:00.000Z',
    'Freight': 3.77,
    'ShipName': 'Wartian Herkku',
    'ShipAddress': 'Torikatu 38',
    'ShipCity': 'Oulu',
    'ShipRegion': null,
    'ShipCountry': 'Finland'
  },
  {
    'OrderID': 10413,
    'CustomerID': 'LAMAI',
    'OrderDate': '1997-01-14T00:00:00.000Z',
    'ShippedDate': '1997-01-16T00:00:00.000Z',
    'Freight': 95.66,

```

```

      'ShipName': 'La maison d\'Asie',
      'ShipAddress': '1 rue Alsace-Lorraine',
      'ShipCity': 'Toulouse',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10414,
      'CustomerID': 'FAMIA',
      'OrderDate': '1997-01-14T00:00:00.000Z',
      'ShippedDate': '1997-01-17T00:00:00.000Z',
      'Freight': 21.48,
      'ShipName': 'Familia Arquibaldo',
      'ShipAddress': 'Rua Orós, 92',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10415,
      'CustomerID': 'HUNGC',
      'OrderDate': '1997-01-15T00:00:00.000Z',
      'ShippedDate': '1997-01-24T00:00:00.000Z',
      'Freight': 0.2,
      'ShipName': 'Hungry Coyote Import Store',
      'ShipAddress': 'City Center Plaza 516 Main St.',
      'ShipCity': 'Elgin',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10416,
      'CustomerID': 'WARTH',
      'OrderDate': '1997-01-16T00:00:00.000Z',
      'ShippedDate': '1997-01-27T00:00:00.000Z',
      'Freight': 22.72,
      'ShipName': 'Wartian Herkku',
      'ShipAddress': 'Torikatu 38',
      'ShipCity': 'Oulu',
      'ShipRegion': null,
      'ShipCountry': 'Finland'
    },
    {
      'OrderID': 10417,
      'CustomerID': 'SIMOB',
      'OrderDate': '1997-01-16T00:00:00.000Z',
      'ShippedDate': '1997-01-28T00:00:00.000Z',
      'Freight': 70.29,
      'ShipName': 'Simons bistro',
      'ShipAddress': 'Vinbæltet 34',
      'ShipCity': 'Kobenhavn',
      'ShipRegion': null,
      'ShipCountry': 'Denmark'
    },
    {
      'OrderID': 10418,
      'CustomerID': 'QUICK',

```



```

      'OrderDate': '1997-01-17T00:00:00.000Z',
      'ShippedDate': '1997-01-24T00:00:00.000Z',
      'Freight': 17.55,
      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10419,
      'CustomerID': 'RICSU',
      'OrderDate': '1997-01-20T00:00:00.000Z',
      'ShippedDate': '1997-01-30T00:00:00.000Z',
      'Freight': 137.35,
      'ShipName': 'Richter Supermarkt',
      'ShipAddress': 'Starenweg 5',
      'ShipCity': 'Genève',
      'ShipRegion': null,
      'ShipCountry': 'Switzerland'
    },
    {
      'OrderID': 10420,
      'CustomerID': 'WELLI',
      'OrderDate': '1997-01-21T00:00:00.000Z',
      'ShippedDate': '1997-01-27T00:00:00.000Z',
      'Freight': 44.12,
      'ShipName': 'Wellington Importadora',
      'ShipAddress': 'Rua do Mercado, 12',
      'ShipCity': 'Resende',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10421,
      'CustomerID': 'QUEDE',
      'OrderDate': '1997-01-21T00:00:00.000Z',
      'ShippedDate': '1997-01-27T00:00:00.000Z',
      'Freight': 99.23,
      'ShipName': 'Que Delícia',
      'ShipAddress': 'Rua da Panificadora, 12',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10422,
      'CustomerID': 'FRANS',
      'OrderDate': '1997-01-22T00:00:00.000Z',
      'ShippedDate': '1997-01-31T00:00:00.000Z',
      'Freight': 3.02,
      'ShipName': 'Franchi S.p.A.',
      'ShipAddress': 'Via Monte Bianco 34',
      'ShipCity': 'Torino',
      'ShipRegion': null,
      'ShipCountry': 'Italy'
    }
  ],

```

```

{
  'OrderID': 10423,
  'CustomerID': 'GOURL',
  'OrderDate': '1997-01-23T00:00:00.000Z',
  'ShippedDate': '1997-02-24T00:00:00.000Z',
  'Freight': 24.5,
  'ShipName': 'Gourmet Lanchonetes',
  'ShipAddress': 'Av. Brasil, 442',
  'ShipCity': 'Campinas',
  'ShipRegion': 'SP',
  'ShipCountry': 'Brazil'
},
{
  'OrderID': 10424,
  'CustomerID': 'MEREP',
  'OrderDate': '1997-01-23T00:00:00.000Z',
  'ShippedDate': '1997-01-27T00:00:00.000Z',
  'Freight': 370.61,
  'ShipName': 'Mère Paillarde',
  'ShipAddress': '43 rue St. Laurent',
  'ShipCity': 'Montréal',
  'ShipRegion': 'Québec',
  'ShipCountry': 'Canada'
},
{
  'OrderID': 10425,
  'CustomerID': 'LAMAI',
  'OrderDate': '1997-01-24T00:00:00.000Z',
  'ShippedDate': '1997-02-14T00:00:00.000Z',
  'Freight': 7.93,
  'ShipName': 'La maison d\'Asie',
  'ShipAddress': '1 rue Alsace-Lorraine',
  'ShipCity': 'Toulouse',
  'ShipRegion': null,
  'ShipCountry': 'France'
},
{
  'OrderID': 10426,
  'CustomerID': 'GALED',
  'OrderDate': '1997-01-27T00:00:00.000Z',
  'ShippedDate': '1997-02-06T00:00:00.000Z',
  'Freight': 18.69,
  'ShipName': 'Galería del gastronómo',
  'ShipAddress': 'Rambla de Cataluña, 23',
  'ShipCity': 'Barcelona',
  'ShipRegion': null,
  'ShipCountry': 'Spain'
},
{
  'OrderID': 10427,
  'CustomerID': 'PICCO',
  'OrderDate': '1997-01-27T00:00:00.000Z',
  'ShippedDate': '1997-03-03T00:00:00.000Z',
  'Freight': 31.29,
  'ShipName': 'Piccolo und mehr',
  'ShipAddress': 'Geislweg 14',
  'ShipCity': 'Salzburg',

```

```

    'ShipRegion': null,
    'ShipCountry': 'Austria'
  },
  {
    'OrderID': 10428,
    'CustomerID': 'REGGC',
    'OrderDate': '1997-01-28T00:00:00.000Z',
    'ShippedDate': '1997-02-04T00:00:00.000Z',
    'Freight': 11.09,
    'ShipName': 'Reggiani Caseifici',
    'ShipAddress': 'Strada Provinciale 124',
    'ShipCity': 'Reggio Emilia',
    'ShipRegion': null,
    'ShipCountry': 'Italy'
  },
  {
    'OrderID': 10429,
    'CustomerID': 'HUNGO',
    'OrderDate': '1997-01-29T00:00:00.000Z',
    'ShippedDate': '1997-02-07T00:00:00.000Z',
    'Freight': 56.63,
    'ShipName': 'Hungry Owl All-Night Grocers',
    'ShipAddress': '8 Johnstown Road',
    'ShipCity': 'Cork',
    'ShipRegion': 'Co. Cork',
    'ShipCountry': 'Ireland'
  },
  {
    'OrderID': 10430,
    'CustomerID': 'ERNSH',
    'OrderDate': '1997-01-30T00:00:00.000Z',
    'ShippedDate': '1997-02-03T00:00:00.000Z',
    'Freight': 458.78,
    'ShipName': 'Ernst Handel',
    'ShipAddress': 'Kirchgasse 6',
    'ShipCity': 'Graz',
    'ShipRegion': null,
    'ShipCountry': 'Austria'
  },
  {
    'OrderID': 10431,
    'CustomerID': 'BOTTM',
    'OrderDate': '1997-01-30T00:00:00.000Z',
    'ShippedDate': '1997-02-07T00:00:00.000Z',
    'Freight': 44.17,
    'ShipName': 'Bottom-Dollar Markets',
    'ShipAddress': '23 Tsawassen Blvd.',
    'ShipCity': 'Tsawassen',
    'ShipRegion': 'BC',
    'ShipCountry': 'Canada'
  },
  {
    'OrderID': 10432,
    'CustomerID': 'SPLIR',
    'OrderDate': '1997-01-31T00:00:00.000Z',
    'ShippedDate': '1997-02-07T00:00:00.000Z',
    'Freight': 4.34,

```

```

      'ShipName': 'Split Rail Beer & Ale',
      'ShipAddress': 'P.O. Box 555',
      'ShipCity': 'Lander',
      'ShipRegion': 'WY',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10433,
      'CustomerID': 'PRINI',
      'OrderDate': '1997-02-03T00:00:00.000Z',
      'ShippedDate': '1997-03-04T00:00:00.000Z',
      'Freight': 73.83,
      'ShipName': 'Princesa Isabel Vinhos',
      'ShipAddress': 'Estrada da saúde n. 58',
      'ShipCity': 'Lisboa',
      'ShipRegion': null,
      'ShipCountry': 'Portugal'
    },
    {
      'OrderID': 10434,
      'CustomerID': 'FOLKO',
      'OrderDate': '1997-02-03T00:00:00.000Z',
      'ShippedDate': '1997-02-13T00:00:00.000Z',
      'Freight': 17.92,
      'ShipName': 'Folk och fä HB',
      'ShipAddress': 'Åkergatan 24',
      'ShipCity': 'Bräcke',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 10435,
      'CustomerID': 'CONSH',
      'OrderDate': '1997-02-04T00:00:00.000Z',
      'ShippedDate': '1997-02-07T00:00:00.000Z',
      'Freight': 9.21,
      'ShipName': 'Consolidated Holdings',
      'ShipAddress': 'Berkeley Gardens 12 Brewery',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10436,
      'CustomerID': 'BLONP',
      'OrderDate': '1997-02-05T00:00:00.000Z',
      'ShippedDate': '1997-02-11T00:00:00.000Z',
      'Freight': 156.66,
      'ShipName': 'Blondel père et fils',
      'ShipAddress': '24, place Kléber',
      'ShipCity': 'Strasbourg',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10437,
      'CustomerID': 'WARTH',

```

```

      'OrderDate': '1997-02-05T00:00:00.000Z',
      'ShippedDate': '1997-02-12T00:00:00.000Z',
      'Freight': 19.97,
      'ShipName': 'Wartian Herkku',
      'ShipAddress': 'Torikatu 38',
      'ShipCity': 'Oulu',
      'ShipRegion': null,
      'ShipCountry': 'Finland'
    },
    {
      'OrderID': 10438,
      'CustomerID': 'TOMSP',
      'OrderDate': '1997-02-06T00:00:00.000Z',
      'ShippedDate': '1997-02-14T00:00:00.000Z',
      'Freight': 8.24,
      'ShipName': 'Toms Spezialitäten',
      'ShipAddress': 'Luisenstr. 48',
      'ShipCity': 'Münster',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10439,
      'CustomerID': 'MEREP',
      'OrderDate': '1997-02-07T00:00:00.000Z',
      'ShippedDate': '1997-02-10T00:00:00.000Z',
      'Freight': 4.07,
      'ShipName': 'Mère Paillarde',
      'ShipAddress': '43 rue St. Laurent',
      'ShipCity': 'Montréal',
      'ShipRegion': 'Québec',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10440,
      'CustomerID': 'SAVEA',
      'OrderDate': '1997-02-10T00:00:00.000Z',
      'ShippedDate': '1997-02-28T00:00:00.000Z',
      'Freight': 86.53,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10441,
      'CustomerID': 'OLDWO',
      'OrderDate': '1997-02-10T00:00:00.000Z',
      'ShippedDate': '1997-03-14T00:00:00.000Z',
      'Freight': 73.02,
      'ShipName': 'Old World Delicatessen',
      'ShipAddress': '2743 Bering St.',
      'ShipCity': 'Anchorage',
      'ShipRegion': 'AK',
      'ShipCountry': 'USA'
    }
  ],

```

```

{
  'OrderID': 10442,
  'CustomerID': 'ERNSH',
  'OrderDate': '1997-02-11T00:00:00.000Z',
  'ShippedDate': '1997-02-18T00:00:00.000Z',
  'Freight': 47.94,
  'ShipName': 'Ernst Handel',
  'ShipAddress': 'Kirchgasse 6',
  'ShipCity': 'Graz',
  'ShipRegion': null,
  'ShipCountry': 'Austria'
},
{
  'OrderID': 10443,
  'CustomerID': 'REGGC',
  'OrderDate': '1997-02-12T00:00:00.000Z',
  'ShippedDate': '1997-02-14T00:00:00.000Z',
  'Freight': 13.95,
  'ShipName': 'Reggiani Caseifici',
  'ShipAddress': 'Strada Provinciale 124',
  'ShipCity': 'Reggio Emilia',
  'ShipRegion': null,
  'ShipCountry': 'Italy'
},
{
  'OrderID': 10444,
  'CustomerID': 'BERGS',
  'OrderDate': '1997-02-12T00:00:00.000Z',
  'ShippedDate': '1997-02-21T00:00:00.000Z',
  'Freight': 3.5,
  'ShipName': 'Berglunds snabbköp',
  'ShipAddress': 'Berguvsvägen 8',
  'ShipCity': 'Luleå',
  'ShipRegion': null,
  'ShipCountry': 'Sweden'
},
{
  'OrderID': 10445,
  'CustomerID': 'BERGS',
  'OrderDate': '1997-02-13T00:00:00.000Z',
  'ShippedDate': '1997-02-20T00:00:00.000Z',
  'Freight': 9.3,
  'ShipName': 'Berglunds snabbköp',
  'ShipAddress': 'Berguvsvägen 8',
  'ShipCity': 'Luleå',
  'ShipRegion': null,
  'ShipCountry': 'Sweden'
},
{
  'OrderID': 10446,
  'CustomerID': 'TOMSP',
  'OrderDate': '1997-02-14T00:00:00.000Z',
  'ShippedDate': '1997-02-19T00:00:00.000Z',
  'Freight': 14.68,
  'ShipName': 'Toms Spezialitäten',
  'ShipAddress': 'Luisenstr. 48',
  'ShipCity': 'Münster',

```

```

    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10447,
    'CustomerID': 'RICAR',
    'OrderDate': '1997-02-14T00:00:00.000Z',
    'ShippedDate': '1997-03-07T00:00:00.000Z',
    'Freight': 68.66,
    'ShipName': 'Ricardo Adocicados',
    'ShipAddress': 'Av. Copacabana, 267',
    'ShipCity': 'Rio de Janeiro',
    'ShipRegion': 'RJ',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10448,
    'CustomerID': 'RANCH',
    'OrderDate': '1997-02-17T00:00:00.000Z',
    'ShippedDate': '1997-02-24T00:00:00.000Z',
    'Freight': 38.82,
    'ShipName': 'Rancho grande',
    'ShipAddress': 'Av. del Libertador 900',
    'ShipCity': 'Buenos Aires',
    'ShipRegion': null,
    'ShipCountry': 'Argentina'
  },
  {
    'OrderID': 10449,
    'CustomerID': 'BLONP',
    'OrderDate': '1997-02-18T00:00:00.000Z',
    'ShippedDate': '1997-02-27T00:00:00.000Z',
    'Freight': 53.3,
    'ShipName': 'Blondel père et fils',
    'ShipAddress': '24, place Kléber',
    'ShipCity': 'Strasbourg',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10450,
    'CustomerID': 'VICTE',
    'OrderDate': '1997-02-19T00:00:00.000Z',
    'ShippedDate': '1997-03-11T00:00:00.000Z',
    'Freight': 7.23,
    'ShipName': 'Victuailles en stock',
    'ShipAddress': '2, rue du Commerce',
    'ShipCity': 'Lyon',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10451,
    'CustomerID': 'QUICK',
    'OrderDate': '1997-02-19T00:00:00.000Z',
    'ShippedDate': '1997-03-12T00:00:00.000Z',
    'Freight': 189.09,

```

```

      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10452,
      'CustomerID': 'SAVEA',
      'OrderDate': '1997-02-20T00:00:00.000Z',
      'ShippedDate': '1997-02-26T00:00:00.000Z',
      'Freight': 140.26,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10453,
      'CustomerID': 'AROUT',
      'OrderDate': '1997-02-21T00:00:00.000Z',
      'ShippedDate': '1997-02-26T00:00:00.000Z',
      'Freight': 25.36,
      'ShipName': 'Around the Horn',
      'ShipAddress': 'Brook Farm Stratford St. Mary',
      'ShipCity': 'Colchester',
      'ShipRegion': 'Essex',
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10454,
      'CustomerID': 'LAMAI',
      'OrderDate': '1997-02-21T00:00:00.000Z',
      'ShippedDate': '1997-02-25T00:00:00.000Z',
      'Freight': 2.74,
      'ShipName': 'La maison d\'Asie',
      'ShipAddress': '1 rue Alsace-Lorraine',
      'ShipCity': 'Toulouse',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10455,
      'CustomerID': 'WARTH',
      'OrderDate': '1997-02-24T00:00:00.000Z',
      'ShippedDate': '1997-03-03T00:00:00.000Z',
      'Freight': 180.45,
      'ShipName': 'Wartian Herkku',
      'ShipAddress': 'Torikatu 38',
      'ShipCity': 'Oulu',
      'ShipRegion': null,
      'ShipCountry': 'Finland'
    },
    {
      'OrderID': 10456,
      'CustomerID': 'KOENE',

```



```

      'OrderDate': '1997-02-25T00:00:00.000Z',
      'ShippedDate': '1997-02-28T00:00:00.000Z',
      'Freight': 8.12,
      'ShipName': 'Königlich Essen',
      'ShipAddress': 'Maubelstr. 90',
      'ShipCity': 'Brandenburg',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10457,
      'CustomerID': 'KOENE',
      'OrderDate': '1997-02-25T00:00:00.000Z',
      'ShippedDate': '1997-03-03T00:00:00.000Z',
      'Freight': 11.57,
      'ShipName': 'Königlich Essen',
      'ShipAddress': 'Maubelstr. 90',
      'ShipCity': 'Brandenburg',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10458,
      'CustomerID': 'SUPRD',
      'OrderDate': '1997-02-26T00:00:00.000Z',
      'ShippedDate': '1997-03-04T00:00:00.000Z',
      'Freight': 147.06,
      'ShipName': 'Suprêmes délices',
      'ShipAddress': 'Boulevard Tirou, 255',
      'ShipCity': 'Charleroi',
      'ShipRegion': null,
      'ShipCountry': 'Belgium'
    },
    {
      'OrderID': 10459,
      'CustomerID': 'VICTE',
      'OrderDate': '1997-02-27T00:00:00.000Z',
      'ShippedDate': '1997-02-28T00:00:00.000Z',
      'Freight': 25.09,
      'ShipName': 'Victuailles en stock',
      'ShipAddress': '2, rue du Commerce',
      'ShipCity': 'Lyon',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10460,
      'CustomerID': 'FOLKO',
      'OrderDate': '1997-02-28T00:00:00.000Z',
      'ShippedDate': '1997-03-03T00:00:00.000Z',
      'Freight': 16.27,
      'ShipName': 'Folk och fä HB',
      'ShipAddress': 'Åkergatan 24',
      'ShipCity': 'Bräcke',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    }
  ],

```

```

{
  'OrderID': 10461,
  'CustomerID': 'LILAS',
  'OrderDate': '1997-02-28T00:00:00.000Z',
  'ShippedDate': '1997-03-05T00:00:00.000Z',
  'Freight': 148.61,
  'ShipName': 'LILA-Supermercado',
  'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
  'ShipCity': 'Barquisimeto',
  'ShipRegion': 'Lara',
  'ShipCountry': 'Venezuela'
},
{
  'OrderID': 10462,
  'CustomerID': 'CONSH',
  'OrderDate': '1997-03-03T00:00:00.000Z',
  'ShippedDate': '1997-03-18T00:00:00.000Z',
  'Freight': 6.17,
  'ShipName': 'Consolidated Holdings',
  'ShipAddress': 'Berkeley Gardens 12 Brewery',
  'ShipCity': 'London',
  'ShipRegion': null,
  'ShipCountry': 'UK'
},
{
  'OrderID': 10463,
  'CustomerID': 'SUPRD',
  'OrderDate': '1997-03-04T00:00:00.000Z',
  'ShippedDate': '1997-03-06T00:00:00.000Z',
  'Freight': 14.78,
  'ShipName': 'Suprêmes délices',
  'ShipAddress': 'Boulevard Tirou, 255',
  'ShipCity': 'Charleroi',
  'ShipRegion': null,
  'ShipCountry': 'Belgium'
},
{
  'OrderID': 10464,
  'CustomerID': 'FURIB',
  'OrderDate': '1997-03-04T00:00:00.000Z',
  'ShippedDate': '1997-03-14T00:00:00.000Z',
  'Freight': 89,
  'ShipName': 'Furia Bacalhau e Frutos do Mar',
  'ShipAddress': 'Jardim das rosas n. 32',
  'ShipCity': 'Lisboa',
  'ShipRegion': null,
  'ShipCountry': 'Portugal'
},
{
  'OrderID': 10465,
  'CustomerID': 'VAFFE',
  'OrderDate': '1997-03-05T00:00:00.000Z',
  'ShippedDate': '1997-03-14T00:00:00.000Z',
  'Freight': 145.04,
  'ShipName': 'Vaffeljernet',
  'ShipAddress': 'Smagsloget 45',
  'ShipCity': 'Århus',

```

```

    'ShipRegion': null,
    'ShipCountry': 'Denmark'
  },
  {
    'OrderID': 10466,
    'CustomerID': 'COMMI',
    'OrderDate': '1997-03-06T00:00:00.000Z',
    'ShippedDate': '1997-03-13T00:00:00.000Z',
    'Freight': 11.93,
    'ShipName': 'Comércio Mineiro',
    'ShipAddress': 'Av. dos Lusíadas, 23',
    'ShipCity': 'Sao Paulo',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10467,
    'CustomerID': 'MAGAA',
    'OrderDate': '1997-03-06T00:00:00.000Z',
    'ShippedDate': '1997-03-11T00:00:00.000Z',
    'Freight': 4.93,
    'ShipName': 'Magazzini Alimentari Riuniti',
    'ShipAddress': 'Via Ludovico il Moro 22',
    'ShipCity': 'Bergamo',
    'ShipRegion': null,
    'ShipCountry': 'Italy'
  },
  {
    'OrderID': 10468,
    'CustomerID': 'KOENE',
    'OrderDate': '1997-03-07T00:00:00.000Z',
    'ShippedDate': '1997-03-12T00:00:00.000Z',
    'Freight': 44.12,
    'ShipName': 'Königlich Essen',
    'ShipAddress': 'Maubelstr. 90',
    'ShipCity': 'Brandenburg',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10469,
    'CustomerID': 'WHITC',
    'OrderDate': '1997-03-10T00:00:00.000Z',
    'ShippedDate': '1997-03-14T00:00:00.000Z',
    'Freight': 60.18,
    'ShipName': 'White Clover Markets',
    'ShipAddress': '1029 - 12th Ave. S.',
    'ShipCity': 'Seattle',
    'ShipRegion': 'WA',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10470,
    'CustomerID': 'BONAP',
    'OrderDate': '1997-03-11T00:00:00.000Z',
    'ShippedDate': '1997-03-14T00:00:00.000Z',
    'Freight': 64.56,

```

```

      'ShipName': 'Bon app',
      'ShipAddress': '12, rue des Bouchers',
      'ShipCity': 'Marseille',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10471,
      'CustomerID': 'BSBEV',
      'OrderDate': '1997-03-11T00:00:00.000Z',
      'ShippedDate': '1997-03-18T00:00:00.000Z',
      'Freight': 45.59,
      'ShipName': 'B\ Beverages',
      'ShipAddress': 'Fauntleroy Circus',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10472,
      'CustomerID': 'SEVES',
      'OrderDate': '1997-03-12T00:00:00.000Z',
      'ShippedDate': '1997-03-19T00:00:00.000Z',
      'Freight': 4.2,
      'ShipName': 'Seven Seas Imports',
      'ShipAddress': '90 Wadhurst Rd.',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10473,
      'CustomerID': 'ISLAT',
      'OrderDate': '1997-03-13T00:00:00.000Z',
      'ShippedDate': '1997-03-21T00:00:00.000Z',
      'Freight': 16.37,
      'ShipName': 'Island Trading',
      'ShipAddress': 'Garden House Crowther Way',
      'ShipCity': 'Cowes',
      'ShipRegion': 'Isle of Wight',
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10474,
      'CustomerID': 'PERIC',
      'OrderDate': '1997-03-13T00:00:00.000Z',
      'ShippedDate': '1997-03-21T00:00:00.000Z',
      'Freight': 83.49,
      'ShipName': 'Pericles Comidas clásicas',
      'ShipAddress': 'Calle Dr. Jorge Cash 321',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10475,
      'CustomerID': 'SUPRD',

```

```

      'OrderDate': '1997-03-14T00:00:00.000Z',
      'ShippedDate': '1997-04-04T00:00:00.000Z',
      'Freight': 68.52,
      'ShipName': 'Suprêmes délices',
      'ShipAddress': 'Boulevard Tirou, 255',
      'ShipCity': 'Charleroi',
      'ShipRegion': null,
      'ShipCountry': 'Belgium'
    },
    {
      'OrderID': 10476,
      'CustomerID': 'HILAA',
      'OrderDate': '1997-03-17T00:00:00.000Z',
      'ShippedDate': '1997-03-24T00:00:00.000Z',
      'Freight': 4.41,
      'ShipName': 'HILARION-Abastos',
      'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
      'ShipCity': 'San Cristóbal',
      'ShipRegion': 'Táchira',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10477,
      'CustomerID': 'PRINI',
      'OrderDate': '1997-03-17T00:00:00.000Z',
      'ShippedDate': '1997-03-25T00:00:00.000Z',
      'Freight': 13.02,
      'ShipName': 'Princesa Isabel Vinhos',
      'ShipAddress': 'Estrada da saúde n. 58',
      'ShipCity': 'Lisboa',
      'ShipRegion': null,
      'ShipCountry': 'Portugal'
    },
    {
      'OrderID': 10478,
      'CustomerID': 'VICTE',
      'OrderDate': '1997-03-18T00:00:00.000Z',
      'ShippedDate': '1997-03-26T00:00:00.000Z',
      'Freight': 4.81,
      'ShipName': 'Victuailles en stock',
      'ShipAddress': '2, rue du Commerce',
      'ShipCity': 'Lyon',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10479,
      'CustomerID': 'RATTC',
      'OrderDate': '1997-03-19T00:00:00.000Z',
      'ShippedDate': '1997-03-21T00:00:00.000Z',
      'Freight': 708.95,
      'ShipName': 'Rattlesnake Canyon Grocery',
      'ShipAddress': '2817 Milton Dr.',
      'ShipCity': 'Albuquerque',
      'ShipRegion': 'NM',
      'ShipCountry': 'USA'
    }
  ],

```

```

{
  'OrderID': 10480,
  'CustomerID': 'FOLIG',
  'OrderDate': '1997-03-20T00:00:00.000Z',
  'ShippedDate': '1997-03-24T00:00:00.000Z',
  'Freight': 1.35,
  'ShipName': 'Folies gourmandes',
  'ShipAddress': '184, chaussée de Tournai',
  'ShipCity': 'Lille',
  'ShipRegion': null,
  'ShipCountry': 'France'
},
{
  'OrderID': 10481,
  'CustomerID': 'RICAR',
  'OrderDate': '1997-03-20T00:00:00.000Z',
  'ShippedDate': '1997-03-25T00:00:00.000Z',
  'Freight': 64.33,
  'ShipName': 'Ricardo Adocicados',
  'ShipAddress': 'Av. Copacabana, 267',
  'ShipCity': 'Rio de Janeiro',
  'ShipRegion': 'RJ',
  'ShipCountry': 'Brazil'
},
{
  'OrderID': 10482,
  'CustomerID': 'LAZYK',
  'OrderDate': '1997-03-21T00:00:00.000Z',
  'ShippedDate': '1997-04-10T00:00:00.000Z',
  'Freight': 7.48,
  'ShipName': 'Lazy K Kountry Store',
  'ShipAddress': '12 Orchestra Terrace',
  'ShipCity': 'Walla Walla',
  'ShipRegion': 'WA',
  'ShipCountry': 'USA'
},
{
  'OrderID': 10483,
  'CustomerID': 'WHITC',
  'OrderDate': '1997-03-24T00:00:00.000Z',
  'ShippedDate': '1997-04-25T00:00:00.000Z',
  'Freight': 15.28,
  'ShipName': 'White Clover Markets',
  'ShipAddress': '1029 - 12th Ave. S.',
  'ShipCity': 'Seattle',
  'ShipRegion': 'WA',
  'ShipCountry': 'USA'
},
{
  'OrderID': 10484,
  'CustomerID': 'BSBEV',
  'OrderDate': '1997-03-24T00:00:00.000Z',
  'ShippedDate': '1997-04-01T00:00:00.000Z',
  'Freight': 6.88,
  'ShipName': 'B\ Beverages',
  'ShipAddress': 'Fauntleroy Circus',
  'ShipCity': 'London',

```

```

      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10485,
      'CustomerID': 'LINOD',
      'OrderDate': '1997-03-25T00:00:00.000Z',
      'ShippedDate': '1997-03-31T00:00:00.000Z',
      'Freight': 64.45,
      'ShipName': 'LINO-Delicatesses',
      'ShipAddress': 'Ave. 5 de Mayo Porlamar',
      'ShipCity': 'I. de Margarita',
      'ShipRegion': 'Nueva Esparta',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10486,
      'CustomerID': 'HILAA',
      'OrderDate': '1997-03-26T00:00:00.000Z',
      'ShippedDate': '1997-04-02T00:00:00.000Z',
      'Freight': 30.53,
      'ShipName': 'HILARION-Abastos',
      'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
      'ShipCity': 'San Cristóbal',
      'ShipRegion': 'Táchira',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10487,
      'CustomerID': 'QUEEN',
      'OrderDate': '1997-03-26T00:00:00.000Z',
      'ShippedDate': '1997-03-28T00:00:00.000Z',
      'Freight': 71.07,
      'ShipName': 'Queen Cozinha',
      'ShipAddress': 'Alameda dos Canários, 891',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10488,
      'CustomerID': 'FRANK',
      'OrderDate': '1997-03-27T00:00:00.000Z',
      'ShippedDate': '1997-04-02T00:00:00.000Z',
      'Freight': 4.93,
      'ShipName': 'Frankenversand',
      'ShipAddress': 'Berliner Platz 43',
      'ShipCity': 'München',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10489,
      'CustomerID': 'PICCO',
      'OrderDate': '1997-03-28T00:00:00.000Z',
      'ShippedDate': '1997-04-09T00:00:00.000Z',
      'Freight': 5.29,

```

```

      'ShipName': 'Piccolo und mehr',
      'ShipAddress': 'Geislweg 14',
      'ShipCity': 'Salzburg',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10490,
      'CustomerID': 'HILAA',
      'OrderDate': '1997-03-31T00:00:00.000Z',
      'ShippedDate': '1997-04-03T00:00:00.000Z',
      'Freight': 210.19,
      'ShipName': 'HILARION-Abastos',
      'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
      'ShipCity': 'San Cristóbal',
      'ShipRegion': 'Táchira',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10491,
      'CustomerID': 'FURIB',
      'OrderDate': '1997-03-31T00:00:00.000Z',
      'ShippedDate': '1997-04-08T00:00:00.000Z',
      'Freight': 16.96,
      'ShipName': 'Furia Bacalhau e Frutos do Mar',
      'ShipAddress': 'Jardim das rosas n. 32',
      'ShipCity': 'Lisboa',
      'ShipRegion': null,
      'ShipCountry': 'Portugal'
    },
    {
      'OrderID': 10492,
      'CustomerID': 'BOTTM',
      'OrderDate': '1997-04-01T00:00:00.000Z',
      'ShippedDate': '1997-04-11T00:00:00.000Z',
      'Freight': 62.89,
      'ShipName': 'Bottom-Dollar Markets',
      'ShipAddress': '23 Tsawassen Blvd.',
      'ShipCity': 'Tsawassen',
      'ShipRegion': 'BC',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10493,
      'CustomerID': 'LAMAI',
      'OrderDate': '1997-04-02T00:00:00.000Z',
      'ShippedDate': '1997-04-10T00:00:00.000Z',
      'Freight': 10.64,
      'ShipName': 'La maison d\'Asie',
      'ShipAddress': '1 rue Alsace-Lorraine',
      'ShipCity': 'Toulouse',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10494,
      'CustomerID': 'COMMI',

```



```

      'OrderDate': '1997-04-02T00:00:00.000Z',
      'ShippedDate': '1997-04-09T00:00:00.000Z',
      'Freight': 65.99,
      'ShipName': 'Comércio Mineiro',
      'ShipAddress': 'Av. dos Lusíadas, 23',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10495,
      'CustomerID': 'LAUGB',
      'OrderDate': '1997-04-03T00:00:00.000Z',
      'ShippedDate': '1997-04-11T00:00:00.000Z',
      'Freight': 4.65,
      'ShipName': 'Laughing Bacchus Wine Cellars',
      'ShipAddress': '2319 Elm St.',
      'ShipCity': 'Vancouver',
      'ShipRegion': 'BC',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10496,
      'CustomerID': 'TRADH',
      'OrderDate': '1997-04-04T00:00:00.000Z',
      'ShippedDate': '1997-04-07T00:00:00.000Z',
      'Freight': 46.77,
      'ShipName': 'Tradição Hipermercados',
      'ShipAddress': 'Av. Inês de Castro, 414',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10497,
      'CustomerID': 'LEHMS',
      'OrderDate': '1997-04-04T00:00:00.000Z',
      'ShippedDate': '1997-04-07T00:00:00.000Z',
      'Freight': 36.21,
      'ShipName': 'Lehmanns Marktstand',
      'ShipAddress': 'Magazinweg 7',
      'ShipCity': 'Frankfurt a.M.',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10498,
      'CustomerID': 'HILAA',
      'OrderDate': '1997-04-07T00:00:00.000Z',
      'ShippedDate': '1997-04-11T00:00:00.000Z',
      'Freight': 29.75,
      'ShipName': 'HILARION-Abastos',
      'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
      'ShipCity': 'San Cristóbal',
      'ShipRegion': 'Táchira',
      'ShipCountry': 'Venezuela'
    }
  ],

```

```

{
  'OrderID': 10499,
  'CustomerID': 'LILAS',
  'OrderDate': '1997-04-08T00:00:00.000Z',
  'ShippedDate': '1997-04-16T00:00:00.000Z',
  'Freight': 102.02,
  'ShipName': 'LILA-Supermercado',
  'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
  'ShipCity': 'Barquisimeto',
  'ShipRegion': 'Lara',
  'ShipCountry': 'Venezuela'
},
{
  'OrderID': 10500,
  'CustomerID': 'LAMAI',
  'OrderDate': '1997-04-09T00:00:00.000Z',
  'ShippedDate': '1997-04-17T00:00:00.000Z',
  'Freight': 42.68,
  'ShipName': 'La maison d\'Asie',
  'ShipAddress': '1 rue Alsace-Lorraine',
  'ShipCity': 'Toulouse',
  'ShipRegion': null,
  'ShipCountry': 'France'
},
{
  'OrderID': 10501,
  'CustomerID': 'BLAUS',
  'OrderDate': '1997-04-09T00:00:00.000Z',
  'ShippedDate': '1997-04-16T00:00:00.000Z',
  'Freight': 8.85,
  'ShipName': 'Blauer See Delikatessen',
  'ShipAddress': 'Forsterstr. 57',
  'ShipCity': 'Mannheim',
  'ShipRegion': null,
  'ShipCountry': 'Germany'
},
{
  'OrderID': 10502,
  'CustomerID': 'PERIC',
  'OrderDate': '1997-04-10T00:00:00.000Z',
  'ShippedDate': '1997-04-29T00:00:00.000Z',
  'Freight': 69.32,
  'ShipName': 'Pericles Comidas clásicas',
  'ShipAddress': 'Calle Dr. Jorge Cash 321',
  'ShipCity': 'México D.F.',
  'ShipRegion': null,
  'ShipCountry': 'Mexico'
},
{
  'OrderID': 10503,
  'CustomerID': 'HUNGO',
  'OrderDate': '1997-04-11T00:00:00.000Z',
  'ShippedDate': '1997-04-16T00:00:00.000Z',
  'Freight': 16.74,
  'ShipName': 'Hungry Owl All-Night Grocers',
  'ShipAddress': '8 Johnstown Road',
  'ShipCity': 'Cork',

```

```

      'ShipRegion': 'Co. Cork',
      'ShipCountry': 'Ireland'
    },
    {
      'OrderID': 10504,
      'CustomerID': 'WHITC',
      'OrderDate': '1997-04-11T00:00:00.000Z',
      'ShippedDate': '1997-04-18T00:00:00.000Z',
      'Freight': 59.13,
      'ShipName': 'White Clover Markets',
      'ShipAddress': '1029 - 12th Ave. S.',
      'ShipCity': 'Seattle',
      'ShipRegion': 'WA',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10505,
      'CustomerID': 'MEREP',
      'OrderDate': '1997-04-14T00:00:00.000Z',
      'ShippedDate': '1997-04-21T00:00:00.000Z',
      'Freight': 7.13,
      'ShipName': 'Mère Paillarde',
      'ShipAddress': '43 rue St. Laurent',
      'ShipCity': 'Montréal',
      'ShipRegion': 'Québec',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10506,
      'CustomerID': 'KOENE',
      'OrderDate': '1997-04-15T00:00:00.000Z',
      'ShippedDate': '1997-05-02T00:00:00.000Z',
      'Freight': 21.19,
      'ShipName': 'Königlich Essen',
      'ShipAddress': 'Maubelstr. 90',
      'ShipCity': 'Brandenburg',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10507,
      'CustomerID': 'ANTON',
      'OrderDate': '1997-04-15T00:00:00.000Z',
      'ShippedDate': '1997-04-22T00:00:00.000Z',
      'Freight': 47.45,
      'ShipName': 'Antonio Moreno Taquería',
      'ShipAddress': 'Mataderos 2312',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10508,
      'CustomerID': 'OTTIK',
      'OrderDate': '1997-04-16T00:00:00.000Z',
      'ShippedDate': '1997-05-13T00:00:00.000Z',
      'Freight': 4.99,

```

```

      'ShipName': 'Ottilies Käseladen',
      'ShipAddress': 'Mehrheimerstr. 369',
      'ShipCity': 'Köln',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10509,
      'CustomerID': 'BLAUS',
      'OrderDate': '1997-04-17T00:00:00.000Z',
      'ShippedDate': '1997-04-29T00:00:00.000Z',
      'Freight': 0.15,
      'ShipName': 'Blauer See Delikatessen',
      'ShipAddress': 'Forsterstr. 57',
      'ShipCity': 'Mannheim',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10510,
      'CustomerID': 'SAVEA',
      'OrderDate': '1997-04-18T00:00:00.000Z',
      'ShippedDate': '1997-04-28T00:00:00.000Z',
      'Freight': 367.63,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10511,
      'CustomerID': 'BONAP',
      'OrderDate': '1997-04-18T00:00:00.000Z',
      'ShippedDate': '1997-04-21T00:00:00.000Z',
      'Freight': 350.64,
      'ShipName': 'Bon app',
      'ShipAddress': '12, rue des Bouchers',
      'ShipCity': 'Marseille',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10512,
      'CustomerID': 'FAMIA',
      'OrderDate': '1997-04-21T00:00:00.000Z',
      'ShippedDate': '1997-04-24T00:00:00.000Z',
      'Freight': 3.53,
      'ShipName': 'Familia Arquibaldo',
      'ShipAddress': 'Rua Orós, 92',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10513,
      'CustomerID': 'WANDK',

```

```

      'OrderDate': '1997-04-22T00:00:00.000Z',
      'ShippedDate': '1997-04-28T00:00:00.000Z',
      'Freight': 105.65,
      'ShipName': 'Die Wandernde Kuh',
      'ShipAddress': 'Adenauerallee 900',
      'ShipCity': 'Stuttgart',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10514,
      'CustomerID': 'ERNSH',
      'OrderDate': '1997-04-22T00:00:00.000Z',
      'ShippedDate': '1997-05-16T00:00:00.000Z',
      'Freight': 789.95,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10515,
      'CustomerID': 'QUICK',
      'OrderDate': '1997-04-23T00:00:00.000Z',
      'ShippedDate': '1997-05-23T00:00:00.000Z',
      'Freight': 204.47,
      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10516,
      'CustomerID': 'HUNGO',
      'OrderDate': '1997-04-24T00:00:00.000Z',
      'ShippedDate': '1997-05-01T00:00:00.000Z',
      'Freight': 62.78,
      'ShipName': 'Hungry Owl All-Night Grocers',
      'ShipAddress': '8 Johnstown Road',
      'ShipCity': 'Cork',
      'ShipRegion': 'Co. Cork',
      'ShipCountry': 'Ireland'
    },
    {
      'OrderID': 10517,
      'CustomerID': 'NORTS',
      'OrderDate': '1997-04-24T00:00:00.000Z',
      'ShippedDate': '1997-04-29T00:00:00.000Z',
      'Freight': 32.07,
      'ShipName': 'North/South',
      'ShipAddress': 'South House 300 Queensbridge',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    }
  ],

```

```

{
  'OrderID': 10518,
  'CustomerID': 'TORTU',
  'OrderDate': '1997-04-25T00:00:00.000Z',
  'ShippedDate': '1997-05-05T00:00:00.000Z',
  'Freight': 218.15,
  'ShipName': 'Tortuga Restaurante',
  'ShipAddress': 'Avda. Azteca 123',
  'ShipCity': 'México D.F.',
  'ShipRegion': null,
  'ShipCountry': 'Mexico'
},
{
  'OrderID': 10519,
  'CustomerID': 'CHOPS',
  'OrderDate': '1997-04-28T00:00:00.000Z',
  'ShippedDate': '1997-05-01T00:00:00.000Z',
  'Freight': 91.76,
  'ShipName': 'Chop-suey Chinese',
  'ShipAddress': 'Hauptstr. 31',
  'ShipCity': 'Bern',
  'ShipRegion': null,
  'ShipCountry': 'Switzerland'
},
{
  'OrderID': 10520,
  'CustomerID': 'SANTG',
  'OrderDate': '1997-04-29T00:00:00.000Z',
  'ShippedDate': '1997-05-01T00:00:00.000Z',
  'Freight': 13.37,
  'ShipName': 'Santé Gourmet',
  'ShipAddress': 'Erling Skakkes gate 78',
  'ShipCity': 'Stavern',
  'ShipRegion': null,
  'ShipCountry': 'Norway'
},
{
  'OrderID': 10521,
  'CustomerID': 'CACTU',
  'OrderDate': '1997-04-29T00:00:00.000Z',
  'ShippedDate': '1997-05-02T00:00:00.000Z',
  'Freight': 17.22,
  'ShipName': 'Cactus Comidas para llevar',
  'ShipAddress': 'Cerrito 333',
  'ShipCity': 'Buenos Aires',
  'ShipRegion': null,
  'ShipCountry': 'Argentina'
},
{
  'OrderID': 10522,
  'CustomerID': 'LEHMS',
  'OrderDate': '1997-04-30T00:00:00.000Z',
  'ShippedDate': '1997-05-06T00:00:00.000Z',
  'Freight': 45.33,
  'ShipName': 'Lehmanns Marktstand',
  'ShipAddress': 'Magazinweg 7',
  'ShipCity': 'Frankfurt a.M.',

```

```

      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10523,
      'CustomerID': 'SEVES',
      'OrderDate': '1997-05-01T00:00:00.000Z',
      'ShippedDate': '1997-05-30T00:00:00.000Z',
      'Freight': 77.63,
      'ShipName': 'Seven Seas Imports',
      'ShipAddress': '90 Wadhurst Rd.',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10524,
      'CustomerID': 'BERGS',
      'OrderDate': '1997-05-01T00:00:00.000Z',
      'ShippedDate': '1997-05-07T00:00:00.000Z',
      'Freight': 244.79,
      'ShipName': 'Berglunds snabbköp',
      'ShipAddress': 'Berguvsvägen 8',
      'ShipCity': 'Luleå',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 10525,
      'CustomerID': 'BONAP',
      'OrderDate': '1997-05-02T00:00:00.000Z',
      'ShippedDate': '1997-05-23T00:00:00.000Z',
      'Freight': 11.06,
      'ShipName': 'Bon app',
      'ShipAddress': '12, rue des Bouchers',
      'ShipCity': 'Marseille',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10526,
      'CustomerID': 'WARTH',
      'OrderDate': '1997-05-05T00:00:00.000Z',
      'ShippedDate': '1997-05-15T00:00:00.000Z',
      'Freight': 58.59,
      'ShipName': 'Wartian Herkku',
      'ShipAddress': 'Torikatu 38',
      'ShipCity': 'Oulu',
      'ShipRegion': null,
      'ShipCountry': 'Finland'
    },
    {
      'OrderID': 10527,
      'CustomerID': 'QUICK',
      'OrderDate': '1997-05-05T00:00:00.000Z',
      'ShippedDate': '1997-05-07T00:00:00.000Z',
      'Freight': 41.9,

```

```

    'ShipName': 'QUICK-Stop',
    'ShipAddress': 'Taucherstraße 10',
    'ShipCity': 'Cunewalde',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10528,
    'CustomerID': 'GREAL',
    'OrderDate': '1997-05-06T00:00:00.000Z',
    'ShippedDate': '1997-05-09T00:00:00.000Z',
    'Freight': 3.35,
    'ShipName': 'Great Lakes Food Market',
    'ShipAddress': '2732 Baker Blvd.',
    'ShipCity': 'Eugene',
    'ShipRegion': 'OR',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10529,
    'CustomerID': 'MAISD',
    'OrderDate': '1997-05-07T00:00:00.000Z',
    'ShippedDate': '1997-05-09T00:00:00.000Z',
    'Freight': 66.69,
    'ShipName': 'Maison Dewey',
    'ShipAddress': 'Rue Joseph-Bens 532',
    'ShipCity': 'Bruxelles',
    'ShipRegion': null,
    'ShipCountry': 'Belgium'
  },
  {
    'OrderID': 10530,
    'CustomerID': 'PICCO',
    'OrderDate': '1997-05-08T00:00:00.000Z',
    'ShippedDate': '1997-05-12T00:00:00.000Z',
    'Freight': 339.22,
    'ShipName': 'Piccolo und mehr',
    'ShipAddress': 'Geislweg 14',
    'ShipCity': 'Salzburg',
    'ShipRegion': null,
    'ShipCountry': 'Austria'
  },
  {
    'OrderID': 10531,
    'CustomerID': 'OCEAN',
    'OrderDate': '1997-05-08T00:00:00.000Z',
    'ShippedDate': '1997-05-19T00:00:00.000Z',
    'Freight': 8.12,
    'ShipName': 'Océano Atlántico Ltda.',
    'ShipAddress': 'Ing. Gustavo Moncada 8585 Piso 20-A',
    'ShipCity': 'Buenos Aires',
    'ShipRegion': null,
    'ShipCountry': 'Argentina'
  },
  {
    'OrderID': 10532,
    'CustomerID': 'EASTC',

```



```

      'OrderDate': '1997-05-09T00:00:00.000Z',
      'ShippedDate': '1997-05-12T00:00:00.000Z',
      'Freight': 74.46,
      'ShipName': 'Eastern Connection',
      'ShipAddress': '35 King George',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10533,
      'CustomerID': 'FOLKO',
      'OrderDate': '1997-05-12T00:00:00.000Z',
      'ShippedDate': '1997-05-22T00:00:00.000Z',
      'Freight': 188.04,
      'ShipName': 'Folk och fä HB',
      'ShipAddress': 'Åkergatan 24',
      'ShipCity': 'Bräcke',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 10534,
      'CustomerID': 'LEHMS',
      'OrderDate': '1997-05-12T00:00:00.000Z',
      'ShippedDate': '1997-05-14T00:00:00.000Z',
      'Freight': 27.94,
      'ShipName': 'Lehmanns Marktstand',
      'ShipAddress': 'Magazinweg 7',
      'ShipCity': 'Frankfurt a.M.',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10535,
      'CustomerID': 'ANTON',
      'OrderDate': '1997-05-13T00:00:00.000Z',
      'ShippedDate': '1997-05-21T00:00:00.000Z',
      'Freight': 15.64,
      'ShipName': 'Antonio Moreno Taquería',
      'ShipAddress': 'Mataderos 2312',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10536,
      'CustomerID': 'LEHMS',
      'OrderDate': '1997-05-14T00:00:00.000Z',
      'ShippedDate': '1997-06-06T00:00:00.000Z',
      'Freight': 58.88,
      'ShipName': 'Lehmanns Marktstand',
      'ShipAddress': 'Magazinweg 7',
      'ShipCity': 'Frankfurt a.M.',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    }
  ],

```

```

{
  'OrderID': 10537,
  'CustomerID': 'RICSU',
  'OrderDate': '1997-05-14T00:00:00.000Z',
  'ShippedDate': '1997-05-19T00:00:00.000Z',
  'Freight': 78.85,
  'ShipName': 'Richter Supermarkt',
  'ShipAddress': 'Starenweg 5',
  'ShipCity': 'Genève',
  'ShipRegion': null,
  'ShipCountry': 'Switzerland'
},
{
  'OrderID': 10538,
  'CustomerID': 'BSBEV',
  'OrderDate': '1997-05-15T00:00:00.000Z',
  'ShippedDate': '1997-05-16T00:00:00.000Z',
  'Freight': 4.87,
  'ShipName': 'B\ ' Beverages',
  'ShipAddress': 'Fauntleroy Circus',
  'ShipCity': 'London',
  'ShipRegion': null,
  'ShipCountry': 'UK'
},
{
  'OrderID': 10539,
  'CustomerID': 'BSBEV',
  'OrderDate': '1997-05-16T00:00:00.000Z',
  'ShippedDate': '1997-05-23T00:00:00.000Z',
  'Freight': 12.36,
  'ShipName': 'B\ ' Beverages',
  'ShipAddress': 'Fauntleroy Circus',
  'ShipCity': 'London',
  'ShipRegion': null,
  'ShipCountry': 'UK'
},
{
  'OrderID': 10540,
  'CustomerID': 'QUICK',
  'OrderDate': '1997-05-19T00:00:00.000Z',
  'ShippedDate': '1997-06-13T00:00:00.000Z',
  'Freight': 1007.64,
  'ShipName': 'QUICK-Stop',
  'ShipAddress': 'Taucherstraße 10',
  'ShipCity': 'Cunewalde',
  'ShipRegion': null,
  'ShipCountry': 'Germany'
},
{
  'OrderID': 10541,
  'CustomerID': 'HANAR',
  'OrderDate': '1997-05-19T00:00:00.000Z',
  'ShippedDate': '1997-05-29T00:00:00.000Z',
  'Freight': 68.65,
  'ShipName': 'Hanari Carnes',
  'ShipAddress': 'Rua do Paço, 67',
  'ShipCity': 'Rio de Janeiro',

```

```

    'ShipRegion': 'RJ',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10542,
    'CustomerID': 'KOENE',
    'OrderDate': '1997-05-20T00:00:00.000Z',
    'ShippedDate': '1997-05-26T00:00:00.000Z',
    'Freight': 10.95,
    'ShipName': 'Königlich Essen',
    'ShipAddress': 'Maubelstr. 90',
    'ShipCity': 'Brandenburg',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10543,
    'CustomerID': 'LILAS',
    'OrderDate': '1997-05-21T00:00:00.000Z',
    'ShippedDate': '1997-05-23T00:00:00.000Z',
    'Freight': 48.17,
    'ShipName': 'LILA-Supermercado',
    'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
    'ShipCity': 'Barquisimeto',
    'ShipRegion': 'Lara',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10544,
    'CustomerID': 'LONEP',
    'OrderDate': '1997-05-21T00:00:00.000Z',
    'ShippedDate': '1997-05-30T00:00:00.000Z',
    'Freight': 24.91,
    'ShipName': 'Lonesome Pine Restaurant',
    'ShipAddress': '89 Chiaroscuro Rd.',
    'ShipCity': 'Portland',
    'ShipRegion': 'OR',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10545,
    'CustomerID': 'LAZYK',
    'OrderDate': '1997-05-22T00:00:00.000Z',
    'ShippedDate': '1997-06-26T00:00:00.000Z',
    'Freight': 11.92,
    'ShipName': 'Lazy K Kountry Store',
    'ShipAddress': '12 Orchestra Terrace',
    'ShipCity': 'Walla Walla',
    'ShipRegion': 'WA',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10546,
    'CustomerID': 'VICTE',
    'OrderDate': '1997-05-23T00:00:00.000Z',
    'ShippedDate': '1997-05-27T00:00:00.000Z',
    'Freight': 194.72,

```

```

      'ShipName': 'Victuailles en stock',
      'ShipAddress': '2, rue du Commerce',
      'ShipCity': 'Lyon',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10547,
      'CustomerID': 'SEVES',
      'OrderDate': '1997-05-23T00:00:00.000Z',
      'ShippedDate': '1997-06-02T00:00:00.000Z',
      'Freight': 178.43,
      'ShipName': 'Seven Seas Imports',
      'ShipAddress': '90 Wadhurst Rd.',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10548,
      'CustomerID': 'TOMSP',
      'OrderDate': '1997-05-26T00:00:00.000Z',
      'ShippedDate': '1997-06-02T00:00:00.000Z',
      'Freight': 1.43,
      'ShipName': 'Toms Spezialitäten',
      'ShipAddress': 'Luisenstr. 48',
      'ShipCity': 'Münster',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10549,
      'CustomerID': 'QUICK',
      'OrderDate': '1997-05-27T00:00:00.000Z',
      'ShippedDate': '1997-05-30T00:00:00.000Z',
      'Freight': 171.24,
      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10550,
      'CustomerID': 'GODOS',
      'OrderDate': '1997-05-28T00:00:00.000Z',
      'ShippedDate': '1997-06-06T00:00:00.000Z',
      'Freight': 4.32,
      'ShipName': 'Godos Cocina Típica',
      'ShipAddress': 'C/ Romero, 33',
      'ShipCity': 'Sevilla',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    },
    {
      'OrderID': 10551,
      'CustomerID': 'FURIB',

```

```

      'OrderDate': '1997-05-28T00:00:00.000Z',
      'ShippedDate': '1997-06-06T00:00:00.000Z',
      'Freight': 72.95,
      'ShipName': 'Furia Bacalhau e Frutos do Mar',
      'ShipAddress': 'Jardim das rosas n. 32',
      'ShipCity': 'Lisboa',
      'ShipRegion': null,
      'ShipCountry': 'Portugal'
    },
    {
      'OrderID': 10552,
      'CustomerID': 'HILAA',
      'OrderDate': '1997-05-29T00:00:00.000Z',
      'ShippedDate': '1997-06-05T00:00:00.000Z',
      'Freight': 83.22,
      'ShipName': 'HILARION-Abastos',
      'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
      'ShipCity': 'San Cristóbal',
      'ShipRegion': 'Táchira',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10553,
      'CustomerID': 'WARTH',
      'OrderDate': '1997-05-30T00:00:00.000Z',
      'ShippedDate': '1997-06-03T00:00:00.000Z',
      'Freight': 149.49,
      'ShipName': 'Wartian Herkku',
      'ShipAddress': 'Torikatu 38',
      'ShipCity': 'Oulu',
      'ShipRegion': null,
      'ShipCountry': 'Finland'
    },
    {
      'OrderID': 10554,
      'CustomerID': 'OTTIK',
      'OrderDate': '1997-05-30T00:00:00.000Z',
      'ShippedDate': '1997-06-05T00:00:00.000Z',
      'Freight': 120.97,
      'ShipName': 'Ottilies Käseladen',
      'ShipAddress': 'Mehrheimerstr. 369',
      'ShipCity': 'Köln',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10555,
      'CustomerID': 'SAVEA',
      'OrderDate': '1997-06-02T00:00:00.000Z',
      'ShippedDate': '1997-06-04T00:00:00.000Z',
      'Freight': 252.49,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    }
  ],

```

```

{
  'OrderID': 10556,
  'CustomerID': 'SIMOB',
  'OrderDate': '1997-06-03T00:00:00.000Z',
  'ShippedDate': '1997-06-13T00:00:00.000Z',
  'Freight': 9.8,
  'ShipName': 'Simons bistro',
  'ShipAddress': 'Vinbæltet 34',
  'ShipCity': 'Kobenhavn',
  'ShipRegion': null,
  'ShipCountry': 'Denmark'
},
{
  'OrderID': 10557,
  'CustomerID': 'LEHMS',
  'OrderDate': '1997-06-03T00:00:00.000Z',
  'ShippedDate': '1997-06-06T00:00:00.000Z',
  'Freight': 96.72,
  'ShipName': 'Lehmanns Marktstand',
  'ShipAddress': 'Magazinweg 7',
  'ShipCity': 'Frankfurt a.M.',
  'ShipRegion': null,
  'ShipCountry': 'Germany'
},
{
  'OrderID': 10558,
  'CustomerID': 'AROUT',
  'OrderDate': '1997-06-04T00:00:00.000Z',
  'ShippedDate': '1997-06-10T00:00:00.000Z',
  'Freight': 72.97,
  'ShipName': 'Around the Horn',
  'ShipAddress': 'Brook Farm Stratford St. Mary',
  'ShipCity': 'Colchester',
  'ShipRegion': 'Essex',
  'ShipCountry': 'UK'
},
{
  'OrderID': 10559,
  'CustomerID': 'BLONP',
  'OrderDate': '1997-06-05T00:00:00.000Z',
  'ShippedDate': '1997-06-13T00:00:00.000Z',
  'Freight': 8.05,
  'ShipName': 'Blondel père et fils',
  'ShipAddress': '24, place Kléber',
  'ShipCity': 'Strasbourg',
  'ShipRegion': null,
  'ShipCountry': 'France'
},
{
  'OrderID': 10560,
  'CustomerID': 'FRANK',
  'OrderDate': '1997-06-06T00:00:00.000Z',
  'ShippedDate': '1997-06-09T00:00:00.000Z',
  'Freight': 36.65,
  'ShipName': 'Frankenversand',
  'ShipAddress': 'Berliner Platz 43',
  'ShipCity': 'München',

```

```

      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10561,
      'CustomerID': 'FOLKO',
      'OrderDate': '1997-06-06T00:00:00.000Z',
      'ShippedDate': '1997-06-09T00:00:00.000Z',
      'Freight': 242.21,
      'ShipName': 'Folk och fä HB',
      'ShipAddress': 'Åkergatan 24',
      'ShipCity': 'Bräcke',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 10562,
      'CustomerID': 'REGGC',
      'OrderDate': '1997-06-09T00:00:00.000Z',
      'ShippedDate': '1997-06-12T00:00:00.000Z',
      'Freight': 22.95,
      'ShipName': 'Reggiani Caseifici',
      'ShipAddress': 'Strada Provinciale 124',
      'ShipCity': 'Reggio Emilia',
      'ShipRegion': null,
      'ShipCountry': 'Italy'
    },
    {
      'OrderID': 10563,
      'CustomerID': 'RICAR',
      'OrderDate': '1997-06-10T00:00:00.000Z',
      'ShippedDate': '1997-06-24T00:00:00.000Z',
      'Freight': 60.43,
      'ShipName': 'Ricardo Adocicados',
      'ShipAddress': 'Av. Copacabana, 267',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10564,
      'CustomerID': 'RATTC',
      'OrderDate': '1997-06-10T00:00:00.000Z',
      'ShippedDate': '1997-06-16T00:00:00.000Z',
      'Freight': 13.75,
      'ShipName': 'Rattlesnake Canyon Grocery',
      'ShipAddress': '2817 Milton Dr.',
      'ShipCity': 'Albuquerque',
      'ShipRegion': 'NM',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10565,
      'CustomerID': 'MEREP',
      'OrderDate': '1997-06-11T00:00:00.000Z',
      'ShippedDate': '1997-06-18T00:00:00.000Z',
      'Freight': 7.15,

```

```

    'ShipName': 'Mère Paillarde',
    'ShipAddress': '43 rue St. Laurent',
    'ShipCity': 'Montréal',
    'ShipRegion': 'Québec',
    'ShipCountry': 'Canada'
  },
  {
    'OrderID': 10566,
    'CustomerID': 'BLONP',
    'OrderDate': '1997-06-12T00:00:00.000Z',
    'ShippedDate': '1997-06-18T00:00:00.000Z',
    'Freight': 88.4,
    'ShipName': 'Blondel père et fils',
    'ShipAddress': '24, place Kléber',
    'ShipCity': 'Strasbourg',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10567,
    'CustomerID': 'HUNGO',
    'OrderDate': '1997-06-12T00:00:00.000Z',
    'ShippedDate': '1997-06-17T00:00:00.000Z',
    'Freight': 33.97,
    'ShipName': 'Hungry Owl All-Night Grocers',
    'ShipAddress': '8 Johnstown Road',
    'ShipCity': 'Cork',
    'ShipRegion': 'Co. Cork',
    'ShipCountry': 'Ireland'
  },
  {
    'OrderID': 10568,
    'CustomerID': 'GALED',
    'OrderDate': '1997-06-13T00:00:00.000Z',
    'ShippedDate': '1997-07-09T00:00:00.000Z',
    'Freight': 6.54,
    'ShipName': 'Galería del gastronómo',
    'ShipAddress': 'Rambla de Cataluña, 23',
    'ShipCity': 'Barcelona',
    'ShipRegion': null,
    'ShipCountry': 'Spain'
  },
  {
    'OrderID': 10569,
    'CustomerID': 'RATTC',
    'OrderDate': '1997-06-16T00:00:00.000Z',
    'ShippedDate': '1997-07-11T00:00:00.000Z',
    'Freight': 58.98,
    'ShipName': 'Rattlesnake Canyon Grocery',
    'ShipAddress': '2817 Milton Dr.',
    'ShipCity': 'Albuquerque',
    'ShipRegion': 'NM',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10570,
    'CustomerID': 'MEREP',

```



```

      'OrderDate': '1997-06-17T00:00:00.000Z',
      'ShippedDate': '1997-06-19T00:00:00.000Z',
      'Freight': 188.99,
      'ShipName': 'Mère Paillarde',
      'ShipAddress': '43 rue St. Laurent',
      'ShipCity': 'Montréal',
      'ShipRegion': 'Québec',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10571,
      'CustomerID': 'ERNSH',
      'OrderDate': '1997-06-17T00:00:00.000Z',
      'ShippedDate': '1997-07-04T00:00:00.000Z',
      'Freight': 26.06,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10572,
      'CustomerID': 'BERGS',
      'OrderDate': '1997-06-18T00:00:00.000Z',
      'ShippedDate': '1997-06-25T00:00:00.000Z',
      'Freight': 116.43,
      'ShipName': 'Berglunds snabbköp',
      'ShipAddress': 'Berguvsvägen 8',
      'ShipCity': 'Luleå',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 10573,
      'CustomerID': 'ANTON',
      'OrderDate': '1997-06-19T00:00:00.000Z',
      'ShippedDate': '1997-06-20T00:00:00.000Z',
      'Freight': 84.84,
      'ShipName': 'Antonio Moreno Taquería',
      'ShipAddress': 'Mataderos 2312',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10574,
      'CustomerID': 'TRAIH',
      'OrderDate': '1997-06-19T00:00:00.000Z',
      'ShippedDate': '1997-06-30T00:00:00.000Z',
      'Freight': 37.6,
      'ShipName': 'Trail\ Head Gourmet Provisioners',
      'ShipAddress': '722 DaVinci Blvd.',
      'ShipCity': 'Kirkland',
      'ShipRegion': 'WA',
      'ShipCountry': 'USA'
    }
  ],

```

```

{
  'OrderID': 10575,
  'CustomerID': 'MORGK',
  'OrderDate': '1997-06-20T00:00:00.000Z',
  'ShippedDate': '1997-06-30T00:00:00.000Z',
  'Freight': 127.34,
  'ShipName': 'Morgenstern Gesundkost',
  'ShipAddress': 'Heerstr. 22',
  'ShipCity': 'Leipzig',
  'ShipRegion': null,
  'ShipCountry': 'Germany'
},
{
  'OrderID': 10576,
  'CustomerID': 'TORTU',
  'OrderDate': '1997-06-23T00:00:00.000Z',
  'ShippedDate': '1997-06-30T00:00:00.000Z',
  'Freight': 18.56,
  'ShipName': 'Tortuga Restaurante',
  'ShipAddress': 'Avda. Azteca 123',
  'ShipCity': 'México D.F.',
  'ShipRegion': null,
  'ShipCountry': 'Mexico'
},
{
  'OrderID': 10577,
  'CustomerID': 'TRAIH',
  'OrderDate': '1997-06-23T00:00:00.000Z',
  'ShippedDate': '1997-06-30T00:00:00.000Z',
  'Freight': 25.41,
  'ShipName': 'Trail\' Head Gourmet Provisioners',
  'ShipAddress': '722 DaVinci Blvd.',
  'ShipCity': 'Kirkland',
  'ShipRegion': 'WA',
  'ShipCountry': 'USA'
},
{
  'OrderID': 10578,
  'CustomerID': 'BSBEV',
  'OrderDate': '1997-06-24T00:00:00.000Z',
  'ShippedDate': '1997-07-25T00:00:00.000Z',
  'Freight': 29.6,
  'ShipName': 'B\' Beverages',
  'ShipAddress': 'Fauntleroy Circus',
  'ShipCity': 'London',
  'ShipRegion': null,
  'ShipCountry': 'UK'
},
{
  'OrderID': 10579,
  'CustomerID': 'LETSS',
  'OrderDate': '1997-06-25T00:00:00.000Z',
  'ShippedDate': '1997-07-04T00:00:00.000Z',
  'Freight': 13.73,
  'ShipName': 'Let\' Stop N Shop',
  'ShipAddress': '87 Polk St. Suite 5',
  'ShipCity': 'San Francisco',

```

```

      'ShipRegion': 'CA',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10580,
      'CustomerID': 'OTTIK',
      'OrderDate': '1997-06-26T00:00:00.000Z',
      'ShippedDate': '1997-07-01T00:00:00.000Z',
      'Freight': 75.89,
      'ShipName': 'Ottilies Käseladen',
      'ShipAddress': 'Mehrheimerstr. 369',
      'ShipCity': 'Köln',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10581,
      'CustomerID': 'FAMIA',
      'OrderDate': '1997-06-26T00:00:00.000Z',
      'ShippedDate': '1997-07-02T00:00:00.000Z',
      'Freight': 3.01,
      'ShipName': 'Familia Arquibaldo',
      'ShipAddress': 'Rua Orós, 92',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10582,
      'CustomerID': 'BLAUS',
      'OrderDate': '1997-06-27T00:00:00.000Z',
      'ShippedDate': '1997-07-14T00:00:00.000Z',
      'Freight': 27.71,
      'ShipName': 'Blauer See Delikatessen',
      'ShipAddress': 'Forsterstr. 57',
      'ShipCity': 'Mannheim',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10583,
      'CustomerID': 'WARTH',
      'OrderDate': '1997-06-30T00:00:00.000Z',
      'ShippedDate': '1997-07-04T00:00:00.000Z',
      'Freight': 7.28,
      'ShipName': 'Wartian Herkku',
      'ShipAddress': 'Torikatu 38',
      'ShipCity': 'Oulu',
      'ShipRegion': null,
      'ShipCountry': 'Finland'
    },
    {
      'OrderID': 10584,
      'CustomerID': 'BLONP',
      'OrderDate': '1997-06-30T00:00:00.000Z',
      'ShippedDate': '1997-07-04T00:00:00.000Z',
      'Freight': 59.14,

```

```

      'ShipName': 'Blondel père et fils',
      'ShipAddress': '24, place Kléber',
      'ShipCity': 'Strasbourg',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10585,
      'CustomerID': 'WELLI',
      'OrderDate': '1997-07-01T00:00:00.000Z',
      'ShippedDate': '1997-07-10T00:00:00.000Z',
      'Freight': 13.41,
      'ShipName': 'Wellington Importadora',
      'ShipAddress': 'Rua do Mercado, 12',
      'ShipCity': 'Resende',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10586,
      'CustomerID': 'REGGC',
      'OrderDate': '1997-07-02T00:00:00.000Z',
      'ShippedDate': '1997-07-09T00:00:00.000Z',
      'Freight': 0.48,
      'ShipName': 'Reggiani Caseifici',
      'ShipAddress': 'Strada Provinciale 124',
      'ShipCity': 'Reggio Emilia',
      'ShipRegion': null,
      'ShipCountry': 'Italy'
    },
    {
      'OrderID': 10587,
      'CustomerID': 'QUEDE',
      'OrderDate': '1997-07-02T00:00:00.000Z',
      'ShippedDate': '1997-07-09T00:00:00.000Z',
      'Freight': 62.52,
      'ShipName': 'Que Delícia',
      'ShipAddress': 'Rua da Panificadora, 12',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10588,
      'CustomerID': 'QUICK',
      'OrderDate': '1997-07-03T00:00:00.000Z',
      'ShippedDate': '1997-07-10T00:00:00.000Z',
      'Freight': 194.67,
      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10589,
      'CustomerID': 'GREAL',

```

```

      'OrderDate': '1997-07-04T00:00:00.000Z',
      'ShippedDate': '1997-07-14T00:00:00.000Z',
      'Freight': 4.42,
      'ShipName': 'Great Lakes Food Market',
      'ShipAddress': '2732 Baker Blvd.',
      'ShipCity': 'Eugene',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10590,
      'CustomerID': 'MEREP',
      'OrderDate': '1997-07-07T00:00:00.000Z',
      'ShippedDate': '1997-07-14T00:00:00.000Z',
      'Freight': 44.77,
      'ShipName': 'Mère Paillarde',
      'ShipAddress': '43 rue St. Laurent',
      'ShipCity': 'Montréal',
      'ShipRegion': 'Québec',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10591,
      'CustomerID': 'VAFFE',
      'OrderDate': '1997-07-07T00:00:00.000Z',
      'ShippedDate': '1997-07-16T00:00:00.000Z',
      'Freight': 55.92,
      'ShipName': 'Vaffeljernet',
      'ShipAddress': 'Smagsloget 45',
      'ShipCity': 'Århus',
      'ShipRegion': null,
      'ShipCountry': 'Denmark'
    },
    {
      'OrderID': 10592,
      'CustomerID': 'LEHMS',
      'OrderDate': '1997-07-08T00:00:00.000Z',
      'ShippedDate': '1997-07-16T00:00:00.000Z',
      'Freight': 32.1,
      'ShipName': 'Lehmanns Marktstand',
      'ShipAddress': 'Magazinweg 7',
      'ShipCity': 'Frankfurt a.M.',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10593,
      'CustomerID': 'LEHMS',
      'OrderDate': '1997-07-09T00:00:00.000Z',
      'ShippedDate': '1997-08-13T00:00:00.000Z',
      'Freight': 174.2,
      'ShipName': 'Lehmanns Marktstand',
      'ShipAddress': 'Magazinweg 7',
      'ShipCity': 'Frankfurt a.M.',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    }
  ],

```

```

{
  'OrderID': 10594,
  'CustomerID': 'OLDWO',
  'OrderDate': '1997-07-09T00:00:00.000Z',
  'ShippedDate': '1997-07-16T00:00:00.000Z',
  'Freight': 5.24,
  'ShipName': 'Old World Delicatessen',
  'ShipAddress': '2743 Bering St.',
  'ShipCity': 'Anchorage',
  'ShipRegion': 'AK',
  'ShipCountry': 'USA'
},
{
  'OrderID': 10595,
  'CustomerID': 'ERNSH',
  'OrderDate': '1997-07-10T00:00:00.000Z',
  'ShippedDate': '1997-07-14T00:00:00.000Z',
  'Freight': 96.78,
  'ShipName': 'Ernst Handel',
  'ShipAddress': 'Kirchgasse 6',
  'ShipCity': 'Graz',
  'ShipRegion': null,
  'ShipCountry': 'Austria'
},
{
  'OrderID': 10596,
  'CustomerID': 'WHITC',
  'OrderDate': '1997-07-11T00:00:00.000Z',
  'ShippedDate': '1997-08-12T00:00:00.000Z',
  'Freight': 16.34,
  'ShipName': 'White Clover Markets',
  'ShipAddress': '1029 - 12th Ave. S.',
  'ShipCity': 'Seattle',
  'ShipRegion': 'WA',
  'ShipCountry': 'USA'
},
{
  'OrderID': 10597,
  'CustomerID': 'PICCO',
  'OrderDate': '1997-07-11T00:00:00.000Z',
  'ShippedDate': '1997-07-18T00:00:00.000Z',
  'Freight': 35.12,
  'ShipName': 'Piccolo und mehr',
  'ShipAddress': 'Geislweg 14',
  'ShipCity': 'Salzburg',
  'ShipRegion': null,
  'ShipCountry': 'Austria'
},
{
  'OrderID': 10598,
  'CustomerID': 'RATTC',
  'OrderDate': '1997-07-14T00:00:00.000Z',
  'ShippedDate': '1997-07-18T00:00:00.000Z',
  'Freight': 44.42,
  'ShipName': 'Rattlesnake Canyon Grocery',
  'ShipAddress': '2817 Milton Dr.',
  'ShipCity': 'Albuquerque',

```

```

      'ShipRegion': 'NM',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10599,
      'CustomerID': 'BSBEV',
      'OrderDate': '1997-07-15T00:00:00.000Z',
      'ShippedDate': '1997-07-21T00:00:00.000Z',
      'Freight': 29.98,
      'ShipName': 'B Beverages',
      'ShipAddress': 'Fauntleroy Circus',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10600,
      'CustomerID': 'HUNGC',
      'OrderDate': '1997-07-16T00:00:00.000Z',
      'ShippedDate': '1997-07-21T00:00:00.000Z',
      'Freight': 45.13,
      'ShipName': 'Hungry Coyote Import Store',
      'ShipAddress': 'City Center Plaza 516 Main St.',
      'ShipCity': 'Elgin',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10601,
      'CustomerID': 'HILAA',
      'OrderDate': '1997-07-16T00:00:00.000Z',
      'ShippedDate': '1997-07-22T00:00:00.000Z',
      'Freight': 58.3,
      'ShipName': 'HILARION-Abastos',
      'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
      'ShipCity': 'San Cristóbal',
      'ShipRegion': 'Táchira',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10602,
      'CustomerID': 'VAFFE',
      'OrderDate': '1997-07-17T00:00:00.000Z',
      'ShippedDate': '1997-07-22T00:00:00.000Z',
      'Freight': 2.92,
      'ShipName': 'Vaffeljernet',
      'ShipAddress': 'Smagsloget 45',
      'ShipCity': 'Århus',
      'ShipRegion': null,
      'ShipCountry': 'Denmark'
    },
    {
      'OrderID': 10603,
      'CustomerID': 'SAVEA',
      'OrderDate': '1997-07-18T00:00:00.000Z',
      'ShippedDate': '1997-08-08T00:00:00.000Z',
      'Freight': 48.77,

```

```

      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10604,
      'CustomerID': 'FURIB',
      'OrderDate': '1997-07-18T00:00:00.000Z',
      'ShippedDate': '1997-07-29T00:00:00.000Z',
      'Freight': 7.46,
      'ShipName': 'Furia Bacalhau e Frutos do Mar',
      'ShipAddress': 'Jardim das rosas n. 32',
      'ShipCity': 'Lisboa',
      'ShipRegion': null,
      'ShipCountry': 'Portugal'
    },
    {
      'OrderID': 10605,
      'CustomerID': 'MEREP',
      'OrderDate': '1997-07-21T00:00:00.000Z',
      'ShippedDate': '1997-07-29T00:00:00.000Z',
      'Freight': 379.13,
      'ShipName': 'Mère Paillard',
      'ShipAddress': '43 rue St. Laurent',
      'ShipCity': 'Montréal',
      'ShipRegion': 'Québec',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10606,
      'CustomerID': 'TRADH',
      'OrderDate': '1997-07-22T00:00:00.000Z',
      'ShippedDate': '1997-07-31T00:00:00.000Z',
      'Freight': 79.4,
      'ShipName': 'Tradição Hipermercados',
      'ShipAddress': 'Av. Inês de Castro, 414',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10607,
      'CustomerID': 'SAVEA',
      'OrderDate': '1997-07-22T00:00:00.000Z',
      'ShippedDate': '1997-07-25T00:00:00.000Z',
      'Freight': 200.24,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10608,
      'CustomerID': 'TOMSP',

```



```

      'OrderDate': '1997-07-23T00:00:00.000Z',
      'ShippedDate': '1997-08-01T00:00:00.000Z',
      'Freight': 27.79,
      'ShipName': 'Toms Spezialitäten',
      'ShipAddress': 'Luisenstr. 48',
      'ShipCity': 'Münster',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10609,
      'CustomerID': 'DUMON',
      'OrderDate': '1997-07-24T00:00:00.000Z',
      'ShippedDate': '1997-07-30T00:00:00.000Z',
      'Freight': 1.85,
      'ShipName': 'Du monde entier',
      'ShipAddress': '67, rue des Cinquante Otages',
      'ShipCity': 'Nantes',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10610,
      'CustomerID': 'LAMAI',
      'OrderDate': '1997-07-25T00:00:00.000Z',
      'ShippedDate': '1997-08-06T00:00:00.000Z',
      'Freight': 26.78,
      'ShipName': 'La maison d\'Asie',
      'ShipAddress': '1 rue Alsace-Lorraine',
      'ShipCity': 'Toulouse',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10611,
      'CustomerID': 'WOLZA',
      'OrderDate': '1997-07-25T00:00:00.000Z',
      'ShippedDate': '1997-08-01T00:00:00.000Z',
      'Freight': 80.65,
      'ShipName': 'Wolski Zajazd',
      'ShipAddress': 'ul. Filtrowa 68',
      'ShipCity': 'Warszawa',
      'ShipRegion': null,
      'ShipCountry': 'Poland'
    },
    {
      'OrderID': 10612,
      'CustomerID': 'SAVEA',
      'OrderDate': '1997-07-28T00:00:00.000Z',
      'ShippedDate': '1997-08-01T00:00:00.000Z',
      'Freight': 544.08,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    }
  ],

```

```

{
  'OrderID': 10613,
  'CustomerID': 'HILAA',
  'OrderDate': '1997-07-29T00:00:00.000Z',
  'ShippedDate': '1997-08-01T00:00:00.000Z',
  'Freight': 8.11,
  'ShipName': 'HILARION-Abastos',
  'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
  'ShipCity': 'San Cristóbal',
  'ShipRegion': 'Táchira',
  'ShipCountry': 'Venezuela'
},
{
  'OrderID': 10614,
  'CustomerID': 'BLAUS',
  'OrderDate': '1997-07-29T00:00:00.000Z',
  'ShippedDate': '1997-08-01T00:00:00.000Z',
  'Freight': 1.93,
  'ShipName': 'Blauer See Delikatessen',
  'ShipAddress': 'Forsterstr. 57',
  'ShipCity': 'Mannheim',
  'ShipRegion': null,
  'ShipCountry': 'Germany'
},
{
  'OrderID': 10615,
  'CustomerID': 'WILMK',
  'OrderDate': '1997-07-30T00:00:00.000Z',
  'ShippedDate': '1997-08-06T00:00:00.000Z',
  'Freight': 0.75,
  'ShipName': 'Wilman Kala',
  'ShipAddress': 'Keskuskatu 45',
  'ShipCity': 'Helsinki',
  'ShipRegion': null,
  'ShipCountry': 'Finland'
},
{
  'OrderID': 10616,
  'CustomerID': 'GREAL',
  'OrderDate': '1997-07-31T00:00:00.000Z',
  'ShippedDate': '1997-08-05T00:00:00.000Z',
  'Freight': 116.53,
  'ShipName': 'Great Lakes Food Market',
  'ShipAddress': '2732 Baker Blvd.',
  'ShipCity': 'Eugene',
  'ShipRegion': 'OR',
  'ShipCountry': 'USA'
},
{
  'OrderID': 10617,
  'CustomerID': 'GREAL',
  'OrderDate': '1997-07-31T00:00:00.000Z',
  'ShippedDate': '1997-08-04T00:00:00.000Z',
  'Freight': 18.53,
  'ShipName': 'Great Lakes Food Market',
  'ShipAddress': '2732 Baker Blvd.',
  'ShipCity': 'Eugene',

```

```

      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10618,
      'CustomerID': 'MEREP',
      'OrderDate': '1997-08-01T00:00:00.000Z',
      'ShippedDate': '1997-08-08T00:00:00.000Z',
      'Freight': 154.68,
      'ShipName': 'Mère Paillarde',
      'ShipAddress': '43 rue St. Laurent',
      'ShipCity': 'Montréal',
      'ShipRegion': 'Québec',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10619,
      'CustomerID': 'MEREP',
      'OrderDate': '1997-08-04T00:00:00.000Z',
      'ShippedDate': '1997-08-07T00:00:00.000Z',
      'Freight': 91.05,
      'ShipName': 'Mère Paillarde',
      'ShipAddress': '43 rue St. Laurent',
      'ShipCity': 'Montréal',
      'ShipRegion': 'Québec',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10620,
      'CustomerID': 'LAUGB',
      'OrderDate': '1997-08-05T00:00:00.000Z',
      'ShippedDate': '1997-08-14T00:00:00.000Z',
      'Freight': 0.94,
      'ShipName': 'Laughing Bacchus Wine Cellars',
      'ShipAddress': '2319 Elm St.',
      'ShipCity': 'Vancouver',
      'ShipRegion': 'BC',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10621,
      'CustomerID': 'ISLAT',
      'OrderDate': '1997-08-05T00:00:00.000Z',
      'ShippedDate': '1997-08-11T00:00:00.000Z',
      'Freight': 23.73,
      'ShipName': 'Island Trading',
      'ShipAddress': 'Garden House Crowther Way',
      'ShipCity': 'Cowes',
      'ShipRegion': 'Isle of Wight',
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10622,
      'CustomerID': 'RICAR',
      'OrderDate': '1997-08-06T00:00:00.000Z',
      'ShippedDate': '1997-08-11T00:00:00.000Z',
      'Freight': 50.97,

```

```

    'ShipName': 'Ricardo Adocicados',
    'ShipAddress': 'Av. Copacabana, 267',
    'ShipCity': 'Rio de Janeiro',
    'ShipRegion': 'RJ',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10623,
    'CustomerID': 'FRANK',
    'OrderDate': '1997-08-07T00:00:00.000Z',
    'ShippedDate': '1997-08-12T00:00:00.000Z',
    'Freight': 97.18,
    'ShipName': 'Frankenversand',
    'ShipAddress': 'Berliner Platz 43',
    'ShipCity': 'München',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10624,
    'CustomerID': 'THECR',
    'OrderDate': '1997-08-07T00:00:00.000Z',
    'ShippedDate': '1997-08-19T00:00:00.000Z',
    'Freight': 94.8,
    'ShipName': 'The Cracker Box',
    'ShipAddress': '55 Grizzly Peak Rd.',
    'ShipCity': 'Butte',
    'ShipRegion': 'MT',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10625,
    'CustomerID': 'ANATR',
    'OrderDate': '1997-08-08T00:00:00.000Z',
    'ShippedDate': '1997-08-14T00:00:00.000Z',
    'Freight': 43.9,
    'ShipName': 'Ana Trujillo Emparedados y helados',
    'ShipAddress': 'Avda. de la Constitución 2222',
    'ShipCity': 'México D.F.',
    'ShipRegion': null,
    'ShipCountry': 'Mexico'
  },
  {
    'OrderID': 10626,
    'CustomerID': 'BERGS',
    'OrderDate': '1997-08-11T00:00:00.000Z',
    'ShippedDate': '1997-08-20T00:00:00.000Z',
    'Freight': 138.69,
    'ShipName': 'Berglunds snabbköp',
    'ShipAddress': 'Berguvsvägen 8',
    'ShipCity': 'Luleå',
    'ShipRegion': null,
    'ShipCountry': 'Sweden'
  },
  {
    'OrderID': 10627,
    'CustomerID': 'SAVEA',

```

```

      'OrderDate': '1997-08-11T00:00:00.000Z',
      'ShippedDate': '1997-08-21T00:00:00.000Z',
      'Freight': 107.46,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10628,
      'CustomerID': 'BLONP',
      'OrderDate': '1997-08-12T00:00:00.000Z',
      'ShippedDate': '1997-08-20T00:00:00.000Z',
      'Freight': 30.36,
      'ShipName': 'Blondel père et fils',
      'ShipAddress': '24, place Kléber',
      'ShipCity': 'Strasbourg',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10629,
      'CustomerID': 'GODOS',
      'OrderDate': '1997-08-12T00:00:00.000Z',
      'ShippedDate': '1997-08-20T00:00:00.000Z',
      'Freight': 85.46,
      'ShipName': 'Godos Cocina Típica',
      'ShipAddress': 'C/ Romero, 33',
      'ShipCity': 'Sevilla',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    },
    {
      'OrderID': 10630,
      'CustomerID': 'KOENE',
      'OrderDate': '1997-08-13T00:00:00.000Z',
      'ShippedDate': '1997-08-19T00:00:00.000Z',
      'Freight': 32.35,
      'ShipName': 'Königlich Essen',
      'ShipAddress': 'Maubelstr. 90',
      'ShipCity': 'Brandenburg',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10631,
      'CustomerID': 'LAMAI',
      'OrderDate': '1997-08-14T00:00:00.000Z',
      'ShippedDate': '1997-08-15T00:00:00.000Z',
      'Freight': 0.87,
      'ShipName': 'La maison d\'Asie',
      'ShipAddress': '1 rue Alsace-Lorraine',
      'ShipCity': 'Toulouse',
      'ShipRegion': null,
      'ShipCountry': 'France'
    }
  ],

```

```

{
  'OrderID': 10632,
  'CustomerID': 'WANDK',
  'OrderDate': '1997-08-14T00:00:00.000Z',
  'ShippedDate': '1997-08-19T00:00:00.000Z',
  'Freight': 41.38,
  'ShipName': 'Die Wandernde Kuh',
  'ShipAddress': 'Adenauerallee 900',
  'ShipCity': 'Stuttgart',
  'ShipRegion': null,
  'ShipCountry': 'Germany'
},
{
  'OrderID': 10633,
  'CustomerID': 'ERNSH',
  'OrderDate': '1997-08-15T00:00:00.000Z',
  'ShippedDate': '1997-08-18T00:00:00.000Z',
  'Freight': 477.9,
  'ShipName': 'Ernst Handel',
  'ShipAddress': 'Kirchgasse 6',
  'ShipCity': 'Graz',
  'ShipRegion': null,
  'ShipCountry': 'Austria'
},
{
  'OrderID': 10634,
  'CustomerID': 'FOLIG',
  'OrderDate': '1997-08-15T00:00:00.000Z',
  'ShippedDate': '1997-08-21T00:00:00.000Z',
  'Freight': 487.38,
  'ShipName': 'Folies gourmandes',
  'ShipAddress': '184, chaussée de Tournai',
  'ShipCity': 'Lille',
  'ShipRegion': null,
  'ShipCountry': 'France'
},
{
  'OrderID': 10635,
  'CustomerID': 'MAGAA',
  'OrderDate': '1997-08-18T00:00:00.000Z',
  'ShippedDate': '1997-08-21T00:00:00.000Z',
  'Freight': 47.46,
  'ShipName': 'Magazzini Alimentari Riuniti',
  'ShipAddress': 'Via Ludovico il Moro 22',
  'ShipCity': 'Bergamo',
  'ShipRegion': null,
  'ShipCountry': 'Italy'
},
{
  'OrderID': 10636,
  'CustomerID': 'WARTH',
  'OrderDate': '1997-08-19T00:00:00.000Z',
  'ShippedDate': '1997-08-26T00:00:00.000Z',
  'Freight': 1.15,
  'ShipName': 'Wartian Herkku',
  'ShipAddress': 'Torikatu 38',
  'ShipCity': 'Oulu',

```

```

    'ShipRegion': null,
    'ShipCountry': 'Finland'
  },
  {
    'OrderID': 10637,
    'CustomerID': 'QUEEN',
    'OrderDate': '1997-08-19T00:00:00.000Z',
    'ShippedDate': '1997-08-26T00:00:00.000Z',
    'Freight': 201.29,
    'ShipName': 'Queen Cozinha',
    'ShipAddress': 'Alameda dos Canários, 891',
    'ShipCity': 'Sao Paulo',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10638,
    'CustomerID': 'LINOD',
    'OrderDate': '1997-08-20T00:00:00.000Z',
    'ShippedDate': '1997-09-01T00:00:00.000Z',
    'Freight': 158.44,
    'ShipName': 'LINO-Delicateses',
    'ShipAddress': 'Ave. 5 de Mayo Porlamar',
    'ShipCity': 'I. de Margarita',
    'ShipRegion': 'Nueva Esparta',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10639,
    'CustomerID': 'SANTG',
    'OrderDate': '1997-08-20T00:00:00.000Z',
    'ShippedDate': '1997-08-27T00:00:00.000Z',
    'Freight': 38.64,
    'ShipName': 'Santé Gourmet',
    'ShipAddress': 'Erling Skakkes gate 78',
    'ShipCity': 'Stavern',
    'ShipRegion': null,
    'ShipCountry': 'Norway'
  },
  {
    'OrderID': 10640,
    'CustomerID': 'WANDK',
    'OrderDate': '1997-08-21T00:00:00.000Z',
    'ShippedDate': '1997-08-28T00:00:00.000Z',
    'Freight': 23.55,
    'ShipName': 'Die Wandernde Kuh',
    'ShipAddress': 'Adenauerallee 900',
    'ShipCity': 'Stuttgart',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10641,
    'CustomerID': 'HILAA',
    'OrderDate': '1997-08-22T00:00:00.000Z',
    'ShippedDate': '1997-08-26T00:00:00.000Z',
    'Freight': 179.61,

```

```

      'ShipName': 'HILARION-Abastos',
      'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
      'ShipCity': 'San Cristóbal',
      'ShipRegion': 'Táchira',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10642,
      'CustomerID': 'SIMOB',
      'OrderDate': '1997-08-22T00:00:00.000Z',
      'ShippedDate': '1997-09-05T00:00:00.000Z',
      'Freight': 41.89,
      'ShipName': 'Simons bistro',
      'ShipAddress': 'Vinbæltet 34',
      'ShipCity': 'Kobenhavn',
      'ShipRegion': null,
      'ShipCountry': 'Denmark'
    },
    {
      'OrderID': 10643,
      'CustomerID': 'ALFKI',
      'OrderDate': '1997-08-25T00:00:00.000Z',
      'ShippedDate': '1997-09-02T00:00:00.000Z',
      'Freight': 29.46,
      'ShipName': 'Alfreds Futterkiste',
      'ShipAddress': 'Obere Str. 57',
      'ShipCity': 'Berlin',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10644,
      'CustomerID': 'WELLI',
      'OrderDate': '1997-08-25T00:00:00.000Z',
      'ShippedDate': '1997-09-01T00:00:00.000Z',
      'Freight': 0.14,
      'ShipName': 'Wellington Importadora',
      'ShipAddress': 'Rua do Mercado, 12',
      'ShipCity': 'Resende',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10645,
      'CustomerID': 'HANAR',
      'OrderDate': '1997-08-26T00:00:00.000Z',
      'ShippedDate': '1997-09-02T00:00:00.000Z',
      'Freight': 12.41,
      'ShipName': 'Hanari Carnes',
      'ShipAddress': 'Rua do Paço, 67',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10646,
      'CustomerID': 'HUNGO',

```



```

      'OrderDate': '1997-08-27T00:00:00.000Z',
      'ShippedDate': '1997-09-03T00:00:00.000Z',
      'Freight': 142.33,
      'ShipName': 'Hungry Owl All-Night Grocers',
      'ShipAddress': '8 Johnstown Road',
      'ShipCity': 'Cork',
      'ShipRegion': 'Co. Cork',
      'ShipCountry': 'Ireland'
    },
    {
      'OrderID': 10647,
      'CustomerID': 'QUEDE',
      'OrderDate': '1997-08-27T00:00:00.000Z',
      'ShippedDate': '1997-09-03T00:00:00.000Z',
      'Freight': 45.54,
      'ShipName': 'Que Delícia',
      'ShipAddress': 'Rua da Panificadora, 12',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10648,
      'CustomerID': 'RICAR',
      'OrderDate': '1997-08-28T00:00:00.000Z',
      'ShippedDate': '1997-09-09T00:00:00.000Z',
      'Freight': 14.25,
      'ShipName': 'Ricardo Adocicados',
      'ShipAddress': 'Av. Copacabana, 267',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10649,
      'CustomerID': 'MAISD',
      'OrderDate': '1997-08-28T00:00:00.000Z',
      'ShippedDate': '1997-08-29T00:00:00.000Z',
      'Freight': 6.2,
      'ShipName': 'Maison Dewey',
      'ShipAddress': 'Rue Joseph-Bens 532',
      'ShipCity': 'Bruxelles',
      'ShipRegion': null,
      'ShipCountry': 'Belgium'
    },
    {
      'OrderID': 10650,
      'CustomerID': 'FAMIA',
      'OrderDate': '1997-08-29T00:00:00.000Z',
      'ShippedDate': '1997-09-03T00:00:00.000Z',
      'Freight': 176.81,
      'ShipName': 'Familia Arquibaldo',
      'ShipAddress': 'Rua Orós, 92',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    }
  ],

```

```

{
  'OrderID': 10651,
  'CustomerID': 'WANDK',
  'OrderDate': '1997-09-01T00:00:00.000Z',
  'ShippedDate': '1997-09-11T00:00:00.000Z',
  'Freight': 20.6,
  'ShipName': 'Die Wandernde Kuh',
  'ShipAddress': 'Adenauerallee 900',
  'ShipCity': 'Stuttgart',
  'ShipRegion': null,
  'ShipCountry': 'Germany'
},
{
  'OrderID': 10652,
  'CustomerID': 'GOURL',
  'OrderDate': '1997-09-01T00:00:00.000Z',
  'ShippedDate': '1997-09-08T00:00:00.000Z',
  'Freight': 7.14,
  'ShipName': 'Gourmet Lanchonetes',
  'ShipAddress': 'Av. Brasil, 442',
  'ShipCity': 'Campinas',
  'ShipRegion': 'SP',
  'ShipCountry': 'Brazil'
},
{
  'OrderID': 10653,
  'CustomerID': 'FRANK',
  'OrderDate': '1997-09-02T00:00:00.000Z',
  'ShippedDate': '1997-09-19T00:00:00.000Z',
  'Freight': 93.25,
  'ShipName': 'Frankenversand',
  'ShipAddress': 'Berliner Platz 43',
  'ShipCity': 'München',
  'ShipRegion': null,
  'ShipCountry': 'Germany'
},
{
  'OrderID': 10654,
  'CustomerID': 'BERGS',
  'OrderDate': '1997-09-02T00:00:00.000Z',
  'ShippedDate': '1997-09-11T00:00:00.000Z',
  'Freight': 55.26,
  'ShipName': 'Berglunds snabbköp',
  'ShipAddress': 'Berguvsvägen 8',
  'ShipCity': 'Luleå',
  'ShipRegion': null,
  'ShipCountry': 'Sweden'
},
{
  'OrderID': 10655,
  'CustomerID': 'REGGC',
  'OrderDate': '1997-09-03T00:00:00.000Z',
  'ShippedDate': '1997-09-11T00:00:00.000Z',
  'Freight': 4.41,
  'ShipName': 'Reggiani Caseifici',
  'ShipAddress': 'Strada Provinciale 124',
  'ShipCity': 'Reggio Emilia',

```

```

    'ShipRegion': null,
    'ShipCountry': 'Italy'
  },
  {
    'OrderID': 10656,
    'CustomerID': 'GREAL',
    'OrderDate': '1997-09-04T00:00:00.000Z',
    'ShippedDate': '1997-09-10T00:00:00.000Z',
    'Freight': 57.15,
    'ShipName': 'Great Lakes Food Market',
    'ShipAddress': '2732 Baker Blvd.',
    'ShipCity': 'Eugene',
    'ShipRegion': 'OR',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10657,
    'CustomerID': 'SAVEA',
    'OrderDate': '1997-09-04T00:00:00.000Z',
    'ShippedDate': '1997-09-15T00:00:00.000Z',
    'Freight': 352.69,
    'ShipName': 'Save-a-lot Markets',
    'ShipAddress': '187 Suffolk Ln.',
    'ShipCity': 'Boise',
    'ShipRegion': 'ID',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10658,
    'CustomerID': 'QUICK',
    'OrderDate': '1997-09-05T00:00:00.000Z',
    'ShippedDate': '1997-09-08T00:00:00.000Z',
    'Freight': 364.15,
    'ShipName': 'QUICK-Stop',
    'ShipAddress': 'Taucherstraße 10',
    'ShipCity': 'Cunewalde',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10659,
    'CustomerID': 'QUEEN',
    'OrderDate': '1997-09-05T00:00:00.000Z',
    'ShippedDate': '1997-09-10T00:00:00.000Z',
    'Freight': 105.81,
    'ShipName': 'Queen Cozinha',
    'ShipAddress': 'Alameda dos Canários, 891',
    'ShipCity': 'Sao Paulo',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10660,
    'CustomerID': 'HUNGC',
    'OrderDate': '1997-09-08T00:00:00.000Z',
    'ShippedDate': '1997-10-15T00:00:00.000Z',
    'Freight': 111.29,

```

```

      'ShipName': 'Hungry Coyote Import Store',
      'ShipAddress': 'City Center Plaza 516 Main St.',
      'ShipCity': 'Elgin',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10661,
      'CustomerID': 'HUNGO',
      'OrderDate': '1997-09-09T00:00:00.000Z',
      'ShippedDate': '1997-09-15T00:00:00.000Z',
      'Freight': 17.55,
      'ShipName': 'Hungry Owl All-Night Grocers',
      'ShipAddress': '8 Johnstown Road',
      'ShipCity': 'Cork',
      'ShipRegion': 'Co. Cork',
      'ShipCountry': 'Ireland'
    },
    {
      'OrderID': 10662,
      'CustomerID': 'LONEP',
      'OrderDate': '1997-09-09T00:00:00.000Z',
      'ShippedDate': '1997-09-18T00:00:00.000Z',
      'Freight': 1.28,
      'ShipName': 'Lonesome Pine Restaurant',
      'ShipAddress': '89 Chiaroscuro Rd.',
      'ShipCity': 'Portland',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10663,
      'CustomerID': 'BONAP',
      'OrderDate': '1997-09-10T00:00:00.000Z',
      'ShippedDate': '1997-10-03T00:00:00.000Z',
      'Freight': 113.15,
      'ShipName': 'Bon app',
      'ShipAddress': '12, rue des Bouchers',
      'ShipCity': 'Marseille',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10664,
      'CustomerID': 'FURIB',
      'OrderDate': '1997-09-10T00:00:00.000Z',
      'ShippedDate': '1997-09-19T00:00:00.000Z',
      'Freight': 1.27,
      'ShipName': 'Furia Bacalhau e Frutos do Mar',
      'ShipAddress': 'Jardim das rosas n. 32',
      'ShipCity': 'Lisboa',
      'ShipRegion': null,
      'ShipCountry': 'Portugal'
    },
    {
      'OrderID': 10665,
      'CustomerID': 'LONEP',

```

```

      'OrderDate': '1997-09-11T00:00:00.000Z',
      'ShippedDate': '1997-09-17T00:00:00.000Z',
      'Freight': 26.31,
      'ShipName': 'Lonesome Pine Restaurant',
      'ShipAddress': '89 Chiaroscuro Rd.',
      'ShipCity': 'Portland',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10666,
      'CustomerID': 'RICSU',
      'OrderDate': '1997-09-12T00:00:00.000Z',
      'ShippedDate': '1997-09-22T00:00:00.000Z',
      'Freight': 232.42,
      'ShipName': 'Richter Supermarkt',
      'ShipAddress': 'Starenweg 5',
      'ShipCity': 'Genève',
      'ShipRegion': null,
      'ShipCountry': 'Switzerland'
    },
    {
      'OrderID': 10667,
      'CustomerID': 'ERNSH',
      'OrderDate': '1997-09-12T00:00:00.000Z',
      'ShippedDate': '1997-09-19T00:00:00.000Z',
      'Freight': 78.09,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10668,
      'CustomerID': 'WANDK',
      'OrderDate': '1997-09-15T00:00:00.000Z',
      'ShippedDate': '1997-09-23T00:00:00.000Z',
      'Freight': 47.22,
      'ShipName': 'Die Wandernde Kuh',
      'ShipAddress': 'Adenauerallee 900',
      'ShipCity': 'Stuttgart',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10669,
      'CustomerID': 'SIMOB',
      'OrderDate': '1997-09-15T00:00:00.000Z',
      'ShippedDate': '1997-09-22T00:00:00.000Z',
      'Freight': 24.39,
      'ShipName': 'Simons bistro',
      'ShipAddress': 'Vinbæltet 34',
      'ShipCity': 'Kobenhavn',
      'ShipRegion': null,
      'ShipCountry': 'Denmark'
    }
  ],

```

```

{
  'OrderID': 10670,
  'CustomerID': 'FRANK',
  'OrderDate': '1997-09-16T00:00:00.000Z',
  'ShippedDate': '1997-09-18T00:00:00.000Z',
  'Freight': 203.48,
  'ShipName': 'Frankenversand',
  'ShipAddress': 'Berliner Platz 43',
  'ShipCity': 'München',
  'ShipRegion': null,
  'ShipCountry': 'Germany'
},
{
  'OrderID': 10671,
  'CustomerID': 'FRANR',
  'OrderDate': '1997-09-17T00:00:00.000Z',
  'ShippedDate': '1997-09-24T00:00:00.000Z',
  'Freight': 30.34,
  'ShipName': 'France restauration',
  'ShipAddress': '54, rue Royale',
  'ShipCity': 'Nantes',
  'ShipRegion': null,
  'ShipCountry': 'France'
},
{
  'OrderID': 10672,
  'CustomerID': 'BERGS',
  'OrderDate': '1997-09-17T00:00:00.000Z',
  'ShippedDate': '1997-09-26T00:00:00.000Z',
  'Freight': 95.75,
  'ShipName': 'Berglunds snabbköp',
  'ShipAddress': 'Berguvsvägen 8',
  'ShipCity': 'Luleå',
  'ShipRegion': null,
  'ShipCountry': 'Sweden'
},
{
  'OrderID': 10673,
  'CustomerID': 'WILMK',
  'OrderDate': '1997-09-18T00:00:00.000Z',
  'ShippedDate': '1997-09-19T00:00:00.000Z',
  'Freight': 22.76,
  'ShipName': 'Wilman Kala',
  'ShipAddress': 'Keskuskatu 45',
  'ShipCity': 'Helsinki',
  'ShipRegion': null,
  'ShipCountry': 'Finland'
},
{
  'OrderID': 10674,
  'CustomerID': 'ISLAT',
  'OrderDate': '1997-09-18T00:00:00.000Z',
  'ShippedDate': '1997-09-30T00:00:00.000Z',
  'Freight': 0.9,
  'ShipName': 'Island Trading',
  'ShipAddress': 'Garden House Crowther Way',
  'ShipCity': 'Cowes',

```

```

      'ShipRegion': 'Isle of Wight',
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10675,
      'CustomerID': 'FRANK',
      'OrderDate': '1997-09-19T00:00:00.000Z',
      'ShippedDate': '1997-09-23T00:00:00.000Z',
      'Freight': 31.85,
      'ShipName': 'Frankenversand',
      'ShipAddress': 'Berliner Platz 43',
      'ShipCity': 'München',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10676,
      'CustomerID': 'TORTU',
      'OrderDate': '1997-09-22T00:00:00.000Z',
      'ShippedDate': '1997-09-29T00:00:00.000Z',
      'Freight': 2.01,
      'ShipName': 'Tortuga Restaurante',
      'ShipAddress': 'Avda. Azteca 123',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10677,
      'CustomerID': 'ANTON',
      'OrderDate': '1997-09-22T00:00:00.000Z',
      'ShippedDate': '1997-09-26T00:00:00.000Z',
      'Freight': 4.03,
      'ShipName': 'Antonio Moreno Taquería',
      'ShipAddress': 'Mataderos 2312',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10678,
      'CustomerID': 'SAVEA',
      'OrderDate': '1997-09-23T00:00:00.000Z',
      'ShippedDate': '1997-10-16T00:00:00.000Z',
      'Freight': 388.98,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10679,
      'CustomerID': 'BLONP',
      'OrderDate': '1997-09-23T00:00:00.000Z',
      'ShippedDate': '1997-09-30T00:00:00.000Z',
      'Freight': 27.94,

```

```

      'ShipName': 'Blondel père et fils',
      'ShipAddress': '24, place Kléber',
      'ShipCity': 'Strasbourg',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10680,
      'CustomerID': 'OLDWO',
      'OrderDate': '1997-09-24T00:00:00.000Z',
      'ShippedDate': '1997-09-26T00:00:00.000Z',
      'Freight': 26.61,
      'ShipName': 'Old World Delicatessen',
      'ShipAddress': '2743 Bering St.',
      'ShipCity': 'Anchorage',
      'ShipRegion': 'AK',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10681,
      'CustomerID': 'GREAL',
      'OrderDate': '1997-09-25T00:00:00.000Z',
      'ShippedDate': '1997-09-30T00:00:00.000Z',
      'Freight': 76.13,
      'ShipName': 'Great Lakes Food Market',
      'ShipAddress': '2732 Baker Blvd.',
      'ShipCity': 'Eugene',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10682,
      'CustomerID': 'ANTON',
      'OrderDate': '1997-09-25T00:00:00.000Z',
      'ShippedDate': '1997-10-01T00:00:00.000Z',
      'Freight': 36.13,
      'ShipName': 'Antonio Moreno Taquería',
      'ShipAddress': 'Mataderos 2312',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10683,
      'CustomerID': 'DUMON',
      'OrderDate': '1997-09-26T00:00:00.000Z',
      'ShippedDate': '1997-10-01T00:00:00.000Z',
      'Freight': 4.4,
      'ShipName': 'Du monde entier',
      'ShipAddress': '67, rue des Cinquante Otages',
      'ShipCity': 'Nantes',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10684,
      'CustomerID': 'OTTIK',

```



```

      'OrderDate': '1997-09-26T00:00:00.000Z',
      'ShippedDate': '1997-09-30T00:00:00.000Z',
      'Freight': 145.63,
      'ShipName': 'Ottilies Käseladen',
      'ShipAddress': 'Mehrheimerstr. 369',
      'ShipCity': 'Köln',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10685,
      'CustomerID': 'GOURL',
      'OrderDate': '1997-09-29T00:00:00.000Z',
      'ShippedDate': '1997-10-03T00:00:00.000Z',
      'Freight': 33.75,
      'ShipName': 'Gourmet Lanchonetes',
      'ShipAddress': 'Av. Brasil, 442',
      'ShipCity': 'Campinas',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10686,
      'CustomerID': 'PICCO',
      'OrderDate': '1997-09-30T00:00:00.000Z',
      'ShippedDate': '1997-10-08T00:00:00.000Z',
      'Freight': 96.5,
      'ShipName': 'Piccolo und mehr',
      'ShipAddress': 'Geislweg 14',
      'ShipCity': 'Salzburg',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10687,
      'CustomerID': 'HUNGO',
      'OrderDate': '1997-09-30T00:00:00.000Z',
      'ShippedDate': '1997-10-30T00:00:00.000Z',
      'Freight': 296.43,
      'ShipName': 'Hungry Owl All-Night Grocers',
      'ShipAddress': '8 Johnstown Road',
      'ShipCity': 'Cork',
      'ShipRegion': 'Co. Cork',
      'ShipCountry': 'Ireland'
    },
    {
      'OrderID': 10688,
      'CustomerID': 'VAFFE',
      'OrderDate': '1997-10-01T00:00:00.000Z',
      'ShippedDate': '1997-10-07T00:00:00.000Z',
      'Freight': 299.09,
      'ShipName': 'Vaffeljernet',
      'ShipAddress': 'Smagsloget 45',
      'ShipCity': 'Århus',
      'ShipRegion': null,
      'ShipCountry': 'Denmark'
    }
  ],

```

```

{
  'OrderID': 10689,
  'CustomerID': 'BERGS',
  'OrderDate': '1997-10-01T00:00:00.000Z',
  'ShippedDate': '1997-10-07T00:00:00.000Z',
  'Freight': 13.42,
  'ShipName': 'Berglunds snabbköp',
  'ShipAddress': 'Berguvsvägen 8',
  'ShipCity': 'Luleå',
  'ShipRegion': null,
  'ShipCountry': 'Sweden'
},
{
  'OrderID': 10690,
  'CustomerID': 'HANAR',
  'OrderDate': '1997-10-02T00:00:00.000Z',
  'ShippedDate': '1997-10-03T00:00:00.000Z',
  'Freight': 15.8,
  'ShipName': 'Hanari Carnes',
  'ShipAddress': 'Rua do Paço, 67',
  'ShipCity': 'Rio de Janeiro',
  'ShipRegion': 'RJ',
  'ShipCountry': 'Brazil'
},
{
  'OrderID': 10691,
  'CustomerID': 'QUICK',
  'OrderDate': '1997-10-03T00:00:00.000Z',
  'ShippedDate': '1997-10-22T00:00:00.000Z',
  'Freight': 810.05,
  'ShipName': 'QUICK-Stop',
  'ShipAddress': 'Taucherstraße 10',
  'ShipCity': 'Cunewalde',
  'ShipRegion': null,
  'ShipCountry': 'Germany'
},
{
  'OrderID': 10692,
  'CustomerID': 'ALFKI',
  'OrderDate': '1997-10-03T00:00:00.000Z',
  'ShippedDate': '1997-10-13T00:00:00.000Z',
  'Freight': 61.02,
  'ShipName': 'Alfred\ Futterkiste',
  'ShipAddress': 'Obere Str. 57',
  'ShipCity': 'Berlin',
  'ShipRegion': null,
  'ShipCountry': 'Germany'
},
{
  'OrderID': 10693,
  'CustomerID': 'WHITC',
  'OrderDate': '1997-10-06T00:00:00.000Z',
  'ShippedDate': '1997-10-10T00:00:00.000Z',
  'Freight': 139.34,
  'ShipName': 'White Clover Markets',
  'ShipAddress': '1029 - 12th Ave. S.',
  'ShipCity': 'Seattle',

```

```

      'ShipRegion': 'WA',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10694,
      'CustomerID': 'QUICK',
      'OrderDate': '1997-10-06T00:00:00.000Z',
      'ShippedDate': '1997-10-09T00:00:00.000Z',
      'Freight': 398.36,
      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10695,
      'CustomerID': 'WILMK',
      'OrderDate': '1997-10-07T00:00:00.000Z',
      'ShippedDate': '1997-10-14T00:00:00.000Z',
      'Freight': 16.72,
      'ShipName': 'Wilman Kala',
      'ShipAddress': 'Keskuskatu 45',
      'ShipCity': 'Helsinki',
      'ShipRegion': null,
      'ShipCountry': 'Finland'
    },
    {
      'OrderID': 10696,
      'CustomerID': 'WHITC',
      'OrderDate': '1997-10-08T00:00:00.000Z',
      'ShippedDate': '1997-10-14T00:00:00.000Z',
      'Freight': 102.55,
      'ShipName': 'White Clover Markets',
      'ShipAddress': '1029 - 12th Ave. S.',
      'ShipCity': 'Seattle',
      'ShipRegion': 'WA',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10697,
      'CustomerID': 'LINOD',
      'OrderDate': '1997-10-08T00:00:00.000Z',
      'ShippedDate': '1997-10-14T00:00:00.000Z',
      'Freight': 45.52,
      'ShipName': 'LINO-Delicateses',
      'ShipAddress': 'Ave. 5 de Mayo Porlamar',
      'ShipCity': 'I. de Margarita',
      'ShipRegion': 'Nueva Esparta',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10698,
      'CustomerID': 'ERNSH',
      'OrderDate': '1997-10-09T00:00:00.000Z',
      'ShippedDate': '1997-10-17T00:00:00.000Z',
      'Freight': 272.47,

```

```

      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10699,
      'CustomerID': 'MORGK',
      'OrderDate': '1997-10-09T00:00:00.000Z',
      'ShippedDate': '1997-10-13T00:00:00.000Z',
      'Freight': 0.58,
      'ShipName': 'Morgenstern Gesundkost',
      'ShipAddress': 'Heerstr. 22',
      'ShipCity': 'Leipzig',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10700,
      'CustomerID': 'SAVEA',
      'OrderDate': '1997-10-10T00:00:00.000Z',
      'ShippedDate': '1997-10-16T00:00:00.000Z',
      'Freight': 65.1,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10701,
      'CustomerID': 'HUNGO',
      'OrderDate': '1997-10-13T00:00:00.000Z',
      'ShippedDate': '1997-10-15T00:00:00.000Z',
      'Freight': 220.31,
      'ShipName': 'Hungry Owl All-Night Grocers',
      'ShipAddress': '8 Johnstown Road',
      'ShipCity': 'Cork',
      'ShipRegion': 'Co. Cork',
      'ShipCountry': 'Ireland'
    },
    {
      'OrderID': 10702,
      'CustomerID': 'ALFKI',
      'OrderDate': '1997-10-13T00:00:00.000Z',
      'ShippedDate': '1997-10-21T00:00:00.000Z',
      'Freight': 23.94,
      'ShipName': 'Alfred Futterkiste',
      'ShipAddress': 'Obere Str. 57',
      'ShipCity': 'Berlin',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10703,
      'CustomerID': 'FOLKO',

```

```

      'OrderDate': '1997-10-14T00:00:00.000Z',
      'ShippedDate': '1997-10-20T00:00:00.000Z',
      'Freight': 152.3,
      'ShipName': 'Folk och fä HB',
      'ShipAddress': 'Åkergatan 24',
      'ShipCity': 'Bräcke',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 10704,
      'CustomerID': 'QUEEN',
      'OrderDate': '1997-10-14T00:00:00.000Z',
      'ShippedDate': '1997-11-07T00:00:00.000Z',
      'Freight': 4.78,
      'ShipName': 'Queen Cozinha',
      'ShipAddress': 'Alameda dos Canários, 891',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10705,
      'CustomerID': 'HILAA',
      'OrderDate': '1997-10-15T00:00:00.000Z',
      'ShippedDate': '1997-11-18T00:00:00.000Z',
      'Freight': 3.52,
      'ShipName': 'HILARION-Abastos',
      'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
      'ShipCity': 'San Cristóbal',
      'ShipRegion': 'Táchira',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10706,
      'CustomerID': 'OLDWO',
      'OrderDate': '1997-10-16T00:00:00.000Z',
      'ShippedDate': '1997-10-21T00:00:00.000Z',
      'Freight': 135.63,
      'ShipName': 'Old World Delicatessen',
      'ShipAddress': '2743 Bering St.',
      'ShipCity': 'Anchorage',
      'ShipRegion': 'AK',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10707,
      'CustomerID': 'AROUT',
      'OrderDate': '1997-10-16T00:00:00.000Z',
      'ShippedDate': '1997-10-23T00:00:00.000Z',
      'Freight': 21.74,
      'ShipName': 'Around the Horn',
      'ShipAddress': 'Brook Farm Stratford St. Mary',
      'ShipCity': 'Colchester',
      'ShipRegion': 'Essex',
      'ShipCountry': 'UK'
    }
  ],

```

```

{
  'OrderID': 10708,
  'CustomerID': 'THEBI',
  'OrderDate': '1997-10-17T00:00:00.000Z',
  'ShippedDate': '1997-11-05T00:00:00.000Z',
  'Freight': 2.96,
  'ShipName': 'The Big Cheese',
  'ShipAddress': '89 Jefferson Way Suite 2',
  'ShipCity': 'Portland',
  'ShipRegion': 'OR',
  'ShipCountry': 'USA'
},
{
  'OrderID': 10709,
  'CustomerID': 'GOURL',
  'OrderDate': '1997-10-17T00:00:00.000Z',
  'ShippedDate': '1997-11-20T00:00:00.000Z',
  'Freight': 210.8,
  'ShipName': 'Gourmet Lanchonetes',
  'ShipAddress': 'Av. Brasil, 442',
  'ShipCity': 'Campinas',
  'ShipRegion': 'SP',
  'ShipCountry': 'Brazil'
},
{
  'OrderID': 10710,
  'CustomerID': 'FRANS',
  'OrderDate': '1997-10-20T00:00:00.000Z',
  'ShippedDate': '1997-10-23T00:00:00.000Z',
  'Freight': 4.98,
  'ShipName': 'Franchi S.p.A.',
  'ShipAddress': 'Via Monte Bianco 34',
  'ShipCity': 'Torino',
  'ShipRegion': null,
  'ShipCountry': 'Italy'
},
{
  'OrderID': 10711,
  'CustomerID': 'SAVEA',
  'OrderDate': '1997-10-21T00:00:00.000Z',
  'ShippedDate': '1997-10-29T00:00:00.000Z',
  'Freight': 52.41,
  'ShipName': 'Save-a-lot Markets',
  'ShipAddress': '187 Suffolk Ln.',
  'ShipCity': 'Boise',
  'ShipRegion': 'ID',
  'ShipCountry': 'USA'
},
{
  'OrderID': 10712,
  'CustomerID': 'HUNGO',
  'OrderDate': '1997-10-21T00:00:00.000Z',
  'ShippedDate': '1997-10-31T00:00:00.000Z',
  'Freight': 89.93,
  'ShipName': 'Hungry Owl All-Night Grocers',
  'ShipAddress': '8 Johnstown Road',
  'ShipCity': 'Cork',

```

```

      'ShipRegion': 'Co. Cork',
      'ShipCountry': 'Ireland'
    },
    {
      'OrderID': 10713,
      'CustomerID': 'SAVEA',
      'OrderDate': '1997-10-22T00:00:00.000Z',
      'ShippedDate': '1997-10-24T00:00:00.000Z',
      'Freight': 167.05,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10714,
      'CustomerID': 'SAVEA',
      'OrderDate': '1997-10-22T00:00:00.000Z',
      'ShippedDate': '1997-10-27T00:00:00.000Z',
      'Freight': 24.49,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10715,
      'CustomerID': 'BONAP',
      'OrderDate': '1997-10-23T00:00:00.000Z',
      'ShippedDate': '1997-10-29T00:00:00.000Z',
      'Freight': 63.2,
      'ShipName': 'Bon app',
      'ShipAddress': '12, rue des Bouchers',
      'ShipCity': 'Marseille',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10716,
      'CustomerID': 'RANCH',
      'OrderDate': '1997-10-24T00:00:00.000Z',
      'ShippedDate': '1997-10-27T00:00:00.000Z',
      'Freight': 22.57,
      'ShipName': 'Rancho grande',
      'ShipAddress': 'Av. del Libertador 900',
      'ShipCity': 'Buenos Aires',
      'ShipRegion': null,
      'ShipCountry': 'Argentina'
    },
    {
      'OrderID': 10717,
      'CustomerID': 'FRANK',
      'OrderDate': '1997-10-24T00:00:00.000Z',
      'ShippedDate': '1997-10-29T00:00:00.000Z',
      'Freight': 59.25,

```

```

      'ShipName': 'Frankenversand',
      'ShipAddress': 'Berliner Platz 43',
      'ShipCity': 'München',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10718,
      'CustomerID': 'KOENE',
      'OrderDate': '1997-10-27T00:00:00.000Z',
      'ShippedDate': '1997-10-29T00:00:00.000Z',
      'Freight': 170.88,
      'ShipName': 'Königlich Essen',
      'ShipAddress': 'Maubelstr. 90',
      'ShipCity': 'Brandenburg',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10719,
      'CustomerID': 'LETSS',
      'OrderDate': '1997-10-27T00:00:00.000Z',
      'ShippedDate': '1997-11-05T00:00:00.000Z',
      'Freight': 51.44,
      'ShipName': 'Let\' Stop N Shop',
      'ShipAddress': '87 Polk St. Suite 5',
      'ShipCity': 'San Francisco',
      'ShipRegion': 'CA',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10720,
      'CustomerID': 'QUEDE',
      'OrderDate': '1997-10-28T00:00:00.000Z',
      'ShippedDate': '1997-11-05T00:00:00.000Z',
      'Freight': 9.53,
      'ShipName': 'Que Delícia',
      'ShipAddress': 'Rua da Panificadora, 12',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10721,
      'CustomerID': 'QUICK',
      'OrderDate': '1997-10-29T00:00:00.000Z',
      'ShippedDate': '1997-10-31T00:00:00.000Z',
      'Freight': 48.92,
      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10722,
      'CustomerID': 'SAVEA',

```



```

      'OrderDate': '1997-10-29T00:00:00.000Z',
      'ShippedDate': '1997-11-04T00:00:00.000Z',
      'Freight': 74.58,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10723,
      'CustomerID': 'WHITC',
      'OrderDate': '1997-10-30T00:00:00.000Z',
      'ShippedDate': '1997-11-25T00:00:00.000Z',
      'Freight': 21.72,
      'ShipName': 'White Clover Markets',
      'ShipAddress': '1029 - 12th Ave. S.',
      'ShipCity': 'Seattle',
      'ShipRegion': 'WA',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10724,
      'CustomerID': 'MEREP',
      'OrderDate': '1997-10-30T00:00:00.000Z',
      'ShippedDate': '1997-11-05T00:00:00.000Z',
      'Freight': 57.75,
      'ShipName': 'Mère Paillarde',
      'ShipAddress': '43 rue St. Laurent',
      'ShipCity': 'Montréal',
      'ShipRegion': 'Québec',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10725,
      'CustomerID': 'FAMIA',
      'OrderDate': '1997-10-31T00:00:00.000Z',
      'ShippedDate': '1997-11-05T00:00:00.000Z',
      'Freight': 10.83,
      'ShipName': 'Familia Arquibaldo',
      'ShipAddress': 'Rua Orós, 92',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10726,
      'CustomerID': 'EASTC',
      'OrderDate': '1997-11-03T00:00:00.000Z',
      'ShippedDate': '1997-12-05T00:00:00.000Z',
      'Freight': 16.56,
      'ShipName': 'Eastern Connection',
      'ShipAddress': '35 King George',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    }
  ],

```

```

{
  'OrderID': 10727,
  'CustomerID': 'REGGC',
  'OrderDate': '1997-11-03T00:00:00.000Z',
  'ShippedDate': '1997-12-05T00:00:00.000Z',
  'Freight': 89.9,
  'ShipName': 'Reggiani Caseifici',
  'ShipAddress': 'Strada Provinciale 124',
  'ShipCity': 'Reggio Emilia',
  'ShipRegion': null,
  'ShipCountry': 'Italy'
},
{
  'OrderID': 10728,
  'CustomerID': 'QUEEN',
  'OrderDate': '1997-11-04T00:00:00.000Z',
  'ShippedDate': '1997-11-11T00:00:00.000Z',
  'Freight': 58.33,
  'ShipName': 'Queen Cozinha',
  'ShipAddress': 'Alameda dos Canários, 891',
  'ShipCity': 'Sao Paulo',
  'ShipRegion': 'SP',
  'ShipCountry': 'Brazil'
},
{
  'OrderID': 10729,
  'CustomerID': 'LINOD',
  'OrderDate': '1997-11-04T00:00:00.000Z',
  'ShippedDate': '1997-11-14T00:00:00.000Z',
  'Freight': 141.06,
  'ShipName': 'LINO-Delicateses',
  'ShipAddress': 'Ave. 5 de Mayo Porlamar',
  'ShipCity': 'I. de Margarita',
  'ShipRegion': 'Nueva Esparta',
  'ShipCountry': 'Venezuela'
},
{
  'OrderID': 10730,
  'CustomerID': 'BONAP',
  'OrderDate': '1997-11-05T00:00:00.000Z',
  'ShippedDate': '1997-11-14T00:00:00.000Z',
  'Freight': 20.12,
  'ShipName': 'Bon app',
  'ShipAddress': '12, rue des Bouchers',
  'ShipCity': 'Marseille',
  'ShipRegion': null,
  'ShipCountry': 'France'
},
{
  'OrderID': 10731,
  'CustomerID': 'CHOPS',
  'OrderDate': '1997-11-06T00:00:00.000Z',
  'ShippedDate': '1997-11-14T00:00:00.000Z',
  'Freight': 96.65,
  'ShipName': 'Chop-suey Chinese',
  'ShipAddress': 'Hauptstr. 31',
  'ShipCity': 'Bern',

```

```

    'ShipRegion': null,
    'ShipCountry': 'Switzerland'
  },
  {
    'OrderID': 10732,
    'CustomerID': 'BONAP',
    'OrderDate': '1997-11-06T00:00:00.000Z',
    'ShippedDate': '1997-11-07T00:00:00.000Z',
    'Freight': 16.97,
    'ShipName': 'Bon app',
    'ShipAddress': '12, rue des Bouchers',
    'ShipCity': 'Marseille',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10733,
    'CustomerID': 'BERGS',
    'OrderDate': '1997-11-07T00:00:00.000Z',
    'ShippedDate': '1997-11-10T00:00:00.000Z',
    'Freight': 110.11,
    'ShipName': 'Berglunds snabbköp',
    'ShipAddress': 'Berguvsvägen 8',
    'ShipCity': 'Luleå',
    'ShipRegion': null,
    'ShipCountry': 'Sweden'
  },
  {
    'OrderID': 10734,
    'CustomerID': 'GOURL',
    'OrderDate': '1997-11-07T00:00:00.000Z',
    'ShippedDate': '1997-11-12T00:00:00.000Z',
    'Freight': 1.63,
    'ShipName': 'Gourmet Lanchonetes',
    'ShipAddress': 'Av. Brasil, 442',
    'ShipCity': 'Campinas',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10735,
    'CustomerID': 'LETSS',
    'OrderDate': '1997-11-10T00:00:00.000Z',
    'ShippedDate': '1997-11-21T00:00:00.000Z',
    'Freight': 45.97,
    'ShipName': 'Let\ Stop N Shop',
    'ShipAddress': '87 Polk St. Suite 5',
    'ShipCity': 'San Francisco',
    'ShipRegion': 'CA',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10736,
    'CustomerID': 'HUNGO',
    'OrderDate': '1997-11-11T00:00:00.000Z',
    'ShippedDate': '1997-11-21T00:00:00.000Z',
    'Freight': 44.1,

```

```

    'ShipName': 'Hungry Owl All-Night Grocers',
    'ShipAddress': '8 Johnstown Road',
    'ShipCity': 'Cork',
    'ShipRegion': 'Co. Cork',
    'ShipCountry': 'Ireland'
  },
  {
    'OrderID': 10737,
    'CustomerID': 'VINET',
    'OrderDate': '1997-11-11T00:00:00.000Z',
    'ShippedDate': '1997-11-18T00:00:00.000Z',
    'Freight': 7.79,
    'ShipName': 'Vins et alcools Chevalier',
    'ShipAddress': '59 rue de l\'Abbaye',
    'ShipCity': 'Reims',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10738,
    'CustomerID': 'SPECB',
    'OrderDate': '1997-11-12T00:00:00.000Z',
    'ShippedDate': '1997-11-18T00:00:00.000Z',
    'Freight': 2.91,
    'ShipName': 'Spécialités du monde',
    'ShipAddress': '25, rue Lauriston',
    'ShipCity': 'Paris',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10739,
    'CustomerID': 'VINET',
    'OrderDate': '1997-11-12T00:00:00.000Z',
    'ShippedDate': '1997-11-17T00:00:00.000Z',
    'Freight': 11.08,
    'ShipName': 'Vins et alcools Chevalier',
    'ShipAddress': '59 rue de l\'Abbaye',
    'ShipCity': 'Reims',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10740,
    'CustomerID': 'WHITC',
    'OrderDate': '1997-11-13T00:00:00.000Z',
    'ShippedDate': '1997-11-25T00:00:00.000Z',
    'Freight': 81.88,
    'ShipName': 'White Clover Markets',
    'ShipAddress': '1029 - 12th Ave. S.',
    'ShipCity': 'Seattle',
    'ShipRegion': 'WA',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10741,
    'CustomerID': 'AROUT',

```

```

      'OrderDate': '1997-11-14T00:00:00.000Z',
      'ShippedDate': '1997-11-18T00:00:00.000Z',
      'Freight': 10.96,
      'ShipName': 'Around the Horn',
      'ShipAddress': 'Brook Farm Stratford St. Mary',
      'ShipCity': 'Colchester',
      'ShipRegion': 'Essex',
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10742,
      'CustomerID': 'BOTTM',
      'OrderDate': '1997-11-14T00:00:00.000Z',
      'ShippedDate': '1997-11-18T00:00:00.000Z',
      'Freight': 243.73,
      'ShipName': 'Bottom-Dollar Markets',
      'ShipAddress': '23 Tsawassen Blvd.',
      'ShipCity': 'Tsawassen',
      'ShipRegion': 'BC',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10743,
      'CustomerID': 'AROUT',
      'OrderDate': '1997-11-17T00:00:00.000Z',
      'ShippedDate': '1997-11-21T00:00:00.000Z',
      'Freight': 23.72,
      'ShipName': 'Around the Horn',
      'ShipAddress': 'Brook Farm Stratford St. Mary',
      'ShipCity': 'Colchester',
      'ShipRegion': 'Essex',
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10744,
      'CustomerID': 'VAFFE',
      'OrderDate': '1997-11-17T00:00:00.000Z',
      'ShippedDate': '1997-11-24T00:00:00.000Z',
      'Freight': 69.19,
      'ShipName': 'Vaffeljernet',
      'ShipAddress': 'Smagsloget 45',
      'ShipCity': 'Århus',
      'ShipRegion': null,
      'ShipCountry': 'Denmark'
    },
    {
      'OrderID': 10745,
      'CustomerID': 'QUICK',
      'OrderDate': '1997-11-18T00:00:00.000Z',
      'ShippedDate': '1997-11-27T00:00:00.000Z',
      'Freight': 3.52,
      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    }
  ],

```

```

{
  'OrderID': 10746,
  'CustomerID': 'CHOPS',
  'OrderDate': '1997-11-19T00:00:00.000Z',
  'ShippedDate': '1997-11-21T00:00:00.000Z',
  'Freight': 31.43,
  'ShipName': 'Chop-suey Chinese',
  'ShipAddress': 'Hauptstr. 31',
  'ShipCity': 'Bern',
  'ShipRegion': null,
  'ShipCountry': 'Switzerland'
},
{
  'OrderID': 10747,
  'CustomerID': 'PICCO',
  'OrderDate': '1997-11-19T00:00:00.000Z',
  'ShippedDate': '1997-11-26T00:00:00.000Z',
  'Freight': 117.33,
  'ShipName': 'Piccolo und mehr',
  'ShipAddress': 'Geislweg 14',
  'ShipCity': 'Salzburg',
  'ShipRegion': null,
  'ShipCountry': 'Austria'
},
{
  'OrderID': 10748,
  'CustomerID': 'SAVEA',
  'OrderDate': '1997-11-20T00:00:00.000Z',
  'ShippedDate': '1997-11-28T00:00:00.000Z',
  'Freight': 232.55,
  'ShipName': 'Save-a-lot Markets',
  'ShipAddress': '187 Suffolk Ln.',
  'ShipCity': 'Boise',
  'ShipRegion': 'ID',
  'ShipCountry': 'USA'
},
{
  'OrderID': 10749,
  'CustomerID': 'ISLAT',
  'OrderDate': '1997-11-20T00:00:00.000Z',
  'ShippedDate': '1997-12-19T00:00:00.000Z',
  'Freight': 61.53,
  'ShipName': 'Island Trading',
  'ShipAddress': 'Garden House Crowther Way',
  'ShipCity': 'Cowes',
  'ShipRegion': 'Isle of Wight',
  'ShipCountry': 'UK'
},
{
  'OrderID': 10750,
  'CustomerID': 'WARTH',
  'OrderDate': '1997-11-21T00:00:00.000Z',
  'ShippedDate': '1997-11-24T00:00:00.000Z',
  'Freight': 79.3,
  'ShipName': 'Wartian Herkku',
  'ShipAddress': 'Torikatu 38',
  'ShipCity': 'Oulu',

```

```

    'ShipRegion': null,
    'ShipCountry': 'Finland'
  },
  {
    'OrderID': 10751,
    'CustomerID': 'RICSU',
    'OrderDate': '1997-11-24T00:00:00.000Z',
    'ShippedDate': '1997-12-03T00:00:00.000Z',
    'Freight': 130.79,
    'ShipName': 'Richter Supermarkt',
    'ShipAddress': 'Starenweg 5',
    'ShipCity': 'Genève',
    'ShipRegion': null,
    'ShipCountry': 'Switzerland'
  },
  {
    'OrderID': 10752,
    'CustomerID': 'NORTS',
    'OrderDate': '1997-11-24T00:00:00.000Z',
    'ShippedDate': '1997-11-28T00:00:00.000Z',
    'Freight': 1.39,
    'ShipName': 'North/South',
    'ShipAddress': 'South House 300 Queensbridge',
    'ShipCity': 'London',
    'ShipRegion': null,
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10753,
    'CustomerID': 'FRANS',
    'OrderDate': '1997-11-25T00:00:00.000Z',
    'ShippedDate': '1997-11-27T00:00:00.000Z',
    'Freight': 7.7,
    'ShipName': 'Franchi S.p.A.',
    'ShipAddress': 'Via Monte Bianco 34',
    'ShipCity': 'Torino',
    'ShipRegion': null,
    'ShipCountry': 'Italy'
  },
  {
    'OrderID': 10754,
    'CustomerID': 'MAGAA',
    'OrderDate': '1997-11-25T00:00:00.000Z',
    'ShippedDate': '1997-11-27T00:00:00.000Z',
    'Freight': 2.38,
    'ShipName': 'Magazzini Alimentari Riuniti',
    'ShipAddress': 'Via Ludovico il Moro 22',
    'ShipCity': 'Bergamo',
    'ShipRegion': null,
    'ShipCountry': 'Italy'
  },
  {
    'OrderID': 10755,
    'CustomerID': 'BONAP',
    'OrderDate': '1997-11-26T00:00:00.000Z',
    'ShippedDate': '1997-11-28T00:00:00.000Z',
    'Freight': 16.71,

```

```

      'ShipName': 'Bon app',
      'ShipAddress': '12, rue des Bouchers',
      'ShipCity': 'Marseille',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10756,
      'CustomerID': 'SPLIR',
      'OrderDate': '1997-11-27T00:00:00.000Z',
      'ShippedDate': '1997-12-02T00:00:00.000Z',
      'Freight': 73.21,
      'ShipName': 'Split Rail Beer & Ale',
      'ShipAddress': 'P.O. Box 555',
      'ShipCity': 'Lander',
      'ShipRegion': 'WY',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10757,
      'CustomerID': 'SAVEA',
      'OrderDate': '1997-11-27T00:00:00.000Z',
      'ShippedDate': '1997-12-15T00:00:00.000Z',
      'Freight': 8.19,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10758,
      'CustomerID': 'RICSU',
      'OrderDate': '1997-11-28T00:00:00.000Z',
      'ShippedDate': '1997-12-04T00:00:00.000Z',
      'Freight': 138.17,
      'ShipName': 'Richter Supermarkt',
      'ShipAddress': 'Starenweg 5',
      'ShipCity': 'Genève',
      'ShipRegion': null,
      'ShipCountry': 'Switzerland'
    },
    {
      'OrderID': 10759,
      'CustomerID': 'ANATR',
      'OrderDate': '1997-11-28T00:00:00.000Z',
      'ShippedDate': '1997-12-12T00:00:00.000Z',
      'Freight': 11.99,
      'ShipName': 'Ana Trujillo Emparedados y helados',
      'ShipAddress': 'Avda. de la Constitución 2222',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10760,
      'CustomerID': 'MAISD',

```



```

      'OrderDate': '1997-12-01T00:00:00.000Z',
      'ShippedDate': '1997-12-10T00:00:00.000Z',
      'Freight': 155.64,
      'ShipName': 'Maison Dewey',
      'ShipAddress': 'Rue Joseph-Bens 532',
      'ShipCity': 'Bruxelles',
      'ShipRegion': null,
      'ShipCountry': 'Belgium'
    },
    {
      'OrderID': 10761,
      'CustomerID': 'RATTC',
      'OrderDate': '1997-12-02T00:00:00.000Z',
      'ShippedDate': '1997-12-08T00:00:00.000Z',
      'Freight': 18.66,
      'ShipName': 'Rattlesnake Canyon Grocery',
      'ShipAddress': '2817 Milton Dr.',
      'ShipCity': 'Albuquerque',
      'ShipRegion': 'NM',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10762,
      'CustomerID': 'FOLKO',
      'OrderDate': '1997-12-02T00:00:00.000Z',
      'ShippedDate': '1997-12-09T00:00:00.000Z',
      'Freight': 328.74,
      'ShipName': 'Folk och fä HB',
      'ShipAddress': 'Åkergatan 24',
      'ShipCity': 'Bräcke',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 10763,
      'CustomerID': 'FOLIG',
      'OrderDate': '1997-12-03T00:00:00.000Z',
      'ShippedDate': '1997-12-08T00:00:00.000Z',
      'Freight': 37.35,
      'ShipName': 'Folies gourmandes',
      'ShipAddress': '184, chaussée de Tournai',
      'ShipCity': 'Lille',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10764,
      'CustomerID': 'ERNSH',
      'OrderDate': '1997-12-03T00:00:00.000Z',
      'ShippedDate': '1997-12-08T00:00:00.000Z',
      'Freight': 145.45,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    }
  ],

```

```

{
  'OrderID': 10765,
  'CustomerID': 'QUICK',
  'OrderDate': '1997-12-04T00:00:00.000Z',
  'ShippedDate': '1997-12-09T00:00:00.000Z',
  'Freight': 42.74,
  'ShipName': 'QUICK-Stop',
  'ShipAddress': 'Taucherstraße 10',
  'ShipCity': 'Cunewalde',
  'ShipRegion': null,
  'ShipCountry': 'Germany'
},
{
  'OrderID': 10766,
  'CustomerID': 'OTTIK',
  'OrderDate': '1997-12-05T00:00:00.000Z',
  'ShippedDate': '1997-12-09T00:00:00.000Z',
  'Freight': 157.55,
  'ShipName': 'Ottilies Käseladen',
  'ShipAddress': 'Mehrheimerstr. 369',
  'ShipCity': 'Köln',
  'ShipRegion': null,
  'ShipCountry': 'Germany'
},
{
  'OrderID': 10767,
  'CustomerID': 'SUPRD',
  'OrderDate': '1997-12-05T00:00:00.000Z',
  'ShippedDate': '1997-12-15T00:00:00.000Z',
  'Freight': 1.59,
  'ShipName': 'Suprêmes délices',
  'ShipAddress': 'Boulevard Tirou, 255',
  'ShipCity': 'Charleroi',
  'ShipRegion': null,
  'ShipCountry': 'Belgium'
},
{
  'OrderID': 10768,
  'CustomerID': 'AROUT',
  'OrderDate': '1997-12-08T00:00:00.000Z',
  'ShippedDate': '1997-12-15T00:00:00.000Z',
  'Freight': 146.32,
  'ShipName': 'Around the Horn',
  'ShipAddress': 'Brook Farm Stratford St. Mary',
  'ShipCity': 'Colchester',
  'ShipRegion': 'Essex',
  'ShipCountry': 'UK'
},
{
  'OrderID': 10769,
  'CustomerID': 'VAFFE',
  'OrderDate': '1997-12-08T00:00:00.000Z',
  'ShippedDate': '1997-12-12T00:00:00.000Z',
  'Freight': 65.06,
  'ShipName': 'Vaffeljernet',
  'ShipAddress': 'Smagsloget 45',
  'ShipCity': 'Århus',

```

```

    'ShipRegion': null,
    'ShipCountry': 'Denmark'
  },
  {
    'OrderID': 10770,
    'CustomerID': 'HANAR',
    'OrderDate': '1997-12-09T00:00:00.000Z',
    'ShippedDate': '1997-12-17T00:00:00.000Z',
    'Freight': 5.32,
    'ShipName': 'Hanari Carnes',
    'ShipAddress': 'Rua do Paço, 67',
    'ShipCity': 'Rio de Janeiro',
    'ShipRegion': 'RJ',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10771,
    'CustomerID': 'ERNSH',
    'OrderDate': '1997-12-10T00:00:00.000Z',
    'ShippedDate': '1998-01-02T00:00:00.000Z',
    'Freight': 11.19,
    'ShipName': 'Ernst Handel',
    'ShipAddress': 'Kirchgasse 6',
    'ShipCity': 'Graz',
    'ShipRegion': null,
    'ShipCountry': 'Austria'
  },
  {
    'OrderID': 10772,
    'CustomerID': 'LEHMS',
    'OrderDate': '1997-12-10T00:00:00.000Z',
    'ShippedDate': '1997-12-19T00:00:00.000Z',
    'Freight': 91.28,
    'ShipName': 'Lehmanns Marktstand',
    'ShipAddress': 'Magazinweg 7',
    'ShipCity': 'Frankfurt a.M.',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10773,
    'CustomerID': 'ERNSH',
    'OrderDate': '1997-12-11T00:00:00.000Z',
    'ShippedDate': '1997-12-16T00:00:00.000Z',
    'Freight': 96.43,
    'ShipName': 'Ernst Handel',
    'ShipAddress': 'Kirchgasse 6',
    'ShipCity': 'Graz',
    'ShipRegion': null,
    'ShipCountry': 'Austria'
  },
  {
    'OrderID': 10774,
    'CustomerID': 'FOLKO',
    'OrderDate': '1997-12-11T00:00:00.000Z',
    'ShippedDate': '1997-12-12T00:00:00.000Z',
    'Freight': 48.2,

```

```

      'ShipName': 'Folk och fä HB',
      'ShipAddress': 'Åkergatan 24',
      'ShipCity': 'Bräcke',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 10775,
      'CustomerID': 'THECR',
      'OrderDate': '1997-12-12T00:00:00.000Z',
      'ShippedDate': '1997-12-26T00:00:00.000Z',
      'Freight': 20.25,
      'ShipName': 'The Cracker Box',
      'ShipAddress': '55 Grizzly Peak Rd.',
      'ShipCity': 'Butte',
      'ShipRegion': 'MT',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10776,
      'CustomerID': 'ERNSH',
      'OrderDate': '1997-12-15T00:00:00.000Z',
      'ShippedDate': '1997-12-18T00:00:00.000Z',
      'Freight': 351.53,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10777,
      'CustomerID': 'GOURL',
      'OrderDate': '1997-12-15T00:00:00.000Z',
      'ShippedDate': '1998-01-21T00:00:00.000Z',
      'Freight': 3.01,
      'ShipName': 'Gourmet Lanchonetes',
      'ShipAddress': 'Av. Brasil, 442',
      'ShipCity': 'Campinas',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10778,
      'CustomerID': 'BERGS',
      'OrderDate': '1997-12-16T00:00:00.000Z',
      'ShippedDate': '1997-12-24T00:00:00.000Z',
      'Freight': 6.79,
      'ShipName': 'Berglunds snabbköp',
      'ShipAddress': 'Berguvsvägen 8',
      'ShipCity': 'Luleå',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 10779,
      'CustomerID': 'MORGK',

```

```

      'OrderDate': '1997-12-16T00:00:00.000Z',
      'ShippedDate': '1998-01-14T00:00:00.000Z',
      'Freight': 58.13,
      'ShipName': 'Morgenstern Gesundkost',
      'ShipAddress': 'Heerstr. 22',
      'ShipCity': 'Leipzig',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10780,
      'CustomerID': 'LILAS',
      'OrderDate': '1997-12-16T00:00:00.000Z',
      'ShippedDate': '1997-12-25T00:00:00.000Z',
      'Freight': 42.13,
      'ShipName': 'LILA-Supermercado',
      'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
      'ShipCity': 'Barquisimeto',
      'ShipRegion': 'Lara',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10781,
      'CustomerID': 'WARTH',
      'OrderDate': '1997-12-17T00:00:00.000Z',
      'ShippedDate': '1997-12-19T00:00:00.000Z',
      'Freight': 73.16,
      'ShipName': 'Wartian Herkku',
      'ShipAddress': 'Torikatu 38',
      'ShipCity': 'Oulu',
      'ShipRegion': null,
      'ShipCountry': 'Finland'
    },
    {
      'OrderID': 10782,
      'CustomerID': 'CACTU',
      'OrderDate': '1997-12-17T00:00:00.000Z',
      'ShippedDate': '1997-12-22T00:00:00.000Z',
      'Freight': 1.1,
      'ShipName': 'Cactus Comidas para llevar',
      'ShipAddress': 'Cerrito 333',
      'ShipCity': 'Buenos Aires',
      'ShipRegion': null,
      'ShipCountry': 'Argentina'
    },
    {
      'OrderID': 10783,
      'CustomerID': 'HANAR',
      'OrderDate': '1997-12-18T00:00:00.000Z',
      'ShippedDate': '1997-12-19T00:00:00.000Z',
      'Freight': 124.98,
      'ShipName': 'Hanari Carnes',
      'ShipAddress': 'Rua do Paço, 67',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    }
  ],

```

```

{
  'OrderID': 10784,
  'CustomerID': 'MAGAA',
  'OrderDate': '1997-12-18T00:00:00.000Z',
  'ShippedDate': '1997-12-22T00:00:00.000Z',
  'Freight': 70.09,
  'ShipName': 'Magazzini Alimentari Riuniti',
  'ShipAddress': 'Via Ludovico il Moro 22',
  'ShipCity': 'Bergamo',
  'ShipRegion': null,
  'ShipCountry': 'Italy'
},
{
  'OrderID': 10785,
  'CustomerID': 'GROSR',
  'OrderDate': '1997-12-18T00:00:00.000Z',
  'ShippedDate': '1997-12-24T00:00:00.000Z',
  'Freight': 1.51,
  'ShipName': 'GROSELLA-Restaurante',
  'ShipAddress': '5a Ave. Los Palos Grandes',
  'ShipCity': 'Caracas',
  'ShipRegion': 'DF',
  'ShipCountry': 'Venezuela'
},
{
  'OrderID': 10786,
  'CustomerID': 'QUEEN',
  'OrderDate': '1997-12-19T00:00:00.000Z',
  'ShippedDate': '1997-12-23T00:00:00.000Z',
  'Freight': 110.87,
  'ShipName': 'Queen Cozinha',
  'ShipAddress': 'Alameda dos Canários, 891',
  'ShipCity': 'Sao Paulo',
  'ShipRegion': 'SP',
  'ShipCountry': 'Brazil'
},
{
  'OrderID': 10787,
  'CustomerID': 'LAMAI',
  'OrderDate': '1997-12-19T00:00:00.000Z',
  'ShippedDate': '1997-12-26T00:00:00.000Z',
  'Freight': 249.93,
  'ShipName': 'La maison d\'Asie',
  'ShipAddress': '1 rue Alsace-Lorraine',
  'ShipCity': 'Toulouse',
  'ShipRegion': null,
  'ShipCountry': 'France'
},
{
  'OrderID': 10788,
  'CustomerID': 'QUICK',
  'OrderDate': '1997-12-22T00:00:00.000Z',
  'ShippedDate': '1998-01-19T00:00:00.000Z',
  'Freight': 42.7,
  'ShipName': 'QUICK-Stop',
  'ShipAddress': 'Taucherstraße 10',
  'ShipCity': 'Cunewalde',

```

```

      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10789,
      'CustomerID': 'FOLIG',
      'OrderDate': '1997-12-22T00:00:00.000Z',
      'ShippedDate': '1997-12-31T00:00:00.000Z',
      'Freight': 100.6,
      'ShipName': 'Folies gourmandes',
      'ShipAddress': '184, chaussée de Tournai',
      'ShipCity': 'Lille',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10790,
      'CustomerID': 'GOURL',
      'OrderDate': '1997-12-22T00:00:00.000Z',
      'ShippedDate': '1997-12-26T00:00:00.000Z',
      'Freight': 28.23,
      'ShipName': 'Gourmet Lanchonetes',
      'ShipAddress': 'Av. Brasil, 442',
      'ShipCity': 'Campinas',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10791,
      'CustomerID': 'FRANK',
      'OrderDate': '1997-12-23T00:00:00.000Z',
      'ShippedDate': '1998-01-01T00:00:00.000Z',
      'Freight': 16.85,
      'ShipName': 'Frankenversand',
      'ShipAddress': 'Berliner Platz 43',
      'ShipCity': 'München',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10792,
      'CustomerID': 'WOLZA',
      'OrderDate': '1997-12-23T00:00:00.000Z',
      'ShippedDate': '1997-12-31T00:00:00.000Z',
      'Freight': 23.79,
      'ShipName': 'Wolski Zajazd',
      'ShipAddress': 'ul. Filtrowa 68',
      'ShipCity': 'Warszawa',
      'ShipRegion': null,
      'ShipCountry': 'Poland'
    },
    {
      'OrderID': 10793,
      'CustomerID': 'AROUT',
      'OrderDate': '1997-12-24T00:00:00.000Z',
      'ShippedDate': '1998-01-08T00:00:00.000Z',
      'Freight': 4.52,

```

```

    'ShipName': 'Around the Horn',
    'ShipAddress': 'Brook Farm Stratford St. Mary',
    'ShipCity': 'Colchester',
    'ShipRegion': 'Essex',
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10794,
    'CustomerID': 'QUEDE',
    'OrderDate': '1997-12-24T00:00:00.000Z',
    'ShippedDate': '1998-01-02T00:00:00.000Z',
    'Freight': 21.49,
    'ShipName': 'Que Delícia',
    'ShipAddress': 'Rua da Panificadora, 12',
    'ShipCity': 'Rio de Janeiro',
    'ShipRegion': 'RJ',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10795,
    'CustomerID': 'ERNSH',
    'OrderDate': '1997-12-24T00:00:00.000Z',
    'ShippedDate': '1998-01-20T00:00:00.000Z',
    'Freight': 126.66,
    'ShipName': 'Ernst Handel',
    'ShipAddress': 'Kirchgasse 6',
    'ShipCity': 'Graz',
    'ShipRegion': null,
    'ShipCountry': 'Austria'
  },
  {
    'OrderID': 10796,
    'CustomerID': 'HILAA',
    'OrderDate': '1997-12-25T00:00:00.000Z',
    'ShippedDate': '1998-01-14T00:00:00.000Z',
    'Freight': 26.52,
    'ShipName': 'HILARION-Abastos',
    'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
    'ShipCity': 'San Cristóbal',
    'ShipRegion': 'Táchira',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10797,
    'CustomerID': 'DRACD',
    'OrderDate': '1997-12-25T00:00:00.000Z',
    'ShippedDate': '1998-01-05T00:00:00.000Z',
    'Freight': 33.35,
    'ShipName': 'Drachenblut Delikatessen',
    'ShipAddress': 'Walserweg 21',
    'ShipCity': 'Aachen',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10798,
    'CustomerID': 'ISLAT',

```



```

      'OrderDate': '1997-12-26T00:00:00.000Z',
      'ShippedDate': '1998-01-05T00:00:00.000Z',
      'Freight': 2.33,
      'ShipName': 'Island Trading',
      'ShipAddress': 'Garden House Crowther Way',
      'ShipCity': 'Cowes',
      'ShipRegion': 'Isle of Wight',
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10799,
      'CustomerID': 'KOENE',
      'OrderDate': '1997-12-26T00:00:00.000Z',
      'ShippedDate': '1998-01-05T00:00:00.000Z',
      'Freight': 30.76,
      'ShipName': 'Königlich Essen',
      'ShipAddress': 'Maubelstr. 90',
      'ShipCity': 'Brandenburg',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10800,
      'CustomerID': 'SEVES',
      'OrderDate': '1997-12-26T00:00:00.000Z',
      'ShippedDate': '1998-01-05T00:00:00.000Z',
      'Freight': 137.44,
      'ShipName': 'Seven Seas Imports',
      'ShipAddress': '90 Wadhurst Rd.',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10801,
      'CustomerID': 'BOLID',
      'OrderDate': '1997-12-29T00:00:00.000Z',
      'ShippedDate': '1997-12-31T00:00:00.000Z',
      'Freight': 97.09,
      'ShipName': 'Bólido Comidas preparadas',
      'ShipAddress': 'C/ Araquil, 67',
      'ShipCity': 'Madrid',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    },
    {
      'OrderID': 10802,
      'CustomerID': 'SIMOB',
      'OrderDate': '1997-12-29T00:00:00.000Z',
      'ShippedDate': '1998-01-02T00:00:00.000Z',
      'Freight': 257.26,
      'ShipName': 'Simons bistro',
      'ShipAddress': 'Vinbæltet 34',
      'ShipCity': 'Kobenhavn',
      'ShipRegion': null,
      'ShipCountry': 'Denmark'
    }
  ],

```

```

{
  'OrderID': 10803,
  'CustomerID': 'WELLI',
  'OrderDate': '1997-12-30T00:00:00.000Z',
  'ShippedDate': '1998-01-06T00:00:00.000Z',
  'Freight': 55.23,
  'ShipName': 'Wellington Importadora',
  'ShipAddress': 'Rua do Mercado, 12',
  'ShipCity': 'Resende',
  'ShipRegion': 'SP',
  'ShipCountry': 'Brazil'
},
{
  'OrderID': 10804,
  'CustomerID': 'SEVES',
  'OrderDate': '1997-12-30T00:00:00.000Z',
  'ShippedDate': '1998-01-07T00:00:00.000Z',
  'Freight': 27.33,
  'ShipName': 'Seven Seas Imports',
  'ShipAddress': '90 Wadhurst Rd.',
  'ShipCity': 'London',
  'ShipRegion': null,
  'ShipCountry': 'UK'
},
{
  'OrderID': 10805,
  'CustomerID': 'THEBI',
  'OrderDate': '1997-12-30T00:00:00.000Z',
  'ShippedDate': '1998-01-09T00:00:00.000Z',
  'Freight': 237.34,
  'ShipName': 'The Big Cheese',
  'ShipAddress': '89 Jefferson Way Suite 2',
  'ShipCity': 'Portland',
  'ShipRegion': 'OR',
  'ShipCountry': 'USA'
},
{
  'OrderID': 10806,
  'CustomerID': 'VICTE',
  'OrderDate': '1997-12-31T00:00:00.000Z',
  'ShippedDate': '1998-01-05T00:00:00.000Z',
  'Freight': 22.11,
  'ShipName': 'Victuailles en stock',
  'ShipAddress': '2, rue du Commerce',
  'ShipCity': 'Lyon',
  'ShipRegion': null,
  'ShipCountry': 'France'
},
{
  'OrderID': 10807,
  'CustomerID': 'FRANS',
  'OrderDate': '1997-12-31T00:00:00.000Z',
  'ShippedDate': '1998-01-30T00:00:00.000Z',
  'Freight': 1.36,
  'ShipName': 'Franchi S.p.A.',
  'ShipAddress': 'Via Monte Bianco 34',
  'ShipCity': 'Torino',

```

```

    'ShipRegion': null,
    'ShipCountry': 'Italy'
  },
  {
    'OrderID': 10808,
    'CustomerID': 'OLDWO',
    'OrderDate': '1998-01-01T00:00:00.000Z',
    'ShippedDate': '1998-01-09T00:00:00.000Z',
    'Freight': 45.53,
    'ShipName': 'Old World Delicatessen',
    'ShipAddress': '2743 Bering St.',
    'ShipCity': 'Anchorage',
    'ShipRegion': 'AK',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10809,
    'CustomerID': 'WELLI',
    'OrderDate': '1998-01-01T00:00:00.000Z',
    'ShippedDate': '1998-01-07T00:00:00.000Z',
    'Freight': 4.87,
    'ShipName': 'Wellington Importadora',
    'ShipAddress': 'Rua do Mercado, 12',
    'ShipCity': 'Resende',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10810,
    'CustomerID': 'LAUGB',
    'OrderDate': '1998-01-01T00:00:00.000Z',
    'ShippedDate': '1998-01-07T00:00:00.000Z',
    'Freight': 4.33,
    'ShipName': 'Laughing Bacchus Wine Cellars',
    'ShipAddress': '2319 Elm St.',
    'ShipCity': 'Vancouver',
    'ShipRegion': 'BC',
    'ShipCountry': 'Canada'
  },
  {
    'OrderID': 10811,
    'CustomerID': 'LINOD',
    'OrderDate': '1998-01-02T00:00:00.000Z',
    'ShippedDate': '1998-01-08T00:00:00.000Z',
    'Freight': 31.22,
    'ShipName': 'LINO-Delicateses',
    'ShipAddress': 'Ave. 5 de Mayo Porlamar',
    'ShipCity': 'I. de Margarita',
    'ShipRegion': 'Nueva Esparta',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 10812,
    'CustomerID': 'REGGC',
    'OrderDate': '1998-01-02T00:00:00.000Z',
    'ShippedDate': '1998-01-12T00:00:00.000Z',
    'Freight': 59.78,

```

```

    'ShipName': 'Reggiani Caseifici',
    'ShipAddress': 'Strada Provinciale 124',
    'ShipCity': 'Reggio Emilia',
    'ShipRegion': null,
    'ShipCountry': 'Italy'
  },
  {
    'OrderID': 10813,
    'CustomerID': 'RICAR',
    'OrderDate': '1998-01-05T00:00:00.000Z',
    'ShippedDate': '1998-01-09T00:00:00.000Z',
    'Freight': 47.38,
    'ShipName': 'Ricardo Adocicados',
    'ShipAddress': 'Av. Copacabana, 267',
    'ShipCity': 'Rio de Janeiro',
    'ShipRegion': 'RJ',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10814,
    'CustomerID': 'VICTE',
    'OrderDate': '1998-01-05T00:00:00.000Z',
    'ShippedDate': '1998-01-14T00:00:00.000Z',
    'Freight': 130.94,
    'ShipName': 'Victuailles en stock',
    'ShipAddress': '2, rue du Commerce',
    'ShipCity': 'Lyon',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10815,
    'CustomerID': 'SAVEA',
    'OrderDate': '1998-01-05T00:00:00.000Z',
    'ShippedDate': '1998-01-14T00:00:00.000Z',
    'Freight': 14.62,
    'ShipName': 'Save-a-lot Markets',
    'ShipAddress': '187 Suffolk Ln.',
    'ShipCity': 'Boise',
    'ShipRegion': 'ID',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10816,
    'CustomerID': 'GREAL',
    'OrderDate': '1998-01-06T00:00:00.000Z',
    'ShippedDate': '1998-02-04T00:00:00.000Z',
    'Freight': 719.78,
    'ShipName': 'Great Lakes Food Market',
    'ShipAddress': '2732 Baker Blvd.',
    'ShipCity': 'Eugene',
    'ShipRegion': 'OR',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10817,
    'CustomerID': 'KOENE',

```

```

      'OrderDate': '1998-01-06T00:00:00.000Z',
      'ShippedDate': '1998-01-13T00:00:00.000Z',
      'Freight': 306.07,
      'ShipName': 'Königlich Essen',
      'ShipAddress': 'Maubelstr. 90',
      'ShipCity': 'Brandenburg',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10818,
      'CustomerID': 'MAGAA',
      'OrderDate': '1998-01-07T00:00:00.000Z',
      'ShippedDate': '1998-01-12T00:00:00.000Z',
      'Freight': 65.48,
      'ShipName': 'Magazzini Alimentari Riuniti',
      'ShipAddress': 'Via Ludovico il Moro 22',
      'ShipCity': 'Bergamo',
      'ShipRegion': null,
      'ShipCountry': 'Italy'
    },
    {
      'OrderID': 10819,
      'CustomerID': 'CACTU',
      'OrderDate': '1998-01-07T00:00:00.000Z',
      'ShippedDate': '1998-01-16T00:00:00.000Z',
      'Freight': 19.76,
      'ShipName': 'Cactus Comidas para llevar',
      'ShipAddress': 'Cerrito 333',
      'ShipCity': 'Buenos Aires',
      'ShipRegion': null,
      'ShipCountry': 'Argentina'
    },
    {
      'OrderID': 10820,
      'CustomerID': 'RATTC',
      'OrderDate': '1998-01-07T00:00:00.000Z',
      'ShippedDate': '1998-01-13T00:00:00.000Z',
      'Freight': 37.52,
      'ShipName': 'Rattlesnake Canyon Grocery',
      'ShipAddress': '2817 Milton Dr.',
      'ShipCity': 'Albuquerque',
      'ShipRegion': 'NM',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10821,
      'CustomerID': 'SPLIR',
      'OrderDate': '1998-01-08T00:00:00.000Z',
      'ShippedDate': '1998-01-15T00:00:00.000Z',
      'Freight': 36.68,
      'ShipName': 'Split Rail Beer & Ale',
      'ShipAddress': 'P.O. Box 555',
      'ShipCity': 'Lander',
      'ShipRegion': 'WY',
      'ShipCountry': 'USA'
    }
  ],

```

```

{
  'OrderID': 10822,
  'CustomerID': 'TRAIH',
  'OrderDate': '1998-01-08T00:00:00.000Z',
  'ShippedDate': '1998-01-16T00:00:00.000Z',
  'Freight': 7,
  'ShipName': 'Trail\' Head Gourmet Provisioners',
  'ShipAddress': '722 DaVinci Blvd.',
  'ShipCity': 'Kirkland',
  'ShipRegion': 'WA',
  'ShipCountry': 'USA'
},
{
  'OrderID': 10823,
  'CustomerID': 'LILAS',
  'OrderDate': '1998-01-09T00:00:00.000Z',
  'ShippedDate': '1998-01-13T00:00:00.000Z',
  'Freight': 163.97,
  'ShipName': 'LILA-Supermercado',
  'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
  'ShipCity': 'Barquisimeto',
  'ShipRegion': 'Lara',
  'ShipCountry': 'Venezuela'
},
{
  'OrderID': 10824,
  'CustomerID': 'FOLKO',
  'OrderDate': '1998-01-09T00:00:00.000Z',
  'ShippedDate': '1998-01-30T00:00:00.000Z',
  'Freight': 1.23,
  'ShipName': 'Folk och fä HB',
  'ShipAddress': 'Åkergratan 24',
  'ShipCity': 'Bräcke',
  'ShipRegion': null,
  'ShipCountry': 'Sweden'
},
{
  'OrderID': 10825,
  'CustomerID': 'DRACD',
  'OrderDate': '1998-01-09T00:00:00.000Z',
  'ShippedDate': '1998-01-14T00:00:00.000Z',
  'Freight': 79.25,
  'ShipName': 'Drachenblut Delikatessen',
  'ShipAddress': 'Walserweg 21',
  'ShipCity': 'Aachen',
  'ShipRegion': null,
  'ShipCountry': 'Germany'
},
{
  'OrderID': 10826,
  'CustomerID': 'BLONP',
  'OrderDate': '1998-01-12T00:00:00.000Z',
  'ShippedDate': '1998-02-06T00:00:00.000Z',
  'Freight': 7.09,
  'ShipName': 'Blondel père et fils',
  'ShipAddress': '24, place Kléber',
  'ShipCity': 'Strasbourg',

```

```

    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10827,
    'CustomerID': 'BONAP',
    'OrderDate': '1998-01-12T00:00:00.000Z',
    'ShippedDate': '1998-02-06T00:00:00.000Z',
    'Freight': 63.54,
    'ShipName': 'Bon app',
    'ShipAddress': '12, rue des Bouchers',
    'ShipCity': 'Marseille',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10828,
    'CustomerID': 'RANCH',
    'OrderDate': '1998-01-13T00:00:00.000Z',
    'ShippedDate': '1998-02-04T00:00:00.000Z',
    'Freight': 90.85,
    'ShipName': 'Rancho grande',
    'ShipAddress': 'Av. del Libertador 900',
    'ShipCity': 'Buenos Aires',
    'ShipRegion': null,
    'ShipCountry': 'Argentina'
  },
  {
    'OrderID': 10829,
    'CustomerID': 'ISLAT',
    'OrderDate': '1998-01-13T00:00:00.000Z',
    'ShippedDate': '1998-01-23T00:00:00.000Z',
    'Freight': 154.72,
    'ShipName': 'Island Trading',
    'ShipAddress': 'Garden House Crowther Way',
    'ShipCity': 'Cowes',
    'ShipRegion': 'Isle of Wight',
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10830,
    'CustomerID': 'TRADH',
    'OrderDate': '1998-01-13T00:00:00.000Z',
    'ShippedDate': '1998-01-21T00:00:00.000Z',
    'Freight': 81.83,
    'ShipName': 'Tradiçao Hipermercados',
    'ShipAddress': 'Av. Inês de Castro, 414',
    'ShipCity': 'Sao Paulo',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10831,
    'CustomerID': 'SANTG',
    'OrderDate': '1998-01-14T00:00:00.000Z',
    'ShippedDate': '1998-01-23T00:00:00.000Z',
    'Freight': 72.19,

```

```

      'ShipName': 'Santé Gourmet',
      'ShipAddress': 'Erling Skakkess gate 78',
      'ShipCity': 'Stavern',
      'ShipRegion': null,
      'ShipCountry': 'Norway'
    },
    {
      'OrderID': 10832,
      'CustomerID': 'LAMAI',
      'OrderDate': '1998-01-14T00:00:00.000Z',
      'ShippedDate': '1998-01-19T00:00:00.000Z',
      'Freight': 43.26,
      'ShipName': 'La maison d\'Asie',
      'ShipAddress': '1 rue Alsace-Lorraine',
      'ShipCity': 'Toulouse',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10833,
      'CustomerID': 'OTTIK',
      'OrderDate': '1998-01-15T00:00:00.000Z',
      'ShippedDate': '1998-01-23T00:00:00.000Z',
      'Freight': 71.49,
      'ShipName': 'Ottilies Käseladen',
      'ShipAddress': 'Mehrheimerstr. 369',
      'ShipCity': 'Köln',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10834,
      'CustomerID': 'TRADH',
      'OrderDate': '1998-01-15T00:00:00.000Z',
      'ShippedDate': '1998-01-19T00:00:00.000Z',
      'Freight': 29.78,
      'ShipName': 'Tradição Hipermercados',
      'ShipAddress': 'Av. Inês de Castro, 414',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10835,
      'CustomerID': 'ALFKI',
      'OrderDate': '1998-01-15T00:00:00.000Z',
      'ShippedDate': '1998-01-21T00:00:00.000Z',
      'Freight': 69.53,
      'ShipName': 'Alfred\' Futterkiste',
      'ShipAddress': 'Obere Str. 57',
      'ShipCity': 'Berlin',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10836,
      'CustomerID': 'ERNSH',

```



```

      'OrderDate': '1998-01-16T00:00:00.000Z',
      'ShippedDate': '1998-01-21T00:00:00.000Z',
      'Freight': 411.88,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10837,
      'CustomerID': 'BERGS',
      'OrderDate': '1998-01-16T00:00:00.000Z',
      'ShippedDate': '1998-01-23T00:00:00.000Z',
      'Freight': 13.32,
      'ShipName': 'Berglunds snabbköp',
      'ShipAddress': 'Berguvsvägen 8',
      'ShipCity': 'Luleå',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 10838,
      'CustomerID': 'LINOD',
      'OrderDate': '1998-01-19T00:00:00.000Z',
      'ShippedDate': '1998-01-23T00:00:00.000Z',
      'Freight': 59.28,
      'ShipName': 'LINO-Delicateses',
      'ShipAddress': 'Ave. 5 de Mayo Porlamar',
      'ShipCity': 'I. de Margarita',
      'ShipRegion': 'Nueva Esparta',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10839,
      'CustomerID': 'TRADH',
      'OrderDate': '1998-01-19T00:00:00.000Z',
      'ShippedDate': '1998-01-22T00:00:00.000Z',
      'Freight': 35.43,
      'ShipName': 'Tradiçao Hipermercados',
      'ShipAddress': 'Av. Inês de Castro, 414',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10840,
      'CustomerID': 'LINOD',
      'OrderDate': '1998-01-19T00:00:00.000Z',
      'ShippedDate': '1998-02-16T00:00:00.000Z',
      'Freight': 2.71,
      'ShipName': 'LINO-Delicateses',
      'ShipAddress': 'Ave. 5 de Mayo Porlamar',
      'ShipCity': 'I. de Margarita',
      'ShipRegion': 'Nueva Esparta',
      'ShipCountry': 'Venezuela'
    }
  ],

```

```

{
  'OrderID': 10841,
  'CustomerID': 'SUPRD',
  'OrderDate': '1998-01-20T00:00:00.000Z',
  'ShippedDate': '1998-01-29T00:00:00.000Z',
  'Freight': 424.3,
  'ShipName': 'Suprêmes délices',
  'ShipAddress': 'Boulevard Tirou, 255',
  'ShipCity': 'Charleroi',
  'ShipRegion': null,
  'ShipCountry': 'Belgium'
},
{
  'OrderID': 10842,
  'CustomerID': 'TORTU',
  'OrderDate': '1998-01-20T00:00:00.000Z',
  'ShippedDate': '1998-01-29T00:00:00.000Z',
  'Freight': 54.42,
  'ShipName': 'Tortuga Restaurante',
  'ShipAddress': 'Avda. Azteca 123',
  'ShipCity': 'México D.F.',
  'ShipRegion': null,
  'ShipCountry': 'Mexico'
},
{
  'OrderID': 10843,
  'CustomerID': 'VICTE',
  'OrderDate': '1998-01-21T00:00:00.000Z',
  'ShippedDate': '1998-01-26T00:00:00.000Z',
  'Freight': 9.26,
  'ShipName': 'Victuailles en stock',
  'ShipAddress': '2, rue du Commerce',
  'ShipCity': 'Lyon',
  'ShipRegion': null,
  'ShipCountry': 'France'
},
{
  'OrderID': 10844,
  'CustomerID': 'PICCO',
  'OrderDate': '1998-01-21T00:00:00.000Z',
  'ShippedDate': '1998-01-26T00:00:00.000Z',
  'Freight': 25.22,
  'ShipName': 'Piccolo und mehr',
  'ShipAddress': 'Geislweg 14',
  'ShipCity': 'Salzburg',
  'ShipRegion': null,
  'ShipCountry': 'Austria'
},
{
  'OrderID': 10845,
  'CustomerID': 'QUICK',
  'OrderDate': '1998-01-21T00:00:00.000Z',
  'ShippedDate': '1998-01-30T00:00:00.000Z',
  'Freight': 212.98,
  'ShipName': 'QUICK-Stop',
  'ShipAddress': 'Taucherstraße 10',
  'ShipCity': 'Cunewalde',

```

```

      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10846,
      'CustomerID': 'SUPRD',
      'OrderDate': '1998-01-22T00:00:00.000Z',
      'ShippedDate': '1998-01-23T00:00:00.000Z',
      'Freight': 56.46,
      'ShipName': 'Suprêmes délices',
      'ShipAddress': 'Boulevard Tirou, 255',
      'ShipCity': 'Charleroi',
      'ShipRegion': null,
      'ShipCountry': 'Belgium'
    },
    {
      'OrderID': 10847,
      'CustomerID': 'SAVEA',
      'OrderDate': '1998-01-22T00:00:00.000Z',
      'ShippedDate': '1998-02-10T00:00:00.000Z',
      'Freight': 487.57,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10848,
      'CustomerID': 'CONSH',
      'OrderDate': '1998-01-23T00:00:00.000Z',
      'ShippedDate': '1998-01-29T00:00:00.000Z',
      'Freight': 38.24,
      'ShipName': 'Consolidated Holdings',
      'ShipAddress': 'Berkeley Gardens 12 Brewery',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10849,
      'CustomerID': 'KOENE',
      'OrderDate': '1998-01-23T00:00:00.000Z',
      'ShippedDate': '1998-01-30T00:00:00.000Z',
      'Freight': 0.56,
      'ShipName': 'Königlich Essen',
      'ShipAddress': 'Maubelstr. 90',
      'ShipCity': 'Brandenburg',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10850,
      'CustomerID': 'VICTE',
      'OrderDate': '1998-01-23T00:00:00.000Z',
      'ShippedDate': '1998-01-30T00:00:00.000Z',
      'Freight': 49.19,

```

```

      'ShipName': 'Victuailles en stock',
      'ShipAddress': '2, rue du Commerce',
      'ShipCity': 'Lyon',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10851,
      'CustomerID': 'RICAR',
      'OrderDate': '1998-01-26T00:00:00.000Z',
      'ShippedDate': '1998-02-02T00:00:00.000Z',
      'Freight': 160.55,
      'ShipName': 'Ricardo Adocicados',
      'ShipAddress': 'Av. Copacabana, 267',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10852,
      'CustomerID': 'RATTC',
      'OrderDate': '1998-01-26T00:00:00.000Z',
      'ShippedDate': '1998-01-30T00:00:00.000Z',
      'Freight': 174.05,
      'ShipName': 'Rattlesnake Canyon Grocery',
      'ShipAddress': '2817 Milton Dr.',
      'ShipCity': 'Albuquerque',
      'ShipRegion': 'NM',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10853,
      'CustomerID': 'BLAUS',
      'OrderDate': '1998-01-27T00:00:00.000Z',
      'ShippedDate': '1998-02-03T00:00:00.000Z',
      'Freight': 53.83,
      'ShipName': 'Blauer See Delikatessen',
      'ShipAddress': 'Forsterstr. 57',
      'ShipCity': 'Mannheim',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10854,
      'CustomerID': 'ERNSH',
      'OrderDate': '1998-01-27T00:00:00.000Z',
      'ShippedDate': '1998-02-05T00:00:00.000Z',
      'Freight': 100.22,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10855,
      'CustomerID': 'OLDWO',

```

```

      'OrderDate': '1998-01-27T00:00:00.000Z',
      'ShippedDate': '1998-02-04T00:00:00.000Z',
      'Freight': 170.97,
      'ShipName': 'Old World Delicatessen',
      'ShipAddress': '2743 Bering St.',
      'ShipCity': 'Anchorage',
      'ShipRegion': 'AK',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10856,
      'CustomerID': 'ANTON',
      'OrderDate': '1998-01-28T00:00:00.000Z',
      'ShippedDate': '1998-02-10T00:00:00.000Z',
      'Freight': 58.43,
      'ShipName': 'Antonio Moreno Taquería',
      'ShipAddress': 'Mataderos 2312',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10857,
      'CustomerID': 'BERGS',
      'OrderDate': '1998-01-28T00:00:00.000Z',
      'ShippedDate': '1998-02-06T00:00:00.000Z',
      'Freight': 188.85,
      'ShipName': 'Berglunds snabbköp',
      'ShipAddress': 'Berguvsvägen 8',
      'ShipCity': 'Luleå',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 10858,
      'CustomerID': 'LACOR',
      'OrderDate': '1998-01-29T00:00:00.000Z',
      'ShippedDate': '1998-02-03T00:00:00.000Z',
      'Freight': 52.51,
      'ShipName': 'La corne d\'abondance',
      'ShipAddress': '67, avenue de l\'Europe',
      'ShipCity': 'Versailles',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10859,
      'CustomerID': 'FRANK',
      'OrderDate': '1998-01-29T00:00:00.000Z',
      'ShippedDate': '1998-02-02T00:00:00.000Z',
      'Freight': 76.1,
      'ShipName': 'Frankenversand',
      'ShipAddress': 'Berliner Platz 43',
      'ShipCity': 'München',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    }
  ],

```

```

{
  'OrderID': 10860,
  'CustomerID': 'FRANR',
  'OrderDate': '1998-01-29T00:00:00.000Z',
  'ShippedDate': '1998-02-04T00:00:00.000Z',
  'Freight': 19.26,
  'ShipName': 'France restauration',
  'ShipAddress': '54, rue Royale',
  'ShipCity': 'Nantes',
  'ShipRegion': null,
  'ShipCountry': 'France'
},
{
  'OrderID': 10861,
  'CustomerID': 'WHITC',
  'OrderDate': '1998-01-30T00:00:00.000Z',
  'ShippedDate': '1998-02-17T00:00:00.000Z',
  'Freight': 14.93,
  'ShipName': 'White Clover Markets',
  'ShipAddress': '1029 - 12th Ave. S.',
  'ShipCity': 'Seattle',
  'ShipRegion': 'WA',
  'ShipCountry': 'USA'
},
{
  'OrderID': 10862,
  'CustomerID': 'LEHMS',
  'OrderDate': '1998-01-30T00:00:00.000Z',
  'ShippedDate': '1998-02-02T00:00:00.000Z',
  'Freight': 53.23,
  'ShipName': 'Lehmanns Marktstand',
  'ShipAddress': 'Magazinweg 7',
  'ShipCity': 'Frankfurt a.M.',
  'ShipRegion': null,
  'ShipCountry': 'Germany'
},
{
  'OrderID': 10863,
  'CustomerID': 'HILAA',
  'OrderDate': '1998-02-02T00:00:00.000Z',
  'ShippedDate': '1998-02-17T00:00:00.000Z',
  'Freight': 30.26,
  'ShipName': 'HILARION-Abastos',
  'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
  'ShipCity': 'San Cristóbal',
  'ShipRegion': 'Táchira',
  'ShipCountry': 'Venezuela'
},
{
  'OrderID': 10864,
  'CustomerID': 'AROUT',
  'OrderDate': '1998-02-02T00:00:00.000Z',
  'ShippedDate': '1998-02-09T00:00:00.000Z',
  'Freight': 3.04,
  'ShipName': 'Around the Horn',
  'ShipAddress': 'Brook Farm Stratford St. Mary',
  'ShipCity': 'Colchester',

```

```

      'ShipRegion': 'Essex',
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10865,
      'CustomerID': 'QUICK',
      'OrderDate': '1998-02-02T00:00:00.000Z',
      'ShippedDate': '1998-02-12T00:00:00.000Z',
      'Freight': 348.14,
      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10866,
      'CustomerID': 'BERGS',
      'OrderDate': '1998-02-03T00:00:00.000Z',
      'ShippedDate': '1998-02-12T00:00:00.000Z',
      'Freight': 109.11,
      'ShipName': 'Berglunds snabbköp',
      'ShipAddress': 'Berguvsvägen 8',
      'ShipCity': 'Luleå',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 10867,
      'CustomerID': 'LONEP',
      'OrderDate': '1998-02-03T00:00:00.000Z',
      'ShippedDate': '1998-02-11T00:00:00.000Z',
      'Freight': 1.93,
      'ShipName': 'Lonesome Pine Restaurant',
      'ShipAddress': '89 Chiaroscuro Rd.',
      'ShipCity': 'Portland',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10868,
      'CustomerID': 'QUEEN',
      'OrderDate': '1998-02-04T00:00:00.000Z',
      'ShippedDate': '1998-02-23T00:00:00.000Z',
      'Freight': 191.27,
      'ShipName': 'Queen Cozinha',
      'ShipAddress': 'Alameda dos Canários, 891',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10869,
      'CustomerID': 'SEVES',
      'OrderDate': '1998-02-04T00:00:00.000Z',
      'ShippedDate': '1998-02-09T00:00:00.000Z',
      'Freight': 143.28,

```

```

      'ShipName': 'Seven Seas Imports',
      'ShipAddress': '90 Wadhurst Rd.',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10870,
      'CustomerID': 'WOLZA',
      'OrderDate': '1998-02-04T00:00:00.000Z',
      'ShippedDate': '1998-02-13T00:00:00.000Z',
      'Freight': 12.04,
      'ShipName': 'Wolski Zajazd',
      'ShipAddress': 'ul. Filtrowa 68',
      'ShipCity': 'Warszawa',
      'ShipRegion': null,
      'ShipCountry': 'Poland'
    },
    {
      'OrderID': 10871,
      'CustomerID': 'BONAP',
      'OrderDate': '1998-02-05T00:00:00.000Z',
      'ShippedDate': '1998-02-10T00:00:00.000Z',
      'Freight': 112.27,
      'ShipName': 'Bon app',
      'ShipAddress': '12, rue des Bouchers',
      'ShipCity': 'Marseille',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10872,
      'CustomerID': 'GODOS',
      'OrderDate': '1998-02-05T00:00:00.000Z',
      'ShippedDate': '1998-02-09T00:00:00.000Z',
      'Freight': 175.32,
      'ShipName': 'Godos Cocina Típica',
      'ShipAddress': 'C/ Romero, 33',
      'ShipCity': 'Sevilla',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    },
    {
      'OrderID': 10873,
      'CustomerID': 'WILMK',
      'OrderDate': '1998-02-06T00:00:00.000Z',
      'ShippedDate': '1998-02-09T00:00:00.000Z',
      'Freight': 0.82,
      'ShipName': 'Wilman Kala',
      'ShipAddress': 'Keskuskatu 45',
      'ShipCity': 'Helsinki',
      'ShipRegion': null,
      'ShipCountry': 'Finland'
    },
    {
      'OrderID': 10874,
      'CustomerID': 'GODOS',

```



```

      'OrderDate': '1998-02-06T00:00:00.000Z',
      'ShippedDate': '1998-02-11T00:00:00.000Z',
      'Freight': 19.58,
      'ShipName': 'Godos Cocina Típica',
      'ShipAddress': 'C/ Romero, 33',
      'ShipCity': 'Sevilla',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    },
    {
      'OrderID': 10875,
      'CustomerID': 'BERGS',
      'OrderDate': '1998-02-06T00:00:00.000Z',
      'ShippedDate': '1998-03-03T00:00:00.000Z',
      'Freight': 32.37,
      'ShipName': 'Berglunds snabbköp',
      'ShipAddress': 'Berguvsvägen 8',
      'ShipCity': 'Luleå',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 10876,
      'CustomerID': 'BONAP',
      'OrderDate': '1998-02-09T00:00:00.000Z',
      'ShippedDate': '1998-02-12T00:00:00.000Z',
      'Freight': 60.42,
      'ShipName': 'Bon app',
      'ShipAddress': '12, rue des Bouchers',
      'ShipCity': 'Marseille',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10877,
      'CustomerID': 'RICAR',
      'OrderDate': '1998-02-09T00:00:00.000Z',
      'ShippedDate': '1998-02-19T00:00:00.000Z',
      'Freight': 38.06,
      'ShipName': 'Ricardo Adocicados',
      'ShipAddress': 'Av. Copacabana, 267',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10878,
      'CustomerID': 'QUICK',
      'OrderDate': '1998-02-10T00:00:00.000Z',
      'ShippedDate': '1998-02-12T00:00:00.000Z',
      'Freight': 46.69,
      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    }
  ],

```

```

{
  'OrderID': 10879,
  'CustomerID': 'WILMK',
  'OrderDate': '1998-02-10T00:00:00.000Z',
  'ShippedDate': '1998-02-12T00:00:00.000Z',
  'Freight': 8.5,
  'ShipName': 'Wilman Kala',
  'ShipAddress': 'Keskuskatu 45',
  'ShipCity': 'Helsinki',
  'ShipRegion': null,
  'ShipCountry': 'Finland'
},
{
  'OrderID': 10880,
  'CustomerID': 'FOLKO',
  'OrderDate': '1998-02-10T00:00:00.000Z',
  'ShippedDate': '1998-02-18T00:00:00.000Z',
  'Freight': 88.01,
  'ShipName': 'Folk och fä HB',
  'ShipAddress': 'Åkergatan 24',
  'ShipCity': 'Bräcke',
  'ShipRegion': null,
  'ShipCountry': 'Sweden'
},
{
  'OrderID': 10881,
  'CustomerID': 'CACTU',
  'OrderDate': '1998-02-11T00:00:00.000Z',
  'ShippedDate': '1998-02-18T00:00:00.000Z',
  'Freight': 2.84,
  'ShipName': 'Cactus Comidas para llevar',
  'ShipAddress': 'Cerrito 333',
  'ShipCity': 'Buenos Aires',
  'ShipRegion': null,
  'ShipCountry': 'Argentina'
},
{
  'OrderID': 10882,
  'CustomerID': 'SAVEA',
  'OrderDate': '1998-02-11T00:00:00.000Z',
  'ShippedDate': '1998-02-20T00:00:00.000Z',
  'Freight': 23.1,
  'ShipName': 'Save-a-lot Markets',
  'ShipAddress': '187 Suffolk Ln.',
  'ShipCity': 'Boise',
  'ShipRegion': 'ID',
  'ShipCountry': 'USA'
},
{
  'OrderID': 10883,
  'CustomerID': 'LONEP',
  'OrderDate': '1998-02-12T00:00:00.000Z',
  'ShippedDate': '1998-02-20T00:00:00.000Z',
  'Freight': 0.53,
  'ShipName': 'Lonesome Pine Restaurant',
  'ShipAddress': '89 Chiaroscuro Rd.',
  'ShipCity': 'Portland',

```

```

      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10884,
      'CustomerID': 'LETSS',
      'OrderDate': '1998-02-12T00:00:00.000Z',
      'ShippedDate': '1998-02-13T00:00:00.000Z',
      'Freight': 90.97,
      'ShipName': 'Let\ Stop N Shop',
      'ShipAddress': '87 Polk St. Suite 5',
      'ShipCity': 'San Francisco',
      'ShipRegion': 'CA',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10885,
      'CustomerID': 'SUPRD',
      'OrderDate': '1998-02-12T00:00:00.000Z',
      'ShippedDate': '1998-02-18T00:00:00.000Z',
      'Freight': 5.64,
      'ShipName': 'Suprêmes délices',
      'ShipAddress': 'Boulevard Tirou, 255',
      'ShipCity': 'Charleroi',
      'ShipRegion': null,
      'ShipCountry': 'Belgium'
    },
    {
      'OrderID': 10886,
      'CustomerID': 'HANAR',
      'OrderDate': '1998-02-13T00:00:00.000Z',
      'ShippedDate': '1998-03-02T00:00:00.000Z',
      'Freight': 4.99,
      'ShipName': 'Hanari Carnes',
      'ShipAddress': 'Rua do Paço, 67',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10887,
      'CustomerID': 'GALED',
      'OrderDate': '1998-02-13T00:00:00.000Z',
      'ShippedDate': '1998-02-16T00:00:00.000Z',
      'Freight': 1.25,
      'ShipName': 'Galería del gastronómo',
      'ShipAddress': 'Rambla de Cataluña, 23',
      'ShipCity': 'Barcelona',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    },
    {
      'OrderID': 10888,
      'CustomerID': 'GODOS',
      'OrderDate': '1998-02-16T00:00:00.000Z',
      'ShippedDate': '1998-02-23T00:00:00.000Z',
      'Freight': 51.87,

```

```

    'ShipName': 'Godos Cocina Típica',
    'ShipAddress': 'C/ Romero, 33',
    'ShipCity': 'Sevilla',
    'ShipRegion': null,
    'ShipCountry': 'Spain'
  },
  {
    'OrderID': 10889,
    'CustomerID': 'RATTC',
    'OrderDate': '1998-02-16T00:00:00.000Z',
    'ShippedDate': '1998-02-23T00:00:00.000Z',
    'Freight': 280.61,
    'ShipName': 'Rattlesnake Canyon Grocery',
    'ShipAddress': '2817 Milton Dr.',
    'ShipCity': 'Albuquerque',
    'ShipRegion': 'NM',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10890,
    'CustomerID': 'DUMON',
    'OrderDate': '1998-02-16T00:00:00.000Z',
    'ShippedDate': '1998-02-18T00:00:00.000Z',
    'Freight': 32.76,
    'ShipName': 'Du monde entier',
    'ShipAddress': '67, rue des Cinquante Otages',
    'ShipCity': 'Nantes',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10891,
    'CustomerID': 'LEHMS',
    'OrderDate': '1998-02-17T00:00:00.000Z',
    'ShippedDate': '1998-02-19T00:00:00.000Z',
    'Freight': 20.37,
    'ShipName': 'Lehmanns Marktstand',
    'ShipAddress': 'Magazinweg 7',
    'ShipCity': 'Frankfurt a.M.',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 10892,
    'CustomerID': 'MAISD',
    'OrderDate': '1998-02-17T00:00:00.000Z',
    'ShippedDate': '1998-02-19T00:00:00.000Z',
    'Freight': 120.27,
    'ShipName': 'Maison Dewey',
    'ShipAddress': 'Rue Joseph-Bens 532',
    'ShipCity': 'Bruxelles',
    'ShipRegion': null,
    'ShipCountry': 'Belgium'
  },
  {
    'OrderID': 10893,
    'CustomerID': 'KOENE',

```

```

      'OrderDate': '1998-02-18T00:00:00.000Z',
      'ShippedDate': '1998-02-20T00:00:00.000Z',
      'Freight': 77.78,
      'ShipName': 'Königlich Essen',
      'ShipAddress': 'Maubelstr. 90',
      'ShipCity': 'Brandenburg',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10894,
      'CustomerID': 'SAVEA',
      'OrderDate': '1998-02-18T00:00:00.000Z',
      'ShippedDate': '1998-02-20T00:00:00.000Z',
      'Freight': 116.13,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10895,
      'CustomerID': 'ERNSH',
      'OrderDate': '1998-02-18T00:00:00.000Z',
      'ShippedDate': '1998-02-23T00:00:00.000Z',
      'Freight': 162.75,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10896,
      'CustomerID': 'MAISD',
      'OrderDate': '1998-02-19T00:00:00.000Z',
      'ShippedDate': '1998-02-27T00:00:00.000Z',
      'Freight': 32.45,
      'ShipName': 'Maison Dewey',
      'ShipAddress': 'Rue Joseph-Bens 532',
      'ShipCity': 'Bruxelles',
      'ShipRegion': null,
      'ShipCountry': 'Belgium'
    },
    {
      'OrderID': 10897,
      'CustomerID': 'HUNGO',
      'OrderDate': '1998-02-19T00:00:00.000Z',
      'ShippedDate': '1998-02-25T00:00:00.000Z',
      'Freight': 603.54,
      'ShipName': 'Hungry Owl All-Night Grocers',
      'ShipAddress': '8 Johnstown Road',
      'ShipCity': 'Cork',
      'ShipRegion': 'Co. Cork',
      'ShipCountry': 'Ireland'
    }
  ],

```

```

{
  'OrderID': 10898,
  'CustomerID': 'OCEAN',
  'OrderDate': '1998-02-20T00:00:00.000Z',
  'ShippedDate': '1998-03-06T00:00:00.000Z',
  'Freight': 1.27,
  'ShipName': 'Océano Atlántico Ltda.',
  'ShipAddress': 'Ing. Gustavo Moncada 8585 Piso 20-A',
  'ShipCity': 'Buenos Aires',
  'ShipRegion': null,
  'ShipCountry': 'Argentina'
},
{
  'OrderID': 10899,
  'CustomerID': 'LILAS',
  'OrderDate': '1998-02-20T00:00:00.000Z',
  'ShippedDate': '1998-02-26T00:00:00.000Z',
  'Freight': 1.21,
  'ShipName': 'LILA-Supermercado',
  'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
  'ShipCity': 'Barquisimeto',
  'ShipRegion': 'Lara',
  'ShipCountry': 'Venezuela'
},
{
  'OrderID': 10900,
  'CustomerID': 'WELLI',
  'OrderDate': '1998-02-20T00:00:00.000Z',
  'ShippedDate': '1998-03-04T00:00:00.000Z',
  'Freight': 1.66,
  'ShipName': 'Wellington Importadora',
  'ShipAddress': 'Rua do Mercado, 12',
  'ShipCity': 'Resende',
  'ShipRegion': 'SP',
  'ShipCountry': 'Brazil'
},
{
  'OrderID': 10901,
  'CustomerID': 'HILAA',
  'OrderDate': '1998-02-23T00:00:00.000Z',
  'ShippedDate': '1998-02-26T00:00:00.000Z',
  'Freight': 62.09,
  'ShipName': 'HILARION-Abastos',
  'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
  'ShipCity': 'San Cristóbal',
  'ShipRegion': 'Táchira',
  'ShipCountry': 'Venezuela'
},
{
  'OrderID': 10902,
  'CustomerID': 'FOLKO',
  'OrderDate': '1998-02-23T00:00:00.000Z',
  'ShippedDate': '1998-03-03T00:00:00.000Z',
  'Freight': 44.15,
  'ShipName': 'Folk och fä HB',
  'ShipAddress': 'Åkergatan 24',
  'ShipCity': 'Bräcke',

```

```

      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 10903,
      'CustomerID': 'HANAR',
      'OrderDate': '1998-02-24T00:00:00.000Z',
      'ShippedDate': '1998-03-04T00:00:00.000Z',
      'Freight': 36.71,
      'ShipName': 'Hanari Carnes',
      'ShipAddress': 'Rua do Paço, 67',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10904,
      'CustomerID': 'WHITC',
      'OrderDate': '1998-02-24T00:00:00.000Z',
      'ShippedDate': '1998-02-27T00:00:00.000Z',
      'Freight': 162.95,
      'ShipName': 'White Clover Markets',
      'ShipAddress': '1029 - 12th Ave. S.',
      'ShipCity': 'Seattle',
      'ShipRegion': 'WA',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10905,
      'CustomerID': 'WELLI',
      'OrderDate': '1998-02-24T00:00:00.000Z',
      'ShippedDate': '1998-03-06T00:00:00.000Z',
      'Freight': 13.72,
      'ShipName': 'Wellington Importadora',
      'ShipAddress': 'Rua do Mercado, 12',
      'ShipCity': 'Resende',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10906,
      'CustomerID': 'WOLZA',
      'OrderDate': '1998-02-25T00:00:00.000Z',
      'ShippedDate': '1998-03-03T00:00:00.000Z',
      'Freight': 26.29,
      'ShipName': 'Wolski Zajazd',
      'ShipAddress': 'ul. Filtrowa 68',
      'ShipCity': 'Warszawa',
      'ShipRegion': null,
      'ShipCountry': 'Poland'
    },
    {
      'OrderID': 10907,
      'CustomerID': 'SPECB',
      'OrderDate': '1998-02-25T00:00:00.000Z',
      'ShippedDate': '1998-02-27T00:00:00.000Z',
      'Freight': 9.19,

```

```

      'ShipName': 'Spécialités du monde',
      'ShipAddress': '25, rue Lauriston',
      'ShipCity': 'Paris',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10908,
      'CustomerID': 'REGGC',
      'OrderDate': '1998-02-26T00:00:00.000Z',
      'ShippedDate': '1998-03-06T00:00:00.000Z',
      'Freight': 32.96,
      'ShipName': 'Reggiani Caseifici',
      'ShipAddress': 'Strada Provinciale 124',
      'ShipCity': 'Reggio Emilia',
      'ShipRegion': null,
      'ShipCountry': 'Italy'
    },
    {
      'OrderID': 10909,
      'CustomerID': 'SANTG',
      'OrderDate': '1998-02-26T00:00:00.000Z',
      'ShippedDate': '1998-03-10T00:00:00.000Z',
      'Freight': 53.05,
      'ShipName': 'Santé Gourmet',
      'ShipAddress': 'Erling Skakkes gate 78',
      'ShipCity': 'Stavern',
      'ShipRegion': null,
      'ShipCountry': 'Norway'
    },
    {
      'OrderID': 10910,
      'CustomerID': 'WILMK',
      'OrderDate': '1998-02-26T00:00:00.000Z',
      'ShippedDate': '1998-03-04T00:00:00.000Z',
      'Freight': 38.11,
      'ShipName': 'Wilman Kala',
      'ShipAddress': 'Keskuskatu 45',
      'ShipCity': 'Helsinki',
      'ShipRegion': null,
      'ShipCountry': 'Finland'
    },
    {
      'OrderID': 10911,
      'CustomerID': 'GODOS',
      'OrderDate': '1998-02-26T00:00:00.000Z',
      'ShippedDate': '1998-03-05T00:00:00.000Z',
      'Freight': 38.19,
      'ShipName': 'Godos Cocina Típica',
      'ShipAddress': 'C/ Romero, 33',
      'ShipCity': 'Sevilla',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    },
    {
      'OrderID': 10912,
      'CustomerID': 'HUNGO',

```



```

      'OrderDate': '1998-02-26T00:00:00.000Z',
      'ShippedDate': '1998-03-18T00:00:00.000Z',
      'Freight': 580.91,
      'ShipName': 'Hungry Owl All-Night Grocers',
      'ShipAddress': '8 Johnstown Road',
      'ShipCity': 'Cork',
      'ShipRegion': 'Co. Cork',
      'ShipCountry': 'Ireland'
    },
    {
      'OrderID': 10913,
      'CustomerID': 'QUEEN',
      'OrderDate': '1998-02-26T00:00:00.000Z',
      'ShippedDate': '1998-03-04T00:00:00.000Z',
      'Freight': 33.05,
      'ShipName': 'Queen Cozinha',
      'ShipAddress': 'Alameda dos Canários, 891',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10914,
      'CustomerID': 'QUEEN',
      'OrderDate': '1998-02-27T00:00:00.000Z',
      'ShippedDate': '1998-03-02T00:00:00.000Z',
      'Freight': 21.19,
      'ShipName': 'Queen Cozinha',
      'ShipAddress': 'Alameda dos Canários, 891',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10915,
      'CustomerID': 'TORTU',
      'OrderDate': '1998-02-27T00:00:00.000Z',
      'ShippedDate': '1998-03-02T00:00:00.000Z',
      'Freight': 3.51,
      'ShipName': 'Tortuga Restaurante',
      'ShipAddress': 'Avda. Azteca 123',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10916,
      'CustomerID': 'RANCH',
      'OrderDate': '1998-02-27T00:00:00.000Z',
      'ShippedDate': '1998-03-09T00:00:00.000Z',
      'Freight': 63.77,
      'ShipName': 'Rancho grande',
      'ShipAddress': 'Av. del Libertador 900',
      'ShipCity': 'Buenos Aires',
      'ShipRegion': null,
      'ShipCountry': 'Argentina'
    }
  ],

```

```

{
  'OrderID': 10917,
  'CustomerID': 'ROMEY',
  'OrderDate': '1998-03-02T00:00:00.000Z',
  'ShippedDate': '1998-03-11T00:00:00.000Z',
  'Freight': 8.29,
  'ShipName': 'Romero y tomillo',
  'ShipAddress': 'Gran Vía, 1',
  'ShipCity': 'Madrid',
  'ShipRegion': null,
  'ShipCountry': 'Spain'
},
{
  'OrderID': 10918,
  'CustomerID': 'BOTTM',
  'OrderDate': '1998-03-02T00:00:00.000Z',
  'ShippedDate': '1998-03-11T00:00:00.000Z',
  'Freight': 48.83,
  'ShipName': 'Bottom-Dollar Markets',
  'ShipAddress': '23 Tsawassen Blvd.',
  'ShipCity': 'Tsawassen',
  'ShipRegion': 'BC',
  'ShipCountry': 'Canada'
},
{
  'OrderID': 10919,
  'CustomerID': 'LINOD',
  'OrderDate': '1998-03-02T00:00:00.000Z',
  'ShippedDate': '1998-03-04T00:00:00.000Z',
  'Freight': 19.8,
  'ShipName': 'LINO-Delicateses',
  'ShipAddress': 'Ave. 5 de Mayo Porlamar',
  'ShipCity': 'I. de Margarita',
  'ShipRegion': 'Nueva Esparta',
  'ShipCountry': 'Venezuela'
},
{
  'OrderID': 10920,
  'CustomerID': 'AROUT',
  'OrderDate': '1998-03-03T00:00:00.000Z',
  'ShippedDate': '1998-03-09T00:00:00.000Z',
  'Freight': 29.61,
  'ShipName': 'Around the Horn',
  'ShipAddress': 'Brook Farm Stratford St. Mary',
  'ShipCity': 'Colchester',
  'ShipRegion': 'Essex',
  'ShipCountry': 'UK'
},
{
  'OrderID': 10921,
  'CustomerID': 'VAFFE',
  'OrderDate': '1998-03-03T00:00:00.000Z',
  'ShippedDate': '1998-03-09T00:00:00.000Z',
  'Freight': 176.48,
  'ShipName': 'Vaffeljernet',
  'ShipAddress': 'Smagsloget 45',
  'ShipCity': 'Århus',

```

```

    'ShipRegion': null,
    'ShipCountry': 'Denmark'
  },
  {
    'OrderID': 10922,
    'CustomerID': 'HANAR',
    'OrderDate': '1998-03-03T00:00:00.000Z',
    'ShippedDate': '1998-03-05T00:00:00.000Z',
    'Freight': 62.74,
    'ShipName': 'Hanari Carnes',
    'ShipAddress': 'Rua do Paço, 67',
    'ShipCity': 'Rio de Janeiro',
    'ShipRegion': 'RJ',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10923,
    'CustomerID': 'LAMAI',
    'OrderDate': '1998-03-03T00:00:00.000Z',
    'ShippedDate': '1998-03-13T00:00:00.000Z',
    'Freight': 68.26,
    'ShipName': 'La maison d\'Asie',
    'ShipAddress': '1 rue Alsace-Lorraine',
    'ShipCity': 'Toulouse',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10924,
    'CustomerID': 'BERGS',
    'OrderDate': '1998-03-04T00:00:00.000Z',
    'ShippedDate': '1998-04-08T00:00:00.000Z',
    'Freight': 151.52,
    'ShipName': 'Berglunds snabbköp',
    'ShipAddress': 'Berguvsvägen 8',
    'ShipCity': 'Luleå',
    'ShipRegion': null,
    'ShipCountry': 'Sweden'
  },
  {
    'OrderID': 10925,
    'CustomerID': 'HANAR',
    'OrderDate': '1998-03-04T00:00:00.000Z',
    'ShippedDate': '1998-03-13T00:00:00.000Z',
    'Freight': 2.27,
    'ShipName': 'Hanari Carnes',
    'ShipAddress': 'Rua do Paço, 67',
    'ShipCity': 'Rio de Janeiro',
    'ShipRegion': 'RJ',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10926,
    'CustomerID': 'ANATR',
    'OrderDate': '1998-03-04T00:00:00.000Z',
    'ShippedDate': '1998-03-11T00:00:00.000Z',
    'Freight': 39.92,

```

```

      'ShipName': 'Ana Trujillo Emparedados y helados',
      'ShipAddress': 'Avda. de la Constitución 2222',
      'ShipCity': 'México D.F.',
      'ShipRegion': null,
      'ShipCountry': 'Mexico'
    },
    {
      'OrderID': 10927,
      'CustomerID': 'LACOR',
      'OrderDate': '1998-03-05T00:00:00.000Z',
      'ShippedDate': '1998-04-08T00:00:00.000Z',
      'Freight': 19.79,
      'ShipName': 'La corne d\'abondance',
      'ShipAddress': '67, avenue de l\'Europe',
      'ShipCity': 'Versailles',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10928,
      'CustomerID': 'GALED',
      'OrderDate': '1998-03-05T00:00:00.000Z',
      'ShippedDate': '1998-03-18T00:00:00.000Z',
      'Freight': 1.36,
      'ShipName': 'Galería del gastronómo',
      'ShipAddress': 'Rambla de Cataluña, 23',
      'ShipCity': 'Barcelona',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    },
    {
      'OrderID': 10929,
      'CustomerID': 'FRANK',
      'OrderDate': '1998-03-05T00:00:00.000Z',
      'ShippedDate': '1998-03-12T00:00:00.000Z',
      'Freight': 33.93,
      'ShipName': 'Frankenversand',
      'ShipAddress': 'Berliner Platz 43',
      'ShipCity': 'München',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10930,
      'CustomerID': 'SUPRD',
      'OrderDate': '1998-03-06T00:00:00.000Z',
      'ShippedDate': '1998-03-18T00:00:00.000Z',
      'Freight': 15.55,
      'ShipName': 'Suprêmes délices',
      'ShipAddress': 'Boulevard Tirou, 255',
      'ShipCity': 'Charleroi',
      'ShipRegion': null,
      'ShipCountry': 'Belgium'
    },
    {
      'OrderID': 10931,
      'CustomerID': 'RICSU',

```

```

      'OrderDate': '1998-03-06T00:00:00.000Z',
      'ShippedDate': '1998-03-19T00:00:00.000Z',
      'Freight': 13.6,
      'ShipName': 'Richter Supermarkt',
      'ShipAddress': 'Starenweg 5',
      'ShipCity': 'Genève',
      'ShipRegion': null,
      'ShipCountry': 'Switzerland'
    },
    {
      'OrderID': 10932,
      'CustomerID': 'BONAP',
      'OrderDate': '1998-03-06T00:00:00.000Z',
      'ShippedDate': '1998-03-24T00:00:00.000Z',
      'Freight': 134.64,
      'ShipName': 'Bon app',
      'ShipAddress': '12, rue des Bouchers',
      'ShipCity': 'Marseille',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10933,
      'CustomerID': 'ISLAT',
      'OrderDate': '1998-03-06T00:00:00.000Z',
      'ShippedDate': '1998-03-16T00:00:00.000Z',
      'Freight': 54.15,
      'ShipName': 'Island Trading',
      'ShipAddress': 'Garden House Crowther Way',
      'ShipCity': 'Cowes',
      'ShipRegion': 'Isle of Wight',
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10934,
      'CustomerID': 'LEHMS',
      'OrderDate': '1998-03-09T00:00:00.000Z',
      'ShippedDate': '1998-03-12T00:00:00.000Z',
      'Freight': 32.01,
      'ShipName': 'Lehmanns Marktstand',
      'ShipAddress': 'Magazinweg 7',
      'ShipCity': 'Frankfurt a.M.',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10935,
      'CustomerID': 'WELLI',
      'OrderDate': '1998-03-09T00:00:00.000Z',
      'ShippedDate': '1998-03-18T00:00:00.000Z',
      'Freight': 47.59,
      'ShipName': 'Wellington Importadora',
      'ShipAddress': 'Rua do Mercado, 12',
      'ShipCity': 'Resende',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    }
  ],

```

```

{
  'OrderID': 10936,
  'CustomerID': 'GREAL',
  'OrderDate': '1998-03-09T00:00:00.000Z',
  'ShippedDate': '1998-03-18T00:00:00.000Z',
  'Freight': 33.68,
  'ShipName': 'Great Lakes Food Market',
  'ShipAddress': '2732 Baker Blvd.',
  'ShipCity': 'Eugene',
  'ShipRegion': 'OR',
  'ShipCountry': 'USA'
},
{
  'OrderID': 10937,
  'CustomerID': 'CACTU',
  'OrderDate': '1998-03-10T00:00:00.000Z',
  'ShippedDate': '1998-03-13T00:00:00.000Z',
  'Freight': 31.51,
  'ShipName': 'Cactus Comidas para llevar',
  'ShipAddress': 'Cerrito 333',
  'ShipCity': 'Buenos Aires',
  'ShipRegion': null,
  'ShipCountry': 'Argentina'
},
{
  'OrderID': 10938,
  'CustomerID': 'QUICK',
  'OrderDate': '1998-03-10T00:00:00.000Z',
  'ShippedDate': '1998-03-16T00:00:00.000Z',
  'Freight': 31.89,
  'ShipName': 'QUICK-Stop',
  'ShipAddress': 'Taucherstraße 10',
  'ShipCity': 'Cunewalde',
  'ShipRegion': null,
  'ShipCountry': 'Germany'
},
{
  'OrderID': 10939,
  'CustomerID': 'MAGAA',
  'OrderDate': '1998-03-10T00:00:00.000Z',
  'ShippedDate': '1998-03-13T00:00:00.000Z',
  'Freight': 76.33,
  'ShipName': 'Magazzini Alimentari Riuniti',
  'ShipAddress': 'Via Ludovico il Moro 22',
  'ShipCity': 'Bergamo',
  'ShipRegion': null,
  'ShipCountry': 'Italy'
},
{
  'OrderID': 10940,
  'CustomerID': 'BONAP',
  'OrderDate': '1998-03-11T00:00:00.000Z',
  'ShippedDate': '1998-03-23T00:00:00.000Z',
  'Freight': 19.77,
  'ShipName': 'Bon app',
  'ShipAddress': '12, rue des Bouchers',
  'ShipCity': 'Marseille',

```

```

    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 10941,
    'CustomerID': 'SAVEA',
    'OrderDate': '1998-03-11T00:00:00.000Z',
    'ShippedDate': '1998-03-20T00:00:00.000Z',
    'Freight': 400.81,
    'ShipName': 'Save-a-lot Markets',
    'ShipAddress': '187 Suffolk Ln.',
    'ShipCity': 'Boise',
    'ShipRegion': 'ID',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10942,
    'CustomerID': 'REGGC',
    'OrderDate': '1998-03-11T00:00:00.000Z',
    'ShippedDate': '1998-03-18T00:00:00.000Z',
    'Freight': 17.95,
    'ShipName': 'Reggiani Caseifici',
    'ShipAddress': 'Strada Provinciale 124',
    'ShipCity': 'Reggio Emilia',
    'ShipRegion': null,
    'ShipCountry': 'Italy'
  },
  {
    'OrderID': 10943,
    'CustomerID': 'BSBEV',
    'OrderDate': '1998-03-11T00:00:00.000Z',
    'ShippedDate': '1998-03-19T00:00:00.000Z',
    'Freight': 2.17,
    'ShipName': 'B\' Beverages',
    'ShipAddress': 'Fauntleroy Circus',
    'ShipCity': 'London',
    'ShipRegion': null,
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10944,
    'CustomerID': 'BOTTM',
    'OrderDate': '1998-03-12T00:00:00.000Z',
    'ShippedDate': '1998-03-13T00:00:00.000Z',
    'Freight': 52.92,
    'ShipName': 'Bottom-Dollar Markets',
    'ShipAddress': '23 Tsawassen Blvd.',
    'ShipCity': 'Tsawassen',
    'ShipRegion': 'BC',
    'ShipCountry': 'Canada'
  },
  {
    'OrderID': 10945,
    'CustomerID': 'MORGK',
    'OrderDate': '1998-03-12T00:00:00.000Z',
    'ShippedDate': '1998-03-18T00:00:00.000Z',
    'Freight': 10.22,

```

```

      'ShipName': 'Morgenstern Gesundkost',
      'ShipAddress': 'Heerstr. 22',
      'ShipCity': 'Leipzig',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10946,
      'CustomerID': 'VAFFE',
      'OrderDate': '1998-03-12T00:00:00.000Z',
      'ShippedDate': '1998-03-19T00:00:00.000Z',
      'Freight': 27.2,
      'ShipName': 'Vaffeljernet',
      'ShipAddress': 'Smagsloget 45',
      'ShipCity': 'Århus',
      'ShipRegion': null,
      'ShipCountry': 'Denmark'
    },
    {
      'OrderID': 10947,
      'CustomerID': 'BSBEV',
      'OrderDate': '1998-03-13T00:00:00.000Z',
      'ShippedDate': '1998-03-16T00:00:00.000Z',
      'Freight': 3.26,
      'ShipName': 'B\ Beverages',
      'ShipAddress': 'Fauntleroy Circus',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10948,
      'CustomerID': 'GODOS',
      'OrderDate': '1998-03-13T00:00:00.000Z',
      'ShippedDate': '1998-03-19T00:00:00.000Z',
      'Freight': 23.39,
      'ShipName': 'Godos Cocina Típica',
      'ShipAddress': 'C/ Romero, 33',
      'ShipCity': 'Sevilla',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    },
    {
      'OrderID': 10949,
      'CustomerID': 'BOTTM',
      'OrderDate': '1998-03-13T00:00:00.000Z',
      'ShippedDate': '1998-03-17T00:00:00.000Z',
      'Freight': 74.44,
      'ShipName': 'Bottom-Dollar Markets',
      'ShipAddress': '23 Tsawassen Blvd.',
      'ShipCity': 'Tsawassen',
      'ShipRegion': 'BC',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 10950,
      'CustomerID': 'MAGAA',

```



```

      'OrderDate': '1998-03-16T00:00:00.000Z',
      'ShippedDate': '1998-03-23T00:00:00.000Z',
      'Freight': 2.5,
      'ShipName': 'Magazzini Alimentari Riuniti',
      'ShipAddress': 'Via Ludovico il Moro 22',
      'ShipCity': 'Bergamo',
      'ShipRegion': null,
      'ShipCountry': 'Italy'
    },
    {
      'OrderID': 10951,
      'CustomerID': 'RICSU',
      'OrderDate': '1998-03-16T00:00:00.000Z',
      'ShippedDate': '1998-04-07T00:00:00.000Z',
      'Freight': 30.85,
      'ShipName': 'Richter Supermarkt',
      'ShipAddress': 'Starenweg 5',
      'ShipCity': 'Genève',
      'ShipRegion': null,
      'ShipCountry': 'Switzerland'
    },
    {
      'OrderID': 10952,
      'CustomerID': 'ALFKI',
      'OrderDate': '1998-03-16T00:00:00.000Z',
      'ShippedDate': '1998-03-24T00:00:00.000Z',
      'Freight': 40.42,
      'ShipName': 'Alfred\ Futterkiste',
      'ShipAddress': 'Obere Str. 57',
      'ShipCity': 'Berlin',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10953,
      'CustomerID': 'AROUT',
      'OrderDate': '1998-03-16T00:00:00.000Z',
      'ShippedDate': '1998-03-25T00:00:00.000Z',
      'Freight': 23.72,
      'ShipName': 'Around the Horn',
      'ShipAddress': 'Brook Farm Stratford St. Mary',
      'ShipCity': 'Colchester',
      'ShipRegion': 'Essex',
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 10954,
      'CustomerID': 'LINOD',
      'OrderDate': '1998-03-17T00:00:00.000Z',
      'ShippedDate': '1998-03-20T00:00:00.000Z',
      'Freight': 27.91,
      'ShipName': 'LINO-Delicateses',
      'ShipAddress': 'Ave. 5 de Mayo Porlamar',
      'ShipCity': 'I. de Margarita',
      'ShipRegion': 'Nueva Esparta',
      'ShipCountry': 'Venezuela'
    }
  ],

```

```

{
  'OrderID': 10955,
  'CustomerID': 'FOLKO',
  'OrderDate': '1998-03-17T00:00:00.000Z',
  'ShippedDate': '1998-03-20T00:00:00.000Z',
  'Freight': 3.26,
  'ShipName': 'Folk och fä HB',
  'ShipAddress': 'Åkergatan 24',
  'ShipCity': 'Bräcke',
  'ShipRegion': null,
  'ShipCountry': 'Sweden'
},
{
  'OrderID': 10956,
  'CustomerID': 'BLAUS',
  'OrderDate': '1998-03-17T00:00:00.000Z',
  'ShippedDate': '1998-03-20T00:00:00.000Z',
  'Freight': 44.65,
  'ShipName': 'Blauer See Delikatessen',
  'ShipAddress': 'Forsterstr. 57',
  'ShipCity': 'Mannheim',
  'ShipRegion': null,
  'ShipCountry': 'Germany'
},
{
  'OrderID': 10957,
  'CustomerID': 'HILAA',
  'OrderDate': '1998-03-18T00:00:00.000Z',
  'ShippedDate': '1998-03-27T00:00:00.000Z',
  'Freight': 105.36,
  'ShipName': 'HILARION-Abastos',
  'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
  'ShipCity': 'San Cristóbal',
  'ShipRegion': 'Táchira',
  'ShipCountry': 'Venezuela'
},
{
  'OrderID': 10958,
  'CustomerID': 'OCEAN',
  'OrderDate': '1998-03-18T00:00:00.000Z',
  'ShippedDate': '1998-03-27T00:00:00.000Z',
  'Freight': 49.56,
  'ShipName': 'Océano Atlántico Ltda.',
  'ShipAddress': 'Ing. Gustavo Moncada 8585 Piso 20-A',
  'ShipCity': 'Buenos Aires',
  'ShipRegion': null,
  'ShipCountry': 'Argentina'
},
{
  'OrderID': 10959,
  'CustomerID': 'GOURL',
  'OrderDate': '1998-03-18T00:00:00.000Z',
  'ShippedDate': '1998-03-23T00:00:00.000Z',
  'Freight': 4.98,
  'ShipName': 'Gourmet Lanchonetes',
  'ShipAddress': 'Av. Brasil, 442',
  'ShipCity': 'Campinas',

```

```

      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10960,
      'CustomerID': 'HILAA',
      'OrderDate': '1998-03-19T00:00:00.000Z',
      'ShippedDate': '1998-04-08T00:00:00.000Z',
      'Freight': 2.08,
      'ShipName': 'HILARION-Abastos',
      'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
      'ShipCity': 'San Cristóbal',
      'ShipRegion': 'Táchira',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10961,
      'CustomerID': 'QUEEN',
      'OrderDate': '1998-03-19T00:00:00.000Z',
      'ShippedDate': '1998-03-30T00:00:00.000Z',
      'Freight': 104.47,
      'ShipName': 'Queen Cozinha',
      'ShipAddress': 'Alameda dos Canários, 891',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10962,
      'CustomerID': 'QUICK',
      'OrderDate': '1998-03-19T00:00:00.000Z',
      'ShippedDate': '1998-03-23T00:00:00.000Z',
      'Freight': 275.79,
      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10963,
      'CustomerID': 'FURIB',
      'OrderDate': '1998-03-19T00:00:00.000Z',
      'ShippedDate': '1998-03-26T00:00:00.000Z',
      'Freight': 2.7,
      'ShipName': 'Furia Bacalhau e Frutos do Mar',
      'ShipAddress': 'Jardim das rosas n. 32',
      'ShipCity': 'Lisboa',
      'ShipRegion': null,
      'ShipCountry': 'Portugal'
    },
    {
      'OrderID': 10964,
      'CustomerID': 'SPECB',
      'OrderDate': '1998-03-20T00:00:00.000Z',
      'ShippedDate': '1998-03-24T00:00:00.000Z',
      'Freight': 87.38,

```

```

      'ShipName': 'Spécialités du monde',
      'ShipAddress': '25, rue Lauriston',
      'ShipCity': 'Paris',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10965,
      'CustomerID': 'OLDWO',
      'OrderDate': '1998-03-20T00:00:00.000Z',
      'ShippedDate': '1998-03-30T00:00:00.000Z',
      'Freight': 144.38,
      'ShipName': 'Old World Delicatessen',
      'ShipAddress': '2743 Bering St.',
      'ShipCity': 'Anchorage',
      'ShipRegion': 'AK',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10966,
      'CustomerID': 'CHOPS',
      'OrderDate': '1998-03-20T00:00:00.000Z',
      'ShippedDate': '1998-04-08T00:00:00.000Z',
      'Freight': 27.19,
      'ShipName': 'Chop-suey Chinese',
      'ShipAddress': 'Hauptstr. 31',
      'ShipCity': 'Bern',
      'ShipRegion': null,
      'ShipCountry': 'Switzerland'
    },
    {
      'OrderID': 10967,
      'CustomerID': 'TOMSP',
      'OrderDate': '1998-03-23T00:00:00.000Z',
      'ShippedDate': '1998-04-02T00:00:00.000Z',
      'Freight': 62.22,
      'ShipName': 'Toms Spezialitäten',
      'ShipAddress': 'Luisenstr. 48',
      'ShipCity': 'Münster',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10968,
      'CustomerID': 'ERNSH',
      'OrderDate': '1998-03-23T00:00:00.000Z',
      'ShippedDate': '1998-04-01T00:00:00.000Z',
      'Freight': 74.6,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10969,
      'CustomerID': 'COMMI',

```

```

      'OrderDate': '1998-03-23T00:00:00.000Z',
      'ShippedDate': '1998-03-30T00:00:00.000Z',
      'Freight': 0.21,
      'ShipName': 'Comércio Mineiro',
      'ShipAddress': 'Av. dos Lusíadas, 23',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10970,
      'CustomerID': 'BOLID',
      'OrderDate': '1998-03-24T00:00:00.000Z',
      'ShippedDate': '1998-04-24T00:00:00.000Z',
      'Freight': 16.16,
      'ShipName': 'Bólido Comidas preparadas',
      'ShipAddress': 'C/ Araquil, 67',
      'ShipCity': 'Madrid',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    },
    {
      'OrderID': 10971,
      'CustomerID': 'FRANR',
      'OrderDate': '1998-03-24T00:00:00.000Z',
      'ShippedDate': '1998-04-02T00:00:00.000Z',
      'Freight': 121.82,
      'ShipName': 'France restauration',
      'ShipAddress': '54, rue Royale',
      'ShipCity': 'Nantes',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10972,
      'CustomerID': 'LACOR',
      'OrderDate': '1998-03-24T00:00:00.000Z',
      'ShippedDate': '1998-03-26T00:00:00.000Z',
      'Freight': 0.02,
      'ShipName': 'La corne d\'abondance',
      'ShipAddress': '67, avenue de l\'Europe',
      'ShipCity': 'Versailles',
      'ShipRegion': null,
      'ShipCountry': 'France'
    },
    {
      'OrderID': 10973,
      'CustomerID': 'LACOR',
      'OrderDate': '1998-03-24T00:00:00.000Z',
      'ShippedDate': '1998-03-27T00:00:00.000Z',
      'Freight': 15.17,
      'ShipName': 'La corne d\'abondance',
      'ShipAddress': '67, avenue de l\'Europe',
      'ShipCity': 'Versailles',
      'ShipRegion': null,
      'ShipCountry': 'France'
    }
  ],

```

```

{
  'OrderID': 10974,
  'CustomerID': 'SPLIR',
  'OrderDate': '1998-03-25T00:00:00.000Z',
  'ShippedDate': '1998-04-03T00:00:00.000Z',
  'Freight': 12.96,
  'ShipName': 'Split Rail Beer & Ale',
  'ShipAddress': 'P.O. Box 555',
  'ShipCity': 'Lander',
  'ShipRegion': 'WY',
  'ShipCountry': 'USA'
},
{
  'OrderID': 10975,
  'CustomerID': 'BOTTM',
  'OrderDate': '1998-03-25T00:00:00.000Z',
  'ShippedDate': '1998-03-27T00:00:00.000Z',
  'Freight': 32.27,
  'ShipName': 'Bottom-Dollar Markets',
  'ShipAddress': '23 Tsawassen Blvd.',
  'ShipCity': 'Tsawassen',
  'ShipRegion': 'BC',
  'ShipCountry': 'Canada'
},
{
  'OrderID': 10976,
  'CustomerID': 'HILAA',
  'OrderDate': '1998-03-25T00:00:00.000Z',
  'ShippedDate': '1998-04-03T00:00:00.000Z',
  'Freight': 37.97,
  'ShipName': 'HILARION-Abastos',
  'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
  'ShipCity': 'San Cristóbal',
  'ShipRegion': 'Táchira',
  'ShipCountry': 'Venezuela'
},
{
  'OrderID': 10977,
  'CustomerID': 'FOLKO',
  'OrderDate': '1998-03-26T00:00:00.000Z',
  'ShippedDate': '1998-04-10T00:00:00.000Z',
  'Freight': 208.5,
  'ShipName': 'Folk och fä HB',
  'ShipAddress': 'Åkergatan 24',
  'ShipCity': 'Bräcke',
  'ShipRegion': null,
  'ShipCountry': 'Sweden'
},
{
  'OrderID': 10978,
  'CustomerID': 'MAISD',
  'OrderDate': '1998-03-26T00:00:00.000Z',
  'ShippedDate': '1998-04-23T00:00:00.000Z',
  'Freight': 32.82,
  'ShipName': 'Maison Dewey',
  'ShipAddress': 'Rue Joseph-Bens 532',
  'ShipCity': 'Bruxelles',

```

```

    'ShipRegion': null,
    'ShipCountry': 'Belgium'
  },
  {
    'OrderID': 10979,
    'CustomerID': 'ERNSH',
    'OrderDate': '1998-03-26T00:00:00.000Z',
    'ShippedDate': '1998-03-31T00:00:00.000Z',
    'Freight': 353.07,
    'ShipName': 'Ernst Handel',
    'ShipAddress': 'Kirchgasse 6',
    'ShipCity': 'Graz',
    'ShipRegion': null,
    'ShipCountry': 'Austria'
  },
  {
    'OrderID': 10980,
    'CustomerID': 'FOLKO',
    'OrderDate': '1998-03-27T00:00:00.000Z',
    'ShippedDate': '1998-04-17T00:00:00.000Z',
    'Freight': 1.26,
    'ShipName': 'Folk och fä HB',
    'ShipAddress': 'Åkergatan 24',
    'ShipCity': 'Bräcke',
    'ShipRegion': null,
    'ShipCountry': 'Sweden'
  },
  {
    'OrderID': 10981,
    'CustomerID': 'HANAR',
    'OrderDate': '1998-03-27T00:00:00.000Z',
    'ShippedDate': '1998-04-02T00:00:00.000Z',
    'Freight': 193.37,
    'ShipName': 'Hanari Carnes',
    'ShipAddress': 'Rua do Paço, 67',
    'ShipCity': 'Rio de Janeiro',
    'ShipRegion': 'RJ',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 10982,
    'CustomerID': 'BOTTM',
    'OrderDate': '1998-03-27T00:00:00.000Z',
    'ShippedDate': '1998-04-08T00:00:00.000Z',
    'Freight': 14.01,
    'ShipName': 'Bottom-Dollar Markets',
    'ShipAddress': '23 Tsawassen Blvd.',
    'ShipCity': 'Tsawassen',
    'ShipRegion': 'BC',
    'ShipCountry': 'Canada'
  },
  {
    'OrderID': 10983,
    'CustomerID': 'SAVEA',
    'OrderDate': '1998-03-27T00:00:00.000Z',
    'ShippedDate': '1998-04-06T00:00:00.000Z',
    'Freight': 657.54,

```

```

    'ShipName': 'Save-a-lot Markets',
    'ShipAddress': '187 Suffolk Ln.',
    'ShipCity': 'Boise',
    'ShipRegion': 'ID',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10984,
    'CustomerID': 'SAVEA',
    'OrderDate': '1998-03-30T00:00:00.000Z',
    'ShippedDate': '1998-04-03T00:00:00.000Z',
    'Freight': 211.22,
    'ShipName': 'Save-a-lot Markets',
    'ShipAddress': '187 Suffolk Ln.',
    'ShipCity': 'Boise',
    'ShipRegion': 'ID',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 10985,
    'CustomerID': 'HUNGO',
    'OrderDate': '1998-03-30T00:00:00.000Z',
    'ShippedDate': '1998-04-02T00:00:00.000Z',
    'Freight': 91.51,
    'ShipName': 'Hungry Owl All-Night Grocers',
    'ShipAddress': '8 Johnstown Road',
    'ShipCity': 'Cork',
    'ShipRegion': 'Co. Cork',
    'ShipCountry': 'Ireland'
  },
  {
    'OrderID': 10986,
    'CustomerID': 'OCEAN',
    'OrderDate': '1998-03-30T00:00:00.000Z',
    'ShippedDate': '1998-04-21T00:00:00.000Z',
    'Freight': 217.86,
    'ShipName': 'Océano Atlántico Ltda.',
    'ShipAddress': 'Ing. Gustavo Moncada 8585 Piso 20-A',
    'ShipCity': 'Buenos Aires',
    'ShipRegion': null,
    'ShipCountry': 'Argentina'
  },
  {
    'OrderID': 10987,
    'CustomerID': 'EASTC',
    'OrderDate': '1998-03-31T00:00:00.000Z',
    'ShippedDate': '1998-04-06T00:00:00.000Z',
    'Freight': 185.48,
    'ShipName': 'Eastern Connection',
    'ShipAddress': '35 King George',
    'ShipCity': 'London',
    'ShipRegion': null,
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 10988,
    'CustomerID': 'RATTC',

```



```

      'OrderDate': '1998-03-31T00:00:00.000Z',
      'ShippedDate': '1998-04-10T00:00:00.000Z',
      'Freight': 61.14,
      'ShipName': 'Rattlesnake Canyon Grocery',
      'ShipAddress': '2817 Milton Dr.',
      'ShipCity': 'Albuquerque',
      'ShipRegion': 'NM',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 10989,
      'CustomerID': 'QUEDE',
      'OrderDate': '1998-03-31T00:00:00.000Z',
      'ShippedDate': '1998-04-02T00:00:00.000Z',
      'Freight': 34.76,
      'ShipName': 'Que Delícia',
      'ShipAddress': 'Rua da Panificadora, 12',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 10990,
      'CustomerID': 'ERNSH',
      'OrderDate': '1998-04-01T00:00:00.000Z',
      'ShippedDate': '1998-04-07T00:00:00.000Z',
      'Freight': 117.61,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 10991,
      'CustomerID': 'QUICK',
      'OrderDate': '1998-04-01T00:00:00.000Z',
      'ShippedDate': '1998-04-07T00:00:00.000Z',
      'Freight': 38.51,
      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 10992,
      'CustomerID': 'THEBI',
      'OrderDate': '1998-04-01T00:00:00.000Z',
      'ShippedDate': '1998-04-03T00:00:00.000Z',
      'Freight': 4.27,
      'ShipName': 'The Big Cheese',
      'ShipAddress': '89 Jefferson Way Suite 2',
      'ShipCity': 'Portland',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    }
  ],

```

```

{
  'OrderID': 10993,
  'CustomerID': 'FOLKO',
  'OrderDate': '1998-04-01T00:00:00.000Z',
  'ShippedDate': '1998-04-10T00:00:00.000Z',
  'Freight': 8.81,
  'ShipName': 'Folk och fä HB',
  'ShipAddress': 'Åkergatan 24',
  'ShipCity': 'Bräcke',
  'ShipRegion': null,
  'ShipCountry': 'Sweden'
},
{
  'OrderID': 10994,
  'CustomerID': 'VAFFE',
  'OrderDate': '1998-04-02T00:00:00.000Z',
  'ShippedDate': '1998-04-09T00:00:00.000Z',
  'Freight': 65.53,
  'ShipName': 'Vaffeljernet',
  'ShipAddress': 'Smagsloget 45',
  'ShipCity': 'Århus',
  'ShipRegion': null,
  'ShipCountry': 'Denmark'
},
{
  'OrderID': 10995,
  'CustomerID': 'PERIC',
  'OrderDate': '1998-04-02T00:00:00.000Z',
  'ShippedDate': '1998-04-06T00:00:00.000Z',
  'Freight': 46,
  'ShipName': 'Pericles Comidas clásicas',
  'ShipAddress': 'Calle Dr. Jorge Cash 321',
  'ShipCity': 'México D.F.',
  'ShipRegion': null,
  'ShipCountry': 'Mexico'
},
{
  'OrderID': 10996,
  'CustomerID': 'QUICK',
  'OrderDate': '1998-04-02T00:00:00.000Z',
  'ShippedDate': '1998-04-10T00:00:00.000Z',
  'Freight': 1.12,
  'ShipName': 'QUICK-Stop',
  'ShipAddress': 'Taucherstraße 10',
  'ShipCity': 'Cunewalde',
  'ShipRegion': null,
  'ShipCountry': 'Germany'
},
{
  'OrderID': 10997,
  'CustomerID': 'LILAS',
  'OrderDate': '1998-04-03T00:00:00.000Z',
  'ShippedDate': '1998-04-13T00:00:00.000Z',
  'Freight': 73.91,
  'ShipName': 'LILA-Supermercado',
  'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
  'ShipCity': 'Barquisimeto',

```

```

      'ShipRegion': 'Lara',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 10998,
      'CustomerID': 'WOLZA',
      'OrderDate': '1998-04-03T00:00:00.000Z',
      'ShippedDate': '1998-04-17T00:00:00.000Z',
      'Freight': 20.31,
      'ShipName': 'Wolski Zajazd',
      'ShipAddress': 'ul. Filtrowa 68',
      'ShipCity': 'Warszawa',
      'ShipRegion': null,
      'ShipCountry': 'Poland'
    },
    {
      'OrderID': 10999,
      'CustomerID': 'OTTIK',
      'OrderDate': '1998-04-03T00:00:00.000Z',
      'ShippedDate': '1998-04-10T00:00:00.000Z',
      'Freight': 96.35,
      'ShipName': 'Ottilies Käseladen',
      'ShipAddress': 'Mehrheimerstr. 369',
      'ShipCity': 'Köln',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 11000,
      'CustomerID': 'RATTC',
      'OrderDate': '1998-04-06T00:00:00.000Z',
      'ShippedDate': '1998-04-14T00:00:00.000Z',
      'Freight': 55.12,
      'ShipName': 'Rattlesnake Canyon Grocery',
      'ShipAddress': '2817 Milton Dr.',
      'ShipCity': 'Albuquerque',
      'ShipRegion': 'NM',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 11001,
      'CustomerID': 'FOLKO',
      'OrderDate': '1998-04-06T00:00:00.000Z',
      'ShippedDate': '1998-04-14T00:00:00.000Z',
      'Freight': 197.3,
      'ShipName': 'Folk och fä HB',
      'ShipAddress': 'Åkergatan 24',
      'ShipCity': 'Bräcke',
      'ShipRegion': null,
      'ShipCountry': 'Sweden'
    },
    {
      'OrderID': 11002,
      'CustomerID': 'SAVEA',
      'OrderDate': '1998-04-06T00:00:00.000Z',
      'ShippedDate': '1998-04-16T00:00:00.000Z',
      'Freight': 141.16,

```

```

      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 11003,
      'CustomerID': 'THECR',
      'OrderDate': '1998-04-06T00:00:00.000Z',
      'ShippedDate': '1998-04-08T00:00:00.000Z',
      'Freight': 14.91,
      'ShipName': 'The Cracker Box',
      'ShipAddress': '55 Grizzly Peak Rd.',
      'ShipCity': 'Butte',
      'ShipRegion': 'MT',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 11004,
      'CustomerID': 'MAISD',
      'OrderDate': '1998-04-07T00:00:00.000Z',
      'ShippedDate': '1998-04-20T00:00:00.000Z',
      'Freight': 44.84,
      'ShipName': 'Maison Dewey',
      'ShipAddress': 'Rue Joseph-Bens 532',
      'ShipCity': 'Bruxelles',
      'ShipRegion': null,
      'ShipCountry': 'Belgium'
    },
    {
      'OrderID': 11005,
      'CustomerID': 'WILMK',
      'OrderDate': '1998-04-07T00:00:00.000Z',
      'ShippedDate': '1998-04-10T00:00:00.000Z',
      'Freight': 0.75,
      'ShipName': 'Wilman Kala',
      'ShipAddress': 'Keskuskatu 45',
      'ShipCity': 'Helsinki',
      'ShipRegion': null,
      'ShipCountry': 'Finland'
    },
    {
      'OrderID': 11006,
      'CustomerID': 'GREAL',
      'OrderDate': '1998-04-07T00:00:00.000Z',
      'ShippedDate': '1998-04-15T00:00:00.000Z',
      'Freight': 25.19,
      'ShipName': 'Great Lakes Food Market',
      'ShipAddress': '2732 Baker Blvd.',
      'ShipCity': 'Eugene',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 11007,
      'CustomerID': 'PRINI',

```

```

      'OrderDate': '1998-04-08T00:00:00.000Z',
      'ShippedDate': '1998-04-13T00:00:00.000Z',
      'Freight': 202.24,
      'ShipName': 'Princesa Isabel Vinhos',
      'ShipAddress': 'Estrada da saúde n. 58',
      'ShipCity': 'Lisboa',
      'ShipRegion': null,
      'ShipCountry': 'Portugal'
    },
    {
      'OrderID': 11008,
      'CustomerID': 'ERNSH',
      'OrderDate': '1998-04-08T00:00:00.000Z',
      'ShippedDate': null,
      'Freight': 79.46,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 11009,
      'CustomerID': 'GODOS',
      'OrderDate': '1998-04-08T00:00:00.000Z',
      'ShippedDate': '1998-04-10T00:00:00.000Z',
      'Freight': 59.11,
      'ShipName': 'Godos Cocina Típica',
      'ShipAddress': 'C/ Romero, 33',
      'ShipCity': 'Sevilla',
      'ShipRegion': null,
      'ShipCountry': 'Spain'
    },
    {
      'OrderID': 11010,
      'CustomerID': 'REGGC',
      'OrderDate': '1998-04-09T00:00:00.000Z',
      'ShippedDate': '1998-04-21T00:00:00.000Z',
      'Freight': 28.71,
      'ShipName': 'Reggiani Caseifici',
      'ShipAddress': 'Strada Provinciale 124',
      'ShipCity': 'Reggio Emilia',
      'ShipRegion': null,
      'ShipCountry': 'Italy'
    },
    {
      'OrderID': 11011,
      'CustomerID': 'ALFKI',
      'OrderDate': '1998-04-09T00:00:00.000Z',
      'ShippedDate': '1998-04-13T00:00:00.000Z',
      'Freight': 1.21,
      'ShipName': 'Alfred\ Futterkiste',
      'ShipAddress': 'Obere Str. 57',
      'ShipCity': 'Berlin',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    }
  ],

```

```

{
  'OrderID': 11012,
  'CustomerID': 'FRANK',
  'OrderDate': '1998-04-09T00:00:00.000Z',
  'ShippedDate': '1998-04-17T00:00:00.000Z',
  'Freight': 242.95,
  'ShipName': 'Frankenversand',
  'ShipAddress': 'Berliner Platz 43',
  'ShipCity': 'München',
  'ShipRegion': null,
  'ShipCountry': 'Germany'
},
{
  'OrderID': 11013,
  'CustomerID': 'ROMEY',
  'OrderDate': '1998-04-09T00:00:00.000Z',
  'ShippedDate': '1998-04-10T00:00:00.000Z',
  'Freight': 32.99,
  'ShipName': 'Romero y tomillo',
  'ShipAddress': 'Gran Vía, 1',
  'ShipCity': 'Madrid',
  'ShipRegion': null,
  'ShipCountry': 'Spain'
},
{
  'OrderID': 11014,
  'CustomerID': 'LINOD',
  'OrderDate': '1998-04-10T00:00:00.000Z',
  'ShippedDate': '1998-04-15T00:00:00.000Z',
  'Freight': 23.6,
  'ShipName': 'LINO-Delicateses',
  'ShipAddress': 'Ave. 5 de Mayo Porlamar',
  'ShipCity': 'I. de Margarita',
  'ShipRegion': 'Nueva Esparta',
  'ShipCountry': 'Venezuela'
},
{
  'OrderID': 11015,
  'CustomerID': 'SANTG',
  'OrderDate': '1998-04-10T00:00:00.000Z',
  'ShippedDate': '1998-04-20T00:00:00.000Z',
  'Freight': 4.62,
  'ShipName': 'Santé Gourmet',
  'ShipAddress': 'Erling Skakkes gate 78',
  'ShipCity': 'Stavern',
  'ShipRegion': null,
  'ShipCountry': 'Norway'
},
{
  'OrderID': 11016,
  'CustomerID': 'AROUT',
  'OrderDate': '1998-04-10T00:00:00.000Z',
  'ShippedDate': '1998-04-13T00:00:00.000Z',
  'Freight': 33.8,
  'ShipName': 'Around the Horn',
  'ShipAddress': 'Brook Farm Stratford St. Mary',
  'ShipCity': 'Colchester',

```

```

      'ShipRegion': 'Essex',
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 11017,
      'CustomerID': 'ERNSH',
      'OrderDate': '1998-04-13T00:00:00.000Z',
      'ShippedDate': '1998-04-20T00:00:00.000Z',
      'Freight': 754.26,
      'ShipName': 'Ernst Handel',
      'ShipAddress': 'Kirchgasse 6',
      'ShipCity': 'Graz',
      'ShipRegion': null,
      'ShipCountry': 'Austria'
    },
    {
      'OrderID': 11018,
      'CustomerID': 'LONEP',
      'OrderDate': '1998-04-13T00:00:00.000Z',
      'ShippedDate': '1998-04-16T00:00:00.000Z',
      'Freight': 11.65,
      'ShipName': 'Lonesome Pine Restaurant',
      'ShipAddress': '89 Chiaroscuro Rd.',
      'ShipCity': 'Portland',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 11019,
      'CustomerID': 'RANCH',
      'OrderDate': '1998-04-13T00:00:00.000Z',
      'ShippedDate': null,
      'Freight': 3.17,
      'ShipName': 'Rancho grande',
      'ShipAddress': 'Av. del Libertador 900',
      'ShipCity': 'Buenos Aires',
      'ShipRegion': null,
      'ShipCountry': 'Argentina'
    },
    {
      'OrderID': 11020,
      'CustomerID': 'OTTIK',
      'OrderDate': '1998-04-14T00:00:00.000Z',
      'ShippedDate': '1998-04-16T00:00:00.000Z',
      'Freight': 43.3,
      'ShipName': 'Ottilies Käseladen',
      'ShipAddress': 'Mehrheimerstr. 369',
      'ShipCity': 'Köln',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 11021,
      'CustomerID': 'QUICK',
      'OrderDate': '1998-04-14T00:00:00.000Z',
      'ShippedDate': '1998-04-21T00:00:00.000Z',
      'Freight': 297.18,

```

```

      'ShipName': 'QUICK-Stop',
      'ShipAddress': 'Taucherstraße 10',
      'ShipCity': 'Cunewalde',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 11022,
      'CustomerID': 'HANAR',
      'OrderDate': '1998-04-14T00:00:00.000Z',
      'ShippedDate': '1998-05-04T00:00:00.000Z',
      'Freight': 6.27,
      'ShipName': 'Hanari Carnes',
      'ShipAddress': 'Rua do Paço, 67',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 11023,
      'CustomerID': 'BSBEV',
      'OrderDate': '1998-04-14T00:00:00.000Z',
      'ShippedDate': '1998-04-24T00:00:00.000Z',
      'Freight': 123.83,
      'ShipName': 'B\' Beverages',
      'ShipAddress': 'Fauntleroy Circus',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 11024,
      'CustomerID': 'EASTC',
      'OrderDate': '1998-04-15T00:00:00.000Z',
      'ShippedDate': '1998-04-20T00:00:00.000Z',
      'Freight': 74.36,
      'ShipName': 'Eastern Connection',
      'ShipAddress': '35 King George',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 11025,
      'CustomerID': 'WARTH',
      'OrderDate': '1998-04-15T00:00:00.000Z',
      'ShippedDate': '1998-04-24T00:00:00.000Z',
      'Freight': 29.17,
      'ShipName': 'Wartian Herkku',
      'ShipAddress': 'Torikatu 38',
      'ShipCity': 'Oulu',
      'ShipRegion': null,
      'ShipCountry': 'Finland'
    },
    {
      'OrderID': 11026,
      'CustomerID': 'FRANS',

```



```

      'OrderDate': '1998-04-15T00:00:00.000Z',
      'ShippedDate': '1998-04-28T00:00:00.000Z',
      'Freight': 47.09,
      'ShipName': 'Franchi S.p.A.',
      'ShipAddress': 'Via Monte Bianco 34',
      'ShipCity': 'Torino',
      'ShipRegion': null,
      'ShipCountry': 'Italy'
    },
    {
      'OrderID': 11027,
      'CustomerID': 'BOTTM',
      'OrderDate': '1998-04-16T00:00:00.000Z',
      'ShippedDate': '1998-04-20T00:00:00.000Z',
      'Freight': 52.52,
      'ShipName': 'Bottom-Dollar Markets',
      'ShipAddress': '23 Tsawassen Blvd.',
      'ShipCity': 'Tsawassen',
      'ShipRegion': 'BC',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 11028,
      'CustomerID': 'KOENE',
      'OrderDate': '1998-04-16T00:00:00.000Z',
      'ShippedDate': '1998-04-22T00:00:00.000Z',
      'Freight': 29.59,
      'ShipName': 'Königlich Essen',
      'ShipAddress': 'Maubelstr. 90',
      'ShipCity': 'Brandenburg',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 11029,
      'CustomerID': 'CHOPS',
      'OrderDate': '1998-04-16T00:00:00.000Z',
      'ShippedDate': '1998-04-27T00:00:00.000Z',
      'Freight': 47.84,
      'ShipName': 'Chop-suey Chinese',
      'ShipAddress': 'Hauptstr. 31',
      'ShipCity': 'Bern',
      'ShipRegion': null,
      'ShipCountry': 'Switzerland'
    },
    {
      'OrderID': 11030,
      'CustomerID': 'SAVEA',
      'OrderDate': '1998-04-17T00:00:00.000Z',
      'ShippedDate': '1998-04-27T00:00:00.000Z',
      'Freight': 830.75,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    }
  ],

```

```

{
  'OrderID': 11031,
  'CustomerID': 'SAVEA',
  'OrderDate': '1998-04-17T00:00:00.000Z',
  'ShippedDate': '1998-04-24T00:00:00.000Z',
  'Freight': 227.22,
  'ShipName': 'Save-a-lot Markets',
  'ShipAddress': '187 Suffolk Ln.',
  'ShipCity': 'Boise',
  'ShipRegion': 'ID',
  'ShipCountry': 'USA'
},
{
  'OrderID': 11032,
  'CustomerID': 'WHITC',
  'OrderDate': '1998-04-17T00:00:00.000Z',
  'ShippedDate': '1998-04-23T00:00:00.000Z',
  'Freight': 606.19,
  'ShipName': 'White Clover Markets',
  'ShipAddress': '1029 - 12th Ave. S.',
  'ShipCity': 'Seattle',
  'ShipRegion': 'WA',
  'ShipCountry': 'USA'
},
{
  'OrderID': 11033,
  'CustomerID': 'RICSU',
  'OrderDate': '1998-04-17T00:00:00.000Z',
  'ShippedDate': '1998-04-23T00:00:00.000Z',
  'Freight': 84.74,
  'ShipName': 'Richter Supermarkt',
  'ShipAddress': 'Starenweg 5',
  'ShipCity': 'Genève',
  'ShipRegion': null,
  'ShipCountry': 'Switzerland'
},
{
  'OrderID': 11034,
  'CustomerID': 'OLDWO',
  'OrderDate': '1998-04-20T00:00:00.000Z',
  'ShippedDate': '1998-04-27T00:00:00.000Z',
  'Freight': 40.32,
  'ShipName': 'Old World Delicatessen',
  'ShipAddress': '2743 Bering St.',
  'ShipCity': 'Anchorage',
  'ShipRegion': 'AK',
  'ShipCountry': 'USA'
},
{
  'OrderID': 11035,
  'CustomerID': 'SUPRD',
  'OrderDate': '1998-04-20T00:00:00.000Z',
  'ShippedDate': '1998-04-24T00:00:00.000Z',
  'Freight': 0.17,
  'ShipName': 'Suprêmes délices',
  'ShipAddress': 'Boulevard Tirou, 255',
  'ShipCity': 'Charleroi',

```

```

    'ShipRegion': null,
    'ShipCountry': 'Belgium'
  },
  {
    'OrderID': 11036,
    'CustomerID': 'DRACD',
    'OrderDate': '1998-04-20T00:00:00.000Z',
    'ShippedDate': '1998-04-22T00:00:00.000Z',
    'Freight': 149.47,
    'ShipName': 'Drachenblut Delikatessen',
    'ShipAddress': 'Walserweg 21',
    'ShipCity': 'Aachen',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 11037,
    'CustomerID': 'GODOS',
    'OrderDate': '1998-04-21T00:00:00.000Z',
    'ShippedDate': '1998-04-27T00:00:00.000Z',
    'Freight': 3.2,
    'ShipName': 'Godos Cocina Típica',
    'ShipAddress': 'C/ Romero, 33',
    'ShipCity': 'Sevilla',
    'ShipRegion': null,
    'ShipCountry': 'Spain'
  },
  {
    'OrderID': 11038,
    'CustomerID': 'SUPRD',
    'OrderDate': '1998-04-21T00:00:00.000Z',
    'ShippedDate': '1998-04-30T00:00:00.000Z',
    'Freight': 29.59,
    'ShipName': 'Suprêmes délices',
    'ShipAddress': 'Boulevard Tirou, 255',
    'ShipCity': 'Charleroi',
    'ShipRegion': null,
    'ShipCountry': 'Belgium'
  },
  {
    'OrderID': 11039,
    'CustomerID': 'LINOD',
    'OrderDate': '1998-04-21T00:00:00.000Z',
    'ShippedDate': null,
    'Freight': 65,
    'ShipName': 'LINO-Delicateses',
    'ShipAddress': 'Ave. 5 de Mayo Porlamar',
    'ShipCity': 'I. de Margarita',
    'ShipRegion': 'Nueva Esparta',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 11040,
    'CustomerID': 'GREAL',
    'OrderDate': '1998-04-22T00:00:00.000Z',
    'ShippedDate': null,
    'Freight': 18.84,

```

```

    'ShipName': 'Great Lakes Food Market',
    'ShipAddress': '2732 Baker Blvd.',
    'ShipCity': 'Eugene',
    'ShipRegion': 'OR',
    'ShipCountry': 'USA'
  },
  {
    'OrderID': 11041,
    'CustomerID': 'CHOPS',
    'OrderDate': '1998-04-22T00:00:00.000Z',
    'ShippedDate': '1998-04-28T00:00:00.000Z',
    'Freight': 48.22,
    'ShipName': 'Chop-suey Chinese',
    'ShipAddress': 'Hauptstr. 31',
    'ShipCity': 'Bern',
    'ShipRegion': null,
    'ShipCountry': 'Switzerland'
  },
  {
    'OrderID': 11042,
    'CustomerID': 'COMMI',
    'OrderDate': '1998-04-22T00:00:00.000Z',
    'ShippedDate': '1998-05-01T00:00:00.000Z',
    'Freight': 29.99,
    'ShipName': 'Comércio Mineiro',
    'ShipAddress': 'Av. dos Lusíadas, 23',
    'ShipCity': 'Sao Paulo',
    'ShipRegion': 'SP',
    'ShipCountry': 'Brazil'
  },
  {
    'OrderID': 11043,
    'CustomerID': 'SPECB',
    'OrderDate': '1998-04-22T00:00:00.000Z',
    'ShippedDate': '1998-04-29T00:00:00.000Z',
    'Freight': 8.8,
    'ShipName': 'Spécialités du monde',
    'ShipAddress': '25, rue Lauriston',
    'ShipCity': 'Paris',
    'ShipRegion': null,
    'ShipCountry': 'France'
  },
  {
    'OrderID': 11044,
    'CustomerID': 'WOLZA',
    'OrderDate': '1998-04-23T00:00:00.000Z',
    'ShippedDate': '1998-05-01T00:00:00.000Z',
    'Freight': 8.72,
    'ShipName': 'Wolski Zajazd',
    'ShipAddress': 'ul. Filtrowa 68',
    'ShipCity': 'Warszawa',
    'ShipRegion': null,
    'ShipCountry': 'Poland'
  },
  {
    'OrderID': 11045,
    'CustomerID': 'BOTTM',

```

```

      'OrderDate': '1998-04-23T00:00:00.000Z',
      'ShippedDate': null,
      'Freight': 70.58,
      'ShipName': 'Bottom-Dollar Markets',
      'ShipAddress': '23 Tsawassen Blvd.',
      'ShipCity': 'Tsawassen',
      'ShipRegion': 'BC',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 11046,
      'CustomerID': 'WANDK',
      'OrderDate': '1998-04-23T00:00:00.000Z',
      'ShippedDate': '1998-04-24T00:00:00.000Z',
      'Freight': 71.64,
      'ShipName': 'Die Wandernde Kuh',
      'ShipAddress': 'Adenauerallee 900',
      'ShipCity': 'Stuttgart',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 11047,
      'CustomerID': 'EASTC',
      'OrderDate': '1998-04-24T00:00:00.000Z',
      'ShippedDate': '1998-05-01T00:00:00.000Z',
      'Freight': 46.62,
      'ShipName': 'Eastern Connection',
      'ShipAddress': '35 King George',
      'ShipCity': 'London',
      'ShipRegion': null,
      'ShipCountry': 'UK'
    },
    {
      'OrderID': 11048,
      'CustomerID': 'BOTTM',
      'OrderDate': '1998-04-24T00:00:00.000Z',
      'ShippedDate': '1998-04-30T00:00:00.000Z',
      'Freight': 24.12,
      'ShipName': 'Bottom-Dollar Markets',
      'ShipAddress': '23 Tsawassen Blvd.',
      'ShipCity': 'Tsawassen',
      'ShipRegion': 'BC',
      'ShipCountry': 'Canada'
    },
    {
      'OrderID': 11049,
      'CustomerID': 'GOURL',
      'OrderDate': '1998-04-24T00:00:00.000Z',
      'ShippedDate': '1998-05-04T00:00:00.000Z',
      'Freight': 8.34,
      'ShipName': 'Gourmet Lanchonetes',
      'ShipAddress': 'Av. Brasil, 442',
      'ShipCity': 'Campinas',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    }
  ],

```

```

{
  'OrderID': 11050,
  'CustomerID': 'FOLKO',
  'OrderDate': '1998-04-27T00:00:00.000Z',
  'ShippedDate': '1998-05-05T00:00:00.000Z',
  'Freight': 59.41,
  'ShipName': 'Folk och fä HB',
  'ShipAddress': 'Åkergatan 24',
  'ShipCity': 'Bräcke',
  'ShipRegion': null,
  'ShipCountry': 'Sweden'
},
{
  'OrderID': 11051,
  'CustomerID': 'LAMAI',
  'OrderDate': '1998-04-27T00:00:00.000Z',
  'ShippedDate': null,
  'Freight': 2.79,
  'ShipName': 'La maison d\'Asie',
  'ShipAddress': '1 rue Alsace-Lorraine',
  'ShipCity': 'Toulouse',
  'ShipRegion': null,
  'ShipCountry': 'France'
},
{
  'OrderID': 11052,
  'CustomerID': 'HANAR',
  'OrderDate': '1998-04-27T00:00:00.000Z',
  'ShippedDate': '1998-05-01T00:00:00.000Z',
  'Freight': 67.26,
  'ShipName': 'Hanari Carnes',
  'ShipAddress': 'Rua do Paço, 67',
  'ShipCity': 'Rio de Janeiro',
  'ShipRegion': 'RJ',
  'ShipCountry': 'Brazil'
},
{
  'OrderID': 11053,
  'CustomerID': 'PICCO',
  'OrderDate': '1998-04-27T00:00:00.000Z',
  'ShippedDate': '1998-04-29T00:00:00.000Z',
  'Freight': 53.05,
  'ShipName': 'Piccolo und mehr',
  'ShipAddress': 'Geislweg 14',
  'ShipCity': 'Salzburg',
  'ShipRegion': null,
  'ShipCountry': 'Austria'
},
{
  'OrderID': 11054,
  'CustomerID': 'CACTU',
  'OrderDate': '1998-04-28T00:00:00.000Z',
  'ShippedDate': null,
  'Freight': 0.33,
  'ShipName': 'Cactus Comidas para llevar',
  'ShipAddress': 'Cerrito 333',
  'ShipCity': 'Buenos Aires',

```

```

    'ShipRegion': null,
    'ShipCountry': 'Argentina'
  },
  {
    'OrderID': 11055,
    'CustomerID': 'HILAA',
    'OrderDate': '1998-04-28T00:00:00.000Z',
    'ShippedDate': '1998-05-05T00:00:00.000Z',
    'Freight': 120.92,
    'ShipName': 'HILARION-Abastos',
    'ShipAddress': 'Carrera 22 con Ave. Carlos Soublette #8-35',
    'ShipCity': 'San Cristóbal',
    'ShipRegion': 'Táchira',
    'ShipCountry': 'Venezuela'
  },
  {
    'OrderID': 11056,
    'CustomerID': 'EASTC',
    'OrderDate': '1998-04-28T00:00:00.000Z',
    'ShippedDate': '1998-05-01T00:00:00.000Z',
    'Freight': 278.96,
    'ShipName': 'Eastern Connection',
    'ShipAddress': '35 King George',
    'ShipCity': 'London',
    'ShipRegion': null,
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 11057,
    'CustomerID': 'NORTS',
    'OrderDate': '1998-04-29T00:00:00.000Z',
    'ShippedDate': '1998-05-01T00:00:00.000Z',
    'Freight': 4.13,
    'ShipName': 'North/South',
    'ShipAddress': 'South House 300 Queensbridge',
    'ShipCity': 'London',
    'ShipRegion': null,
    'ShipCountry': 'UK'
  },
  {
    'OrderID': 11058,
    'CustomerID': 'BLAUS',
    'OrderDate': '1998-04-29T00:00:00.000Z',
    'ShippedDate': null,
    'Freight': 31.14,
    'ShipName': 'Blauer See Delikatessen',
    'ShipAddress': 'Forsterstr. 57',
    'ShipCity': 'Mannheim',
    'ShipRegion': null,
    'ShipCountry': 'Germany'
  },
  {
    'OrderID': 11059,
    'CustomerID': 'RICAR',
    'OrderDate': '1998-04-29T00:00:00.000Z',
    'ShippedDate': null,
    'Freight': 85.8,

```

```

      'ShipName': 'Ricardo Adocicados',
      'ShipAddress': 'Av. Copacabana, 267',
      'ShipCity': 'Rio de Janeiro',
      'ShipRegion': 'RJ',
      'ShipCountry': 'Brazil'
    },
    {
      'OrderID': 11060,
      'CustomerID': 'FRANS',
      'OrderDate': '1998-04-30T00:00:00.000Z',
      'ShippedDate': '1998-05-04T00:00:00.000Z',
      'Freight': 10.98,
      'ShipName': 'Franchi S.p.A.',
      'ShipAddress': 'Via Monte Bianco 34',
      'ShipCity': 'Torino',
      'ShipRegion': null,
      'ShipCountry': 'Italy'
    },
    {
      'OrderID': 11061,
      'CustomerID': 'GREAL',
      'OrderDate': '1998-04-30T00:00:00.000Z',
      'ShippedDate': null,
      'Freight': 14.01,
      'ShipName': 'Great Lakes Food Market',
      'ShipAddress': '2732 Baker Blvd.',
      'ShipCity': 'Eugene',
      'ShipRegion': 'OR',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 11062,
      'CustomerID': 'REGGC',
      'OrderDate': '1998-04-30T00:00:00.000Z',
      'ShippedDate': null,
      'Freight': 29.93,
      'ShipName': 'Reggiani Caseifici',
      'ShipAddress': 'Strada Provinciale 124',
      'ShipCity': 'Reggio Emilia',
      'ShipRegion': null,
      'ShipCountry': 'Italy'
    },
    {
      'OrderID': 11063,
      'CustomerID': 'HUNGO',
      'OrderDate': '1998-04-30T00:00:00.000Z',
      'ShippedDate': '1998-05-06T00:00:00.000Z',
      'Freight': 81.73,
      'ShipName': 'Hungry Owl All-Night Grocers',
      'ShipAddress': '8 Johnstown Road',
      'ShipCity': 'Cork',
      'ShipRegion': 'Co. Cork',
      'ShipCountry': 'Ireland'
    },
    {
      'OrderID': 11064,
      'CustomerID': 'SAVEA',

```



```

      'OrderDate': '1998-05-01T00:00:00.000Z',
      'ShippedDate': '1998-05-04T00:00:00.000Z',
      'Freight': 30.09,
      'ShipName': 'Save-a-lot Markets',
      'ShipAddress': '187 Suffolk Ln.',
      'ShipCity': 'Boise',
      'ShipRegion': 'ID',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 11065,
      'CustomerID': 'LILAS',
      'OrderDate': '1998-05-01T00:00:00.000Z',
      'ShippedDate': null,
      'Freight': 12.91,
      'ShipName': 'LILA-Supermercado',
      'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
      'ShipCity': 'Barquisimeto',
      'ShipRegion': 'Lara',
      'ShipCountry': 'Venezuela'
    },
    {
      'OrderID': 11066,
      'CustomerID': 'WHITC',
      'OrderDate': '1998-05-01T00:00:00.000Z',
      'ShippedDate': '1998-05-04T00:00:00.000Z',
      'Freight': 44.72,
      'ShipName': 'White Clover Markets',
      'ShipAddress': '1029 - 12th Ave. S.',
      'ShipCity': 'Seattle',
      'ShipRegion': 'WA',
      'ShipCountry': 'USA'
    },
    {
      'OrderID': 11067,
      'CustomerID': 'DRACD',
      'OrderDate': '1998-05-04T00:00:00.000Z',
      'ShippedDate': '1998-05-06T00:00:00.000Z',
      'Freight': 7.98,
      'ShipName': 'Drachenblut Delikatessen',
      'ShipAddress': 'Walserweg 21',
      'ShipCity': 'Aachen',
      'ShipRegion': null,
      'ShipCountry': 'Germany'
    },
    {
      'OrderID': 11068,
      'CustomerID': 'QUEEN',
      'OrderDate': '1998-05-04T00:00:00.000Z',
      'ShippedDate': null,
      'Freight': 81.75,
      'ShipName': 'Queen Cozinha',
      'ShipAddress': 'Alameda dos Canários, 891',
      'ShipCity': 'Sao Paulo',
      'ShipRegion': 'SP',
      'ShipCountry': 'Brazil'
    }
  ],

```

```

{
  'OrderID': 11069,
  'CustomerID': 'TORTU',
  'OrderDate': '1998-05-04T00:00:00.000Z',
  'ShippedDate': '1998-05-06T00:00:00.000Z',
  'Freight': 15.67,
  'ShipName': 'Tortuga Restaurante',
  'ShipAddress': 'Avda. Azteca 123',
  'ShipCity': 'México D.F.',
  'ShipRegion': null,
  'ShipCountry': 'Mexico'
},
{
  'OrderID': 11070,
  'CustomerID': 'LEHMS',
  'OrderDate': '1998-05-05T00:00:00.000Z',
  'ShippedDate': null,
  'Freight': 136,
  'ShipName': 'Lehmanns Marktstand',
  'ShipAddress': 'Magazinweg 7',
  'ShipCity': 'Frankfurt a.M.',
  'ShipRegion': null,
  'ShipCountry': 'Germany'
},
{
  'OrderID': 11071,
  'CustomerID': 'LILAS',
  'OrderDate': '1998-05-05T00:00:00.000Z',
  'ShippedDate': null,
  'Freight': 0.93,
  'ShipName': 'LILA-Supermercado',
  'ShipAddress': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
  'ShipCity': 'Barquisimeto',
  'ShipRegion': 'Lara',
  'ShipCountry': 'Venezuela'
},
{
  'OrderID': 11072,
  'CustomerID': 'ERNSH',
  'OrderDate': '1998-05-05T00:00:00.000Z',
  'ShippedDate': null,
  'Freight': 258.64,
  'ShipName': 'Ernst Handel',
  'ShipAddress': 'Kirchgasse 6',
  'ShipCity': 'Graz',
  'ShipRegion': null,
  'ShipCountry': 'Austria'
},
{
  'OrderID': 11073,
  'CustomerID': 'PERIC',
  'OrderDate': '1998-05-05T00:00:00.000Z',
  'ShippedDate': null,
  'Freight': 24.95,
  'ShipName': 'Pericles Comidas clásicas',
  'ShipAddress': 'Calle Dr. Jorge Cash 321',
  'ShipCity': 'México D.F.',

```

```

        'ShipRegion': null,
        'ShipCountry': 'Mexico'
    },
    {
        'OrderID': 11074,
        'CustomerID': 'SIMOB',
        'OrderDate': '1998-05-06T00:00:00.000Z',
        'ShippedDate': null,
        'Freight': 18.44,
        'ShipName': 'Simons bistro',
        'ShipAddress': 'Vindbæltet 34',
        'ShipCity': 'Kobenhavn',
        'ShipRegion': null,
        'ShipCountry': 'Denmark'
    },
    {
        'OrderID': 11075,
        'CustomerID': 'RICSU',
        'OrderDate': '1998-05-06T00:00:00.000Z',
        'ShippedDate': null,
        'Freight': 6.19,
        'ShipName': 'Richter Supermarkt',
        'ShipAddress': 'Starenweg 5',
        'ShipCity': 'Genève',
        'ShipRegion': null,
        'ShipCountry': 'Switzerland'
    },
    {
        'OrderID': 11076,
        'CustomerID': 'BONAP',
        'OrderDate': '1998-05-06T00:00:00.000Z',
        'ShippedDate': null,
        'Freight': 38.28,
        'ShipName': 'Bon app',
        'ShipAddress': '12, rue des Bouchers',
        'ShipCity': 'Marseille',
        'ShipRegion': null,
        'ShipCountry': 'France'
    },
    {
        'OrderID': 11077,
        'CustomerID': 'RATTC',
        'OrderDate': '1998-05-06T00:00:00.000Z',
        'ShippedDate': null,
        'Freight': 8.53,
        'ShipName': 'Rattlesnake Canyon Grocery',
        'ShipAddress': '2817 Milton Dr.',
        'ShipCity': 'Albuquerque',
        'ShipRegion': 'NM',
        'ShipCountry': 'USA'
    }
  ];
export const orderData: Object[] = JSON.parse(stringData, (field: any,
value: any) => {
  let dupValue: any = value;
  if (typeof value === 'string' && /^(\\d{4}\\-\\d\\d\\-\\d\\d([tT][\\d:\\.]*){1})([zZ]|([+\\-])(\\d\\d):?(\\d\\d))?$/\\.test(value)) {

```

```

    let arr: any = dupValue.split(/^[0-9]/);
    let arg: any = parseInt(arr[4], 10);
    let arg1: any = parseInt(arr[5], 10);
    value = new Date(parseInt(arr[0], 10), parseInt(arr[1], 10) - 1,
    parseInt(arr[2], 10), parseInt(arr[3], 10), arg, arg1);
  }
  return value;
});
export const categoryData: Object[] = [
  {
    'CategoryName': 'Beverages',
    'ProductName': 'Chai',
    'QuantityPerUnit': '10 boxes x 20 bags',
    'UnitsInStock': 39,
    'Discontinued': true
  },
  {
    'CategoryName': 'Beverages',
    'ProductName': 'Chang',
    'QuantityPerUnit': '24 - 12 oz bottles',
    'UnitsInStock': 17,
    'Discontinued': true
  },
  {
    'CategoryName': 'Beverages',
    'ProductName': 'Chartreuse verte',
    'QuantityPerUnit': '750 cc per bottle',
    'UnitsInStock': 69,
    'Discontinued': true
  },
  {
    'CategoryName': 'Beverages',
    'ProductName': 'C\u00f4te de Blaye',
    'QuantityPerUnit': '12 - 75 cl bottles',
    'UnitsInStock': 17,
    'Discontinued': false
  },
  {
    'CategoryName': 'Beverages',
    'ProductName': 'Ipoh Coffee',
    'QuantityPerUnit': '16 - 500 g tins',
    'UnitsInStock': 17,
    'Discontinued': true
  },
  {
    'CategoryName': 'Beverages',
    'ProductName': 'Lakkalik\u00f6\u00f6ri',
    'QuantityPerUnit': '500 ml',
    'UnitsInStock': 57,
    'Discontinued': true
  },
  {
    'CategoryName': 'Beverages',
    'ProductName': 'Laughing Lumberjack Lager',
    'QuantityPerUnit': '24 - 12 oz bottles',
    'UnitsInStock': 52,
    'Discontinued': false
  }
]

```

```

    },
    {
      'CategoryName': 'Beverages',
      'ProductName': 'Outback Lager',
      'QuantityPerUnit': '24 - 355 ml bottles',
      'UnitsInStock': 15,
      'Discontinued': false
    },
    {
      'CategoryName': 'Beverages',
      'ProductName': 'Rh\u00f6nbr\u00e4 Klosterbier',
      'QuantityPerUnit': '24 - 0.5 l bottles',
      'UnitsInStock': 125,
      'Discontinued': false
    },
    {
      'CategoryName': 'Beverages',
      'ProductName': 'Sasquatch Ale',
      'QuantityPerUnit': '24 - 12 oz bottles',
      'UnitsInStock': 111,
      'Discontinued': true
    },
    {
      'CategoryName': 'Beverages',
      'ProductName': 'Steeleye Stout',
      'QuantityPerUnit': '24 - 12 oz bottles',
      'UnitsInStock': 20,
      'Discontinued': true
    },
    {
      'CategoryName': 'Condiments',
      'ProductName': 'Aniseed Syrup',
      'QuantityPerUnit': '12 - 550 ml bottles',
      'UnitsInStock': 13,
      'Discontinued': true
    },
    {
      'CategoryName': 'Condiments',
      'ProductName': 'Chef Anton\'s Cajun Seasoning',
      'QuantityPerUnit': '48 - 6 oz jars',
      'UnitsInStock': 53,
      'Discontinued': false
    },
    {
      'CategoryName': 'Condiments',
      'ProductName': 'Genen Shouyu',
      'QuantityPerUnit': '24 - 250 ml bottles',
      'UnitsInStock': 39,
      'Discontinued': false
    },
    {
      'CategoryName': 'Condiments',
      'ProductName': 'Grandma\'s Boysenberry Spread',
      'QuantityPerUnit': '12 - 8 oz jars',
      'UnitsInStock': 120,
      'Discontinued': false
    },
  ],

```

```

{
  'CategoryName': 'Condiments',
  'ProductName': 'Gula Malacca',
  'QuantityPerUnit': '20 - 2 kg bags',
  'UnitsInStock': 27,
  'Discontinued': true
},
{
  'CategoryName': 'Condiments',
  'ProductName': 'Louisiana Fiery Hot Pepper Sauce',
  'QuantityPerUnit': '32 - 8 oz bottles',
  'UnitsInStock': 76,
  'Discontinued': false
},
{
  'CategoryName': 'Condiments',
  'ProductName': 'Louisiana Hot Spiced Okra',
  'QuantityPerUnit': '24 - 8 oz jars',
  'UnitsInStock': 4,
  'Discontinued': false
},
{
  'CategoryName': 'Condiments',
  'ProductName': 'Northwoods Cranberry Sauce',
  'QuantityPerUnit': '12 - 12 oz jars',
  'UnitsInStock': 6,
  'Discontinued': true
},
{
  'CategoryName': 'Condiments',
  'ProductName': 'Original Frankfurter gr\u00fcne So\u00dfe',
  'QuantityPerUnit': '12 boxes',
  'UnitsInStock': 32,
  'Discontinued': false
},
{
  'CategoryName': 'Condiments',
  'ProductName': 'Sirop d'\u00e9rable',
  'QuantityPerUnit': '24 - 500 ml bottles',
  'UnitsInStock': 113,
  'Discontinued': false
},
{
  'CategoryName': 'Condiments',
  'ProductName': 'Vegie-spread',
  'QuantityPerUnit': '15 - 625 g jars',
  'UnitsInStock': 24,
  'Discontinued': false
},
{
  'CategoryName': 'Confections',
  'ProductName': 'Chocolate',
  'QuantityPerUnit': '10 pkgs.',
  'UnitsInStock': 15,
  'Discontinued': true
},
{

```

```

    'CategoryName': 'Confections',
    'ProductName': 'Gumb\u00e4r Gummib\u00e4rchen',
    'QuantityPerUnit': '100 - 250 g bags',
    'UnitsInStock': 15,
    'Discontinued': false
  },
  {
    'CategoryName': 'Confections',
    'ProductName': 'Maxilaku',
    'QuantityPerUnit': '24 - 50 g pkgs.',
    'UnitsInStock': 10,
    'Discontinued': false
  },
  {
    'CategoryName': 'Confections',
    'ProductName': 'NuNuCa Nu\u00df-Nougat-Creme',
    'QuantityPerUnit': '20 - 450 g glasses',
    'UnitsInStock': 76,
    'Discontinued': true
  },
  {
    'CategoryName': 'Confections',
    'ProductName': 'Pavlova',
    'QuantityPerUnit': '32 - 500 g boxes',
    'UnitsInStock': 29,
    'Discontinued': true
  },
  {
    'CategoryName': 'Confections',
    'ProductName': 'Schoggi Schokolade',
    'QuantityPerUnit': '100 - 100 g pieces',
    'UnitsInStock': 49,
    'Discontinued': true
  },
  {
    'CategoryName': 'Confections',
    'ProductName': 'Scottish Longbreads',
    'QuantityPerUnit': '10 boxes x 8 pieces',
    'UnitsInStock': 6,
    'Discontinued': true
  },
  {
    'CategoryName': 'Confections',
    'ProductName': 'Sir Rodney\'s Marmalade',
    'QuantityPerUnit': '30 gift boxes',
    'UnitsInStock': 40,
    'Discontinued': true
  },
  {
    'CategoryName': 'Confections',
    'ProductName': 'Sir Rodney\'s Scones',
    'QuantityPerUnit': '24 pkgs. x 4 pieces',
    'UnitsInStock': 3,
    'Discontinued': true
  },
  {
    'CategoryName': 'Confections',

```

```

    'ProductName': 'Tarte au sucre',
    'QuantityPerUnit': '48 pies',
    'UnitsInStock': 17,
    'Discontinued': false
  },
  {
    'CategoryName': 'Confections',
    'ProductName': 'Teatime Chocolate Biscuits',
    'QuantityPerUnit': '10 boxes x 12 pieces',
    'UnitsInStock': 25,
    'Discontinued': false
  },
  {
    'CategoryName': 'Confections',
    'ProductName': 'Valkoinen suklaa',
    'QuantityPerUnit': '12 - 100 g bars',
    'UnitsInStock': 65,
    'Discontinued': false
  },
  {
    'CategoryName': 'Confections',
    'ProductName': 'Zaanse koeken',
    'QuantityPerUnit': '10 - 4 oz boxes',
    'UnitsInStock': 36,
    'Discontinued': true
  },
  {
    'CategoryName': 'Dairy Products',
    'ProductName': 'Camembert Pierrot',
    'QuantityPerUnit': '15 - 300 g rounds',
    'UnitsInStock': 19,
    'Discontinued': false
  },
  {
    'CategoryName': 'Dairy Products',
    'ProductName': 'Flotemysost',
    'QuantityPerUnit': '10 - 500 g pkgs.',
    'UnitsInStock': 26,
    'Discontinued': false
  },
  {
    'CategoryName': 'Dairy Products',
    'ProductName': 'Geitost',
    'QuantityPerUnit': '500 g',
    'UnitsInStock': 112,
    'Discontinued': false
  },
  {
    'CategoryName': 'Dairy Products',
    'ProductName': 'Gorgonzola Telino',
    'QuantityPerUnit': '12 - 100 g pkgs',
    'UnitsInStock': 0,
    'Discontinued': true
  },
  {
    'CategoryName': 'Dairy Products',
    'ProductName': 'Gudbrandsdalsost',

```



```

    'QuantityPerUnit': '10 kg pkg.',
    'UnitsInStock': 26,
    'Discontinued': true
  },
  {
    'CategoryName': 'Dairy Products',
    'ProductName': 'Mascarpone Fabioli',
    'QuantityPerUnit': '24 - 200 g pkgs.',
    'UnitsInStock': 9,
    'Discontinued': true
  },
  {
    'CategoryName': 'Dairy Products',
    'ProductName': 'Mozzarella di Giovanni',
    'QuantityPerUnit': '24 - 200 g pkgs.',
    'UnitsInStock': 14,
    'Discontinued': true
  },
  {
    'CategoryName': 'Dairy Products',
    'ProductName': 'Queso Cabrales',
    'QuantityPerUnit': '1 kg pkg.',
    'UnitsInStock': 22,
    'Discontinued': true
  },
  {
    'CategoryName': 'Dairy Products',
    'ProductName': 'Queso Manchego La Pastora',
    'QuantityPerUnit': '10 - 500 g pkgs.',
    'UnitsInStock': 86,
    'Discontinued': true
  },
  {
    'CategoryName': 'Dairy Products',
    'ProductName': 'Raclette Courdavault',
    'QuantityPerUnit': '5 kg pkg.',
    'UnitsInStock': 79,
    'Discontinued': false
  },
  {
    'CategoryName': 'Grains/Cereals',
    'ProductName': 'Filo Mix',
    'QuantityPerUnit': '16 - 2 kg boxes',
    'UnitsInStock': 38,
    'Discontinued': false
  },
  {
    'CategoryName': 'Grains/Cereals',
    'ProductName': 'Gnocchi di nonna Alice',
    'QuantityPerUnit': '24 - 250 g pkgs.',
    'UnitsInStock': 21,
    'Discontinued': false
  },
  {
    'CategoryName': 'Grains/Cereals',
    'ProductName': 'Gustaf\'s Kn\u00e4ckebr\u00f6d',
    'QuantityPerUnit': '24 - 500 g pkgs.',

```

```

    'UnitsInStock': 104,
    'Discontinued': true
  },
  {
    'CategoryName': 'Grains/Cereals',
    'ProductName': 'Ravioli Angelo',
    'QuantityPerUnit': '24 - 250 g pkgs.',
    'UnitsInStock': 36,
    'Discontinued': true
  },
  {
    'CategoryName': 'Grains/Cereals',
    'ProductName': 'Tunnbr\u00f6d',
    'QuantityPerUnit': '12 - 250 g pkgs.',
    'UnitsInStock': 61,
    'Discontinued': true
  },
  {
    'CategoryName': 'Grains/Cereals',
    'ProductName': 'Wimmers gute Semmelkn\u00f6del',
    'QuantityPerUnit': '20 bags x 4 pieces',
    'UnitsInStock': 22,
    'Discontinued': true
  },
  {
    'CategoryName': 'Meat/Poultry',
    'ProductName': 'P\u00e4\u00e7\u00e9 chinois',
    'QuantityPerUnit': '24 boxes x 2 pies',
    'UnitsInStock': 115,
    'Discontinued': false
  },
  {
    'CategoryName': 'Meat/Poultry',
    'ProductName': 'Tourti\u00e8re',
    'QuantityPerUnit': '16 pies',
    'UnitsInStock': 21,
    'Discontinued': false
  },
  {
    'CategoryName': 'Produce',
    'ProductName': 'Longlife Tofu',
    'QuantityPerUnit': '5 kg pkg.',
    'UnitsInStock': 4,
    'Discontinued': false
  },
  {
    'CategoryName': 'Produce',
    'ProductName': 'Manjimup Dried Apples',
    'QuantityPerUnit': '50 - 300 g pkgs.',
    'UnitsInStock': 20,
    'Discontinued': false
  },
  {
    'CategoryName': 'Produce',
    'ProductName': 'Tofu',
    'QuantityPerUnit': '40 - 100 g pkgs.',
    'UnitsInStock': 35,

```

```

    'Discontinued': true
  },
  {
    'CategoryName': 'Produce',
    'ProductName': 'Uncle Bob\'s Organic Dried Pears',
    'QuantityPerUnit': '12 - 1 lb pkgs.',
    'UnitsInStock': 15,
    'Discontinued': true
  },
  {
    'CategoryName': 'Seafood',
    'ProductName': 'Boston Crab Meat',
    'QuantityPerUnit': '24 - 4 oz tins',
    'UnitsInStock': 123,
    'Discontinued': true
  },
  {
    'CategoryName': 'Seafood',
    'ProductName': 'Carnarvon Tigers',
    'QuantityPerUnit': '16 kg pkg.',
    'UnitsInStock': 42,
    'Discontinued': true
  },
  {
    'CategoryName': 'Seafood',
    'ProductName': 'Escargots de Bourgogne',
    'QuantityPerUnit': '24 pieces',
    'UnitsInStock': 62,
    'Discontinued': true
  },
  {
    'CategoryName': 'Seafood',
    'ProductName': 'Gravad lax',
    'QuantityPerUnit': '12 - 500 g pkgs.',
    'UnitsInStock': 11,
    'Discontinued': true
  },
  {
    'CategoryName': 'Seafood',
    'ProductName': 'Ikura',
    'QuantityPerUnit': '12 - 200 ml jars',
    'UnitsInStock': 31,
    'Discontinued': false
  },
  {
    'CategoryName': 'Seafood',
    'ProductName': 'Inlagd Sill',
    'QuantityPerUnit': '24 - 250 g jars',
    'UnitsInStock': 112,
    'Discontinued': false
  },
  {
    'CategoryName': 'Seafood',
    'ProductName': 'Jack\'s New England Clam Chowder',
    'QuantityPerUnit': '12 - 12 oz cans',
    'UnitsInStock': 85,
    'Discontinued': false
  }

```

```

    },
    {
      'CategoryName': 'Seafood',
      'ProductName': 'Konbu',
      'QuantityPerUnit': '2 kg box',
      'UnitsInStock': 24,
      'Discontinued': false
    },
    {
      'CategoryName': 'Seafood',
      'ProductName': 'Nord-Ost Matjeshering',
      'QuantityPerUnit': '10 - 200 g glasses',
      'UnitsInStock': 10,
      'Discontinued': false
    },
    {
      'CategoryName': 'Seafood',
      'ProductName': 'R\u00f6d Kavlar',
      'QuantityPerUnit': '24 - 150 g jars',
      'UnitsInStock': 101,
      'Discontinued': false
    },
    {
      'CategoryName': 'Seafood',
      'ProductName': 'Rogede sild',
      'QuantityPerUnit': '1k pkg.',
      'UnitsInStock': 5,
      'Discontinued': false
    },
    {
      'CategoryName': 'Seafood',
      'ProductName': 'Spegesild',
      'QuantityPerUnit': '4 - 450 g glasses',
      'UnitsInStock': 95,
      'Discontinued': true
    }
  ];
export const customerData: Object[] = [
  {
    'CustomerID': 'ALFKI',
    'ContactName': 'Maria ',
    'CompanyName': 'Alfreds Futterkiste',
    'Address': 'Obere Str. 57',
    'Country': 'Germany'
  },
  {
    'CustomerID': 'ANATR',
    'ContactName': 'Ana Trujillo',
    'CompanyName': 'Ana Trujillo Emparedados y helados',
    'Address': 'Avda. de la Constitución 2222',
    'Country': 'Mexico'
  },
  {
    'CustomerID': 'ANTON',
    'ContactName': 'Antonio Moreno',
    'CompanyName': 'Antonio Moreno Taquería',
    'Address': 'Mataderos 2312',
  }
];

```

```

    'Country': 'Mexico'
  },
  {
    'CustomerID': 'AROUT',
    'ContactName': 'Thomas Hardy',
    'CompanyName': 'Around the Horn',
    'Address': '120 Hanover Sq.',
    'Country': 'UK'
  },
  {
    'CustomerID': 'BERGS',
    'ContactName': 'Christina Berglund',
    'CompanyName': 'Berglunds snabbköp',
    'Address': 'Berguvsvägen 8',
    'Country': 'Sweden'
  },
  {
    'CustomerID': 'BLAUS',
    'ContactName': 'Hanna Moos',
    'CompanyName': 'Blauer See Delikatessen',
    'Address': 'Forsterstr. 57',
    'Country': 'Germany'
  },
  {
    'CustomerID': 'BLONP',
    'ContactName': 'Frédérique Citeaux',
    'CompanyName': 'Blondesddsl père et fils',
    'Address': '24, place Kléber',
    'Country': 'France'
  },
  {
    'CustomerID': 'BOLID',
    'ContactName': 'Martín Sommer',
    'CompanyName': 'Bólido Comidas preparadas',
    'Address': 'C/ Araquil, 67',
    'Country': 'Spain'
  },
  {
    'CustomerID': 'BONAP',
    'ContactName': 'Laurence Lebihan',
    'CompanyName': 'Bon app',
    'Address': '12, rue des Bouchers',
    'Country': 'France'
  },
  {
    'CustomerID': 'BOTTM',
    'ContactName': 'Elizabeth Lincoln',
    'CompanyName': 'Bottom-Dollar Markets',
    'Address': '23 Tsawassen Blvd.',
    'Country': 'Canada'
  },
  {
    'CustomerID': 'BSBEV',
    'ContactName': 'Victoria Ashworth',
    'CompanyName': 'B\'s Beverages',
    'Address': 'Fauntleroy Circus',
    'Country': 'UK'
  }

```

```

    },
    {
      'CustomerID': 'CACTU',
      'ContactName': 'Patricio Simpson',
      'CompanyName': 'Cactus Comidas para llevar',
      'Address': 'Cerrito 333',
      'Country': 'Argentina'
    },
    {
      'CustomerID': 'CENTC',
      'ContactName': 'Francisco Chang',
      'CompanyName': 'Centro comercial Moctezuma',
      'Address': 'Sierras de Granada 9993',
      'Country': 'Mexico'
    },
    {
      'CustomerID': 'CHOPS',
      'ContactName': 'Yang Wang',
      'CompanyName': 'Chop-suey Chinese',
      'Address': 'Hauptstr. 29',
      'Country': 'Switzerland'
    },
    {
      'CustomerID': 'COMMI',
      'ContactName': 'Pedro Afonso',
      'CompanyName': 'Comércio Mineiro',
      'Address': 'Av. dos Lusíadas, 23',
      'Country': 'Brazil'
    },
    {
      'CustomerID': 'CONSH',
      'ContactName': 'Elizabeth Brown',
      'CompanyName': 'Consolidated Holdings',
      'Address': 'Berkeley Gardens 12 Brewery',
      'Country': 'UK'
    },
    {
      'CustomerID': 'DRACD',
      'ContactName': 'Sven Ottlieb',
      'CompanyName': 'Drachenblut Delikatessen',
      'Address': 'Walserweg 21',
      'Country': 'Germany'
    },
    {
      'CustomerID': 'DUMON',
      'ContactName': 'Janine Labrune',
      'CompanyName': 'Du monde entier',
      'Address': '67, rue des Cinquante Otages',
      'Country': 'France'
    },
    {
      'CustomerID': 'EASTC',
      'ContactName': 'Ann Devon',
      'CompanyName': 'Eastern Connection',
      'Address': '35 King George',
      'Country': 'UK'
    },
  ],

```

```

{
  'CustomerID': 'ERNSH',
  'ContactName': 'Roland Mendel',
  'CompanyName': 'Ernst Handel',
  'Address': 'Kirchgasse 6',
  'Country': 'Austria'
},
{
  'CustomerID': 'FAMIA',
  'ContactName': 'Aria Cruz',
  'CompanyName': 'Familia Arquibaldo',
  'Address': 'Rua Orós, 92',
  'Country': 'Brazil'
},
{
  'CustomerID': 'FISSA',
  'ContactName': 'Diego Roel',
  'CompanyName': 'FISSA Fabrica Inter. Salchichas S.A.',
  'Address': 'C/ Moralzarzal, 86',
  'Country': 'Spain'
},
{
  'CustomerID': 'FOLIG',
  'ContactName': 'Martine Rancé',
  'CompanyName': 'Folies gourmandes',
  'Address': '184, chaussée de Tournai',
  'Country': 'France'
},
{
  'CustomerID': 'FOLKO',
  'ContactName': 'Maria Larsson',
  'CompanyName': 'Folk och fä HB',
  'Address': 'Åkergatan 24',
  'Country': 'Sweden'
},
{
  'CustomerID': 'FRANK',
  'ContactName': 'Peter Franken',
  'CompanyName': 'Frankenversand',
  'Address': 'Berliner Platz 43',
  'Country': 'Germany'
},
{
  'CustomerID': 'FRANR',
  'ContactName': 'Carine Schmitt',
  'CompanyName': 'France restauration',
  'Address': '54, rue Royale',
  'Country': 'France'
},
{
  'CustomerID': 'FRANS',
  'ContactName': 'Paolo Accorti',
  'CompanyName': 'Franchi S.p.A.',
  'Address': 'Via Monte Bianco 34',
  'Country': 'Italy'
},
{

```

```

    'CustomerID': 'FURIB',
    'ContactName': 'Lino Rodriguez',
    'CompanyName': 'Furia Bacalhau e Frutos do Mar',
    'Address': 'Jardim das rosas n. 32',
    'Country': 'Portugal'
  },
  {
    'CustomerID': 'GALED',
    'ContactName': 'Eduardo Saavedra',
    'CompanyName': 'Galería del gastrónomo',
    'Address': 'Rambla de Cataluña, 23',
    'Country': 'Spain'
  },
  {
    'CustomerID': 'GODOS',
    'ContactName': 'José Pedro Freyre',
    'CompanyName': 'Godos Cocina Típica',
    'Address': 'C/ Romero, 33',
    'Country': 'Spain'
  },
  {
    'CustomerID': 'GOURL',
    'ContactName': 'André Fonseca',
    'CompanyName': 'Gourmet Lanchonetes',
    'Address': 'Av. Brasil, 442',
    'Country': 'Brazil'
  },
  {
    'CustomerID': 'GREAL',
    'ContactName': 'Howard Snyder',
    'CompanyName': 'Great Lakes Food Market',
    'Address': '2732 Baker Blvd.',
    'Country': 'USA'
  },
  {
    'CustomerID': 'GROSR',
    'ContactName': 'Manuel Pereira',
    'CompanyName': 'GROSELLA-Restaurante',
    'Address': '5ª Ave. Los Palos Grandes',
    'Country': 'Venezuela'
  },
  {
    'CustomerID': 'HANAR',
    'ContactName': 'Mario Pontes',
    'CompanyName': 'Hanari Carnes',
    'Address': 'Rua do Paço, 67',
    'Country': 'Brazil'
  },
  {
    'CustomerID': 'HILAA',
    'ContactName': 'Carlos Hernández',
    'CompanyName': 'HILARION-Abastos',
    'Address': 'Carrera 22 con Ave. Carlos Soublette #8-35',
    'Country': 'Venezuela'
  },
  {
    'CustomerID': 'HUNGC',

```



```

    'ContactName': 'Yoshi Latimer',
    'CompanyName': 'Hungry Coyote Import Store',
    'Address': 'City Center Plaza 516 Main St.',
    'Country': 'USA'
  },
  {
    'CustomerID': 'HUNGO',
    'ContactName': 'Patricia McKenna',
    'CompanyName': 'Hungry Owl All-Night Grocers',
    'Address': '8 Johnstown Road',
    'Country': 'Ireland'
  },
  {
    'CustomerID': 'ISLAT',
    'ContactName': 'Helen Bennett',
    'CompanyName': 'Island Trading',
    'Address': 'Garden House Crowther Way',
    'Country': 'UK'
  },
  {
    'CustomerID': 'KOENE',
    'ContactName': 'Philip Cramer',
    'CompanyName': 'Königlich Essen',
    'Address': 'Maubelstr. 90',
    'Country': 'Germany'
  },
  {
    'CustomerID': 'LACOR',
    'ContactName': 'Daniel Tonini',
    'CompanyName': 'La corne d\'abondance',
    'Address': '67, avenue de l\'Europe',
    'Country': 'France'
  },
  {
    'CustomerID': 'LAMAI',
    'ContactName': 'Annette Roulet',
    'CompanyName': 'La maison d\'Asie',
    'Address': '1 rue Alsace-Lorraine',
    'Country': 'France'
  },
  {
    'CustomerID': 'LAUGB',
    'ContactName': 'Yoshi Tannamuri',
    'CompanyName': 'Laughing Bacchus Wine Cellars',
    'Address': '1900 Oak St.',
    'Country': 'Canada'
  },
  {
    'CustomerID': 'LAZYK',
    'ContactName': 'John Steel',
    'CompanyName': 'Lazy K Kountry Store',
    'Address': '12 Orchestra Terrace',
    'Country': 'USA'
  },
  {
    'CustomerID': 'LEHMS',
    'ContactName': 'Renate Messner',

```

```

    'CompanyName': 'Lehmanns Marktstand',
    'Address': 'Magazinweg 7',
    'Country': 'Germany'
  },
  {
    'CustomerID': 'LETSS',
    'ContactName': 'Jaime Yorres',
    'CompanyName': 'Let\'s Stop N Shop',
    'Address': '87 Polk St. Suite 5',
    'Country': 'USA'
  },
  {
    'CustomerID': 'LILAS',
    'ContactName': 'Carlos González',
    'CompanyName': 'LILA-Supermercado',
    'Address': 'Carrera 52 con Ave. Bolívar #65-98 Llano Largo',
    'Country': 'Venezuela'
  },
  {
    'CustomerID': 'LINOD',
    'ContactName': 'Felipe Izquierdo',
    'CompanyName': 'LINO-Delicateses',
    'Address': 'Ave. 5 de Mayo Porlamar',
    'Country': 'Venezuela'
  },
  {
    'CustomerID': 'LONEP',
    'ContactName': 'Fran Wilson',
    'CompanyName': 'Lonesome Pine Restaurant',
    'Address': '89 Chiaroscuro Rd.',
    'Country': 'USA'
  },
  {
    'CustomerID': 'MAGAA',
    'ContactName': 'Giovanni Rovelli',
    'CompanyName': 'Magazzini Alimentari Riuniti',
    'Address': 'Via Ludovico il Moro 22',
    'Country': 'Italy'
  },
  {
    'CustomerID': 'MAISD',
    'ContactName': 'Catherine Dewey',
    'CompanyName': 'Maison Dewey',
    'Address': 'Rue Joseph-Bens 532',
    'Country': 'Belgium'
  },
  {
    'CustomerID': 'MEREP',
    'ContactName': 'Jean Fresnière',
    'CompanyName': 'Mère Paillarde',
    'Address': '43 rue St. Laurent',
    'Country': 'Canada'
  },
  {
    'CustomerID': 'MORGK',
    'ContactName': 'Alexander Feuer',
    'CompanyName': 'Morgenstern Gesundkost',

```

```

      'Address': 'Heerstr. 22',
      'Country': 'Germany'
    },
    {
      'CustomerID': 'NORTS',
      'ContactName': 'Simon Crowther',
      'CompanyName': 'North/South',
      'Address': 'South House 300 Queensbridge',
      'Country': 'UK'
    },
    {
      'CustomerID': 'OCEAN',
      'ContactName': 'Yvonne Moncada',
      'CompanyName': 'Océano Atlántico Ltda.',
      'Address': 'Ing. Gustavo Moncada 8585 Piso 20-A',
      'Country': 'Argentina'
    },
    {
      'CustomerID': 'OLDWO',
      'ContactName': 'Rene Phillips',
      'CompanyName': 'Old World Delicatessen',
      'Address': '2743 Bering St.',
      'Country': 'USA'
    },
    {
      'CustomerID': 'OTTIK',
      'ContactName': 'Henriette Pfalzheim',
      'CompanyName': 'Ottilies Käseladen',
      'Address': 'Mehrheimerstr. 369',
      'Country': 'Germany'
    },
    {
      'CustomerID': 'PARIS',
      'ContactName': 'Marie Bertrand',
      'CompanyName': 'Paris spécialités',
      'Address': '265, boulevard Charonne',
      'Country': 'France'
    },
    {
      'CustomerID': 'PERIC',
      'ContactName': 'Guillermo Fernández',
      'CompanyName': 'Pericles Comidas clásicas',
      'Address': 'Calle Dr. Jorge Cash 321',
      'Country': 'Mexico'
    },
    {
      'CustomerID': 'PICCO',
      'ContactName': 'Georg Pipps',
      'CompanyName': 'Piccolo und mehr',
      'Address': 'Geislweg 14',
      'Country': 'Austria'
    },
    {
      'CustomerID': 'PRINI',
      'ContactName': 'Isabel de Castro',
      'CompanyName': 'Princesa Isabel Vinhos',
      'Address': 'Estrada da saúde n. 58',

```

```

    'Country': 'Portugal'
  },
  {
    'CustomerID': 'QUEDE',
    'ContactName': 'Bernardo Batista',
    'CompanyName': 'Que Delícia',
    'Address': 'Rua da Panificadora, 12',
    'Country': 'Brazil'
  },
  {
    'CustomerID': 'QUEEN',
    'ContactName': 'Lúcia Carvalho',
    'CompanyName': 'Queen Cozinha',
    'Address': 'Alameda dos Canários, 891',
    'Country': 'Brazil'
  },
  {
    'CustomerID': 'QUICK',
    'ContactName': 'Horst Kloss',
    'CompanyName': 'QUICK-Stop',
    'Address': 'Taucherstraße 10',
    'Country': 'Germany'
  },
  {
    'CustomerID': 'RANCH',
    'ContactName': 'Sergio Gutiérrez',
    'CompanyName': 'Rancho grande',
    'Address': 'Av. del Libertador 900',
    'Country': 'Argentina'
  },
  {
    'CustomerID': 'RATTC',
    'ContactName': 'Paula Wilson',
    'CompanyName': 'Rattlesnake Canyon Grocery',
    'Address': '2817 Milton Dr.',
    'Country': 'USA'
  },
  {
    'CustomerID': 'REGGC',
    'ContactName': 'Maurizio Moroni',
    'CompanyName': 'Reggiani Caseifici',
    'Address': 'Strada Provinciale 124',
    'Country': 'Italy'
  },
  {
    'CustomerID': 'RICAR',
    'ContactName': 'Janete Limeira',
    'CompanyName': 'Ricardo Adocicados',
    'Address': 'Av. Copacabana, 267',
    'Country': 'Brazil'
  },
  {
    'CustomerID': 'RICSU',
    'ContactName': 'Michael Holz',
    'CompanyName': 'Richter Supermarkt',
    'Address': 'Grenzacherweg 237',
    'Country': 'Switzerland'
  }

```

```

    },
    {
      'CustomerID': 'ROMEY',
      'ContactName': 'Alejandra Camino',
      'CompanyName': 'Romero y tomillo',
      'Address': 'Gran Vía, 1',
      'Country': 'Spain'
    },
    {
      'CustomerID': 'SANTG',
      'ContactName': 'Jonas Bergulfsen',
      'CompanyName': 'Santé Gourmet',
      'Address': 'Erling Skakkes gate 78',
      'Country': 'Norway'
    },
    {
      'CustomerID': 'SAVEA',
      'ContactName': 'Jose Pavarotti',
      'CompanyName': 'Save-a-lot Markets',
      'Address': '187 Suffolk Ln.',
      'Country': 'USA'
    },
    {
      'CustomerID': 'SEVES',
      'ContactName': 'Hari Kumar',
      'CompanyName': 'Seven Seas Imports',
      'Address': '90 Wadhurst Rd.',
      'Country': 'UK'
    },
    {
      'CustomerID': 'SIMOB',
      'ContactName': 'Jytte Petersen',
      'CompanyName': 'Simons bistro',
      'Address': 'Vinbæltet 34',
      'Country': 'Denmark'
    },
    {
      'CustomerID': 'SPECB',
      'ContactName': 'Dominique Perrier',
      'CompanyName': 'Spécialités du monde',
      'Address': '25, rue Lauriston',
      'Country': 'France'
    },
    {
      'CustomerID': 'SPLIR',
      'ContactName': 'Art Braunschweiger',
      'CompanyName': 'Split Rail Beer & Ale',
      'Address': 'P.O. Box 555',
      'Country': 'USA'
    },
    {
      'CustomerID': 'SUPRD',
      'ContactName': 'Pascale Cartrain',
      'CompanyName': 'Suprêmes délices',
      'Address': 'Boulevard Tirou, 255',
      'Country': 'Belgium'
    },
  ],

```

```

{
  'CustomerID': 'THEBI',
  'ContactName': 'Liz Nixon',
  'CompanyName': 'The Big Cheese',
  'Address': '89 Jefferson Way Suite 2',
  'Country': 'USA'
},
{
  'CustomerID': 'THECR',
  'ContactName': 'Liu Wong',
  'CompanyName': 'The Cracker Box',
  'Address': '55 Grizzly Peak Rd.',
  'Country': 'USA'
},
{
  'CustomerID': 'TOMSP',
  'ContactName': 'Karin Josephs',
  'CompanyName': 'Toms Spezialitäten',
  'Address': 'Luisenstr. 48',
  'Country': 'Germany'
},
{
  'CustomerID': 'TORTU',
  'ContactName': 'Miguel Angel Paolino',
  'CompanyName': 'Tortuga Restaurante',
  'Address': 'Avda. Azteca 123',
  'Country': 'Mexico'
},
{
  'CustomerID': 'TRADH',
  'ContactName': 'Anabela Domingues',
  'CompanyName': 'Tradição Hipermercados',
  'Address': 'Av. Inês de Castro, 414',
  'Country': 'Brazil'
},
{
  'CustomerID': 'TRAIH',
  'ContactName': 'Helvetius Nagy',
  'CompanyName': 'Trail\'s Head Gourmet Provisioners',
  'Address': '722 DaVinci Blvd.',
  'Country': 'USA'
},
{
  'CustomerID': 'VAFFE',
  'ContactName': 'Palle Ibsen',
  'CompanyName': 'Vaffeljernet',
  'Address': 'Smagsloget 45',
  'Country': 'Denmark'
},
{
  'CustomerID': 'VICTE',
  'ContactName': 'Mary Saveley',
  'CompanyName': 'Victuailles en stock',
  'Address': '2, rue du Commerce',
  'Country': 'France'
},
{

```

```

        'CustomerID': 'VINET',
        'ContactName': 'Paul Henriot',
        'CompanyName': 'Vins et alcools Chevalier',
        'Address': '59 rue de l\'Abbaye',
        'Country': 'France'
    },
    {
        'CustomerID': 'WANDK',
        'ContactName': 'Rita Müller',
        'CompanyName': 'Die Wandernde Kuh',
        'Address': 'Adenauerallee 900',
        'Country': 'Germany'
    },
    {
        'CustomerID': 'WARTH',
        'ContactName': 'Pirkko Koskitalo',
        'CompanyName': 'Wartian Herkku',
        'Address': 'Torikatu 38',
        'Country': 'Finland'
    },
    {
        'CustomerID': 'WELLI',
        'ContactName': 'Paula Parente',
        'CompanyName': 'Wellington Importadora',
        'Address': 'Rua do Mercado, 12',
        'Country': 'Brazil'
    },
    {
        'CustomerID': 'WHITC',
        'ContactName': 'Karl Jablonski',
        'CompanyName': 'White Clover Markets',
        'Address': '305 - 14th Ave. S. Suite 3B',
        'Country': 'USA'
    },
    {
        'CustomerID': 'WILMK',
        'ContactName': 'Matti Karttunen',
        'CompanyName': 'Wilman Kala',
        'Address': 'Keskuskatu 45',
        'Country': 'Finland'
    },
    {
        'CustomerID': 'WOLZA',
        'ContactName': 'Zbyszek Piestrzeniewicz',
        'CompanyName': 'Wolski Zajazd',
        'Address': 'ul. Filtrowa 68',
        'Country': 'Poland'
    }
];
type cType = { CustomerID: string, ContactName: string, CustomerName: string };
export const data: Object[] = orderData.map((item: cType) => {
    let name: cType = (<cType[]>customerData).filter((cItem: cType) => {
        return cItem.CustomerID === item.CustomerID;
    })[0];
    item.CustomerName = (name || <cType>{}).ContactName;
    return item;
});

```

```

});
export const inventoryData: Object[] = [
  {
    'Inventor': 'Kia Silverbrook',
    'NumberofPatentFamilies': 4737,
    'Country': 'Australia',
    'Number of INPADOC patents': 9839,
    'Active': '1994-2016',
    'Mainfieldsofinvention': 'Printing, Digital paper, Internet,
Electronics, Lab-on-a-chip, MEMS, Mechanical, VLSI',
  },
  {
    'Inventor': 'Shunpei Yamazaki',
    'NumberofPatentFamilies': 4677,
    'Country': 'Japan',
    'Number of INPADOC patents': '10000+',
    'Active': '1976-2016',
    'Mainfieldsofinvention': 'Thin film transistors, Liquid crystal
displays, Solar cells, Flash memory, OLED',
  },
  {
    'Inventor': 'Lowell L. Wood, Jr.',
    'NumberofPatentFamilies': 1419,
    'Country': 'USA',
    'Number of INPADOC patents': 1332,
    'Active': '1977-2016',
    'Mainfieldsofinvention': 'Mosquito laser, Nuclear weapons',
  },
  {
    'Inventor': 'Paul Lapstun',
    'NumberofPatentFamilies': 1281,
    'Country': 'Australia',
    'Number of INPADOC patents': 3099,
    'Active': '2000-2016',
    'Mainfieldsofinvention': 'Printing, Digital paper, Internet,
Electronics, CGI, VLSI',
  },
  {
    'Inventor': 'Gurtej Sandhu',
    'NumberofPatentFamilies': 1255,
    'Country': 'India',
    'Number of INPADOC patents': 2038,
    'Active': '1991-2016',
    'Mainfieldsofinvention': 'Thin film processes and materials, VLSI,
Semiconductor device fabrication',
  },
  {
    'Inventor': 'Jun Koyama',
    'NumberofPatentFamilies': 1240,
    'Country': 'Japan',
    'Number of INPADOC patents': 4126,
    'Active': '1991-2016',
    'Mainfieldsofinvention': 'Thin film transistors, Liquid crystal
displays, OLED',
  },
  {
    'Inventor': 'Roderick A. Hyde',

```



```

    'NumberofPatentFamilies': 1240,
    'Country': 'USA',
    'Number of INPADOC patents': 3360,
    'Active': '2001-2016',
    'Mainfieldsofinvention': 'Various',
  },
  {
    'Inventor': 'Leonard Forbes',
    'NumberofPatentFamilies': 1093,
    'Country': 'Canada',
    'Number of INPADOC patents': 1398,
    'Active': '1991-2016',
    'Mainfieldsofinvention': 'Semiconductor Memories, CCDs, Thin film
processes and materials, VLSI',
  },
  {
    'Inventor': 'Thomas Edison',
    'NumberofPatentFamilies': 1084,
    'Country': 'USA',
    'Number of INPADOC patents': 2332,
    'Active': '1847(b)-1931(d)',
    'Mainfieldsofinvention': 'Electric power, Lighting, Batteries,
Phonograph, Cement, Telegraphy, Mining',
  },
  {
    'Inventor': 'Donald E. Weder',
    'NumberofPatentFamilies': 999,
    'Country': 'USA',
    'Number of INPADOC patents': 1993,
    'Active': '1976-2015',
    'Mainfieldsofinvention': 'Florist supplies',
  },
  {
    'Inventor': 'George Albert Lyon',
    'NumberofPatentFamilies': 993,
    'Country': 'Canada',
    'Number of INPADOC patents': 'NA',
    'Active': '1882(b)-1961(d)',
    'Mainfieldsofinvention': 'Automotive, Stainless steel products',
  },
  {
    'Inventor': 'John F. O\'Connor',
    'NumberofPatentFamilies': 949,
    'Country': 'USA',
    'Number of INPADOC patents': 'NA',
    'Active': '1864(b)-1938(d)',
    'Mainfieldsofinvention': 'Railway draft gearing',
  },
  {
    'Inventor': 'Melvin De Groote',
    'NumberofPatentFamilies': 925,
    'Country': 'USA',
    'Number of INPADOC patents': 'NA',
    'Active': '1895(b)-1963(d)',
    'Mainfieldsofinvention': 'Chemical de-emulsifiers',
  },
  {

```

```

    'Inventor': 'Jay S. Walker',
    'NumberofPatentFamilies': 918,
    'Country': 'USA',
    'Number of INPADOC patents': 2206,
    'Active': '1998-2016',
    'Mainfieldsofinvention': 'Gaming machines',
  },
  {
    'Inventor': 'Edward K. Y. Jung',
    'NumberofPatentFamilies': 911,
    'Country': 'USA',
    'Number of INPADOC patents': 2254,
    'Active': '1996-2016',
    'Mainfieldsofinvention': 'Various',
  },
  {
    'Inventor': 'Francis H. Richards',
    'NumberofPatentFamilies': 894,
    'Country': 'USA',
    'Number of INPADOC patents': 'NA',
    'Active': '1850(b)-19??(d)',
    'Mainfieldsofinvention': 'Mechanical, automation',
  },
  {
    'Inventor': 'Kangguo Cheng',
    'NumberofPatentFamilies': 884,
    'Country': 'USA',
    'Number of INPADOC patents': 1314,
    'Active': '2004-2016',
    'Mainfieldsofinvention': 'Semiconductor device fabrication,
Semiconductor memory, Semiconductor device',
  },
  {
    'Inventor': 'Clarence T. Tegreene',
    'NumberofPatentFamilies': 872,
    'Country': 'USA',
    'Number of INPADOC patents': 2255,
    'Active': '2000-2016',
    'Mainfieldsofinvention': 'Various',
  },
  {
    'Inventor': 'Ahmadreza Rofougaran',
    'NumberofPatentFamilies': 808,
    'Country': 'USA',
    'Number of INPADOC patents': 1396,
    'Active': '2002-2016',
    'Mainfieldsofinvention': 'Radio Frequency Integrated Circuits',
  },
  {
    'Inventor': 'Shou-Shan Fan',
    'NumberofPatentFamilies': 805,
    'Country': 'China',
    'Number of INPADOC patents': 2120,
    'Active': '2006-2016',
    'Mainfieldsofinvention': 'Carbon nanotubes and applications of
carbon nanotubes',
  },

```

```

{
  'Inventor': 'Michael J. Sullivan',
  'NumberofPatentFamilies': 788,
  'Country': 'USA',
  'Number of INPADOC patents': 1560,
  'Active': '1977-2016',
  'Mainfieldsofinvention': 'Golf balls',
},
{
  'Inventor': 'Rick Allen Hamilton II',
  'NumberofPatentFamilies': 773,
  'Country': 'USA',
  'Number of INPADOC patents': 1064,
  'Active': '1999-2016',
  'Mainfieldsofinvention': 'Various',
},
{
  'Inventor': 'Warren Farnworth',
  'NumberofPatentFamilies': 770,
  'Country': 'USA',
  'Number of INPADOC patents': 931,
  'Active': '1990-2016',
  'Mainfieldsofinvention': 'Semiconductor packaging',
},
{
  'Inventor': 'Carleton Ellis',
  'NumberofPatentFamilies': 753,
  'Country': 'USA',
  'Number of INPADOC patents': 'NA',
  'Active': '1876(b)-1941(d)',
  'Mainfieldsofinvention': 'Margarine, Polyester, Anti-knock gasoline,
Paint stripper',
},
{
  'Inventor': 'William H. Eby',
  'NumberofPatentFamilies': 733,
  'Country': 'USA',
  'Number of INPADOC patents': 758,
  'Active': '1994-2016',
  'Mainfieldsofinvention': 'Transgenic soybeans',
},
{
  'Inventor': 'Hideo Ando',
  'NumberofPatentFamilies': 728,
  'Country': 'Japan',
  'Number of INPADOC patents': 2588,
  'Active': '1983-2016',
  'Mainfieldsofinvention': 'Optical recording',
},
{
  'Inventor': 'Salman Akram',
  'NumberofPatentFamilies': 728,
  'Country': 'USA',
  'Number of INPADOC patents': 915,
  'Active': '1995-2016',
  'Mainfieldsofinvention': 'Semiconductor packaging',
},
},

```

```

{
  'Inventor': 'George Spector',
  'NumberofPatentFamilies': 722,
  'Country': 'USA',
  'Number of INPADOC patents': 747,
  'Active': '1976-1998',
  'Mainfieldsofinvention': 'Gadgets, Toys',
},
{
  'Inventor': 'Jeyhan Karaoguz',
  'NumberofPatentFamilies': 721,
  'Country': 'USA',
  'Number of INPADOC patents': 1530,
  'Active': '1996-2016',
  'Mainfieldsofinvention': 'Wireless communications, Computer
networks',
},
{
  'Inventor': 'Elihu Thomson',
  'NumberofPatentFamilies': 696,
  'Country': 'UK',
  'Number of INPADOC patents': 'NA',
  'Active': '1853(b)-1937(d)',
  'Mainfieldsofinvention': 'Electric power, Arc lamp, Electric motors,
Lightning arrester, Arc welder',
},
{
  'Inventor': 'Austin L. Gurney',
  'NumberofPatentFamilies': 695,
  'Country': 'USA',
  'Number of INPADOC patents': 3909,
  'Active': '1999-2016',
  'Mainfieldsofinvention': 'Proteins, Antibodies',
},
{
  'Inventor': 'Tetsujiro Kondo',
  'NumberofPatentFamilies': 684,
  'Country': 'Japan',
  'Number of INPADOC patents': 4158,
  'Active': '1987-2015',
  'Mainfieldsofinvention': 'Signal processing, Image processing',
},
{
  'Inventor': 'Nathan Myhrvold',
  'NumberofPatentFamilies': 661,
  'Country': 'USA',
  'Number of INPADOC patents': 1690,
  'Active': '1994-2016',
  'Mainfieldsofinvention': 'Various',
},
{
  'Inventor': 'William I. Wood',
  'NumberofPatentFamilies': 653,
  'Country': 'USA',
  'Number of INPADOC patents': 3560,
  'Active': '1981-2016',
  'Mainfieldsofinvention': 'Proteins, Antibodies',
}

```

```

    },
    {
      'Inventor': 'Simon R. Walmsley',
      'NumberofPatentFamilies': 651,
      'Country': 'Australia',
      'Number of INPADOC patents': 1249,
      'Active': '1995-2015',
      'Mainfieldsofinvention': 'Printing, Electronics, VLSI,
Cryptography',
    },
    {
      'Inventor': 'Mark Malamud',
      'NumberofPatentFamilies': 632,
      'Country': 'USA',
      'Number of INPADOC patents': 1759,
      'Active': '1997-2016',
      'Mainfieldsofinvention': 'Various',
    },
    {
      'Inventor': 'Royce A. Levien',
      'NumberofPatentFamilies': 630,
      'Country': 'USA',
      'Number of INPADOC patents': 1799,
      'Active': '1997-2016',
      'Mainfieldsofinvention': 'Various',
    },
    {
      'Inventor': 'Audrey D. Goddard',
      'NumberofPatentFamilies': 622,
      'Country': 'USA',
      'Number of INPADOC patents': 3416,
      'Active': '1997-2014',
      'Mainfieldsofinvention': 'Proteins, Antibodies',
    },
    {
      'Inventor': 'Muriel Y. Ishikawa',
      'NumberofPatentFamilies': 619,
      'Country': 'USA',
      'Number of INPADOC patents': 1660,
      'Active': '2002-2016',
      'Mainfieldsofinvention': 'Various',
    },
    {
      'Inventor': 'Robert W. Lord',
      'NumberofPatentFamilies': 618,
      'Country': 'USA',
      'Number of INPADOC patents': 1708,
      'Active': '2003-2016',
      'Mainfieldsofinvention': 'Various',
    },
    {
      'Inventor': 'Jerome Lemelson',
      'NumberofPatentFamilies': 606,
      'Country': 'USA',
      'Number of INPADOC patents': 'NA',
      'Active': '1923(b)-1997(d)',
    }
  ],

```

```

    'Mainfieldsofinvention': 'Toys, Industrial robots, Cordless
    telephones, Fax machines, Videocassette recorders',
  },
  {
    'Inventor': 'Béla Barényi',
    'NumberofPatentFamilies': 595,
    'Country': 'Austria',
    'Number of INPADOC patents': 1244,
    'Active': '1907(b)-1997(d)',
    'Mainfieldsofinvention': 'Passive safety in automobiles',
  },
  {
    'Inventor': 'Kie Y Ahn',
    'NumberofPatentFamilies': 593,
    'Country': 'USA',
    'Number of INPADOC patents': 709,
    'Active': '1976-2016',
    'Mainfieldsofinvention': 'Thin film processes and materials, VLSI,
    Semiconductor device fabrication',
  },
  {
    'Inventor': 'Tadahiro Ohmi',
    'NumberofPatentFamilies': 592,
    'Country': 'Japan',
    'Number of INPADOC patents': 2691,
    'Active': '1981-2016',
    'Mainfieldsofinvention': 'Thin film processes and materials,
    Semiconductor device fabrication',
  },
  {
    'Inventor': 'Jordin T. Kare',
    'NumberofPatentFamilies': 585,
    'Country': 'USA',
    'Number of INPADOC patents': 1559,
    'Active': '1992-2016',
    'Mainfieldsofinvention': 'Various',
  },
  {
    'Inventor': 'Paul J. Godowski',
    'NumberofPatentFamilies': 579,
    'Country': 'USA',
    'Number of INPADOC patents': 2605,
    'Active': '1994-2014',
    'Mainfieldsofinvention': 'Proteins, Antibodies',
  },
  {
    'Inventor': 'Artur Fischer',
    'NumberofPatentFamilies': 570,
    'Country': 'Germany',
    'Number of INPADOC patents': 3097,
    'Active': '1976-2002',
    'Mainfieldsofinvention': 'Fasteners, Construction toys',
  },
  {
    'Inventor': 'Edward J. Nowak',
    'NumberofPatentFamilies': 564,
    'Country': 'USA',

```

```

    'Number of INPADOC patents': 1145,
    'Active': '1979-2016',
    'Mainfieldsofinvention': 'Semiconductor device fabrication,
Semiconductor memory, Semiconductor device',
  },
  {
    'Inventor': 'Louis L. Hsu',
    'NumberofPatentFamilies': 551,
    'Country': 'USA',
    'Number of INPADOC patents': 914,
    'Active': '1988-2016',
    'Mainfieldsofinvention': 'Various',
  },
  {
    'Inventor': 'Edwin H. Land',
    'NumberofPatentFamilies': 535,
    'Country': 'USA',
    'Number of INPADOC patents': 1236,
    'Active': '1909(b)-1991(d)',
    'Mainfieldsofinvention': 'Instant photography, Polarizing film',
  },
  {
    'Inventor': 'Henri Dreyfus',
    'NumberofPatentFamilies': 524,
    'Country': 'Switzerland',
    'Number of INPADOC patents': 2117,
    'Active': '1882(b)-1944(d)',
    'Mainfieldsofinvention': 'Polymers, Synthetic fibers, Dyes',
  },
  {
    'Inventor': 'Bruce B. Doris',
    'NumberofPatentFamilies': 522,
    'Country': 'USA',
    'Number of INPADOC patents': 867,
    'Active': '1995-2016',
    'Mainfieldsofinvention': 'Integrated Circuits, CMOS, DRAM,
Semiconductor device fabrication',
  },
  {
    'Inventor': 'Clyde C. Farmer',
    'NumberofPatentFamilies': 513,
    'Country': 'USA',
    'Number of INPADOC patents': 830,
    'Active': '18??(b)-19??(d)',
    'Mainfieldsofinvention': 'Railway air brakes',
  },
  {
    'Inventor': 'Heinz Focke',
    'NumberofPatentFamilies': 512,
    'Country': 'Germany',
    'Number of INPADOC patents': 2896,
    'Active': '1976-2013',
    'Mainfieldsofinvention': 'Cigarette packaging',
  },
  {
    'Inventor': 'Mark I. Gardner',
    'NumberofPatentFamilies': 511,

```

```

    'Country': 'USA',
    'Number of INPADOC patents': 587,
    'Active': '1994-2010',
    'Mainfieldsofinvention': 'Consumer electronics, Energy, Computers,
Semiconductors, Physics',
  },
  {
    'Inventor': 'Ravi K. Arimilli',
    'NumberofPatentFamilies': 506,
    'Country': 'India',
    'Number of INPADOC patents': 767,
    'Active': '1992-2016',
    'Mainfieldsofinvention': 'Computer architecture, Semiconductor
memory, Cache coherence, Symmetric multiprocessing',
  },
  {
    'Inventor': 'Louis H. Morin',
    'NumberofPatentFamilies': 503,
    'Country': 'USA',
    'Number of INPADOC patents': 720,
    'Active': '18??(b)-19??(d)',
    'Mainfieldsofinvention': 'Fasteners, Locks, Bobbins',
  },
  {
    'Inventor': 'Tobin A. King',
    'NumberofPatentFamilies': 497,
    'Country': 'Australia',
    'Number of INPADOC patents': 1218,
    'Active': '2000-2015',
    'Mainfieldsofinvention': 'Printing, Digital paper, Mechanical',
  },
  {
    'Inventor': 'Eric C. Leuthardt',
    'NumberofPatentFamilies': 495,
    'Country': 'USA',
    'Number of INPADOC patents': 1274,
    'Active': '2006-2016',
    'Mainfieldsofinvention': 'Medical devices',
  },
  {
    'Inventor': 'Ali Khakifirooz',
    'NumberofPatentFamilies': 489,
    'Country': 'USA',
    'Number of INPADOC patents': 737,
    'Active': '2011-2016',
    'Mainfieldsofinvention': 'Integrated Circuits, CMOS, Semiconductor
device fabrication',
  },
  {
    'Inventor': 'Jack A. Mandelman',
    'NumberofPatentFamilies': 481,
    'Country': 'USA',
    'Number of INPADOC patents': 889,
    'Active': '1987-2014',
    'Mainfieldsofinvention': 'Various',
  },
  {

```



```

    'Inventor': 'Jeffrey P. Gambino',
    'NumberofPatentFamilies': 479,
    'Country': 'USA',
    'Number of INPADOC patents': 798,
    'Active': '1992-2016',
    'Mainfieldsofinvention': 'MEMS, CMOS, BiCMOS, DRAM, Image Sensors,
RF, Biosensors, 3D Integrated Circuits',
  },
  {
    'Inventor': 'John M. Santosuosso',
    'NumberofPatentFamilies': 473,
    'Country': 'USA',
    'Number of INPADOC patents': 683,
    'Active': '2001-2016',
    'Mainfieldsofinvention': 'Various',
  },
  {
    'Inventor': 'James M. Hart',
    'NumberofPatentFamilies': 464,
    'Country': 'USA',
    'Number of INPADOC patents': 1145,
    'Active': '1988-2016',
    'Mainfieldsofinvention': 'Motor vehicle transmission',
  },
  {
    'Inventor': 'Eberhard Ammermann',
    'NumberofPatentFamilies': 451,
    'Country': 'Germany',
    'Number of INPADOC patents': 5178,
    'Active': '1979-2015',
    'Mainfieldsofinvention': 'Fungicides',
  },
  {
    'Inventor': 'Thomas E. Murray',
    'NumberofPatentFamilies': 449,
    'Country': 'USA',
    'Number of INPADOC patents': 462,
    'Active': '1860(b)-1929(d)',
    'Mainfieldsofinvention': 'Electrical, HVAC, Wheels, Metal working,
Light dimmer',
  },
  {
    'Inventor': 'Akira Nakazawa',
    'NumberofPatentFamilies': 445,
    'Country': 'Australia',
    'Number of INPADOC patents': 1340,
    'Active': '1980-2016',
    'Mainfieldsofinvention': 'Printing, Mechanical',
  },
  {
    'Inventor': 'Hongyong Zhang',
    'NumberofPatentFamilies': 440,
    'Country': 'Japan',
    'Number of INPADOC patents': 858,
    'Active': '1993-2016',
    'Mainfieldsofinvention': 'Thin film transistors, Liquid crystal
displays',
  },

```

```

    },
    {
      'Inventor': 'Ronald S. Cok',
      'NumberofPatentFamilies': 436,
      'Country': 'USA',
      'Number of INPADOC patents': 747,
      'Active': '1986-2016',
      'Mainfieldsofinvention': 'OLED displays; image processing',
    },
    {
      'Inventor': 'fe',
      'NumberofPatentFamilies': 430,
      'Country': 'USA',
      'Number of INPADOC patents': 1759,
      'Active': '1983-2016',
      'Mainfieldsofinvention': 'Biotechnology, Drug delivery, Tissue
engineering',
    },
    {
      'Inventor': 'Scott H. Wittkopp',
      'NumberofPatentFamilies': 429,
      'Country': 'USA',
      'Number of INPADOC patents': 1010,
      'Active': '2001-2016',
      'Mainfieldsofinvention': 'Motor vehicle transmission',
    },
    {
      'Inventor': 'John Hays Hammond, Jr.',
      'NumberofPatentFamilies': 417,
      'Country': 'USA',
      'Number of INPADOC patents': 460,
      'Active': '1888(b)-1965(d)',
      'Mainfieldsofinvention': 'Radio control, Radio communications,
Torpedoes',
    },
    {
      'Inventor': 'Wilhelm Brandes',
      'NumberofPatentFamilies': 411,
      'Country': 'Germany',
      'Number of INPADOC patents': 2923,
      'Active': '1976-2010',
      'Mainfieldsofinvention': 'Fungicides',
    },
    {
      'Inventor': 'Anthony K. Stamper',
      'NumberofPatentFamilies': 411,
      'Country': 'USA',
      'Number of INPADOC patents': 726,
      'Active': '1998-2016',
      'Mainfieldsofinvention': 'MEMS, CMOS, BiCMOS, Silicon-germanium',
    },
    {
      'Inventor': 'Hossein Eslambolchi',
      'NumberofPatentFamilies': 410,
      'Country': 'USA',
      'Number of INPADOC patents': 631,
      'Active': '1993-2016',
    }
  ],

```

```

    'Mainfieldsofinvention': 'Telecommunications, Network intelligence,
information Technology, communications technology',
  },
  {
    'Inventor': 'Stanford R. Ovshinsky',
    'NumberofPatentFamilies': 400,
    'Country': 'USA',
    'Number of INPADOC patents': 1649,
    'Active': '1922(b)-2012(d)',
    'Mainfieldsofinvention': 'Batteries, Solar cells, Liquid crystal
displays, Hydrogen fuel cells, Computer data storage',
  },
  {
    'Inventor': 'Victoria Y. H. Wood',
    'NumberofPatentFamilies': 400,
    'Country': 'USA',
    'Number of INPADOC patents': 1045,
    'Active': '2009-2016',
    'Mainfieldsofinvention': 'Various',
  },
  {
    'Inventor': 'Josef Theurer',
    'NumberofPatentFamilies': 388,
    'Country': 'Austria',
    'Number of INPADOC patents': 5085,
    'Active': '1976-2016',
    'Mainfieldsofinvention': 'Railroad maintenance machines',
  },
  {
    'Inventor': 'Cary L. Bates',
    'NumberofPatentFamilies': 384,
    'Country': 'USA',
    'Number of INPADOC patents': 570,
    'Active': '1994-2016',
    'Mainfieldsofinvention': 'Programming tools, DBX, Memory debuggers',
  },
  {
    'Inventor': 'David V. Horak',
    'NumberofPatentFamilies': 380,
    'Country': 'USA',
    'Number of INPADOC patents': 616,
    'Active': '1992-2016',
    'Mainfieldsofinvention': 'Various',
  },
  {
    'Inventor': 'Kai-Li Jiang',
    'NumberofPatentFamilies': 379,
    'Country': 'China',
    'Number of INPADOC patents': 829,
    'Active': '2006-2016',
    'Mainfieldsofinvention': 'Carbon nanotubes and applications of
carbon nanotubes',
  },
  {
    'Inventor': 'Hans-Joachim Santel',
    'NumberofPatentFamilies': 377,
    'Country': 'Germany',

```

```

      'Number of INPADOC patents': 2623,
      'Active': '1986-2013',
      'Mainfieldsofinvention': 'Herbicides, Pesticides, Organic
chemistry',
    },
    {
      'Inventor': 'Xuemin (Sherman) Chen',
      'NumberofPatentFamilies': 377,
      'Country': 'USA',
      'Number of INPADOC patents': 1151,
      'Active': '1997-2017',
      'Mainfieldsofinvention': 'Computer networks, Integrated Circuits,
Signal Processing',
    },
    {
      'Inventor': 'George P. Liang',
      'NumberofPatentFamilies': 375,
      'Country': 'China',
      'Number of INPADOC patents': 508,
      'Active': '1983-2016',
      'Mainfieldsofinvention': 'Gas turbine cooling',
    },
    {
      'Inventor': 'Gisela Lorenz',
      'NumberofPatentFamilies': 374,
      'Country': 'Germany',
      'Number of INPADOC patents': 4155,
      'Active': '1990-2015',
      'Mainfieldsofinvention': 'Fungicides, Organic chemistry',
    },
    {
      'Inventor': 'Garry R. Jackson',
      'NumberofPatentFamilies': 367,
      'Country': 'Australia',
      'Number of INPADOC patents': 656,
      'Active': '2001-2016',
      'Mainfieldsofinvention': 'Printing, Mechanical',
    },
    {
      'Inventor': 'Paul W. Dent',
      'NumberofPatentFamilies': 362,
      'Country': 'USA',
      'Number of INPADOC patents': 2252,
      'Active': '1984-2015',
      'Mainfieldsofinvention': 'Wireless communications',
    },
    {
      'Inventor': 'George Westinghouse',
      'NumberofPatentFamilies': 361,
      'Country': 'USA',
      'Number of INPADOC patents': 'NA',
      'Active': '1846(b)-1914(d)',
      'Mainfieldsofinvention': 'Electric power, Electricity meter, Railway
air brake, Steam engines',
    },
    {
      'Inventor': 'Wael W. Diab',

```

```

      'NumberofPatentFamilies': 358,
      'Country': 'USA',
      'Number of INPADOC patents': 774,
      'Active': '2003-2016',
      'Mainfieldsofinvention': 'Computer networks',
    },
    {
      'Inventor': 'Devendra K. Sadana',
      'NumberofPatentFamilies': 356,
      'Country': 'India',
      'Number of INPADOC patents': 794,
      'Active': '1983-2016',
      'Mainfieldsofinvention': 'Solar cells, OLED, Integrated Circuits,
CMOS, DRAM, LEDs',
    },
    {
      'Inventor': 'Vincent J. Zimmer',
      'NumberofPatentFamilies': 354,
      'Country': 'USA',
      'Number of INPADOC patents': 972,
      'Active': '1999-2016',
      'Mainfieldsofinvention': 'Computer software and firmware',
    },
    {
      'Inventor': 'Robert R. Schmidt',
      'NumberofPatentFamilies': 350,
      'Country': 'Germany',
      'Number of INPADOC patents': 2467,
      'Active': '1971-2005',
      'Mainfieldsofinvention': 'Herbicides, Fungicides, Organic
chemistry',
    },
    {
      'Inventor': 'Norman M. Berry',
      'NumberofPatentFamilies': 347,
      'Country': 'Australia',
      'Number of INPADOC patents': 516,
      'Active': '2006-2016',
      'Mainfieldsofinvention': 'Printing, Mechanical',
    },
    {
      'Inventor': 'Chih-Chao Yang',
      'NumberofPatentFamilies': 345,
      'Country': 'USA',
      'Number of INPADOC patents': 690,
      'Active': '2003-2016',
      'Mainfieldsofinvention': 'Integrated Circuits',
    },
    {
      'Inventor': 'Gregory J. Boss',
      'NumberofPatentFamilies': 345,
      'Country': 'USA',
      'Number of INPADOC patents': 588,
      'Active': '2008-2016',
      'Mainfieldsofinvention': 'Various'
    },
  ],
  {

```

```

    'Inventor': 'Mark W. Kroll',
    'NumberofPatentFamilies': 343,
    'Country': 'USA',
    'Number of INPADOC patents': 460,
    'Active': '1987-2016',
    'Mainfieldsofinvention': 'Implantable medical devices',
  },
  {
    'Inventor': 'Brian M. O\'Connell',
    'NumberofPatentFamilies': 331,
    'Country': 'USA',
    'Number of INPADOC patents': 592,
    'Active': '2009-2016',
    'Mainfieldsofinvention': 'Various',
  },
  {
    'Inventor': 'William Daniel Hillis',
    'NumberofPatentFamilies': 328,
    'Country': 'USA',
    'Number of INPADOC patents': 229,
    'Active': '1986-2016',
    'Mainfieldsofinvention': 'Various',
  },
  {
    'Inventor': 'Brent A. Anderson',
    'NumberofPatentFamilies': 323,
    'Country': 'USA',
    'Number of INPADOC patents': 454,
    'Active': '2001-2016',
    'Mainfieldsofinvention': 'Semiconductor device fabrication,
Semiconductor memory, Semiconductor device',
  },
  {
    'Inventor': 'Jeffrey E. Stahmann',
    'NumberofPatentFamilies': 321,
    'Country': 'USA',
    'Number of INPADOC patents': 640,
    'Active': '1994-2016',
    'Mainfieldsofinvention': 'Medical devices',
  },
  {
    'Inventor': 'Carl J. Radens',
    'NumberofPatentFamilies': 317,
    'Country': 'USA',
    'Number of INPADOC patents': 636,
    'Active': '1994-2016',
    'Mainfieldsofinvention': 'Integrated Circuits, CMOS, DRAM,
Semiconductor device fabrication',
  },
  {
    'Inventor': 'Clifford A. Pickover',
    'NumberofPatentFamilies': 317,
    'Country': 'USA',
    'Number of INPADOC patents': 653,
    'Active': '1992-2016',
    'Mainfieldsofinvention': 'Various',
  },

```

```

{
  'Inventor': 'Liang Liu',
  'NumberofPatentFamilies': 310,
  'Country': 'China',
  'Number of INPADOC patents': 777,
  'Active': '2005-2016',
  'Mainfieldsofinvention': 'Carbon nanotubes and applications of
carbon nanotubes',
},
{
  'Inventor': 'Steven L. Teig',
  'NumberofPatentFamilies': 307,
  'Country': 'USA',
  'Number of INPADOC patents': 366,
  'Active': '1995-2016',
  'Mainfieldsofinvention': 'Integrated Circuits',
},
{
  'Inventor': 'Victoria Smith',
  'NumberofPatentFamilies': 305,
  'Country': 'USA',
  'Number of INPADOC patents': 2040,
  'Active': '2006-2016',
  'Mainfieldsofinvention': 'Proteins, Antibodies',
},
{
  'Inventor': 'Robert G. LeTourneau',
  'NumberofPatentFamilies': 299,
  'Country': 'USA',
  'Number of INPADOC patents': 'NA',
  'Active': '1888(b)-1969(d)',
  'Mainfieldsofinvention': 'Earthworks (engineering), Heavy Equipment,
Industrial Machinery',
},
{
  'Inventor': 'William R. Tonti',
  'NumberofPatentFamilies': 291,
  'Country': 'USA',
  'Number of INPADOC patents': 441,
  'Active': '1994-2016',
  'Mainfieldsofinvention': 'Integrated Circuits, CMOS, DRAM,
Semiconductor device fabrication',
},
{
  'Inventor': 'Keith R. Walker',
  'NumberofPatentFamilies': 282,
  'Country': 'Saudi Arabia',
  'Number of INPADOC patents': 318,
  'Active': '2003-2016',
  'Mainfieldsofinvention': 'Various',
},
{
  'Inventor': 'Victor S. Moore',
  'NumberofPatentFamilies': 280,
  'Country': 'USA',
  'Number of INPADOC patents': 428,
  'Active': '1982-2016',
}

```

```

    'Mainfieldsofinvention': 'Various',
  },
  {
    'Inventor': 'Hanson S. Gifford III',
    'NumberofPatentFamilies': 276,
    'Country': 'USA',
    'Number of INPADOC patents': 795,
    'Active': '1987-2016',
    'Mainfieldsofinvention': 'Medical Devices',
  },
  {
    'Inventor': 'Daniel J. Winarski',
    'NumberofPatentFamilies': 275,
    'Country': 'USA',
    'Number of INPADOC patents': 506,
    'Active': '1982-2016',
    'Mainfieldsofinvention': 'Various',
  },
  {
    'Inventor': 'Adam Heller',
    'NumberofPatentFamilies': 272,
    'Country': 'Romania',
    'Number of INPADOC patents': 711,
    'Active': '1968-2016',
    'Mainfieldsofinvention': 'Solar cells, Glucose meters, Lasers',
  },
  {
    'Inventor': 'Lisa Seacat DeLuca',
    'NumberofPatentFamilies': 271,
    'Country': 'USA',
    'Number of INPADOC patents': 385,
    'Active': '2009-2016',
    'Mainfieldsofinvention': 'Various',
  },
  {
    'Inventor': 'Brent Keeth',
    'NumberofPatentFamilies': 270,
    'Country': 'USA',
    'Number of INPADOC patents': 470,
    'Active': '1994-2016',
    'Mainfieldsofinvention': 'Integrated Circuits, CMOS, DRAM',
  },
  {
    'Inventor': 'Hartley Owen',
    'NumberofPatentFamilies': 267,
    'Country': 'USA',
    'Number of INPADOC patents': 751,
    'Active': '1976-2010',
    'Mainfieldsofinvention': 'Fluid catalytic cracking',
  },
  {
    'Inventor': 'Michael A. Rothman',
    'NumberofPatentFamilies': 256,
    'Country': 'USA',
    'Number of INPADOC patents': 687,
    'Active': '2001-2017',
    'Mainfieldsofinvention': 'Computer software and firmware',
  },

```



```

    },
    {
      'Inventor': 'Yoshihiro Kikuchi',
      'NumberofPatentFamilies': 255,
      'Country': 'Japan',
      'Number of INPADOC patents': 1120,
      'Active': '1994-2015',
      'Mainfieldsofinvention': 'Video processing',
    },
    {
      'Inventor': 'Kulvir S. Bhogal',
      'NumberofPatentFamilies': 252,
      'Country': 'USA',
      'Number of INPADOC patents': 486,
      'Active': '2003-2016',
      'Mainfieldsofinvention': 'Various',
    },
    {
      'Inventor': 'Bengt Lindoff',
      'NumberofPatentFamilies': 248,
      'Country': 'Sweden',
      'Number of INPADOC patents': 1658,
      'Active': '2000-2017',
      'Mainfieldsofinvention': 'Wireless communications',
    },
    {
      'Inventor': 'Nobuyuki Taniguchi',
      'NumberofPatentFamilies': 245,
      'Country': 'Japan',
      'Number of INPADOC patents': 967,
      'Active': '1979-2015',
      'Mainfieldsofinvention': 'Cameras',
    },
    {
      'Inventor': 'Dean L. Kamen',
      'NumberofPatentFamilies': 243,
      'Country': 'USA',
      'Number of INPADOC patents': 1186,
      'Active': '1979-2016',
      'Mainfieldsofinvention': 'Battery-powered electric vehicles, Medical
devices, Stirling engines, Water purification, Wheelchairs',
    },
    {
      'Inventor': 'Philip S. Yu',
      'NumberofPatentFamilies': 236,
      'Country': 'USA',
      'Number of INPADOC patents': 158,
      'Active': '1982-2016',
      'Mainfieldsofinvention': 'Various',
    },
    {
      'Inventor': 'Rajiv V. Joshi',
      'NumberofPatentFamilies': 235,
      'Country': 'USA',
      'Number of INPADOC patents': 354,
      'Active': '1986-2016',
      'Mainfieldsofinvention': 'Electronics, analytics',
    }
  ],

```

```

    },
    {
      'Inventor': 'Lawrence A. Clevenger',
      'NumberofPatentFamilies': 235,
      'Country': 'USA',
      'Number of INPADOC patents': 526,
      'Active': '1996-2017',
      'Mainfieldsofinvention': 'Semiconductor, Cognitive, Memory,
Security, Analytics',
    },
    {
      'Inventor': 'Johnny M. Shieh',
      'NumberofPatentFamilies': 231,
      'Country': 'USA',
      'Number of INPADOC patents': 444,
      'Active': '1996-2016',
      'Mainfieldsofinvention': 'Various',
    },
    {
      'Inventor': 'Takeshi Chujoh',
      'NumberofPatentFamilies': 229,
      'Country': 'Japan',
      'Number of INPADOC patents': 1065,
      'Active': '1995-2016',
      'Mainfieldsofinvention': 'Video processing',
    },
    {
      'Inventor': 'Bran Ferren',
      'NumberofPatentFamilies': 225,
      'Country': 'USA',
      'Number of INPADOC patents': 589,
      'Active': '1986-2017',
      'Mainfieldsofinvention': 'Computers, Consumer Electronics, Optical
Systems, Medical, User Interfaces, Automotive',
    },
    {
      'Inventor': 'Paul Ian Mackey',
      'NumberofPatentFamilies': 220,
      'Country': 'Australia',
      'Number of INPADOC patents': 246,
      'Active': '2008-2016',
      'Mainfieldsofinvention': 'Printing, Mechanical',
    },
    {
      'Inventor': 'Louis Rosenberg',
      'NumberofPatentFamilies': 218,
      'Country': 'USA',
      'Number of INPADOC patents': 444,
      'Active': '1995-2016',
      'Mainfieldsofinvention': 'Augmented Reality, Virtual Reality, A.I.,
HCI',
    },
    {
      'Inventor': 'Thomas J. Kennedy III',
      'NumberofPatentFamilies': 218,
      'Country': 'USA',
      'Number of INPADOC patents': 513,
    }
  ],

```

```

    'Active': '1992-2016',
    'Mainfieldsofinvention': 'Sporting Goods, Wind Turbines',
  },
  {
    'Inventor': 'Gerald F. McBrearty',
    'NumberofPatentFamilies': 213,
    'Country': 'USA',
    'Number of INPADOC patents': 387,
    'Active': '1997-2016',
    'Mainfieldsofinvention': 'Various',
  },
  {
    'Inventor': 'Esmael H. Dinan',
    'NumberofPatentFamilies': 208,
    'Country': 'USA',
    'Number of INPADOC patents': 344,
    'Active': '2000-2017',
    'Mainfieldsofinvention': 'Communication Networks',
  },
  {
    'Inventor': 'Imad Libbus',
    'NumberofPatentFamilies': 207,
    'Country': 'USA',
    'Number of INPADOC patents': 472,
    'Active': '2007-2017',
    'Mainfieldsofinvention': 'Medical devices',
  },
  {
    'Inventor': 'Hiroshi (You) Yoshioka',
    'NumberofPatentFamilies': 205,
    'Country': 'Japan',
    'Number of INPADOC patents': 181,
    'Active': '1997-2015',
    'Mainfieldsofinvention': 'Cameras',
  },
  {
    'Inventor': 'Patrick B. Usoro',
    'NumberofPatentFamilies': 205,
    'Country': 'USA',
    'Number of INPADOC patents': 343,
    'Active': '1999-2016',
    'Mainfieldsofinvention': 'Transmissions, Hybrid Powertrains, Vehicle
Thermal Management',
  },
  {
    'Inventor': 'Gregory McAvoy',
    'NumberofPatentFamilies': 205,
    'Country': 'Australia',
    'Number of INPADOC patents': 433,
    'Active': '2003-2014',
    'Mainfieldsofinvention': 'Printing, MEMS',
  },
  {
    'Inventor': 'Sebastian T Ventrone',
    'NumberofPatentFamilies': 204,
    'Country': 'USA',
    'Number of INPADOC patents': 283,
  }

```

```

        'Active': '1989-2017',
        'Mainfieldsofinvention': 'Semiconductor, Logic, Architecture',
    },
    {
        'Inventor': 'Dorin Comaniciu',
        'NumberofPatentFamilies': 200,
        'Country': 'USA',
        'Number of INPADOC patents': 452,
        'Active': '2003-2017',
        'Mainfieldsofinvention': 'Machine Intelligence, Medical Imaging,
Image-Guided Surgery, Computer Vision',
    }
];
export interface ColumnSpanDataType {
    EmployeeID: number;
    EmployeeName: string;
    '9:00': string;
    '9:30': string;
    '10:00': string;
    '10:30': string;
    '11:00': string;
    '11:30': string;
    '12:00': string;
    '12:30': string;
    '1:00': string;
    '1:30': string;
    '2:00': string;
    '2:30': string;
    '3:00': string;
    '3:30': string;
    '4:00': string;
    '4:30': string;
    '5:00': string;
}
export let columnSpanData: ColumnSpanDataType[] = [
    {
        EmployeeID: 10001,
        EmployeeName: 'Davolio',
        '9:00': 'Analysis Tasks',
        '9:30': 'Analysis Tasks',
        '10:00': 'Team Meeting',
        '10:30': 'Testing',
        '11:00': 'Development',
        '11:30': 'Development',
        '12:00': 'Development',
        '12:30': 'Support',
        '1:00': 'Lunch Break',
        '1:30': 'Lunch Break',
        '2:00': 'Lunch Break',
        '2:30': 'Testing',
        '3:00': 'Testing',
        '3:30': 'Development',
        '4:00': 'Conference',
        '4:30': 'Team Meeting',
        '5:00': 'Team Meeting'
    },
    {

```

```

    EmployeeID: 10002,
    EmployeeName: 'Buchanan',
    '9:00': 'Task Assign',
    '9:30': 'Support',
    '10:00': 'Support',
    '10:30': 'Support',
    '11:00': 'Testing',
    '11:30': 'Testing',
    '12:00': 'Testing',
    '12:30': 'Testing',
    '1:00': 'Lunch Break',
    '1:30': 'Lunch Break',
    '2:00': 'Lunch Break',
    '2:30': 'Development',
    '3:00': 'Development',
    '3:30': 'Check Mail',
    '4:00': 'Check Mail',
    '4:30': 'Team Meeting',
    '5:00': 'Team Meeting'
  },
  {
    EmployeeID: 10003,
    EmployeeName: 'Fuller',
    '9:00': 'Check Mail',
    '9:30': 'Check Mail',
    '10:00': 'Check Mail',
    '10:30': 'Analysis Tasks',
    '11:00': 'Analysis Tasks',
    '11:30': 'Support',
    '12:00': 'Support',
    '12:30': 'Support',
    '1:00': 'Lunch Break',
    '1:30': 'Lunch Break',
    '2:00': 'Lunch Break',
    '2:30': 'Development',
    '3:00': 'Development',
    '3:30': 'Team Meeting',
    '4:00': 'Team Meeting',
    '4:30': 'Development',
    '5:00': 'Development'
  },
  {
    EmployeeID: 10004,
    EmployeeName: 'Leverling',
    '9:00': 'Testing',
    '9:30': 'Check Mail',
    '10:00': 'Check Mail',
    '10:30': 'Support',
    '11:00': 'Testing',
    '11:30': 'Testing',
    '12:00': 'Testing',
    '12:30': 'Testing',
    '1:00': 'Lunch Break',
    '1:30': 'Lunch Break',
    '2:00': 'Lunch Break',
    '2:30': 'Development',
    '3:00': 'Development',
  }

```

```

      '3:30': 'Check Mail',
      '4:00': 'Conference',
      '4:30': 'Conference',
      '5:00': 'Team Meeting'
    },
    {
      EmployeeID: 10005,
      EmployeeName: 'Peacock',
      '9:00': 'Task Assign',
      '9:30': 'Task Assign',
      '10:00': 'Task Assign',
      '10:30': 'Task Assign',
      '11:00': 'Check Mail',
      '11:30': 'Support',
      '12:00': 'Support',
      '12:30': 'Support',
      '1:00': 'Lunch Break',
      '1:30': 'Lunch Break',
      '2:00': 'Lunch Break',
      '2:30': 'Development',
      '3:00': 'Development',
      '3:30': 'Team Meeting',
      '4:00': 'Team Meeting',
      '4:30': 'Testing',
      '5:00': 'Testing'
    },
    {
      EmployeeID: 10006,
      EmployeeName: 'Janet',
      '9:00': 'Testing',
      '9:30': 'Testing',
      '10:00': 'Support',
      '10:30': 'Support',
      '11:00': 'Support',
      '11:30': 'Team Meeting',
      '12:00': 'Team Meeting',
      '12:30': 'Team Meeting',
      '1:00': 'Lunch Break',
      '1:30': 'Lunch Break',
      '2:00': 'Lunch Break',
      '2:30': 'Development',
      '3:00': 'Development',
      '3:30': 'Team Meeting',
      '4:00': 'Team Meeting',
      '4:30': 'Development',
      '5:00': 'Development'
    },
    {
      EmployeeID: 10007,
      EmployeeName: 'Suyama',
      '9:00': 'Analysis Tasks',
      '9:30': 'Analysis Tasks',
      '10:00': 'Testing',
      '10:30': 'Development',
      '11:00': 'Development',
      '11:30': 'Testing',
      '12:00': 'Testing',

```

```

        '12:30': 'Testing',
        '1:00': 'Lunch Break',
        '1:30': 'Lunch Break',
        '2:00': 'Lunch Break',
        '2:30': 'Support',
        '3:00': 'Build',
        '3:30': 'Build',
        '4:00': 'Check Mail',
        '4:30': 'Check Mail',
        '5:00': 'Check Mail'
    },
    {
        EmployeeID: 10008,
        EmployeeName: 'Robert',
        '9:00': 'Task Assign',
        '9:30': 'Task Assign',
        '10:00': 'Task Assign',
        '10:30': 'Development',
        '11:00': 'Development',
        '11:30': 'Development',
        '12:00': 'Testing',
        '12:30': 'Support',
        '1:00': 'Lunch Break',
        '1:30': 'Lunch Break',
        '2:00': 'Lunch Break',
        '2:30': 'Check Mail',
        '3:00': 'Check Mail',
        '3:30': 'Check Mail',
        '4:00': 'Team Meeting',
        '4:30': 'Team Meeting',
        '5:00': 'Build'
    },
    {
        EmployeeID: 10009,
        EmployeeName: 'Andrew',
        '9:00': 'Check Mail',
        '9:30': 'Team Meeting',
        '10:00': 'Team Meeting',
        '10:30': 'Support',
        '11:00': 'Testing',
        '11:30': 'Development',
        '12:00': 'Development',
        '12:30': 'Development',
        '1:00': 'Lunch Break',
        '1:30': 'Lunch Break',
        '2:00': 'Lunch Break',
        '2:30': 'Check Mail',
        '3:00': 'Check Mail',
        '3:30': 'Check Mail',
        '4:00': 'Team Meeting',
        '4:30': 'Development',
        '5:00': 'Development'
    },
    {
        EmployeeID: 10010,
        EmployeeName: 'Michael',
        '9:00': 'Task Assign',

```

```

    '9:30': 'Task Assign',
    '10:00': 'Task Assign',
    '10:30': 'Analysis Tasks',
    '11:00': 'Analysis Tasks',
    '11:30': 'Development',
    '12:00': 'Development',
    '12:30': 'Development',
    '1:00': 'Lunch Break',
    '1:30': 'Lunch Break',
    '2:00': 'Lunch Break',
    '2:30': 'Testing',
    '3:00': 'Testing',
    '3:30': 'Testing',
    '4:00': 'Build',
    '4:30': 'Build',
    '5:00': 'Build'
  }
];

```

INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, getElement, Inject, AccumulationDataLabel,
AccumulationTooltip, PieSeries, AccumulationAnnotation, } from
"@syncfusion/ej2-react-charts";
import { ColumnDirective, ColumnsDirective, GridComponent } from
"@syncfusion/ej2-react-grids";
export class Drilldown extends React.Component {
  data = [
    {
      x: 'SUV',
      y: 25,
      z: [
        {
          title: 'Automobile Sales in the SUV Segment',
          x: 'Toyota',
          y: 8,
          z: [
            { x: '2000', y: 20 },
            { x: '2001', y: 30 },
            { x: '2002', y: 40 },
          ],
        },
        { x: 'Ford', y: 12 },
        { x: 'GM', y: 17 },
        { x: 'Renault', y: 6 },
      ],
    },
    {
      x: 'Car',
      y: 37,
      z: [

```



```

        { title: 'Automobile Sales in the Car Segment', x: 'Toyota',
y: 7 },
        { x: 'Chrysler', y: 12 },
        { x: 'Nissan', y: 9 },
        { x: 'Ford', y: 15 },
    ],
    },
    {
        x: 'Pickup',
        y: 15,
        z: [
            { title: 'Automobile Sales in the Pickup Segment', x:
'Nissan', y: 9 },
            { x: 'Chrysler', y: 4 },
            { x: 'Ford', y: 7 },
            { x: 'Toyota', y: 20 },
        ],
    },
    {
        x: 'Minivan',
        y: 23,
        z: [
            { title: 'Automobile Sales in the Minivan Segment', x:
'Hummer', y: 11 },
            { x: 'Ford', y: 5 },
            { x: 'GM', y: 12 },
            { x: 'Chrysler', y: 3 },
        ],
    },
    ],
    grid;
    pointIndex = -1;
    dataLabel = {
        visible: true, position: 'Inside', connectorStyle: { type: 'Curve',
length: '5%' }, font: { size: '14px', color: 'white' }
    };
    pointIndex = -1;
    startAngle = 0;
    //public explodeIndex: number = 2;
    endAngle = 360;
    title = 'Automobile Sales by Category';
    pie;
    isparent = true;
    render() {
        return (<div className='control-pane'>
            <div className='control-section'>
                <AccumulationChartComponent id='pie-chart' ref={pie =>
this.pie = pie} title='Automobile Sales by Category'
enableSmartLabels={false} legendSettings={{ visible: false }} tooltip={{
enable: false, format: '${point.x} <br> ${point.y} %' }}
chartMouseClicked={this.onChartMouseClicked.bind(this)}
pointClick={this.onPointClick.bind(this)}
textRender={this.onTextRender.bind(this)}>
                    <Inject services={[AccumulationDataLabel,
AccumulationTooltip, PieSeries, AccumulationAnnotation]}/>
                    <AccumulationSeriesCollectionDirective>

```

```

        <AccumulationSeriesDirective
dataSource={this.data} xName='x' yName='y' dataLabel={this.dataLabel}
radius='70%' explode={false}>
        </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
</AccumulationChartComponent>
    <GridComponent ref={g => this.grid = g}
dataSource={data} height='350'>
        <ColumnsDirective>
            <ColumnDirective field='x' headerText='Vehicle'
type='string'></ColumnDirective>
            <ColumnDirective field='y' headerText='Sales'
type='string'></ColumnDirective>
        </ColumnsDirective>
    </GridComponent>
</div>
</div>);
    }
    onTextRender(args) {
        args.text = args.point.x + ' ' + args.point.y + ' %';
    }
    onPointClick(args) {
        if (getElement(pie.element.id + '_Series_' + args.seriesIndex +
'_Point_' + args.pointIndex)) {
            this.pie.series[0].dataSource = data[args.pointIndex].z;
            this.pie.title = data[args.pointIndex].z[0].title;
            pointIndex = args.pointIndex;
            this.pie.series[0].name = 'Level 2';
            this.pie.series[0].innerRadius = '30%';
            this.pie.annotations = [
                {
                    content: '<div id="back"
style="cursor:pointer;padding:3px;width:30px; height:30px;">' +
                        '',
                    region: 'Series',
                    x: '50%',
                    y: '50%',
                },
            ];
            this.pie.pointClick = click;
        }
        this.grid.dataSource = pie.series[0].dataSource;
        this.grid.columns[0].headerText = data[args.pointIndex].x;
        this.grid.refresh();
        this.pie.refresh();
    }
    onChartMouseClicked(args) {
        if (args.target.indexOf('back') > -1) {
            if (pie.series[0].name === 'Level 3') {
                pie.series[0].dataSource = data[pointIndex].z;
                pie.series[0].name = 'Level 2';
                pie.title = data[pointIndex].z[0].title;
                pie.series[0].innerRadius = '30%';
                grid.dataSource = pie.series[0].dataSource;
                grid.columns[0].headerText = data[pointIndex].x;
            }
        }
    }
}

```

```

        grid.refresh();
        pie.refresh();
    }
    else if (pie.series[0].name === 'Level 2') {
        pie.series[0].dataSource = data;
        pie.series[0].name = 'Level 1';
        pie.series[0].innerRadius = '0%';
        pie.title = 'Automobile Sales by Category';
        pie.annotations = null;
        pie.pointClick = pointClick;
        grid.dataSource = pie.series[0].dataSource;
        grid.columns[0].headerText = 'Vehicle';
        grid.refresh();
        pie.refresh();
    }
    }
    grid.dataSource = pie.series[0].dataSource;
}
pointClick;
tooltip;
title;
click(args) {
    if (pie.series[0].name !== 'Level 3') {
        switch (args.pointIndex) {
            case 0:
                pie.series[0].dataSource = data[0].z[0].z;
                pie.title = 'SUV Sales by Years';
                pie.series[0].name = 'Level 3';
                grid.columns[0].headerText = 'Year';
                grid.refresh();
                pie.refresh();
                break;
        }
        grid.dataSource = pie.series[0].dataSource;
    }
}
;
}
ReactDOM.render(<Drilldown />, document.getElementById("charts"));
{% endraw %}

```

INDEX.TSX

```

{% raw %}
import * as ReactDOM from 'react-dom';
import * as React from 'react';
import { Browser, EmitType } from '@syncfusion/ej2-base';
import {
    AccumulationChartComponent, AccumulationSeriesCollectionDirective,
    AccumulationSeriesDirective, indexFinder, Index,
    Inject, AccumulationDataLabel, AccumulationTooltip, PieSeries,
    IAccLoadedEventArgs, IAccTextRenderEventArgs,
    IMouseEventArgs, AccumulationAnnotation, AccumulationTheme,
    AccumulationChart,
} from '@syncfusion/ej2-react-charts';
const SAMPLE_CSS = `

```

```

#category: hover {
  cursor: pointer;
} `;
function Drilldown() {
  let data: Object[] = [
    { x: 'SUV', y: 25 }, { x: 'Car', y: 37 }, { x: 'Pickup', y: 15 },
    { x: 'Minivan', y: 23 }
  ];
  let suvs: Object = [{ x: 'Toyota', y: 8 }, { x: 'Ford', y: 12 }, { x:
'GM', y: 17 }, { x: 'Renault', y: 6 }, { x: 'Fiat', y: 3 },
  { x: 'Hyundai', y: 16 }, { x: 'Honda', y: 8 }, { x: 'Maruthi', y: 10 },
  { x: 'BMW', y: 20 }];
  let cars: Object = [{ x: 'Toyota', y: 7 }, { x: 'Chrysler', y: 12 }, {
x: 'Nissan', y: 9 }, { x: 'Ford', y: 15 },
  { x: 'Tata', y: 10 },
  { x: 'Mahindra', y: 7 }, { x: 'Renault', y: 8 }, { x: 'Skoda', y: 5 }, {
x: 'Volkswagen', y: 15 }, { x: 'Fiat', y: 3 }];
  let pickups: Object = [{ x: 'Nissan', y: 9 }, { x: 'Chrysler', y: 4 }, {
x: 'Ford', y: 7 }, { x: 'Toyota', y: 20 },
  { x: 'Suzuki', y: 13 }, { x: 'Lada', y: 12 }, { x: 'Bentley', y: 6 }, {
x: 'Volvo', y: 10 }, { x: 'Audi', y: 19 }];
  let minivans: Object = [{ x: 'Hummer', y: 11 }, { x: 'Ford', y: 5 }, {
x: 'GM', y: 12 }, { x: 'Chrysler', y: 3 },
  { x: 'Jaguar', y: 9 },
  { x: 'Fiat', y: 8 }, { x: 'Honda', y: 15 }, { x: 'Hyundai', y: 4 }, { x:
'Scion', y: 11 }, { x: 'Toyota', y: 17 }];
  let dataLabel: Object = {
    visible: true, position: 'Inside', enableRotation: false,
connectorStyle: { type: 'Curve', length: '5%' }, font: { fontWeight: '600',
color: 'white' }
  };
  let startAngle: number = 0;
  let endAngle: number = 360;
  let title: string = 'Automobile Sales by Category';
  let pie: AccumulationChart;
  let isparent: boolean = true;
  return (
    <div className='control-pane'>
      <style>
        {SAMPLE_CSS}
      </style>
      <div className='control-section'>
        <div id="link">
          <a id="category" onClick={onClick.bind(this)}
style={{ visibility: 'hidden', display: 'inline-block' }}>
            Sales by Category
          </a>
          <p style={{ visibility: 'hidden', display: 'inline-block' }} id="symbol">
            &#160;&#62;&#62;&#160;</p>
          <p id="text" style={{ display: 'inline-block'
}}></p>
        </div>
        <AccumulationChartComponent id='pie-chart' ref={ (scope)
=> { pie = scope } }
          title='Automobile Sales by Category'
          enableSmartLabels={false} legendSettings={{ visible: false }}
          enableBorderOnMouseMove={false}

```

```

        tooltip={{ enable: false, format: '${point.x} <br>
        ${point.y} %' }}

        chartMouseClick={onChartMouseClick.bind(this) }
        textRender={onTextRender.bind(this) }
        load={load.bind(this) }
        loaded={onChartLoad.bind(this) }
    >
        <Inject services={[AccumulationDataLabel,
AccumulationTooltip, PieSeries, AccumulationAnnotation]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={data}
xName='x' yName='y' dataLabel={dataLabel} radius='70%'
            explode={false} >
                </AccumulationSeriesDirective>
            </AccumulationSeriesCollectionDirective>
        </AccumulationChartComponent>
    </div>
</div>
)
function onTextRender(args: IAccTextRenderEventArgs): void {
    args.text = args.point.x + ' ' + args.point.y + '%';
}
function onChartMouseClick(args: IMouseEventArgs): void {
    let index: Index = indexFinder(args.target);
    if (isparent && document.getElementById('pie-chart_Series_' +
index.series + '_Point_' + index.point)) {
        isparent = false;
        switch (index.point) {
            case 0:
                pie.series[0].dataSource = suvs;
                pie.title = 'Automobile Sales in the SUV Segment';
                document.getElementById('text').innerHTML = 'SUV';
                break;
            case 1:
                pie.series[0].dataSource = cars;
                pie.title = 'Automobile Sales in the Car Segment';
                document.getElementById('text').innerHTML = 'Car';
                break;
            case 2:
                pie.series[0].dataSource = pickups;
                pie.title = 'Automobile Sales in the Pickup Segment';
                document.getElementById('text').innerHTML = 'Pickup';
                break;
            case 3:
                pie.series[0].dataSource = minivans;
                pie.title = 'Automobile Sales in the Minivan Segment';
                document.getElementById('text').innerHTML = 'Minivan';
                break;
        }
        pie.annotations = [{
            content: '<div id="back" style="cursor:pointer; padding:
3px; width: 30px; height: 30px;">' +
                '', region: 'Series', x: '50%', y: '50%'
        }];
        pie.series[0].innerRadius = '30%';
        pie.series[0].radius = Browser.isDevice ? '90%' : '80%';
    }
}

```

```

        pie.series[0].explode = false;
        pie.series[0].dataLabel.connectorStyle.length = '20px'
        pie.series[0].dataLabel.position = Browser.isDevice ? 'Inside' :
'Outside';
        pie.series[0].dataLabel.position = Browser.isDevice ? 'Inside' :
'Outside';
        pie.series[0].dataLabel.enableRotation = true;
        pie.series[0].dataLabel.font.color = '';
        pie.legendSettings.visible = false;
        pie.visibleSeries[0].explodeIndex = null;
        pie.enableSmartLabels = true;
        pie.refresh();
        document.getElementById('category').style.visibility =
'visible';
        document.getElementById('symbol').style.visibility = 'visible';
        document.getElementById('text').style.visibility = 'visible';
    }
    if (args.target.indexOf('back') > -1) {
        hide(document.getElementById(args.target));
    }
}
function onClick(e: MouseEvent): void {
    hide(e.target as Element);
}
function hide(target: Element): void {
    pie.series[0].dataSource = data;
    pie.series[0].innerRadius = '0%';
    isparent = true;
    pie.series[0].explode = false;
    pie.annotations = [];
    pie.annotationModule['annotations'] = [];
    pie.series[0].dataLabel = dataLabel;
    pie.title = title;
    pie.legendSettings.visible = false;
    //pie.visibleSeries[0].explodeIndex = explodeIndex;
    pie.enableSmartLabels = false;
    pie.refresh();
    (target as HTMLButtonElement).style.visibility = 'hidden';
    document.getElementById('category').style.visibility = 'hidden';
    document.getElementById('symbol').style.visibility = 'hidden';
    document.getElementById('text').style.visibility = 'hidden';
}
function onChartLoad(args: IAccLoadedEventArgs): void {
    document.getElementById('pie-chart').setAttribute('title', '');
};
function load(args: IAccLoadedEventArgs): void {
    let selectedTheme: string = location.hash.split('/')[1];
    selectedTheme = selectedTheme ? selectedTheme : 'Material';
    args.accumulation.theme = (selectedTheme.charAt(0).toUpperCase() +
selectedTheme.slice(1)).
        replace(/-dark/i, "Dark") as AccumulationTheme;
    if (selectedTheme === 'highcontrast' ||
selectedTheme.indexOf('dark') > -1) {
        args.accumulation.series[0].dataLabel.font.color = "white";
        if (args.accumulation.annotations[0] && !isparent) {

```

```

        args.accumulation.annotations[0].content = '<div id= "white"
style="cursor:pointer;padding:3px;width:30px; height:30px;"></div>';
    }
    };
}
export default Drilldown;
ReactDOM.render(<Drilldown />, document.getElementById('charts'));
{% enddraw %}

```

See Also

- [Data label](#)
- [Grouping](#)

Pyramid in React Accumulation chart component

To render a pyramid series, use the series [type](#) as `Pyramid` and inject `PyramidSeries` module into the services.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend, PyramidSeries,
AccumulationTooltip, AccumulationDataLabel } from '@syncfusion/ej2-react-
charts';
import { pyramidData } from 'datasource.ts';
function App() {
    return <AccumulationChartComponent id='charts'>
        <Inject services={[AccumulationLegend, PyramidSeries,
AccumulationTooltip, AccumulationDataLabel]}/>
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={pyramidData} xName='x'
yName='y' type='Pyramid'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import {
    AccumulationChartComponent, AccumulationSeriesCollectionDirective,
    AccumulationSeriesDirective,
    Inject, AccumulationLegend, PyramidSeries, AccumulationTooltip,
    IAccLoadedEventArgs,
    AccumulationDataLabel, IAccResizeEventArgs, AccumulationTheme
}

```

```

} from '@syncfusion/ej2-react-charts';
import { pyramidData } from 'datasource.ts';
function App() {
    return <AccumulationChartComponent id='charts'>
        <Inject services={[AccumulationLegend, PyramidSeries,
AccumulationTooltip, AccumulationDataLabel]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={pyramidData} xName='x'
yName='y' type='Pyramid'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Syncfusion React Accumulation Series</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" />
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></sc
ript>
    <script src="systemjs.config.js"></script>
    <style>
        #loader {
            color: #008cff;
            height: 40px;
            left: 45%;
            position: absolute;
            top: 45%;
            width: 30%;
        }
        #charts {
            height : 350px;
            display: block;
        }
    </style>
</head>
<body>
    <div id='charts'>
        <div id='loader'>Loading....</div>
    </div>
</body>
</html>

```


Mode

The Pyramid chart supports linear and surface modes of rendering. The default type of the `pyramidMode` is `linear`.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend, PyramidSeries,
AccumulationTooltip, AccumulationDataLabel } from '@syncfusion/ej2-react-
charts';
import { pyramidData } from 'datasource.ts';
function App() {
    return <AccumulationChartComponent id='charts'>
        <Inject services={[AccumulationLegend, PyramidSeries,
AccumulationTooltip, AccumulationDataLabel]}/>
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={pyramidData} xName='x'
yName='y' type='Pyramid' pyramidMode='Surface'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import {
    AccumulationChartComponent, AccumulationSeriesCollectionDirective,
    AccumulationSeriesDirective,
    Inject, AccumulationLegend, PyramidSeries, AccumulationTooltip,
    IAccLoadedEventArgs,
    AccumulationDataLabel, IAccResizeEventArgs, AccumulationTheme
} from '@syncfusion/ej2-react-charts';
import { pyramidData } from 'datasource.ts';
function App() {
    return <AccumulationChartComponent id='charts'>
        <Inject services={[AccumulationLegend, PyramidSeries,
AccumulationTooltip, AccumulationDataLabel]}/>
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={pyramidData} xName='x'
yName='y' type='Pyramid'
            pyramidMode='Surface' >
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Accumulation Series</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></sc
ript>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
    #charts {
      height : 350px;
      display: block;
    }
  </style>
</head>
<body>
  <div id='charts'>
    <div id='loader'>Loading....</div>
  </div>
</body>
</html>

```

Size

The size of the pyramid chart can be customized by using the **width** and **height** properties.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend, PyramidSeries,
AccumulationTooltip, AccumulationDataLabel } from '@syncfusion/ej2-react-
charts';
import { pyramidData } from 'datasource.ts';
function App() {
  return <AccumulationChartComponent id='charts'>
    <Inject services={[AccumulationLegend, PyramidSeries,
AccumulationTooltip, AccumulationDataLabel]}/>
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={pyramidData} xName='x'
yName='y' type='Pyramid' width='60%' height='80%'>

```

```

        </AccumulationSeriesDirective>
      </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import {
  AccumulationChartComponent, AccumulationSeriesCollectionDirective,
  AccumulationSeriesDirective,
  Inject, AccumulationLegend, PyramidSeries, AccumulationTooltip,
  IAccLoadedEventArgs,
  AccumulationDataLabel, IAccResizeEventArgs, AccumulationTheme
} from '@syncfusion/ej2-react-charts';
import { pyramidData } from 'datasource.ts';
function App() {
  return <AccumulationChartComponent id='charts'>
    <Inject services={[AccumulationLegend, PyramidSeries,
    AccumulationTooltip, AccumulationDataLabel]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={pyramidData} xName='x'
yName='y' type='Pyramid' width='60%' height='80%'>
    </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Accumulation Series</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></sc
ript>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;

```

```

        width: 30%;
    }
    #charts {
        height : 350px;
        display: block;
    }
</style>
</head>
<body>
    <div id='charts'>
        <div id='loader'>Loading....</div>
    </div>
</body>
</html>

```

Gap Between the Segments

Pyramid chart provides options to customize the space between the segments by using the `gapRatio` property of the series. It ranges from 0 to 1.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend, PyramidSeries,
AccumulationTooltip, AccumulationDataLabel } from '@syncfusion/ej2-react-
charts';
import { pyramidData } from 'datasource.ts';
function App() {
    return <AccumulationChartComponent id='charts'>
        <Inject services={[AccumulationLegend, PyramidSeries,
AccumulationTooltip, AccumulationDataLabel]}/>
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={pyramidData} xName='x'
yName='y' type='Pyramid' gapRatio={0.2}>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import {
    AccumulationChartComponent, AccumulationSeriesCollectionDirective,
    AccumulationSeriesDirective,
    Inject, AccumulationLegend, PyramidSeries, AccumulationTooltip,
    IAccLoadedEventArgs,
    AccumulationDataLabel, IAccResizeEventArgs, AccumulationTheme
} from '@syncfusion/ej2-react-charts';
import { pyramidData } from 'datasource.ts';

```

```
function App() {
  return <AccumulationChartComponent id='charts'>
    <Inject services={[AccumulationLegend, PyramidSeries,
AccumulationTooltip, AccumulationDataLabel]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={pyramidData} xName='x'
yName='y' type='Pyramid' gapRatio={0.2}>
    </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Accumulation Series</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></sc
ript>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
    #charts {
      height : 350px;
      display: block;
    }
  </style>
</head>
<body>
  <div id='charts'>
    <div id='loader'>Loading....</div>
  </div>
</body>
</html>
```

Explode

Points can be exploded on mouse click by setting the `explode` property to true. You can also explode the point on load using `explodeIndex`. Explode distance can be set by using `explodeOffset` property.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend, PyramidSeries,
AccumulationTooltip, AccumulationDataLabel } from '@syncfusion/ej2-react-
charts';
import { pyramidData } from 'datasource.ts';
function App() {
    return <AccumulationChartComponent id='charts'>
        <Inject services={[AccumulationLegend, PyramidSeries,
AccumulationTooltip, AccumulationDataLabel]}/>
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={pyramidData} xName='x'
yName='y' type='Pyramid' explode={true} explodeOffset='10'
explodeAll={false} explodeIndex={3}>
                </AccumulationSeriesDirective>
            </AccumulationSeriesCollectionDirective>
        </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import {
    AccumulationChartComponent, AccumulationSeriesCollectionDirective,
    AccumulationSeriesDirective,
    Inject, AccumulationLegend, PyramidSeries, AccumulationTooltip,
    IAccLoadedEventArgs,
    AccumulationDataLabel, IAccResizeEventArgs, AccumulationTheme
} from '@syncfusion/ej2-react-charts';
import { pyramidData } from 'datasource.ts';
function App() {
    return <AccumulationChartComponent id='charts'>
        <Inject services={[AccumulationLegend, PyramidSeries,
AccumulationTooltip, AccumulationDataLabel]}/>
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={pyramidData} xName='x'
yName='y' type='Pyramid'
                explode={true} explodeOffset='10' explodeAll={false}
explodeIndex={3}>
                </AccumulationSeriesDirective>
            </AccumulationSeriesCollectionDirective>
        </AccumulationChartComponent>
    </>;
}
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.HTML

```
<!DOCTYPE html>
```

```

<html lang="en">
<head>
  <title>Syncfusion React Accumulation Series</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></sc
ript>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
    #charts {
      height : 350px;
      display: block;
    }
  </style>
</head>
<body>
  <div id='charts'>
    <div id='loader'>Loading....</div>
  </div>
</body>
</html>

```

Customization

Individual points can be customized using the `pointRender` event.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend, PyramidSeries,
AccumulationTooltip, AccumulationDataLabel } from '@syncfusion/ej2-react-
charts';
import { pyramidData } from 'datasource.ts';
function App() {
  const onPointRender = (args) => {
    if (args.point.x.indexOf('India') > -1) {
      args.fill = '#f4bc42';
    }
    else {
      args.fill = '#597cf9';
    }
  };
};

```

```

    return <AccumulationChartComponent id='charts'
pointRender={onPointRender}>
    <Inject services={[AccumulationLegend, PyramidSeries,
AccumulationTooltip, AccumulationDataLabel]}/>
    <AccumulationSeriesCollectionDirective>
        <AccumulationSeriesDirective dataSource={pyramidData} xName='x'
yName='y' type='Pyramid' gapRatio={0.2}>
        </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
</AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import {
    AccumulationChartComponent, AccumulationSeriesCollectionDirective,
    AccumulationSeriesDirective,
    Inject, AccumulationLegend, PyramidSeries, AccumulationTooltip,
    IAccLoadedEventArgs,
    AccumulationDataLabel, IAccResizeEventArgs, AccumulationTheme,
    IAccPointRenderEventArgs
} from '@syncfusion/ej2-react-charts';
import { pyramidData } from 'datasource.ts';
function App() {
    const onPointRender: EmitType<IAccPointRenderEventArgs> = (args:
    IAccPointRenderEventArgs): void => {
        if ((args.point.x as string).indexOf('India') > -1) {
            args.fill = '#f4bc42';
        }
        else {
            args.fill = '#597cf9';
        }
    };
    return <AccumulationChartComponent id='charts'
pointRender={onPointRender}>
        <Inject services={[AccumulationLegend, PyramidSeries,
AccumulationTooltip, AccumulationDataLabel]}/>
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={pyramidData} xName='x'
yName='y' type='Pyramid'
            gapRatio={0.2}>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.HTML

```
<!DOCTYPE html>
```



```

<html lang="en">
<head>
  <title>Syncfusion React Accumulation Series</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></sc
ript>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
    #charts {
      height : 350px;
      display: block;
    }
  </style>
</head>
<body>
  <div id='charts'>
    <div id='loader'>Loading....</div>
  </div>
</body>
</html>

```

See Also

- [Data label](#)
- [Grouping](#)

Funnel in React Accumulation chart component

To render a funnel series, use the series [type](#) as `Funnel` and inject the `FunnelSeries` module into the services.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend, FunnelSeries,
AccumulationTooltip, AccumulationDataLabel } from '@syncfusion/ej2-react-
charts';
import { data1 } from 'datasource.ts';
function App() {
  return <AccumulationChartComponent id='charts'>

```

```

    <Inject services={[AccumulationLegend, FunnelSeries,
AccumulationTooltip, AccumulationDataLabel]}/>
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={data1} xName='x' yName='y'
type='Funnel'>
        </AccumulationSeriesDirective>
      </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import {
  AccumulationChartComponent, AccumulationSeriesCollectionDirective,
  AccumulationSeriesDirective, Inject, AccumulationLegend, FunnelSeries, AccumulationTooltip,
  IAccLoadedEventArgs,
  AccumulationDataLabel, IAccResizeEventArgs, AccumulationTheme
} from '@syncfusion/ej2-react-charts';
import { data1 } from 'datasource.ts';
function App() {
  return <AccumulationChartComponent id='charts'>
    <Inject services={[AccumulationLegend, FunnelSeries,
AccumulationTooltip, AccumulationDataLabel]}/>
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={data1} xName='x' yName='y'
type='Funnel'>
        </AccumulationSeriesDirective>
      </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
  };
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Size

The size of the funnel chart can be customized by using the `width` and `height` properties.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
  AccumulationSeriesDirective, Inject, AccumulationLegend, FunnelSeries,
  AccumulationTooltip, AccumulationDataLabel } from '@syncfusion/ej2-react-
charts';
import { data1 } from 'datasource.ts';
function App() {
  return <AccumulationChartComponent id='charts'>
    <Inject services={[AccumulationLegend, FunnelSeries,
AccumulationTooltip, AccumulationDataLabel]}/>

```

```

        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={data1} xName='x' yName='y'
type='Funnel' width='60%' height='80%'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import {
    AccumulationChartComponent, AccumulationSeriesCollectionDirective,
    AccumulationSeriesDirective,
    Inject, AccumulationLegend, FunnelSeries, AccumulationTooltip,
    IAccLoadedEventArgs,
    AccumulationDataLabel, IAccResizeEventArgs, AccumulationTheme
} from '@syncfusion/ej2-react-charts';
import { data1 } from 'datasource.ts';
function App() {
    return <AccumulationChartComponent id='charts'>
        <Inject services={[AccumulationLegend, FunnelSeries,
AccumulationTooltip, AccumulationDataLabel]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={data1} xName='x' yName='y'
type='Funnel' width='60%' height='80%'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Syncfusion React Accumulation Series</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" />
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></sc
ript>
    <script src="systemjs.config.js"></script>
    <style>
        #loader {
            color: #008cff;
            height: 40px;

```

```

        left: 45%;
        position: absolute;
        top: 45%;
        width: 30%;
    }
    #charts {
        height : 350px;
        display: block;
    }
</style>
</head>
<body>
    <div id='charts'>
        <div id='loader'>Loading....</div>
    </div>
</body>
</html>

```

Neck Size

The funnel's neck size can be customized by using the `neckWidth` and `neckHeight` properties.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend, FunnelSeries,
AccumulationTooltip, AccumulationDataLabel } from '@syncfusion/ej2-react-
charts';
import { data1 } from 'datasource.ts';
function App() {
    return <AccumulationChartComponent id='charts'>
        <Inject services={[AccumulationLegend, FunnelSeries,
AccumulationTooltip, AccumulationDataLabel]}/>
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={data1} xName='x' yName='y'
type='Funnel' neckWidth='25%' neckHeight='5%'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import {
    AccumulationChartComponent, AccumulationSeriesCollectionDirective,
    AccumulationSeriesDirective,
    Inject, AccumulationLegend, FunnelSeries, AccumulationTooltip,
    IAccLoadedEventArgs,
    AccumulationDataLabel, IAccResizeEventArgs, AccumulationTheme
}

```

```

} from '@syncfusion/ej2-react-charts';
import { data1 } from 'datasource.ts';
function App() {
    return <AccumulationChartComponent id='charts'>
        <Inject services={[AccumulationLegend, FunnelSeries,
AccumulationTooltip, AccumulationDataLabel]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={data1} xName='x' yName='y'
type='Funnel'
            neckWidth='25%' neckHeight='5%'>
        </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Syncfusion React Accumulation Series</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" />
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></sc
ript>
    <script src="systemjs.config.js"></script>
    <style>
        #loader {
            color: #008cff;
            height: 40px;
            left: 45%;
            position: absolute;
            top: 45%;
            width: 30%;
        }
        #charts {
            height : 350px;
            display: block;
        }
    </style>
</head>
<body>
    <div id='charts'>
        <div id='loader'>Loading....</div>
    </div>
</body>
</html>

```

Gap between the segments

Funnel chart provides options to customize the space between the segments by using the `gapRatio` property of the series. It ranges from 0 to 1.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend, FunnelSeries,
AccumulationTooltip, AccumulationDataLabel } from '@syncfusion/ej2-react-charts';
import { data1 } from 'datasource.ts';
function App() {
    return <AccumulationChartComponent id='charts'>
        <Inject services={[AccumulationLegend, FunnelSeries,
AccumulationTooltip, AccumulationDataLabel]}/>
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={data1} xName='x' yName='y'
type='Funnel' gapRatio={0.08}>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import {
    AccumulationChartComponent, AccumulationSeriesCollectionDirective,
    AccumulationSeriesDirective,
    Inject, AccumulationLegend, FunnelSeries, AccumulationTooltip,
    IAccLoadedEventArgs,
    AccumulationDataLabel, IAccResizeEventArgs, AccumulationTheme
} from '@syncfusion/ej2-react-charts';
import { data1 } from 'datasource.ts';
function App() {
    return <AccumulationChartComponent id='charts'>
        <Inject services={[AccumulationLegend, FunnelSeries,
AccumulationTooltip, AccumulationDataLabel]}/>
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={data1} xName='x' yName='y'
type='Funnel'
            gapRatio={0.08}>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Accumulation Series</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></sc
ript>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
    #charts {
      height : 350px;
      display: block;
    }
  </style>
</head>
<body>
  <div id='charts'>
    <div id='loader'>Loading....</div>
  </div>
</body>
</html>

```

Explode

Points can be exploded on mouse click by setting the **explode** property to true. You can also explode the point on load using **explodeIndex**. Explode distance can be set by using **explodeOffset** property.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend, FunnelSeries,
AccumulationTooltip, AccumulationDataLabel } from '@syncfusion/ej2-react-
charts';
import { data1 } from 'datasource.ts';
function App() {
  return <AccumulationChartComponent id='charts'>
    <Inject services={[AccumulationLegend, FunnelSeries,
AccumulationTooltip, AccumulationDataLabel]}/>
    <AccumulationSeriesCollectionDirective>

```

```

        <AccumulationSeriesDirective dataSource={data1} xName='x' yName='y'
type='Funnel' explode={true} explodeOffset='10' explodeAll={false}
explodeIndex={3}>
        </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
</AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import {
    AccumulationChartComponent, AccumulationSeriesCollectionDirective,
    AccumulationSeriesDirective,
    Inject, AccumulationLegend, FunnelSeries, AccumulationTooltip,
    IAccLoadedEventArgs,
    AccumulationDataLabel, IAccResizeEventArgs, AccumulationTheme
} from '@syncfusion/ej2-react-charts';
import { data1 } from 'datasource.ts';
function App() {
    return <AccumulationChartComponent id='charts'>
        <Inject services={[AccumulationLegend, FunnelSeries,
    AccumulationTooltip, AccumulationDataLabel]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={data1} xName='x' yName='y'
type='Funnel'
                explode={true} explodeOffset='10' explodeAll={false}
explodeIndex={3}>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Syncfusion React Accumulation Series</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" />
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></sc
ript>
    <script src="systemjs.config.js"></script>
    <style>
        #loader {

```



```

        color: #008cff;
        height: 40px;
        left: 45%;
        position: absolute;
        top: 45%;
        width: 30%;
    }
    #charts {
        height : 350px;
        display: block;
    }
</style>
</head>
<body>
    <div id='charts'>
        <div id='loader'>Loading....</div>
    </div>
</body>
</html>

```

Smart Data Label

It provides the data label smart arrangement of the funnel and pyramid series. The overlap data label will be placed on left side of the funnel/pyramid series.

INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend, FunnelSeries,
AccumulationTooltip, AccumulationDataLabel } from '@syncfusion/ej2-react-
charts';
import { funnelData } from 'datasource.ts';
function App() {
    function onLoad(args) {
        if (args.accumulation.availableSize) {
            if (args.accumulation.availableSize.width <
args.accumulation.availableSize.height) {
                args.accumulation.series[0].width = '80%';
                args.accumulation.series[0].height = '70%';
            }
        }
    }
    function onChartResized(args) {
        let bounds =
document.getElementById('charts').getBoundingClientRect();
        if (bounds.width < bounds.height) {
            args.accumulation.series[0].width = '80%';
            args.accumulation.series[0].height = '70%';
        }
        else {
            args.accumulation.series[0].width = '60%';
            args.accumulation.series[0].height = '80%';
        }
    }
}

```

```

    return <AccumulationChartComponent id='charts' legendSettings={{
visible: false }} tooltip={{ enable: true, format: '${point.x} :
<b>${point.y}</b>' }} title='Top population countries in the world 2017'
resized={onChartResized.bind(this)} load={onLoad.bind(this)}>
    <Inject services={[AccumulationLegend, FunnelSeries,
AccumulationTooltip, AccumulationDataLabel]} />
    <AccumulationSeriesCollectionDirective>
        <AccumulationSeriesDirective dataSource={funnelData} xName='x'
yName='y' type='Funnel' name='2017 Population' dataLabel={{
            visible: true, position: 'Outside',
            connectorStyle: { length: '6%' }, name: 'text'
        }} explode={false}>
        </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
</AccumulationChartComponent>;
}
;
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 {
    AccumulationChartComponent, AccumulationSeriesCollectionDirective,
    AccumulationSeriesDirective,
    Inject, AccumulationLegend, FunnelSeries, AccumulationTooltip,
    IAccLoadedEventArgs,
    AccumulationDataLabel, IAccResizeEventArgs
} from '@syncfusion/ej2-react-charts';
import { funnelData } from 'datasource.ts';
function App() {
    function onLoad(args: IAccLoadedEventArgs): void {
        if (args.accumulation.availableSize) {
            if (args.accumulation.availableSize.width <
args.accumulation.availableSize.height) {
                args.accumulation.series[0].width = '80%';
                args.accumulation.series[0].height = '70%';
            }
        }
    }
    function onChartResized(args: IAccResizeEventArgs): void {
        let bounds: ClientRect =
document.getElementById('charts').getBoundingClientRect();
        if (bounds.width < bounds.height) {
            args.accumulation.series[0].width = '80%';
            args.accumulation.series[0].height = '70%';
        } else {
            args.accumulation.series[0].width = '60%';
            args.accumulation.series[0].height = '80%';
        }
    }
}

```

```

    return <AccumulationChartComponent id='charts' legendSettings={{ visible:
false }} tooltip={{ enable: true, format: '${point.x} : <b>${point.y}</b>'
}} title='Top population countries in the world 2017'
resized={onChartResized.bind(this)} load={onLoad.bind(this)}>
    <Inject services={[AccumulationLegend, FunnelSeries,
AccumulationTooltip, AccumulationDataLabel]} />
    <AccumulationSeriesCollectionDirective>
        <AccumulationSeriesDirective dataSource={funnelData} xName='x'
yName='y' type='Funnel' name='2017 Population' dataLabel={{
    visible: true, position: 'Outside',
    connectorStyle: { length: '6%' }, name: 'text',
    }} explode={false}>
        </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
</AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
[% endraw %]

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Syncfusion React Accumulation Series</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" />
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></sc
ript>
    <script src="systemjs.config.js"></script>
    <style>
        #loader {
            color: #008cff;
            height: 40px;
            left: 45%;
            position: absolute;
            top: 45%;
            width: 30%;
        }
        #charts {
            height : 350px;
            display: block;
        }
    </style>
</head>
<body>
    <div id='charts'>
        <div id='loader'>Loading....</div>
    </div>
</body>
</html>

```

Customization

Individual points can be customized using the `pointRender` event.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend, FunnelSeries,
AccumulationTooltip, AccumulationDataLabel } from '@syncfusion/ej2-react-
charts';
import { data1 } from 'datasource.ts';
function App() {
    const onPointRender = (args) => {
        if (args.point.x.indexOf('Downloaded') > -1) {
            args.fill = '#f4bc42';
        }
        else {
            args.fill = '#597cf9';
        }
    };
    return <AccumulationChartComponent id='charts'
pointRender={onPointRender}>
        <Inject services={[AccumulationLegend, FunnelSeries,
AccumulationTooltip, AccumulationDataLabel]}/>
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={data1} xName='x' yName='y'
type='Funnel' gapRatio={0.08}>
                </AccumulationSeriesDirective>
            </AccumulationSeriesCollectionDirective>
        </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import {
    AccumulationChartComponent, AccumulationSeriesCollectionDirective,
    AccumulationSeriesDirective,
    Inject, AccumulationLegend, FunnelSeries, AccumulationTooltip,
    IAccLoadedEventArgs,
    AccumulationDataLabel, IAccResizeEventArgs, AccumulationTheme,
    IAccPointRenderEventArgs
} from '@syncfusion/ej2-react-charts';
import { data1 } from 'datasource.ts';
function App() {
    const onPointRender: EmitType<IAccPointRenderEventArgs> = (args:
IAccPointRenderEventArgs): void => {
        if ((args.point.x as string).indexOf('Downloaded') > -1) {
            args.fill = '#f4bc42';
        }
    }
}
```

```

    else {
      args.fill = '#597cf9';
    }
  };
  return <AccumulationChartComponent id='charts'
pointRender={onPointRender}>
    <Inject services={[AccumulationLegend, FunnelSeries,
AccumulationTooltip, AccumulationDataLabel]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={data1} xName='x' yName='y'
type='Funnel'
        gapRatio={0.08}>
        </AccumulationSeriesDirective>
      </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
  };
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Accumulation Series</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></sc
ript>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
    #charts {
      height : 350px;
      display: block;
    }
  </style>
</head>
<body>
  <div id='charts'>
    <div id='loader'>Loading....</div>
  </div>
</body>
</html>

```

See Also

- [Data label](#)
- [Grouping](#)

Data label in React Accumulation chart component

Data label can be added to a chart series by enabling the [visible](#) option in the dataLabel property.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationDataLabel } from
'@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    const datalabel = { visible: true };
    return <AccumulationChartComponent id='charts' enableSmartLabels='true'>
        <Inject services={[AccumulationDataLabel]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' dataLabel={datalabel}></AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective,
Inject, AccumulationDataLabel, AccumulationDataLabelSettingsModel }
from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    const datalabel: AccumulationDataLabelSettingsModel = { visible: true }
    return <AccumulationChartComponent id='charts' enableSmartLabels='true'>
        <Inject services={[AccumulationDataLabel]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y'
                dataLabel={datalabel}></AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Note: To use the data label feature, inject the `DataLabel` module into the `services`.

Positioning

Accumulation chart provides support for placing the data label either **inside** or **outside** the chart.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
  AccumulationSeriesDirective, Inject, AccumulationDataLabel } from
  '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
  const datalabel = { visible: true, position: 'Outside' };
  return <AccumulationChartComponent id='charts' enableSmartLabels='true'>
    <Inject services={[AccumulationDataLabel]}/>
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' dataLabel={datalabel}></AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
  AccumulationSeriesDirective,
  Inject, AccumulationDataLabel, AccumulationDataLabelSettingsModel }
  from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
  const datalabel: AccumulationDataLabelSettingsModel = { visible: true,
position: 'Outside' };
  return <AccumulationChartComponent id='charts' enableSmartLabels='true'>
    <Inject services={[AccumulationDataLabel]}/>
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y'
      dataLabel={datalabel}></AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

DataLabel rotation

Using **angle** property, you can rotate the data label by its given angle.

INDEX.JSX

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

```
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationDataLabel } from
'@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    const datalabel = { visible: true, angle: 90, enableRotation: true };
    return <AccumulationChartComponent id='charts' enableSmartLabels='true'>
        <Inject services={[AccumulationDataLabel]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' dataLabel={datalabel}></AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective,
Inject, AccumulationDataLabel, AccumulationDataLabelSettingsModel }
from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    const datalabel: AccumulationDataLabelSettingsModel = { visible: true,
angle: 90, enableRotation: true };

    return <AccumulationChartComponent id='charts' enableSmartLabels='true'>
        <Inject services={[AccumulationDataLabel]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y'
            dataLabel={datalabel}></AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Note: By default, when the `enableRotation` is true, the datalabel is rotated along the slice.

Smart labels

Data labels will be arranged smartly without overlapping with each other. You can enable or disable this feature using the [enableSmartLabels](#) property.

INDEX.JSX

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



```
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationDataLabel } from
'@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    const datalabel = { visible: true, name: 'text', position: 'Outside' };
    return <AccumulationChartComponent id='charts' enableSmartLabels='true'>
        <Inject services={[AccumulationDataLabel]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' dataLabel={datalabel}></AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective,
Inject, AccumulationDataLabel, AccumulationDataLabelSettingsModel }
from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    const datalabel: AccumulationDataLabelSettingsModel = { visible: true,
name: 'text', position: 'Outside' };
    return <AccumulationChartComponent id='charts' enableSmartLabels='true'>
        <Inject services={[AccumulationDataLabel]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y'
            dataLabel={datalabel}></AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Format

Data label for the accumulation chart can be formatted using [format](#) property. You can use the global formatting options, such as 'n', 'p', and 'c'.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationDataLabel } from
'@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
```

```

const datalabel = { visible: true, format: 'n2' };
return <AccumulationChartComponent id='charts' enableSmartLabels='true'>
  <Inject services={[AccumulationDataLabel]}/>
  <AccumulationSeriesCollectionDirective>
    <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' dataLabel={datalabel}></AccumulationSeriesDirective>
  </AccumulationSeriesCollectionDirective>
</AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective,
Inject, AccumulationDataLabel, AccumulationDataLabelSettingsModel }
from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
  const datalabel: AccumulationDataLabelSettingsModel = { visible: true,
format: 'n2' };
  return <AccumulationChartComponent id='charts' enableSmartLabels='true'>
    <Inject services={[AccumulationDataLabel]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y'
      dataLabel={datalabel}></AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Value	Format	Resultant Value	Description
1000	n1	1000.0	The number is rounded to 1 decimal place.
1000	n2	1000.00	The number is rounded to 2 decimal places.
1000	n3	1000.000	The number is rounded to 3 decimal place.
0.01	p1	1.0%	The number is converted to percentage with 1 decimal place.
0.01	p2	1.00%	The number is converted to percentage with 2 decimal place.
0.01	p3	1.000%	The number is converted to percentage with 3 decimal place.
1000	c1	\$1000.0	The currency symbol is appended to number and number is rounded to 1 decimal place.

1000	c2	\$1000.00	The currency symbol is appended to number and number is rounded to 2 decimal place.
------	----	-----------	---

Datalabel template

Label content can be formatted by using the template option. Inside the template, you can add the placeholder text `${point.x}` and `${point.y}` to display corresponding data points x & y value. Using [template](#) property, you can set data label template in chart.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationDataLabel } from
'@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    const template = chartTemplate;
    const datalabel = { visible: true, name: 'text', position: 'Outside',
template: template };
    function chartTemplate(args) {
        return (<div id="templateWrap">
            <div>{args.point.x}</div>
            <div>{args.point.y}</div>
        </div>);
    }
    return <AccumulationChartComponent id='charts'>
        <Inject services={[AccumulationDataLabel]}/>
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' dataLabel={datalabel}>
        </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective,
Inject, AccumulationDataLabel, AccumulationDataLabelSettingsModel}
from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    const template:any = chartTemplate;
    const datalabel: AccumulationDataLabelSettingsModel = { visible: true,
name: 'text', position: 'Outside', template: template };
    function chartTemplate(args:any) {
        return (<div id="templateWrap">
```

```

        <div>{args.point.x}</div>
        <div>{args.point.y}</div>
    </div>);
}
return <AccumulationChartComponent id='charts'>
    <Inject services={[AccumulationDataLabel]} />
    <AccumulationSeriesCollectionDirective>
        <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' dataLabel={datalabel}>
        </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
</AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Connector Line

Connector line will be visible when the data label is placed **outside** the chart.

The connector line can be customized using the **type**, **color**, **width**, **length** and **dashArray** properties

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationDataLabel } from
'@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    const datalabel = {
        visible: true, name: 'text', position: 'Outside',
        connectorStyle: {
            //Length of the connector line in pixels
            length: '50px',
            //Width of the connector line in pixels
            width: 2,
            //dashArray of the connector line
            dashArray: '5,3',
            //Color of the connector line
            color: '#f4429e',
            //Specifies the type of the connector line either Line or Curve
            type: 'Curve'
        }
    };
    return <AccumulationChartComponent id='charts' enableSmartLabels='true'>
        <Inject services={[AccumulationDataLabel]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' dataLabel={datalabel}>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
export default App;

```

```
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective,
Inject, AccumulationDataLabel, AccumulationDataLabelSettingsModel }
from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
  const datalabel: AccumulationDataLabelSettingsModel = {
    visible: true, name: 'text', position: 'Outside',
    connectorStyle: {
      //Length of the connector line in pixels
      length: '50px',
      //Width of the connector line in pixels
      width: 2,
      //dashArray of the connector line
      dashArray: '5,3',
      //Color of the connector line
      color: '#f4429e',
      //Specifies the type of the connector line either Line or Curve
      type: 'Curve'
    }
  };
  return <AccumulationChartComponent id='charts' enableSmartLabels='true'>
    <Inject services={[AccumulationDataLabel]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' dataLabel={datalabel}>
        </AccumulationSeriesDirective>
      </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
  </return>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Text Mapping

The fill color and the text in the data source can be mapped to the chart using `pointColorMapping` and `name` properties, respectively.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationDataLabel } from
'@syncfusion/ej2-react-charts';
import { dataMapping } from 'datasource.ts';
function App() {
  const datalabel = { visible: true, name: 'text', position: 'Outside' };
  return <AccumulationChartComponent id='charts'>
    <Inject services={[AccumulationDataLabel]} />
  </return>
};
```

```

    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={dataMapping} xName='x'
yName='y' dataLabel={datalabel}>
    </AccumulationSeriesDirective>
  </AccumulationSeriesCollectionDirective>
</AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective,
  Inject, AccumulationDataLabel, AccumulationDataLabelSettingsModel }
from '@syncfusion/ej2-react-charts';
import { dataMapping } from 'datasource.ts';
function App() {
  const datalabel: AccumulationDataLabelSettingsModel = { visible: true,
name: 'text', position: 'Outside' };
  return <AccumulationChartComponent id='charts'>
    <Inject services={[AccumulationDataLabel]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={dataMapping} xName='x'
yName='y' dataLabel={datalabel}>
    </AccumulationSeriesDirective>
  </AccumulationSeriesCollectionDirective>
</AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Customization

Individual text can be customized using the `textRender` event.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationDataLabel } from
'@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
  const textRender = (args) => {
    if (args.text === '7') {
      args.color = 'red';
      args.border.width = 1;
    }
  };
  const datalabel = { visible: true };

```

```

    return <AccumulationChartComponent id='charts' enableSmartLabels='true'
textRender={textRender}>
    <Inject services={[AccumulationDataLabel]}/>
    <AccumulationSeriesCollectionDirective>
    <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' dataLabel={datalabel}>
    </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent,
AccumulationSeriesCollectionDirective, AccumulationDataLabelSettingsModel,
IAccTextRenderEventArgs, AccumulationSeriesDirective, Inject,
AccumulationDataLabel } from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    const textRender = (args: IAccTextRenderEventArgs): void => {
        if (args.text === '7') {
            args.color = 'red';
            args.border.width = 1;
        }
    };
    const datalabel: AccumulationDataLabelSettingsModel = { visible: true };
    return <AccumulationChartComponent id='charts' enableSmartLabels='true'
textRender={textRender}>
    <Inject services={[AccumulationDataLabel]}/>
    <AccumulationSeriesCollectionDirective>
    <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' dataLabel={datalabel}>
    </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Text wrap

When the data label text exceeds the container, the text can be wrapped by using [textWrap](#) property. End user can also wrap the data label text based on [maxWidth](#) property.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
PieSeries, AccumulationSeriesDirective, Inject, AccumulationDataLabel } from
'@syncfusion/ej2-react-charts';

```

```
function App() {
  const datalabel = {
    visible: true, osition: 'Inside', maxWidth: 100, textWrap: 'Wrap',
    name: 'text', enableRotation: true
  };
  const data = [
    { x: 'Chrome', y: 100, text: 'Chrome (100M)<br>40%',
      tooltipMappingName: '40%' },
    { x: 'UC Browser', y: 40, text: 'UC Browser (40M)<br>16%',
      tooltipMappingName: '16%' },
    { x: 'Opera', y: 30, text: 'Opera (30M)<br>12%', tooltipMappingName:
      '12%' },
    { x: 'Safari', y: 30, text: 'Safari (30M)<br>12%',
      tooltipMappingName: '12%' },
    { x: 'Firefox', y: 25, text: 'Firefox (25M)<br>10%',
      tooltipMappingName: '10%' },
    { x: 'Others', y: 25, text: 'Others (25M)<br>10%',
      tooltipMappingName: '10%' }
  ];
  return <AccumulationChartComponent id='charts' enableSmartLabels='true'>
    <Inject services={[AccumulationDataLabel, PieSeries]}/>
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={data} xName='x' yName='y'
startAngle={270} endAngle={90} dataLabel={datalabel} innerRadius='40%'
radius='100%'>
      </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent,
  AccumulationSeriesCollectionDirective, AccumulationDataLabelSettingsModel, Pie
  Series, IAccTextRenderEventArgs, AccumulationSeriesDirective, Inject,
  AccumulationDataLabel } from '@syncfusion/ej2-react-charts';
function App() {
  const datalabel: AccumulationDataLabelSettingsModel = {
    visible: true, osition: 'Inside', maxWidth: 100, textWrap: 'Wrap',
    name: 'text', enableRotation: true
  };
  const data: any[] = [
    { x: 'Chrome', y: 100, text: 'Chrome (100M)<br>40%',
      tooltipMappingName: '40%' },
    { x: 'UC Browser', y: 40, text: 'UC Browser
(40M)<br>16%', tooltipMappingName: '16%' },
    { x: 'Opera', y: 30, text: 'Opera (30M)<br>12%',
      tooltipMappingName: '12%' },
    { x: 'Safari', y: 30, text: 'Safari (30M)<br>12%',
      tooltipMappingName: '12%' },
  ];
```



```

    { x: 'Firefox', y: 25, text: 'Firefox (25M)<br>10%',
    tooltipMappingName: '10%' },
    { x: 'Others', y: 25, text: 'Others (25M)<br>10%',
    tooltipMappingName: '10%' }]];

    return <AccumulationChartComponent id='charts' enableSmartLabels='true'
  >
    <Inject services={[AccumulationDataLabel, PieSeries]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={data} xName='x' yName='y'
startAngle={270} endAngle={90} dataLabel={datalabel} innerRadius='40%'
radius='100%'>
        </AccumulationSeriesDirective>
      </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
  </>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Show percentages in data labels of pie chart

You can show the percentages in data labels of pie chart using `textRender` event and `template` option.

Using `textRender` event

You can customize the data label of pie chart using `textRender` event as follows to show percentage.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationDataLabel } from
'@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
  const textRender = (args) => {
    args.text = args.point.percentage + "%";
  };
  const datalabel = { visible: true };
  return <AccumulationChartComponent id='charts' enableSmartLabels='true'
textRender={textRender}>
    <Inject services={[AccumulationDataLabel]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' dataLabel={datalabel}>
        </AccumulationSeriesDirective>
      </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import * as React from "react";

```

```
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent,
  AccumulationSeriesCollectionDirective, AccumulationDataLabelSettingsModel,
  IAccTextRenderEventArgs, AccumulationSeriesDirective, Inject,
  AccumulationDataLabel } from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
  const textRender = (args: IAccTextRenderEventArgs): void => {
    args.text = args.point.percentage + "%";
  };
  const datalabel: AccumulationDataLabelSettingsModel = { visible: true };
  return <AccumulationChartComponent id='charts' enableSmartLabels='true'
  textRender={textRender}>
    <Inject services={[AccumulationDataLabel]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={accData} xName='x'
      yName='y' dataLabel={datalabel}>
    </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Using template

You can display the percentage values in data label of pie chart using **template** option.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
  AccumulationSeriesDirective, Inject, AccumulationDataLabel } from
  '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
  const template = chartTemplate;
  const datalabel = { visible: true, template: template };
  function chartTemplate(args) {
    return (<div id="templateWrap">
      <div>{args.point.percentage}%</div>
    </div>);
  }
  return <AccumulationChartComponent id='charts'>
    <Inject services={[AccumulationDataLabel]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={accData} xName='x'
      yName='y' dataLabel={datalabel}>
    </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationDataLabel, AccumulationDataLabelSettingsModel }
from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    const template:any = chartTemplate;
    const datalabel: AccumulationDataLabelSettingsModel = { visible: true,
template: template };
    function chartTemplate(args:any) {
        return (<div id="templateWrap">
            <div>{args.point.percentage}%</div>
        </div>);
    }
    return <AccumulationChartComponent id='charts'>
        <Inject services={[AccumulationDataLabel]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' dataLabel={datalabel}>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
    </>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Grouping in React Accumulation chart component

You can club/group few points of the series based on [groupTo](#) property. For example, if the club value is 11, then the points with value less than 11 is grouped together and will be showed as a single point with label **others**. The property also takes value in percentage (percentage of total data points value).

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationDataLabel } from
'@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    const datalabel = { visible: true, name: 'text', position: 'Outside' };
    return <AccumulationChartComponent id='charts'>
        <Inject services={[AccumulationDataLabel]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' groupTo='11' dataLabel={datalabel}></AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
    </>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import {
  AccumulationChartComponent, AccumulationSeriesCollectionDirective,
  AccumulationSeriesDirective,
  Inject, AccumulationDataLabel, AccumulationDataLabelSettingsModel,
  IAccTextRenderEventArgs, IAccPointRenderEventArgs
} from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
  const datalabel: AccumulationDataLabelSettingsModel = { visible: true,
  name: 'text', position: 'Outside' };
  return <AccumulationChartComponent id='charts' >
    <Inject services={[AccumulationDataLabel]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' groupTo='11'
      datalabel={datalabel}></AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>
  </>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Broken slice

You can visualize all points available in club/group points by clicking on the grouped point. For example, if 5 points are grouped together it will be showed as a single slice with label **others**. If we click on **others** slice it will explode and broke into 5 separate slices.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
  AccumulationSeriesDirective, Inject, AccumulationDataLabel } from
  '@syncfusion/ej2-react-charts';
function App() {
  const dataSource = [
    { x: 'Jan', y: 3, text: 'Jan: 3' }, { x: 'Feb', y: 3.5, text: 'Feb:
3.5' },
    { x: 'Mar', y: 7, text: 'Mar: 7' }, { x: 'Apr', y: 13.5, text: 'Apr:
13.5' },
    { x: 'May', y: 19, text: 'May: 19' }, { x: 'Jun', y: 23.5, text:
'Jun: 23.5' },
    { x: 'Jul', y: 26, text: 'Jul: 26' }, { x: 'Aug', y: 25, text: 'Aug:
25' },
    { x: 'Sep', y: 21, text: 'Sep: 21' }, { x: 'Oct', y: 15, text: 'Oct:
15' },
    { x: 'Nov', y: 9, text: 'Nov: 9' }, { x: 'Dec', y: 3.5, text: 'Dec:
3.5' }
  ];
  const datalabel = { visible: true, name: 'text', position: 'Outside' };
```

```

    return <AccumulationChartComponent id='charts'>
      <Inject services={[AccumulationDataLabel]} />
      <AccumulationSeriesCollectionDirective>
        <AccumulationSeriesDirective dataSource={dataSource} xName='x'
yName='y' groupTo='11' explode={true} explodeOffset='10'
dataLabel={datalabel}></AccumulationSeriesDirective>
      </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import {
  AccumulationChartComponent, AccumulationSeriesCollectionDirective,
  AccumulationSeriesDirective,
  Inject, AccumulationDataLabel, AccumulationDataLabelSettingsModel,
  IAccTextRenderEventArgs, IAccPointRenderEventArgs
} from '@syncfusion/ej2-react-charts';
function App() {
  const dataSource:any[] = [
    { x: 'Jan', y: 3, text: 'Jan: 3' }, { x: 'Feb', y: 3.5,
text: 'Feb: 3.5' },
    { x: 'Mar', y: 7, text: 'Mar: 7' }, { x: 'Apr', y: 13.5,
text: 'Apr: 13.5' },
    { x: 'May', y: 19, text: 'May: 19' }, { x: 'Jun', y:
23.5, text: 'Jun: 23.5' },
    { x: 'Jul', y: 26, text: 'Jul: 26' }, { x: 'Aug', y: 25,
text: 'Aug: 25' },
    { x: 'Sep', y: 21, text: 'Sep: 21' }, { x: 'Oct', y: 15,
text: 'Oct: 15' },
    { x: 'Nov', y: 9, text: 'Nov: 9' }, { x: 'Dec', y: 3.5,
text: 'Dec: 3.5' }
  ];
  const datalabel: AccumulationDataLabelSettingsModel = { visible: true,
name: 'text', position: 'Outside' };
  return <AccumulationChartComponent id='charts' >
    <Inject services={[AccumulationDataLabel]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={dataSource} xName='x'
yName='y' groupTo='11' explode={true} explodeOffset='10'
dataLabel={datalabel}></AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

GroupMode

Slice can also be grouped based on number of points by specifying the `groupMode` to `Point`. For example, if the group to value is 11, accumulation chart will show 1st 11 points and will group remaining entries in the collection as a single point.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
  AccumulationSeriesDirective, Inject, AccumulationDataLabel } from
  '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
  const datalabel = { visible: true, name: 'text', position: 'Outside' };
  return <AccumulationChartComponent id='charts'>
    <Inject services={[AccumulationDataLabel]}/>
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' groupTo='4' groupMode='Point'
dataLabel={datalabel}></AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import {
  AccumulationChartComponent, AccumulationSeriesCollectionDirective,
  AccumulationSeriesDirective, Inject, AccumulationDataLabel,
  IAccTextRenderEventArgs,
  IAccPointRenderEventArgs, AccumulationDataLabelSettingsModel
}
from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
  const datalabel: AccumulationDataLabelSettingsModel = { visible: true,
name: 'text', position: 'Outside' };
  return <AccumulationChartComponent id='charts'>
    <Inject services={[AccumulationDataLabel]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' groupTo='4' groupMode='Point'
dataLabel={datalabel}></AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Customization

You can customize the grouped point and its data label using `pointRender` and `textRender` event.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
  AccumulationSeriesDirective, Inject, AccumulationDataLabel } from
  '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
  const onTextRender = (args) => {
    if (args.text.indexOf('Others') > -1) {
      args.color = 'red';
      args.border.width = 1;
    }
  };
  const onPointRender = (args) => {
    if (args.point.x.indexOf('Others') > -1) {
      args.fill = '#D3D3D3';
    }
  };
  const datalabel = { visible: true, name: 'text', position: 'Outside' };
  return <AccumulationChartComponent id='charts' textRender={onTextRender}
    pointRender={onPointRender}>
    <Inject services={[AccumulationDataLabel]}/>
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={accData} xName='x'
        yName='y' groupTo='11' dataLabel={datalabel}></AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import {
  AccumulationChartComponent, AccumulationSeriesCollectionDirective,
  AccumulationSeriesDirective, Inject, AccumulationDataLabel,
  IAccTextRenderEventArgs,
  IAccPointRenderEventArgs, AccumulationDataLabelSettingsModel
} from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
  const onTextRender: EmitType<IAccTextRenderEventArgs> = (args:
    IAccTextRenderEventArgs): void => {
    if (args.text.indexOf('Others') > -1) {
      args.color = 'red';
      args.border.width = 1;
    }
  };
};
```

```

const onPointRender: EmitType<IAccPointRenderEventArgs> = (args:
IAccPointRenderEventArgs): void => {
    if ((args.point.x as string).indexOf('Others') > -1) {
        args.fill = '#D3D3D3';
    }
};

const datalabel: AccumulationDataLabelSettingsModel = { visible: true,
name: 'text', position: 'Outside' };

return <AccumulationChartComponent id='charts' textRender={onTextRender}
pointRender={onPointRender}>
    <Inject services={[AccumulationDataLabel]} />
    <AccumulationSeriesCollectionDirective>
        <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' groupTo='11'
        dataLabel={datalabel}></AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
</AccumulationChartComponent>
};

export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Empty points in React Accumulation chart component

The data points those uses the `null` or `undefined` as value are considered as empty points. The empty data points are ignored and not plotted in the chart. You can customize those points, using the `emptyPointSettings` property in series. The default mode of the empty point is `Gap`. Other supported modes are `Average` and `Zero`.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, AccumulationDataLabel, Inject } from
'@syncfusion/ej2-react-charts';

function App() {
    const data = [
        { x: 'Jan', y: null, text: 'Jan: 3' }, { x: 'Feb', y: 3.5, text:
'Feb: 3.5' },
        { x: 'Mar', y: 7, text: 'Mar: 7' }, { x: 'Apr', y: 13.5, text: 'Apr:
13.5' },
        { x: 'May', y: 19, text: 'May: 19' }, { x: 'Jun', y: undefined,
text: 'Jun: 23.5' },
        { x: 'Jul', y: 26, text: 'Jul: 26' }, { x: 'Aug', y: 25, text: 'Aug:
25' },
        { x: 'Sep', y: 21, text: 'Sep: 21' }, { x: 'Oct', y: 15, text: 'Oct:
15' }
    ];

    const datalabel = { visible: true };
    const emptypointsettings = { mode: 'Zero', fill: 'pink' };
    return <AccumulationChartComponent id='charts'>
        <Inject services={[AccumulationDataLabel]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={data} xName='x' yName='y'
dataLabel={datalabel} emptyPointSettings={emptypointsettings}>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>

```



```

    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
  AccumulationSeriesDirective,
  AccumulationDataLabel, Inject, AccumulationDataLabelSettingsModel }
from '@syncfusion/ej2-react-charts';
function App() {
  const data: any[] = [
    { x: 'Jan', y: null, text: 'Jan: 3' }, { x: 'Feb', y: 3.5, text: 'Feb: 3.5' },
    { x: 'Mar', y: 7, text: 'Mar: 7' }, { x: 'Apr', y: 13.5, text: 'Apr: 13.5' },
    { x: 'May', y: 19, text: 'May: 19' }, { x: 'Jun', y: undefined, text: 'Jun: 23.5' },
    { x: 'Jul', y: 26, text: 'Jul: 26' }, { x: 'Aug', y: 25, text: 'Aug: 25' },
    { x: 'Sep', y: 21, text: 'Sep: 21' }, { x: 'Oct', y: 15, text: 'Oct: 15' }
  ];
  const datalabel: AccumulationDataLabelSettingsModel = { visible: true };
  const emptypointsettings: EmptyPointSettingsModel = { mode: 'Zero', fill: 'pink' };
  return <AccumulationChartComponent id='charts'>
    <Inject services={[AccumulationDataLabel]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={data} xName='x' yName='y'
        dataLabel={datalabel} emptyPointSettings={emptypointsettings} >
    </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Customization

Specific color for an empty point can be set by using the **fill** property in **emptyPointSettings** and the border for an empty point can be set by using the **border** property.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
  AccumulationSeriesDirective, Inject, AccumulationDataLabel } from
  '@syncfusion/ej2-react-charts';
function App() {
  const data = [

```

```

    { x: 'Jan', y: null, text: 'Jan: 3' }, { x: 'Feb', y: 3.5, text:
'Feb: 3.5' },
    { x: 'Mar', y: 7, text: 'Mar: 7' }, { x: 'Apr', y: 13.5, text: 'Apr:
13.5' },
    { x: 'May', y: 19, text: 'May: 19' }, { x: 'Jun', y: undefined,
text: 'Jun: 23.5' },
    { x: 'Jul', y: 26, text: 'Jul: 26' }, { x: 'Aug', y: 25, text: 'Aug:
25' },
    { x: 'Sep', y: 21, text: 'Sep: 21' }, { x: 'Oct', y: 15, text: 'Oct:
15' }
  ];
  const datalabel = { visible: true };
  const emptypointsettings = { mode: 'Average', fill: 'black' };
  return <AccumulationChartComponent id='charts'>
    <Inject services={[AccumulationDataLabel]}/>
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={data} xName='x' yName='y'
emptyPointSettings={emptypointsettings} dataLabel={datalabel}>
    </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective,
Inject,
AccumulationDataLabel, EmptyPointSettingsModel, AccumulationDataLabelSettingsM
odel } from '@syncfusion/ej2-react-charts';
function App() {
  const data: any[] = [
    { x: 'Jan', y: null, text: 'Jan: 3' }, { x: 'Feb', y: 3.5, text: 'Feb:
3.5' },
    { x: 'Mar', y: 7, text: 'Mar: 7' }, { x: 'Apr', y: 13.5, text: 'Apr:
13.5' },
    { x: 'May', y: 19, text: 'May: 19' }, { x: 'Jun', y: undefined, text:
'Jun: 23.5' },
    { x: 'Jul', y: 26, text: 'Jul: 26' }, { x: 'Aug', y: 25, text: 'Aug: 25'
},
    { x: 'Sep', y: 21, text: 'Sep: 21' }, { x: 'Oct', y: 15, text: 'Oct: 15'
}];
  const datalabel: AccumulationDataLabelSettingsModel = { visible: true };
  const emptypointsettings: EmptyPointSettingsModel = { mode: 'Average',
fill: 'black' };
  return <AccumulationChartComponent id='charts'>
    <Inject services={[AccumulationDataLabel]}/>
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={data} xName='x' yName='y'
emptyPointSettings={emptypointsettings} dataLabel={datalabel}>
    </AccumulationSeriesDirective>
  </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>

```

```

    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Annotation in React Accumulation chart component

The annotations are used to mark the specific area of interest in the chart area with texts, shapes or images.

<!-- markdownlint-disable MD033 -->

By using the `<content>` option of annotation property, you can specify the Id of the element that needs to be displayed in the chart area.

INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationAnnotation,
AccumulationAnnotationsDirective, AccumulationAnnotationDirective } from
'@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
  const template = chartTemplate;
  function chartTemplate() {
    return (<div className='template'>
      <div style={{ border: '1px solid black', backgroundColor: '#f5f5f5',
padding: '5px 5px 5px 5px' }}>13.5</div>
    </div>);
  }
  ;
  return <AccumulationChartComponent id='charts'>
    <Inject services={[AccumulationAnnotation]} />
    <AccumulationAnnotationsDirective>
      <AccumulationAnnotationDirective content={template} region='Series'
coordinateUnits='Point' x='Feb' y={13.5}>
        </AccumulationAnnotationDirective>
      </AccumulationAnnotationsDirective>
      <AccumulationSeriesCollectionDirective>
        <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y'>
          </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
      </AccumulationChartComponent>;
    }
  ;
  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 { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationAnnotation,
AccumulationAnnotationsDirective, AccumulationAnnotationDirective }
from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
  const template: any = chartTemplate;
  function chartTemplate(): any {
    return (<div className='template'>
      <div style={{ border: '1px solid black', backgroundColor: '#f5f5f5',
padding: '5px 5px 5px 5px' }}>13.5</div>
    </div>);
  };
  return <AccumulationChartComponent id='charts' >
    <Inject services={[AccumulationAnnotation]} />
    <AccumulationAnnotationsDirective>
      <AccumulationAnnotationDirective content={template} region='Series'
coordinateUnits='Point' x='Feb' y={13.5}>
    </AccumulationAnnotationDirective>
    </AccumulationAnnotationsDirective>
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y'>
    </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>
  </>
  export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

*Note: To use annotation feature in accumulation chart, we need to inject **AccumulationAnnotation** module into the **services**

Region

The annotation can be placed with respect to either **Series** or **Chart**.

INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationAnnotation,
AccumulationAnnotationsDirective, AccumulationAnnotationDirective } from
'@syncfusion/ej2-react-charts';
import { labelData } from 'datasource.ts';
function App() {
  const template = chartTemplate;
  function chartTemplate() {
    return (<div className='template'>
      <div style={{ border: '1px solid black',
backgroundColor: '#f5f5f5', padding: '5px 5px 5px 5px' }}>13.5</div>
    </div>);
  };

```

```

    }
    ;
    return <AccumulationChartComponent id='charts'>
        <Inject services={[AccumulationAnnotation]}/>
        <AccumulationAnnotationsDirective>
            <AccumulationAnnotationDirective content={template}
region='Chart' coordinateUnits='Pixel' x={550} y={150}>
            </AccumulationAnnotationDirective>
        </AccumulationAnnotationsDirective>
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={labelData}
xName='x' yName='y'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
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 { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationAnnotation,
AccumulationAnnotationsDirective, AccumulationAnnotationDirective }
from '@syncfusion/ej2-react-charts';
import { labelData } from 'datasource.ts';
function App() {
    const template:any=chartTemplate;
    function chartTemplate(): any {
        return (<div className='template'>
            <div style={{ border: '1px solid black'
,backgroundColor:'#f5f5f5', padding: '5px 5px 5px 5px'}}>13.5</div>
            </div>);
    };
    return <AccumulationChartComponent id='charts' >
        <Inject services={ [AccumulationAnnotation] } />
        <AccumulationAnnotationsDirective >
            <AccumulationAnnotationDirective content={ template } region
= 'Chart' coordinateUnits= 'Pixel' x= { 550} y= { 150}>
            </AccumulationAnnotationDirective>
        </AccumulationAnnotationsDirective>
        <AccumulationSeriesCollectionDirective >
            <AccumulationSeriesDirective dataSource ={ labelData } xName
= 'x' yName= 'y' >
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

Co-ordinate Units

Specifies the coordinate units of an annotation either in **Pixel** or **Point**.

INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationAnnotation,
AccumulationAnnotationsDirective, AccumulationAnnotationDirective } from
'@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
  const template = chartTemplate;
  function chartTemplate() {
    return (<div className='template'>
      <div style={{ border: '1px solid black',
backgroundColor: '#f5f5f5', padding: '5px 5px 5px 5px' }}>13.5</div>
    </div>);
  }
  ;
  return <AccumulationChartComponent id='charts'>
    <Inject services={[AccumulationAnnotation]}/>
    <AccumulationAnnotationsDirective>
      <AccumulationAnnotationDirective content={template}
region='Series' coordinateUnits='Point' x='Feb' y={13.5}>
    </AccumulationAnnotationDirective>
    </AccumulationAnnotationsDirective>
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y'>
    </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>;
}
;
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 { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationAnnotation,
AccumulationAnnotationsDirective, AccumulationAnnotationDirective}
from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
  const template:any=chartTemplate;
  function chartTemplate(): any {
    return (<div className='template'>
```

```

        <div style={{ border: '1px solid black'
,backgroundColor:'#f5f5f5', padding: '5px 5px 5px 5px'}}>13.5</div>
        </div>);
    };
    return <AccumulationChartComponent id='charts' >
        <Inject services={[AccumulationAnnotation]}/>
        <AccumulationAnnotationsDirective>
            <AccumulationAnnotationDirective content={ template}
region='Series' coordinateUnits='Point' x='Feb' y={13.5}>
            </AccumulationAnnotationDirective>
        </AccumulationAnnotationsDirective>
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource ={accData}
xName='x' yName='y'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
    };
    export default App;
    ReactDOM.render(<App />, document.getElementById("charts"));
    {% endraw %}

```

Alignment

The annotations can be moved vertically and horizontally from its default position by using `verticalAlignment` or `horizontalAlignment` properties. The `verticalAlignment` property takes value as `Top`, `Bottom` or `Middle` and the `horizontalAlignment` property takes value as `Near`, `Far` or `Center`.

INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationAnnotation,
AccumulationAnnotationsDirective, AccumulationAnnotationDirective } from
'@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    const template = chartTemplate;
    function chartTemplate() {
        return (<div className='template'>
            <div style={{ border: '1px solid black',
backgroundColor: '#f5f5f5', padding: '5px 5px 5px 5px' }}>13.5</div>
            </div>);
    }
    ;
    return <AccumulationChartComponent id='charts'>
        <Inject services={[AccumulationAnnotation]}/>
        <AccumulationAnnotationsDirective>
            <AccumulationAnnotationDirective content={template}
region='Series' coordinateUnits='Point' x='Feb' y={13.5}
verticalAlignment='Top' horizontalAlignment='Near'>
            </AccumulationAnnotationDirective>
        </AccumulationAnnotationsDirective>
        <AccumulationSeriesCollectionDirective>

```

```

        <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y'>
        </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
</AccumulationChartComponent>;
}
;
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 { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationAnnotation,
AccumulationAnnotationsDirective, AccumulationAnnotationDirective }
from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    const template:any=chartTemplate;
    function chartTemplate(): any {
        return (<div className='template'>
            <div style={{ border: '1px solid black'
,backgroundColor:'#f5f5f5', padding: '5px 5px 5px 5px'}}>13.5</div>
            </div>);
    };
    return <AccumulationChartComponent id='charts' >
        <Inject services={[AccumulationAnnotation]}/>
        <AccumulationAnnotationsDirective>
            <AccumulationAnnotationDirective content={ template}
region='Series' coordinateUnits='Point' x='Feb' y={13.5}
verticalAlignment='Top' horizontalAlignment='Near'>
                </AccumulationAnnotationDirective>
            </AccumulationAnnotationsDirective>
            <AccumulationSeriesCollectionDirective>
                <AccumulationSeriesDirective dataSource = {accData}
xName='x' yName='y'>
                    </AccumulationSeriesDirective>
                </AccumulationSeriesCollectionDirective>
            </AccumulationChartComponent>
        </AccumulationChartComponent>
    };
    export default App;
    ReactDOM.render(<App />, document.getElementById("charts"));
    {% endraw %}

```

Tooltip in React Accumulation chart component

Tooltip for the accumulation chart can be enabled by using the [enable](#) property.

INDEX.JSX

```

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

```



```
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, AccumulationTooltip, Inject } from
'@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    return <AccumulationChartComponent id='charts' tooltip={{ enable: true
}}>
        <Inject services={[AccumulationTooltip]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
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 { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective,
    AccumulationTooltip, Inject, TooltipSettingsModel } from '@syncfusion/ej2-
react-charts';
import { accData } from 'datasource.ts';
function App() {

    return <AccumulationChartComponent id='charts' tooltip={{ enable: true
}}>
        <Inject services={[AccumulationTooltip]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

Note: To use tooltip feature, we need to inject **AccumulationTooltip** module into the **services**.

Header

We can specify header for the tooltip using [header](#) property.

INDEX.JSX

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

```
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, AccumulationTooltip, Inject } from
'@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    return <AccumulationChartComponent id='charts' tooltip={{ enable: true,
header: "Pie Chart" }}>
        <Inject services={[AccumulationTooltip]}/>
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
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 { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective,
AccumulationTooltip, Inject }
from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {

    return <AccumulationChartComponent id='charts' tooltip={{ enable: true,
header: "Pie Chart" }}>
        <Inject services={[AccumulationTooltip]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

Format

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 { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, AccumulationTooltip, Inject, AccumulationLegend
} from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    return <AccumulationChartComponent id='charts' tooltip={{ enable: true,
format: '${point.x} : <b>${point.y}</b>' }}>
        <Inject services={[AccumulationLegend, AccumulationTooltip]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
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 { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, AccumulationTooltip, Inject,
AccumulationLegend} from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {

    return <AccumulationChartComponent id='charts' tooltip={{ enable: true,
format: '${point.x} : <b>${point.y}</b>' }}>
        <Inject services={[AccumulationLegend, AccumulationTooltip]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

Tooltip mapping name

By default, tooltip shows information of x and y value in points. You can show more information from datasource in tooltip by using the [tooltipMappingName](#) property of the tooltip. You can use the `${point.tooltip}` as place holders to display the specified tooltip content.

INDEX.JSX

```
{% raw %}
```

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, AccumulationTooltip, Inject, AccumulationLegend
} from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    return <AccumulationChartComponent id='charts' tooltip={{ enable: true,
format: '${point.tooltip}' }}>
        <Inject services={[AccumulationLegend, AccumulationTooltip]}/>
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' tooltipMappingName='text'>
                </AccumulationSeriesDirective>
            </AccumulationSeriesCollectionDirective>
        </AccumulationChartComponent>;
}
;
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 { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, AccumulationTooltip, Inject,
AccumulationLegend
} from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {

    return <AccumulationChartComponent id='charts' tooltip={{ enable: true,
format: '${point.tooltip}' }}>
        <Inject services={[AccumulationLegend, AccumulationTooltip]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' tooltipMappingName='text'>
                </AccumulationSeriesDirective>
            </AccumulationSeriesCollectionDirective>
        </AccumulationChartComponent>
    };
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

Tooltip template

Any HTML element can be displayed in the tooltip by using the [template](#) property.

INDEX.JSX

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

```

import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, AccumulationTooltip, Inject } from
'@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    const template = chartTemplate;
    function chartTemplate(args) {
        return (<div style={{ border: '1px solid black', backgroundColor:
'red', padding: '2px', float: 'right', lineHeight: '20px', textAlign:
'center' }}>
            <img src='sun_annotation.png' />
            <div style={{ color: 'blue', padding: '2px', fontSize: '14px',
fontStyle: 'medium', fontFamily: 'Roboto', float: 'right', lineHeight:
'20px', textAlign: 'center', paddingRight: '6px' }}>
                <span>{args.y}</span>
            </div>
        </div>);
    }
    return <AccumulationChartComponent id='charts' tooltip={{ enable: true,
template: template }}>
        <Inject services={[AccumulationTooltip]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
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 { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective,
AccumulationTooltip, Inject } from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    const template:any =chartTemplate;
    function chartTemplate(args:any){
        return (<div style={{ border:'1px solid black', backgroundColor:'red',
padding:'2px',float:'right', lineHeight:'20px',textAlign:'center'}}>
            <img src='sun_annotation.png' />
            <div style={{ color:'blue', padding:'2px',fontSize:'14px',
fontStyle:'medium',fontFamily:'Roboto'
,float:'right',lineHeight:'20px',textAlign:'center',paddingRight:'6px'}}>
                <span>{args.y}</span>
            </div>
        </div>
    );
}

```

```

    return <AccumulationChartComponent id='charts' tooltip={{ enable: true,
template: template }}>
      <Inject services={[AccumulationTooltip]} />
      <AccumulationSeriesCollectionDirective>
        <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y'>
          </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
      </AccumulationChartComponent>
    };
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

Fixed tooltip

By default, tooltip track the mouse movement, but you can set a fixed position for the tooltip by using the [location](#) property.

INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, AccumulationTooltip, Inject } from
'@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
  return <AccumulationChartComponent id='charts' tooltip={{ enable: true,
location: { x: 200, y: 20 } }}>
    <Inject services={[AccumulationTooltip]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y'>
        </AccumulationSeriesDirective>
      </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
  }
;
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 { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective,
AccumulationTooltip, Inject } from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
  return <AccumulationChartComponent id='charts' tooltip={{ enable: true,
location: { x: 200, y: 20 } }}>

```

```

    <Inject services={[AccumulationTooltip]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y'>
    </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

Customization

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. The [highlightColor](#) property can be used to change the color of the data point when hovering.

INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, AccumulationTooltip, Inject } from
"@syncfusion/ej2-react-charts";
import { accData } from "datasource.ts";
function App() {
  return <AccumulationChartComponent id='charts' tooltip={{
    enable: true, format: '${series.name} ${point.x} : ${point.y}',
    //fill for tooltip
    fill: '#7bb4eb',
    //border for tooltip
    border: {
      width: 2,
      color: 'grey'
    }
  }} highlightColor='red'>
    <Inject services={[AccumulationTooltip]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y'>
    </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>;
}
;
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 { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
IAccTooltipRenderEventArgs,
  AccumulationSeriesDirective, AccumulationTooltip, Inject}
from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
  return <AccumulationChartComponent id='charts' tooltip={{
    enable: true, format: '${series.name} ${point.x} : ${point.y}',
    //fill for tooltip
    fill: '#7bb4eb',
    //border for tooltip
    border: {
      width: 2,
      color: 'grey'
    }
  }} highlightColor='red'>
    <Inject services={[AccumulationTooltip]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y'>
        </AccumulationSeriesDirective>
      </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
  };
  export default App;
  ReactDOM.render(<App />, document.getElementById("charts"));
  {% endraw %}

```

To customize individual tooltip

Using [tooltipRender](#) event, you can customize a tooltip for particular point. event, you can customize a tooltip for particular point.

INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, AccumulationTooltip, Inject, AccumulationLegend
} from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
  const tooltipRender = (args) => {
    if (args.point.index === 3) {
      args.text = args.point.x + ' ' + ':' + args.point.y + ' ' + ' ' +
'customtext';
      args.textStyle.color = '#f48042';
    }
  };
  return <AccumulationChartComponent id='charts' tooltip={{ enable: true
}} tooltipRender={tooltipRender}>
    <Inject services={[AccumulationLegend, AccumulationTooltip]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y'>
        </AccumulationSeriesDirective>

```



```

    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% enddraw %}

```

INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
IAccTooltipRenderEventArgs,
AccumulationSeriesDirective, AccumulationTooltip, Inject,
AccumulationLegend, TooltipSettingsModel } from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
  const tooltipRender: EmitType<IAccTooltipRenderEventArgs> = (args:
IAccTooltipRenderEventArgs): void => {
    if (args.point.index === 3) {
      args.text = args.point.x + ' ' + ':' + args.point.y + ' ' + ' ' +
'customtext';
      args.textStyle.color = '#f48042';
    }
  };

  return <AccumulationChartComponent id='charts' tooltip={{ enable: true }}
tooltipRender={tooltipRender}>
    <Inject services={[AccumulationLegend, AccumulationTooltip]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y'>
        </AccumulationSeriesDirective>
      </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
  </>;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% enddraw %}

```

Legend in React Accumulation chart component

As like a chart, the legend is also available for accumulation charts, which gives information about the points.

By default, the legend will be placed on the right, if the width of the chart is high or will be placed on the bottom, if the height of the chart is high. Other customization features regarding the legend element are same as the [chart legend](#).

Here, the legend for a point can be collapsed by giving the empty string to the x value of the point.

INDEX.JSX

```

import * as React from "react";

```

```
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend } from
'@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    const legendSettings = { visible: true };
    return <AccumulationChartComponent id='charts'
legendSettings={legendSettings}>
        <Inject services={[AccumulationLegend]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective,
Inject, AccumulationLegend, LegendSettingsModel } from '@syncfusion/ej2-react-
charts';
import { accData } from 'datasource.ts';
function App() {
    const legendSettings: LegendSettingsModel = { visible: true };
    return <AccumulationChartComponent id='charts'
legendSettings={legendSettings}>
        <Inject services={[AccumulationLegend]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Note: To use the legends feature, inject the **AccumulationLegend** using the module into the **services**.

Position and Alignment

By using the position property, you can position the legend at the **left**, **right**, **top** or **bottom** of the chart. You can also align the legend to **center**, **far** or **near** of the chart using the alignment property.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend } from
'@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    const legendSettings = { position: 'Right', visible: true, height: '40',
width: '160' };
    return <AccumulationChartComponent id='charts'
legendSettings={legendSettings}>
        <Inject services={[AccumulationLegend]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective,
Inject, AccumulationLegend, LegendSettingsModel } from '@syncfusion/ej2-
react-charts';
import { accData } from 'datasource.ts';
function App() {
    const legendSettings: LegendSettingsModel = { position: 'Right', visible:
true, height: '40', width: '160' };
    return <AccumulationChartComponent id='charts'
legendSettings={legendSettings}>
        <Inject services={[AccumulationLegend]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Legend Reverse

You can reverse the order of the legend items by using the [reverse](#) property. By default, legend for the first series in the collection will be placed first.

INDEX.JSX

```
{% raw %}
```

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, AccumulationLegend, PieSeries,
AccumulationDataLabel, AccumulationTooltip, Inject } from '@syncfusion/ej2-react-charts';
function App() {
    const data = [
        { x: 'Argentina', y: 505370, r: '50%' },
        { x: 'Belgium', y: 551500, r: '70%' },
        { x: 'Cuba', y: 312685, r: '84%' },
        { x: 'Dominican Republic', y: 350000, r: '97%' },
        { x: 'Egypt', y: 301000, r: '84%' },
        { x: 'Kazakhstan', y: 300000, r: '70%' },
        { x: 'Somalia', y: 357022, r: '90%' },
    ];
    return (<AccumulationChartComponent id="charts" ref={(pie) => (pie =
pie)} legendSettings={{
        visible: true,
        reverse: true,
    }} useGroupingSeparator={true} enableSmartLabels={true}
enableAnimation={true} tooltip={{ enable: true }}>
        <Inject services={[
            AccumulationLegend,
            PieSeries,
            AccumulationDataLabel,
            AccumulationTooltip,
        ]}/>
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={data1} xName="x"
yName="y" innerRadius="20%" dataLabel={{
                visible: true,
                position: 'Outside',
                name: 'x',
            }} radius="r"></AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
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 {
    AccumulationChartComponent,
    AccumulationSeriesCollectionDirective,
    AccumulationSeriesDirective,
    AccumulationLegend,
    PieSeries,
    AccumulationDataLabel,
    AccumulationTooltip,
}

```

```

Inject } from '@syncfusion/ej2-react-charts';
function App() {
  const datal: any = [
    { x: 'Argentina', y: 505370, r: '50%' },
    { x: 'Belgium', y: 551500, r: '70%' },
    { x: 'Cuba', y: 312685, r: '84%' },
    { x: 'Dominican Republic', y: 350000, r: '97%' },
    { x: 'Egypt', y: 301000, r: '84%' },
    { x: 'Kazakhstan', y: 300000, r: '70%' },
    { x: 'Somalia', y: 357022, r: '90%' },
  ];

  return (
    <AccumulationChartComponent
      id="charts"
      ref={(pie) => (pie = pie)}
      legendSettings={{
        visible: true,
        reverse: true,
      }}
      useGroupingSeparator={true}
      enableSmartLabels={true}
      enableAnimation={true}
      tooltip={{ enable: true }}
    >
      <Inject
        services={[
          AccumulationLegend,
          PieSeries,
          AccumulationDataLabel,
          AccumulationTooltip,
        ]}
      />
      <AccumulationSeriesCollectionDirective>
        <AccumulationSeriesDirective
          dataSource={datal}
          xName="x"
          yName="y"
          innerRadius="20%"
          dataLabel={{
            visible: true,
            position: 'Outside',
            name: 'x',
          }}
          radius="r"
        ></AccumulationSeriesDirective>
      </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
  );
};

export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% enddraw %}

```

Legend Shape

To change the legend icon shape, use the `legendShape` property in the `series`. By default, legend icon shape is `seriesType`.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend } from
'@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    const legendSettings = { position: 'Right', visible: true, height: '40',
width: '160' };
    return <AccumulationChartComponent id='charts'
legendSettings={legendSettings}>
        <Inject services={[AccumulationLegend]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' legendShape='Rectangle'></AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend, LegendSettingsModel }
from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    const legendSettings: LegendSettingsModel = { position: 'Right', visible:
true, height: '40', width: '160' };
    return <AccumulationChartComponent id='charts'
legendSettings={legendSettings}>
        <Inject services={[AccumulationLegend]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' legendShape='Rectangle'></AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Legend Size

The legend size can be changed by using the `width` and `height` properties of the `legendSettings`.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend } from
'@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    const legendSettings = { width: '150', height: '100', border: { width:
1, color: 'pink' } };
    return <AccumulationChartComponent id='charts'
legendSettings={legendSettings}>
        <Inject services={[AccumulationLegend]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' legendShape='Circle'></AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend, LegendSettingsModel }
from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    const legendSettings: LegendSettingsModel = { width: '150', height: '100',
border: { width: 1, color: 'pink' } };
    return <AccumulationChartComponent id='charts'
legendSettings={legendSettings}>
        <Inject services={[AccumulationLegend]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' legendShape='Circle'></AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Legend Item Size

You can customize the size of the legend items by using the `shapeHeight` and `shapeWidth` properties.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend } from
'@syncfusion/ej2-react-charts';
```

```
import { accData } from 'datasource.ts';
function App() {
  const legendSettings = { shapeHeight: 15, shapeWidth: 15 };
  return <AccumulationChartComponent id='charts'
  legendSettings={legendSettings}>
    <Inject services={[AccumulationLegend]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={accData} xName='x'
      yName='y' legendShape='Circle'></AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
  AccumulationSeriesDirective, Inject, AccumulationLegend, LegendSettingsModel }
  from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
  const legendSettings: LegendSettingsModel = { shapeHeight: 15, shapeWidth:
  15 };
  return <AccumulationChartComponent id='charts'
  legendSettings={legendSettings}>
    <Inject services={[AccumulationLegend]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={accData} xName='x'
      yName='y' legendShape='Circle'></AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Paging for Legend

Paging will be enabled by default, when the legend items exceeds the legend bounds. You can view the each legend item by navigating between the pages using the navigation buttons.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
  AccumulationSeriesDirective, Inject, AccumulationLegend } from
  '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
  const legendSettings = { height: '150', width: '80' };
  return <AccumulationChartComponent id='charts'
  legendSettings={legendSettings}>
```



```

    <Inject services={[AccumulationLegend]}/>
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' legendShape='Circle'></AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend, LegendSettingsModel }
from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
  const legendSettings: LegendSettingsModel = { height: '150', width: '80'
};
  return <AccumulationChartComponent id='charts'
legendSettings={legendSettings}>
    <Inject services={[AccumulationLegend]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' legendShape='Circle'></AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Legend Text Wrap

When the legend text exceeds the container, the text can be wrapped by using `textWrap` Property. End user can also wrap the legend text based on the `maximumLabelWidth` property.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend } from
 '@syncfusion/ej2-react-charts';
import { piechart } from 'datasource.ts';
function App() {
  const legendSettings = { visible: true, position: 'Right', textWrap:
'Wrap', maximumLabelWidth: 60, height: '44%', width: '64%' };
  return <AccumulationChartComponent id='charts'
legendSettings={legendSettings}>
    <Inject services={[AccumulationLegend]}/>
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={piechart} xName='x'
yName='y'></AccumulationSeriesDirective>

```

```

    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend, LegendSettingsModel }
from '@syncfusion/ej2-react-charts';
import { piechart } from 'datasource.ts';
function App() {
  const legendSettings: LegendSettingsModel = { visible: true,
position: 'Right', textWrap: 'Wrap', maximumLabelWidth: 60, height: '44%',
width: '64%' };
  return <AccumulationChartComponent id='charts'
legendSettings={legendSettings}>
    <Inject services={[AccumulationLegend]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={piechart} xName='x'
yName='y'></AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

Enable Animation

You can customize the animation while clicking legend by setting enableAnimation as true or false using enableAnimation property in Accumulation Chart.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend,
AccumulationTooltip, AccumulationDataLabel } from '@syncfusion/ej2-react-
charts';
import { data1 } from 'datasource.ts';
function App() {
  const legendSettings = { height: '150', width: '80' };
  const datalabel = { visible: true,
    name: 'text',
    position: 'Inside',
    font: {
      fontWeight: '600',
      color: '#ffffff'
    }
  };
  const tooltip = { enable: false };

```

```

    return <AccumulationChartComponent id='charts'
    legendSettings={legendSettings} tooltip={tooltip} title='Project Cost
    Breakdown'>
      <Inject services={[AccumulationLegend, AccumulationDataLabel,
    AccumulationTooltip]}/>
      <AccumulationSeriesCollectionDirective>
        <AccumulationSeriesDirective dataSource={data1} xName='x' yName='y'
    legendShape='Circle' dataLabel={datalabel} startAngle='0' endAngle='360'
    explode={true} explodeOffset='10%' explodeIndex={3} name='Project'
    radius='70%' innerRadius='40%'>
          </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
      </AccumulationChartComponent>;
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
    AccumulationSeriesDirective, Inject,
    AccumulationLegend, TooltipSettingsModel, AccumulationTooltip, LegendSettingsMo
    del, AccumulationDataLabelSettingsModel, AccumulationDataLabel }
    from '@syncfusion/ej2-react-charts';
import { data1 } from 'datasource.ts';
function App() {
  const legendSettings: LegendSettingsModel = { height: '150', width: '80'
};
  const datalabel: AccumulationDataLabelSettingsModel = { visible: true,
    name: 'text',
    position: 'Inside',
    font: {
      fontWeight: '600',
      color: '#ffffff'
    }
  };
  const tooltip: TooltipSettingsModel = { enable: false };
  return <AccumulationChartComponent id='charts'
    legendSettings={legendSettings} tooltip={tooltip} title='Project Cost
    Breakdown' >
    <Inject
    services={[AccumulationLegend, AccumulationDataLabel, AccumulationTooltip]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={data1} xName='x' yName='y'
    legendShape='Circle' dataLabel={datalabel}
      startAngle='0' endAngle='360' explode={true} explodeOffset='10%'
    explodeIndex={3} name='Project' radius='70%' innerRadius='40%'>
        </AccumulationSeriesDirective>
      </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
  </>;
  export default App;
  ReactDOM.render(<App />, document.getElementById("charts"));
}

```

Legend Title

You can set title for legend using `title` property in `legendSettings`. You can also 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 { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend } from
'@syncfusion/ej2-react-charts';
import { data } from 'datasource.ts';
function App() {
    const legendSettings = { title: 'Months', position: 'Bottom' };
    return <AccumulationChartComponent id='charts'
legendSettings={legendSettings}>
        <Inject services={[AccumulationLegend]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={data} xName='x'
yName='y'></AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend, LegendSettingsModel }
from '@syncfusion/ej2-react-charts';
import { data } from 'datasource.ts';
function App() {
    const legendSettings: LegendSettingsModel = { title: 'Months', position:
'Bottom' };
    return <AccumulationChartComponent id='charts'
legendSettings={legendSettings}>
        <Inject services={[AccumulationLegend]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={data} xName='x'
yName='y'></AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Arrow Page Navigation

By default, the page number will be enabled while legend paging. Now, you can disable that page number and also you can get left and right arrows for page navigation. You have to set `false` value to `enablePages` to get this support.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend } from
'@syncfusion/ej2-react-charts';
import { data } from 'datasource.ts';
function App() {
    const legendSettings = { width: '260px', height: '50px', enablePages:
false, position: 'Bottom' };
    return <AccumulationChartComponent id='charts'
legendSettings={legendSettings}>
        <Inject services={[AccumulationLegend]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={data} xName='x'
yName='y'></AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend, LegendSettingsModel }
from '@syncfusion/ej2-react-charts';
import { data } from 'datasource.ts';
function App() {
    const legendSettings: LegendSettingsModel = { width: '260px', height:
'50px', enablePages: false, position: 'Bottom' };
    return <AccumulationChartComponent id='charts'
legendSettings={legendSettings}>
        <Inject services={[AccumulationLegend]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={data} xName='x'
yName='y'></AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Legend Item Padding

The `itemPadding` property can be used to adjust the space between the legend items.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend } from
'@syncfusion/ej2-react-charts';
import { data } from 'datasource.ts';
function App() {
    const legendSettings = { width: '260px', height: '50px', position:
'Bottom', itemPadding: 30 };
    return <AccumulationChartComponent id='charts'
legendSettings={legendSettings}>
        <Inject services={[AccumulationLegend]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={data} xName='x'
yName='y'></AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend, LegendSettingsModel }
from '@syncfusion/ej2-react-charts';
import { data } from 'datasource.ts';
function App() {
    const legendSettings: LegendSettingsModel = { width: '260px', height:
'50px', position: 'Bottom', itemPadding: 30 };
    return <AccumulationChartComponent id='charts'
legendSettings={legendSettings}>
        <Inject services={[AccumulationLegend]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={data} xName='x'
yName='y'></AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Center label in React Accumulation chart component

Center label

Using [centerLabel](#) it is now possible to place a label at the center of a pie or donut chart. To configure the default text rendered on the center label for the pie and doughnut charts, use the [text](#) property in the [centerLabel](#).

INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective } from '@syncfusion/ej2-react-charts';
import { centerData } from 'datasource.ts';
function App() {
    return <AccumulationChartComponent id='charts' centerLabel={{text :
'Browsers Statistics'}}>
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={centerData} xName='x'
yName='y' innerRadius='65%'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
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 { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective} from '@syncfusion/ej2-react-charts';
import { centerData } from 'datasource.ts';
function App() {
    return <AccumulationChartComponent id='charts' centerLabel={{text :
'Browsers Statistics'}}>
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={centerData} xName='x'
yName='y' innerRadius='65%'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

Hover text

The default text in the center label can be changed when the mouse pointer hovers over the pie and doughnut charts slice using the [hoverTextFormat](#) property.

INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective } from '@syncfusion/ej2-react-charts';
import { centerData } from 'datasource.ts';
```

```
function App() {
    return <AccumulationChartComponent id='charts' centerLabel={{text :
'Browsers Statistics', hoverTextFormat: '${point.x} <br> Browser Share <br>
${point.y}%'}}>
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={centerData} xName='x'
yName='y' innerRadius='65%'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
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 { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective } from '@syncfusion/ej2-react-charts';
import { centerData } from 'datasource.ts';
function App() {
    return <AccumulationChartComponent id='charts' centerLabel={{text :
'Browsers Statistics', hoverTextFormat: '${point.x} <br> Browser Share <br>
${point.y}%'}}>
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={centerData} xName='x'
yName='y' innerRadius='65%'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

Customization

Customize the center label text using the [textStyle](#) property.

INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective } from '@syncfusion/ej2-react-charts';
import { centerData } from 'datasource.ts';
function App() {
    return <AccumulationChartComponent id='charts' centerLabel={{text :
'Mobile<br>Browsers<br>Statistics',
    hoverTextFormat: '${point.x} <br> Browser Share <br> ${point.y}%',
    textStyle:{ fontWeight: '900',
    size: '15px',

```



```

        color: 'grey',
        fontFamily: 'Roboto',
        fontStyle: 'Italic'}
    }}>
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={centerData} xName='x'
yName='y' innerRadius='65%'>
      </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>;
}
;
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 { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective } from '@syncfusion/ej2-react-charts';
import { centerData } from 'datasource.ts';
function App() {
  return <AccumulationChartComponent id='charts' centerLabel={{text :
'Mobile<br>Browsers<br>Statistics',
  hoverTextFormat: '${point.x} <br> Browser Share <br> ${point.y}%',
  textStyle:{ fontWeight: '900',
    size: '15px',
    color: 'grey',
    fontFamily: 'Roboto',
    fontStyle: 'Italic'}}}}>
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={centerData} xName='x'
yName='y' innerRadius='65%'>
      </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

Title and sub title in React Accumulation chart component

Title

Accumulation 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 { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend } from
'@syncfusion/ej2-react-charts';
function App() {
  const data = [
    { x: 'Saudi Arabia', y: 58, text: '58%' },
    { x: 'Persian Gulf', y: 15, text: '15%' },
    { x: 'Canada', y: 13, text: '13%' },
    { x: 'Venezuela', y: 8, text: '8%' },
    { x: 'Mexico', y: 3, text: '3%' },
    { x: 'Russia', y: 2, text: '2%' },
    { x: 'Miscellaneous', y: 1, text: '1%' }
  ];
  const legendSettings = { visible: true };
  return <AccumulationChartComponent id='charts'
legendSettings={legendSettings} title='Oil and other liquid imports in USA'>
    <Inject services={[AccumulationLegend]}>
      <AccumulationSeriesCollectionDirective>
        <AccumulationSeriesDirective dataSource={data} xName='x' yName='y'
dataLabel={{
          visible: true,
          name: 'text',
          font: {
            fontWeight: '600'
          }
        }}>
      </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>;
}
;
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 { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective,
  Inject, LegendSettingsModel, AccumulationLegend } from '@syncfusion/ej2-
react-charts';
function App() {
  const data: any[] = [
    { x: 'Saudi Arabia', y: 58, text: '58%' },
    { x: 'Persian Gulf', y: 15, text: '15%' },
    { x: 'Canada', y: 13, text: '13%' },
    { x: 'Venezuela', y: 8, text: '8%' },
    { x: 'Mexico', y: 3, text: '3%' },
    { x: 'Russia', y: 2, text: '2%' },
    { x: 'Miscellaneous', y: 1, text: '1%' }];
  const legendSettings: LegendSettingsModel = { visible: true };
  return <AccumulationChartComponent id='charts'
legendSettings={legendSettings} title='Oil and other liquid imports in USA'>

```

```

<Inject services={[AccumulationLegend]} />
  <AccumulationSeriesCollectionDirective>
    <AccumulationSeriesDirective dataSource={data} xName='x' yName='y'
dataLabel={{
      visible: true,
      name: 'text',
      font: {
        fontWeight: '600'
      }
    }}>
    </AccumulationSeriesDirective>
  </AccumulationSeriesCollectionDirective>
</AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

Title customization

Accumulation Chart can be customizing a title using [titleStyle](#) property.

INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend } from
"@syncfusion/ej2-react-charts";
function App() {
  const data = [
    { x: 'Saudi Arabia', y: 58, text: '58%' },
    { x: 'Persian Gulf', y: 15, text: '15%' },
    { x: 'Canada', y: 13, text: '13%' },
    { x: 'Venezuela', y: 8, text: '8%' },
    { x: 'Mexico', y: 3, text: '3%' },
    { x: 'Russia', y: 2, text: '2%' },
    { x: 'Miscellaneous', y: 1, text: '1%' }
  ];
  const legendSettings = { visible: true };
  const title = {
    fontFamily: "Arial",
    fontStyle: 'italic',
    fontWeight: 'regular',
    color: "#E27F2D",
    size: '23px'
  };
  return <AccumulationChartComponent id='charts'
legendSettings={legendSettings} title='Oil and other liquid imports in USA'
subTitle='In the year 2014 - 2015' titleStyle={title}>
    <Inject services={[AccumulationLegend]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={data} xName='x' yName='y'
dataLabel={{
        visible: true,
        name: 'text',

```

```

        font: {
            fontWeight: '600'
        }
    }}>
</AccumulationSeriesDirective>
</AccumulationSeriesCollectionDirective>
</AccumulationChartComponent>;
}
;
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 { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective,
Inject, LegendSettingsModel, AccumulationLegend } from '@syncfusion/ej2-
react-charts';
function App() {
    const data: any[] = [
        { x: 'Saudi Arabia', y: 58, text: '58%' },
        { x: 'Persian Gulf', y: 15, text: '15%' },
        { x: 'Canada', y: 13, text: '13%' },
        { x: 'Venezuela', y: 8, text: '8%' },
        { x: 'Mexico', y: 3, text: '3%' },
        { x: 'Russia', y: 2, text: '2%' },
        { x: 'Miscellaneous', y: 1, text: '1%' }];
    const legendSettings: LegendSettingsModel = { visible: true };
    const title: any = {
        fontFamily: "Arial",
        fontStyle: 'italic',
        fontWeight: 'regular',
        color: "#E27F2D",
        size: '23px'
    };
    return <AccumulationChartComponent id='charts'
legendSettings={legendSettings} title='Oil and other liquid imports in USA'
subTitle='In the year 2014 - 2015' titleStyle={title}>
        <Inject services={[AccumulationLegend]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={data} xName='x' yName='y'
dataLabel={{
                visible: true,
                name: 'text',
                font: {
                    fontWeight: '600'
                }
            }}>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
};

```

```
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

SubTitle

Accumulation Chart can be given a subtitle using [subTitle](#) property, to show the information about the data plotted.

INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend } from
'@syncfusion/ej2-react-charts';
function App() {
  const data = [
    { x: 'Saudi Arabia', y: 58, text: '58%' },
    { x: 'Persian Gulf', y: 15, text: '15%' },
    { x: 'Canada', y: 13, text: '13%' },
    { x: 'Venezuela', y: 8, text: '8%' },
    { x: 'Mexico', y: 3, text: '3%' },
    { x: 'Russia', y: 2, text: '2%' },
    { x: 'Miscellaneous', y: 1, text: '1%' }
  ];
  const legendSettings = { visible: true };
  return <AccumulationChartComponent id='charts'
  legendSettings={legendSettings} title='Oil and other liquid imports in USA'
  subTitle='In the year 2014 - 2015'>
    <Inject services={[AccumulationLegend]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={data} xName='x' yName='y'
dataLabel={{
        visible: true,
        name: 'text',
        font: {
          fontWeight: '600'
        }
      }}>
    </AccumulationSeriesDirective>
  </AccumulationSeriesCollectionDirective>
</AccumulationChartComponent>;
}
;
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 { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective,
Inject, LegendSettingsModel, AccumulationLegend } from '@syncfusion/ej2-
react-charts';
function App() {
  const data: any[] = [
    { x: 'Saudi Arabia', y: 58, text: '58%' },
    { x: 'Persian Gulf', y: 15, text: '15%' },
    { x: 'Canada', y: 13, text: '13%' },
    { x: 'Venezuela', y: 8, text: '8%' },
    { x: 'Mexico', y: 3, text: '3%' },
    { x: 'Russia', y: 2, text: '2%' },
    { x: 'Miscellaneous', y: 1, text: '1%' }];
  const legendSettings: LegendSettingsModel = { visible: true };
  return <AccumulationChartComponent id='charts'
legendSettings={legendSettings} title='Oil and other liquid imports in USA'
subTitle='In the year 2014 - 2015'>
    <Inject services={[AccumulationLegend]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={data} xName='x' yName='y'
dataLabel={{
        visible: true,
        name: 'text',
        font: {
          fontWeight: '600'
        }
      }}>
    </AccumulationSeriesDirective>
  </AccumulationSeriesCollectionDirective>
</AccumulationChartComponent>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

SubTitle customization

Accumulation Chart can be customizing a subtitle using [subTitleStyle](#) property, to show the information about the data plotted.

INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationLegend } from
 '@syncfusion/ej2-react-charts';
function App() {
  const data = [
    { x: 'Saudi Arabia', y: 58, text: '58%' },
    { x: 'Persian Gulf', y: 15, text: '15%' },
    { x: 'Canada', y: 13, text: '13%' },
    { x: 'Venezuela', y: 8, text: '8%' },
    { x: 'Mexico', y: 3, text: '3%' },
    { x: 'Russia', y: 2, text: '2%' },
    { x: 'Miscellaneous', y: 1, text: '1%' }

```

```

];
const legendSettings = { visible: true };
const subTitle = {
  fontFamily: "Arial",
  fontStyle: 'italic',
  fontWeight: 'regular',
  color: "#E27F2D",
  size: '13px'
};
return <AccumulationChartComponent id='charts'
legendSettings={legendSettings} title='Oil and other liquid imports in USA'
subTitle='In the year 2014 - 2015' subTitleStyle={subTitle}>
  <Inject services={[AccumulationLegend]}/>
  <AccumulationSeriesCollectionDirective>
    <AccumulationSeriesDirective dataSource={data} xName='x' yName='y'
dataLabel={{
  visible: true,
  name: 'text',
  font: {
    fontWeight: '600'
  }
}}>
    </AccumulationSeriesDirective>
  </AccumulationSeriesCollectionDirective>
</AccumulationChartComponent>;
}
;
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 { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective,
Inject, LegendSettingsModel, AccumulationLegend } from '@syncfusion/ej2-
react-charts';
function App() {
  const data: any[] = [
    { x: 'Saudi Arabia', y: 58, text: '58%' },
    { x: 'Persian Gulf', y: 15, text: '15%' },
    { x: 'Canada', y: 13, text: '13%' },
    { x: 'Venezuela', y: 8, text: '8%' },
    { x: 'Mexico', y: 3, text: '3%' },
    { x: 'Russia', y: 2, text: '2%' },
    { x: 'Miscellaneous', y: 1, text: '1%' }];
  const legendSettings: LegendSettingsModel = { visible: true };
  const subTitle: any = {
    fontFamily: "Arial",
    fontStyle: 'italic',
    fontWeight: 'regular',
    color: "#E27F2D",
    size: '13px'
  }

```

```

    };
    return <AccumulationChartComponent id='charts'
    legendSettings={legendSettings} title='Oil and other liquid imports in USA'
    subTitle = 'In the year 2014 - 2015' subTitleStyle={subTitle}>
    <Inject services={[AccumulationLegend]} />
    <AccumulationSeriesCollectionDirective>
    <AccumulationSeriesDirective dataSource={data} xName='x' yName='y'
    dataLabel={{
        visible: true,
        name: 'text',
        font: {
            fontWeight: '600'
        }
    }}
    </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
    };
    export default App;
    ReactDOM.render(<App />, document.getElementById("charts"));
    {% enddraw %}

```

Chart print in React Accumulation chart component

Print

The rendered chart can be printed directly from the browser by calling the public method print.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective } from '@syncfusion/ej2-react-charts';
function App() {
    const data = [
        { month: 'Jan', sales: 35 }, { month: 'Feb', sales: 28 },
        { month: 'Mar', sales: 34 }, { month: 'Apr', sales: 32 },
        { month: 'May', sales: 40 }, { month: 'Jun', sales: 32 },
        { month: 'Jul', sales: 35 }, { month: 'Aug', sales: 55 },
        { month: 'Sep', sales: 38 }, { month: 'Oct', sales: 30 },
        { month: 'Nov', sales: 25 }, { month: 'Dec', sales: 32 }
    ];
    let chartInstance;
    function clickHandler() {
        chartInstance.print();
    }
    return (<div>
    <button value='print' onClick={clickHandler.bind(this)}>Print</button>
    <AccumulationChartComponent id='charts' ref={chart => chartInstance =
    chart}>
    <AccumulationSeriesCollectionDirective>
    <AccumulationSeriesDirective dataSource={data} xName='month'
    yName='sales' radius='100%'>
    </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent></div>);
}

```



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

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject } from '@syncfusion/ej2-react-charts';
function App() {
  const data: any[] = [
    { month: 'Jan', sales: 35 }, { month: 'Feb', sales: 28 },
    { month: 'Mar', sales: 34 }, { month: 'Apr', sales: 32 },
    { month: 'May', sales: 40 }, { month: 'Jun', sales: 32 },
    { month: 'Jul', sales: 35 }, { month: 'Aug', sales: 55 },
    { month: 'Sep', sales: 38 }, { month: 'Oct', sales: 30 },
    { month: 'Nov', sales: 25 }, { month: 'Dec', sales: 32 }
  ];
  let chartInstance: AccumulationChartComponent;
  function clickHandler() {
    chartInstance.print();
  }
  return (<div>
    <button value='print' onClick={clickHandler.bind(this)}>Print</button>
    <AccumulationChartComponent id='charts' ref={chart => chartInstance =
chart}>
      <AccumulationSeriesCollectionDirective>
        <AccumulationSeriesDirective dataSource={data} xName='month'
yName='sales' radius='100%'>
        </AccumulationSeriesDirective>
      </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent></div>)
};
export default App;
ReactDOM.render(<App />, document.getElementById('charts'));
```

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Chart-Category Axis</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></sc
ript>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
```

```

        left: 45%;
        position: absolute;
        top: 45%;
        width: 30%;
    }
    #charts {
        height : 350px;
        display: block;
    }
</style>
</head>
<body>
    <div id='charts'>
        <div id='loader'>Loading....</div>
    </div>
</body>
</html>

```

Export

The rendered chart can be exported to Image(jpeg or png) or SVG or PDF format by using the export method. Input parameters for this method are Export Type for format and fileName of result.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Export, Inject } from '@syncfusion/ej2-react-charts';
function App() {
    const data = [
        { month: 'Jan', sales: 35 }, { month: 'Feb', sales: 28 },
        { month: 'Mar', sales: 34 }, { month: 'Apr', sales: 32 },
        { month: 'May', sales: 40 }, { month: 'Jun', sales: 32 },
        { month: 'Jul', sales: 35 }, { month: 'Aug', sales: 55 },
        { month: 'Sep', sales: 38 }, { month: 'Oct', sales: 30 },
        { month: 'Nov', sales: 25 }, { month: 'Dec', sales: 32 }
    ];
    let chartInstance;
    function clickHandler() {
        chartInstance.exportModule.export('PNG', 'charts');
    }
    return (<div><button value='print'
onClick={clickHandler.bind(this)}>Export</button>
    <AccumulationChartComponent id='charts' ref={chart => chartInstance =
chart}>
        <Inject services={[Export]}>
            <AccumulationSeriesCollectionDirective>
                <AccumulationSeriesDirective dataSource={data} xName='month'
yName='sales' radius='100%'>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent></div>);
}
;
export default App;

```

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

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Export, Inject } from '@syncfusion/ej2-react-charts';
function App() {
  const data: any[] = [
    { month: 'Jan', sales: 35 }, { month: 'Feb', sales: 28 },
    { month: 'Mar', sales: 34 }, { month: 'Apr', sales: 32 },
    { month: 'May', sales: 40 }, { month: 'Jun', sales: 32 },
    { month: 'Jul', sales: 35 }, { month: 'Aug', sales: 55 },
    { month: 'Sep', sales: 38 }, { month: 'Oct', sales: 30 },
    { month: 'Nov', sales: 25 }, { month: 'Dec', sales: 32 }
  ];
  let chartInstance: AccumulationChartComponent;
  function clickHandler() {
    chartInstance.exportModule.export('PNG', 'charts');
  }
  return (<div><button value='print'
onClick={clickHandler.bind(this)}>Export</button>
  <AccumulationChartComponent id='charts' ref={chart => chartInstance =
chart}>
    <Inject services={[Export]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={data} xName='month'
yName='sales' radius='100%'>
      </AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent></div>);
};
export default App;
ReactDOM.render(<App />, document.getElementById('charts'));
```

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Chart-Category Axis</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></sc
ript>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
```

```

        left: 45%;
        position: absolute;
        top: 45%;
        width: 30%;
    }
    #charts {
        height : 350px;
        display: block;
    }
</style>
</head>
<body>
    <div id='charts'>
        <div id='loader'>Loading....</div>
    </div>
</body>
</html>

```

Appearance in React Accumulation chart component

Custom Color Palette

You can customize the default color of series or points by providing a custom color palette of your choice by using the [palettes](#) property.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, PieSeries, Inject } from '@syncfusion/ej2-
react-charts';
import { accData } from 'datasource.ts';
function App() {
    const palettes = [
        '#E94649',
        '#F6B53F',
        '#6FAAB0',
        '#FF33F3',
        '#228B22',
        '#3399FF',
    ];
    return <AccumulationChartComponent id='charts'>
        <Inject services={[PieSeries]}/>
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' type='Pie' palettes={palettes}>
            </AccumulationSeriesDirective>
        </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, PieSeries, Inject } from '@syncfusion/ej2-react-charts';
import { accData } from 'datasource.ts';
function App() {
    const palettes: any[] = [
        '#E94649',
        '#F6B53F',
        '#6FAAB0',
        '#FF33F3',
        '#228B22',
        '#3399FF',
    ];
    return <AccumulationChartComponent id='charts'>
        <Inject services={[PieSeries]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' type='Pie' palettes= {palettes}>
                </AccumulationSeriesDirective>
            </AccumulationSeriesCollectionDirective>
        </AccumulationChartComponent>
    </>
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

Animation

Fluid Animation

Fluid animation used to animate series with updated dataSource continues animation rather than animation whole series. You can customize animation for a particular series using `animate` method.

INDEX.JSX

```
{% raw %}
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationChart, PieSeries,
AccumulationLegend, AccumulationDataLabel, AccumulationTooltip,
AccumulationSelection } from '@syncfusion/ej2-react-charts';
import { getInstance } from '@syncfusion/ej2-base';
import { accData } from 'datasource.ts';
function App() {
    let count = 0;
    function onChartLoad(args) {
        let pie = getInstance('#pie-chart2', AccumulationChart);
        document.getElementById('pie-chart2').setAttribute('title', '');
        pie.loaded = null;
        let pieinterval = setInterval(() => {
            if (document.getElementById('pie-chart2')) {
                if (count === 0) {
                    pie.series[0].dataSource = [
                        { x: 'Net-tution and Fees', y: 10 },
                        { x: 'Self-supporting Operations', y: 10 },
                    ];
                }
            }
        }, 1000);
    }
    return <AccumulationChartComponent id='pie-chart2'>
        <Inject services={[PieSeries]} />
        <AccumulationSeriesCollectionDirective>
            <AccumulationSeriesDirective dataSource={accData} xName='x'
yName='y' type='Pie' palettes= {palettes}>
                </AccumulationSeriesDirective>
            </AccumulationSeriesCollectionDirective>
        </AccumulationChartComponent>
    </>
};
export default App;
ReactDOM.render(<App />, document.getElementById("pie-chart2"));
```

```

        { x: 'Private Gifts', y: 13 },
        { x: 'All Other', y: 14 },
        { x: 'Local Revenue', y: 9 },
        { x: 'State Revenue', y: 13 },
        { x: 'Federal Revenue', y: 8 }
    ];
    pie.animate();
    count++;
}
else if (count === 1) {
    pie.series[0].dataSource = [
        { x: 'Net-tution and Fees', y: 120 },
        { x: 'Self-supporting Operations', y: 31 },
        { x: 'Private Gifts', y: 6 },
        { x: 'All Other', y: 12 },
        { x: 'Local Revenue', y: 25 },
        { x: 'State Revenue', y: 11 },
        { x: 'Federal Revenue', y: 12 }
    ];
    pie.animate();
    count++;
}
else if (count === 2) {
    pie.series[0].dataSource = [
        { x: 'Net-tution and Fees', y: 6 },
        { x: 'Self-supporting Operations', y: 22 },
        { x: 'Private Gifts', y: 11 },
        { x: 'All Other', y: 15 },
        { x: 'Local Revenue', y: 13 },
        { x: 'State Revenue', y: 10 },
        { x: 'Federal Revenue', y: 8 }
    ];
    pie.animate();
    count++;
}
else if (count === 3) {
    pie.series[0].dataSource = [
        { x: 'Net-tution and Fees', y: 15 },
        { x: 'Self-supporting Operations', y: 10 },
        { x: 'Private Gifts', y: 18 },
        { x: 'All Other', y: 20 },
        { x: 'Local Revenue', y: 30 },
        { x: 'State Revenue', y: 20 },
        { x: 'Federal Revenue', y: 25 }
    ];
    pie.animate();
    count++;
}
else if (count === 4) {
    pie.series[0].dataSource = [
        { x: 'Net-tution and Fees', y: 21 },
        { x: 'Self-supporting Operations', y: 10 },
        { x: 'Private Gifts', y: 15 },
        { x: 'All Other', y: 15 },
        { x: 'Local Revenue', y: 11 },
        { x: 'State Revenue', y: 20 },
        { x: 'Federal Revenue', y: 60 }
    ];

```

```

        ];
        pie.animate();
        count = 0;
    }
}
else {
    clearInterval(pieinterval);
}
}, 3000);
}
return (<AccumulationChartComponent id="pie-chart2" title="Education
Institutional Revenue" loaded={onChartLoad.bind(this)}>
    <Inject services=[
        AccumulationLegend,
        PieSeries,
        AccumulationDataLabel,
        AccumulationTooltip,
        AccumulationSelection
    ]/>
    <AccumulationSeriesCollectionDirective>
        <AccumulationSeriesDirective name="Revenue" dataSource={accData}
xName="x" yName="y" innerRadius="40%" startAngle={0} endAngle={360}
dataLabel={{
            visible: true,
            position: 'Inside',
        }}></AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
</AccumulationChartComponent>);
}
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 {
    AccumulationChartComponent,
    AccumulationSeriesCollectionDirective,
    AccumulationSeriesDirective,
    Inject,
    AccumulationChart,
    PieSeries,
    AccumulationLegend,
    AccumulationDataLabel,
    AccumulationTooltip,
    AccumulationSelection,
    ILoadedEventArgs
} from '@syncfusion/ej2-react-charts';
import { getInstance } from '@syncfusion/ej2-base';
import { accData } from 'datasource.ts';
function App() {
    let count = 0;
    function onChartLoad(args: ILoadedEventArgs): void {

```

```

let pie = getInstance('#pie-chart2', AccumulationChart);
document.getElementById('pie-chart2').setAttribute('title', '');
pie.loaded = null;
let pieinterval = setInterval(() => {
  if (document.getElementById('pie-chart2')) {
    if (count === 0) {
      pie.series[0].dataSource = [
        { x: 'Net-tution and Fees', y: 10 },
        { x: 'Self-supporting Operations', y: 10 },
        { x: 'Private Gifts', y: 13 },
        { x: 'All Other', y: 14 },
        { x: 'Local Revenue', y: 9 },
        { x: 'State Revenue', y: 13 },
        { x: 'Federal Revenue', y: 8 }
      ];
      pie.animate();
      count++;
    } else if (count === 1) {
      pie.series[0].dataSource = [
        { x: 'Net-tution and Fees', y: 120 },
        { x: 'Self-supporting Operations', y: 31 },
        { x: 'Private Gifts', y: 6 },
        { x: 'All Other', y: 12 },
        { x: 'Local Revenue', y: 25 },
        { x: 'State Revenue', y: 11 },
        { x: 'Federal Revenue', y: 12 }
      ];
      pie.animate();
      count++;
    } else if (count === 2) {
      pie.series[0].dataSource = [
        { x: 'Net-tution and Fees', y: 6 },
        { x: 'Self-supporting Operations', y: 22 },
        { x: 'Private Gifts', y: 11 },
        { x: 'All Other', y: 15 },
        { x: 'Local Revenue', y: 13 },
        { x: 'State Revenue', y: 10 },
        { x: 'Federal Revenue', y: 8 }
      ];
      pie.animate();
      count++;
    } else if (count === 3) {
      pie.series[0].dataSource = [
        { x: 'Net-tution and Fees', y: 15 },
        { x: 'Self-supporting Operations', y: 10 },
        { x: 'Private Gifts', y: 18 },
        { x: 'All Other', y: 20 },
        { x: 'Local Revenue', y: 30 },
        { x: 'State Revenue', y: 20 },
        { x: 'Federal Revenue', y: 25 }
      ];
      pie.animate();
      count++;
    } else if (count === 4) {
      pie.series[0].dataSource = [
        { x: 'Net-tution and Fees', y: 21 },
        { x: 'Self-supporting Operations', y: 10 },

```



```

        { x: 'Private Gifts', y: 15 },
        { x: 'All Other', y: 15 },
        { x: 'Local Revenue', y: 11 },
        { x: 'State Revenue', y: 20 },
        { x: 'Federal Revenue', y: 60 }
    ];
    pie.animate();
    count = 0;
  }
} else {
  clearInterval(pieinterval);
}
}, 3000);
}
return (
  <AccumulationChartComponent
    id="pie-chart2"
    title="Education Institutional Revenue"
    loaded={onChartLoad.bind(this)}
  >
    <Inject
      services={[
        AccumulationLegend,
        PieSeries,
        AccumulationDataLabel,
        AccumulationTooltip,
        AccumulationSelection
      ]}
    />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective
        name="Revenue"
        dataSource={accData}
        xName="x"
        yName="y"
        innerRadius="40%"
        startAngle={0}
        endAngle={360}
        dataLabel={{
          visible: true,
          position: 'Inside',
        }}
      ></AccumulationSeriesDirective>
    </AccumulationSeriesCollectionDirective>
  </AccumulationChartComponent>
);
}
export default App;
ReactDOM.render(<App />, document.getElementById('charts'));
{% enddraw %}

```

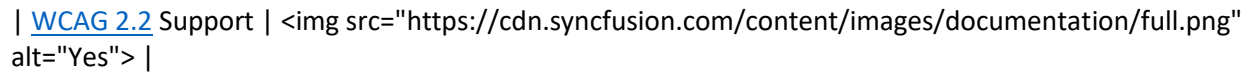
Accessibility in React Accumulation chart component

The Accumulation 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 Accumulation chart component is outlined below.

| Accessibility Criteria | Compatibility |

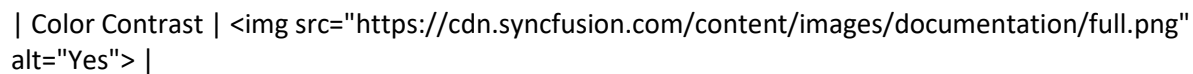
| -- | -- |

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

- `img (role)`

- button (role)
- region (role)
- aria-label (attribute)
- aria-hidden (attribute)
- aria-pressed (attribute)

Keyboard interaction

The Accumulation 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 Accumulation chart component.

| Press | To do this |

| --- | --- |

| Alt + J | Moves the focus to the Accumulation chart element. |

| Tab | Moves the focus to the next element in the Accumulation chart. |

| Shift + Tab | Moves the focus to the previous element in the Accumulation 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. |

| Down/Left Arrow | Moves the focus to the legend left side from the selected legend. |

| Up/Right Arrow | Moves the focus to the legend right side from the selected legend. |

| Enter/Space | Toggles the visibility of the corresponding series. |

| Ctrl + P | Prints the Accumulation chart. |

Ensuring accessibility

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

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

See also

- [Accessibility in Syncfusion React components](#)

Ej1 api migration in React Accumulation chart component

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

Accumulation Chart

<!-- markdownlint-disable MD033 -->

<!-- markdownlint-disable MD038 -->

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

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

```

|Annotations| Property: annotations.content <br/><br/>annotations: [{ content: "watermark"
}],<br/><EJ.Chart <br/> annotations={annotations}> <br/></EJ.Chart>| Property:
annotations.content <br/><br/><AccumulationChartComponent
id='charts'><br/><AccumulationAnnotationDirective content='<div>Highest Medal
Count</div>'><br/></AccumulationAnnotationDirective><br/></AccumulationChartComponent
>|

|background| Property: background <br/><br/><EJ.Chart <br/> background='#DDCCEE'>
<br/></EJ.Chart>| Property: annotations.content <br/><br/><AccumulationChartComponent
id='charts' background='DDCCEE'><br/></AccumulationChartComponent>|

|border| Property: border <br/><br/><EJ.Chart <br/> border: { }><br/></EJ.Chart>| Property: border
<br/><br/><AccumulationChartComponent id='charts' border: { color: 'red', width:
2}><br/></AccumulationChartComponent>|

|dataSource| Not applicable| Property: dataSource <br/><br/><AccumulationChartComponent
id='charts' dataSource: [ ]><br/></AccumulationChartComponent>|

|Animation after legend click| Not applicable| Property: enableAnimation
<br/><br/><AccumulationChartComponent id='charts' enableAnimation=
true;><br/></AccumulationChartComponent>|

|Persisting component's state between page reloads| Not applicable| Property: enablePersistence
<br/><br/><AccumulationChartComponent id='charts'
enablePersistence={true};><br/></AccumulationChartComponent>|

|Enabling smart labels| Property: series.enableSmartLabels <br/><br/>var series= [{ enableSmartLabels:
true }],<br/><EJ.Chart <br/> series={series}><br/></EJ.Chart>| Property:
series.enableSmartLabels<br/><br/><AccumulationChartComponent
id='charts';><br/><AccumulationSeriesDirective<br/>enableSmartLabels={true}><br/></Accumul
ationSeriesDirective><br/></AccumulationChartComponent>`|

|Height of Chart| Property: size.height <br/><br/><EJ.Chart <br/> size={ height: '400'
}><br/></EJ.Chart>| Property: height<br/><br/><AccumulationChartComponent id='charts'
height='400';><br/></AccumulationChartComponent>|

|Multi selection| Property: selectionSettings.type <br/><br/>var selectionSettings= [{ type:
'multiple'}],<br/><EJ.Chart <br/> selectionSettings={selectionSettings}><br/></EJ.Chart>| Property:
isMultiSelect<br/><br/><AccumulationChartComponent id='charts' isMultiSelect:
true;><br/></AccumulationChartComponent>|

|legend Settings| Property: legend <br/><br/><EJ.Chart <br/> legend: { }><br/></EJ.Chart>| Property:
isMultiSelect<br/><br/>public legendSettings: LegendSettingsModel = { visible: true
};<br/><AccumulationChartComponent id='charts'
legendSettings={this.legendSettings};><br/></AccumulationChartComponent>|

|Margin for the chart| Property: margin <br/><br/><EJ.Chart <br/> margin: { top: 20, bottom: 23,
right: 15, left: 10 }><br/></EJ.Chart>| Property: isMultiSelect<br/><br/>public margin: { top: 20,
bottom: 23, right: 15, left: 10 };<br/><AccumulationChartComponent id='charts'
margin={this.margin};><br/></AccumulationChartComponent>|

```

```

|SelectedDataIndexes| Property: selectedDataPointIndexes <br/><br/><EJ.Chart <br/>
selectedDataPointIndexes : [{}]><br/></EJ.Chart>| Property: selectedDataIndexes<br/><br/>public
selectedDataIndexes : [ { series: 0, point: 1}];<br/><AccumulationChartComponent id='charts'
selectedDataIndexes={this.selectedDataIndexes};><br/></AccumulationChartComponent>|

|Selection Mode| Property: selectionSettings.mode <br/><br/><EJ.Chart <br/> selectionSettings: {
mode: 'Point'}><br/></EJ.Chart>| Property:
selectedDataIndexes<br/><br/><AccumulationChartComponent id='charts' selectionMode :
'Point';><br/></AccumulationChartComponent>|

|Series| Property: series <br/><br/><EJ.Chart <br/> series: []><br/></EJ.Chart>| Property:
series<br/><br/><AccumulationChartComponent
id='charts';><br/><AccumulationSeriesDirective><br/></AccumulationSeriesDirective><br/></Ac
cumulationChartComponent>|

|Title text| Property: title.text <br/><br/><EJ.Chart <br/> title: { text: 'Pie Chart'
}><br/></EJ.Chart>| Property: title<br/><br/><AccumulationChartComponent id='charts' title: 'Pie
Chart';><br/></AccumulationChartComponent>|

|Title Styles| Property: title <br/><br/><EJ.Chart <br/> title: { text: 'Pie Chart'
}><br/></EJ.Chart>| Property: titleStyle<br/><br/><AccumulationChartComponent id='charts'
titleStyle: { fontFamily: 'SegoeUI'};><br/></AccumulationChartComponent>|

|Title Styles| Property: title <br/><br/><EJ.Chart <br/> title: { text: 'Pie Chart'
}><br/></EJ.Chart>| Property: titleStyle<br/><br/><AccumulationChartComponent id='charts'
titleStyle: { fontFamily: 'SegoeUI'};><br/></AccumulationChartComponent>|

|Sub Title text| Property: subTitle.text <br/><br/>var subTitle={ text: 'Pie Chart' }<br/><EJ.Chart <br/>
subTitle={subTitle}><br/></EJ.Chart>| Property: subTitle<br/><br/>public subTitle='Pie
Chart';<br/><AccumulationChartComponent id='charts'
subTitle={this.subTitle};><br/></AccumulationChartComponent>|

|Sub title Styles| Property: subTitleStyle <br/><br/>var subTitleStyle={ fontFamily: 'SegoeUI'
}><br/><EJ.Chart <br/> subTitleStyle={subTitleStyle}><br/></EJ.Chart>| Property:
subTitleStyle<br/><br/>public subTitleStyle={ fontFamily: 'SegoeUI'
};<br/><AccumulationChartComponent id='charts'
subTitleStyle={this.subTitleStyle};><br/></AccumulationChartComponent>|

|tooltip| Property: series.tooltip <br/><br/>var series=[ { tooltip: { }}<br/><EJ.Chart <br/>
series={series}><br/></EJ.Chart>| Property: tooltip<br/><br/><AccumulationChartComponent
id='charts' tooltip={enable: true };><br/></AccumulationChartComponent>|

|Width of Chart| Property: size.width <br/><br/><EJ.Chart <br/> size={ width: '400'
}><br/></EJ.Chart>| Property: width<br/><br/><AccumulationChartComponent id='charts'
width='400';><br/></AccumulationChartComponent>|

```

Annotation

```
<!-- markdownlint-disable MD033 -->
```

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

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

| Annotations | **Property:** *annotations.content*

annotations: [{ content: "watermark" }],
<EJ.Chart
 annotations={annotations}>
</EJ.Chart> | **Property:** *annotations.content*

<AccumulationChartComponent id='charts'>
<AccumulationAnnotationDirective content='<div>Highest Medal Count</div>'>
</AccumulationAnnotationDirective>
</AccumulationChartComponent> |

| coordinate unit for annotation | **Property:** *annotations.coordinateUnit*

annotations: [{ coordinateUnit : "pixels" }],
<EJ.Chart
 annotations={annotations}>
</EJ.Chart> | **Property:** *annotations.coordinateUnits*

<AccumulationChartComponent id='charts'>
<AccumulationAnnotationDirective coordinateUnits='Pixels'>
</AccumulationAnnotationsDirective>
</AccumulationChartComponent> |

| description | Not Applicable | **Property:** *description*

<AccumulationChartComponent id='charts'>
<AccumulationAnnotationDirective description='Pixel'>
</AccumulationAnnotationsDirective>
</AccumulationChartComponent> |

| horizontalAlignment for annotation | **Property:** *annotations.horizontalAlignment*

annotations=[{ horizontalAlignment: "middle" }],
<EJ.Chart
 annotations={annotations}>
</EJ.Chart> | **Property:** *annotations.horizontalAlignment*

<AccumulationChartComponent id='charts'>
<AccumulationAnnotationDirective horizontalAlignment='Near'>
</AccumulationAnnotationsDirective>
</AccumulationChartComponent> |

| margin for annotation | **Property:** *annotations.margin*

annotations: [margin: { right: 40 }],
<EJ.Chart
 annotations={annotations}>
</EJ.Chart> | Not applicable |

| Opacity for annotation | **Property:** *annotations.opacity*

annotations: [{ opacity: 0.4 }],
<EJ.Chart
 annotations={annotations}>
</EJ.Chart> | Not applicable |

| Region for annotation with respect to chart or series | **Property:** *annotations.region*

annotations: [{ region: "series" }],
<EJ.Chart
 annotations={annotations}>
</EJ.Chart> | **Property:** *annotations.region*

<AccumulationChartComponent id='charts'>
<AccumulationAnnotationDirective region='Series'>
</AccumulationAnnotationsDirective>
</AccumulationChartComponent> |

| verticalAlignment for annotation | **Property:** *annotations.verticalAlignment*

annotations: [{ verticalAlignment: "middle" }],
<EJ.Chart
 annotations={annotations}>
</EJ.Chart> | **Property:** *annotations.verticalAlignment*

<AccumulationChartComponent id='charts'>
<AccumulationAnnotationDirective verticalAlignment='Middle'>
</AccumulationAnnotationsDirective>
</AccumulationChartComponent> |

| Visibility of annotations | **Property:** *annotations.visible*

annotations: [{ visible: true }],
<EJ.Chart
 annotations={annotations}>
</EJ.Chart> | Not applicable

|X offset for annotation| **Property:** *annotations.x*

annotations: [{ x: 170 }],
<EJ.Chart

 annotations={annotations}>
</EJ.Chart>| **Property:** *annotations.x*

<AccumulationChartComponent id='charts'>
<AccumulationAnnotationDirective
x={170}>
</AccumulationAnnotationsDirective>
</AccumulationChartComponent>|

|X axis name in which annotation to be rendered| **Property:** *annotations.xAxisName*

annotations: [{ xAxisName : "xAxis" }],
<EJ.Chart
 annotations={annotations}>

</EJ.Chart>| **Property:** *annotations.xAxisName*

<AccumulationChartComponent
id='charts'>
<AccumulationAnnotationDirective
xAxisName='xAxis'>
</AccumulationAnnotationsDirective>
</AccumulationChartComp
onent>|

|Y offset for annotation| **Property:** *annotations.y*

annotations: [{ y: 170 }],
<EJ.Chart

 annotations={annotations}>
</EJ.Chart>| **Property:** *annotations.y*

<AccumulationChartComponent id='charts'>
<AccumulationAnnotationDirective
y={170}>
</AccumulationAnnotationsDirective>
</AccumulationChartComponent>|

|Y axis name in which annotation to be rendered| **Property:** *annotations.yAxisName*

annotations: [{ yAxisName : "yAxis" }],
<EJ.Chart
 annotations={annotations}>

</EJ.Chart>| **Property:** *annotations.yAxisName*

<AccumulationChartComponent
id='charts'>
<AccumulationAnnotationDirective
yAxisName='yAxis'>
</AccumulationAnnotationsDirective>
</AccumulationChartComp
onent>|

Series

<!-- markdownlint-disable MD033 -->

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

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

|series| **Property:** *series*

var series =[{ bearFillColor: 'red'}]
<EJ.Chart

series={series}>
</EJ.Chart>| **Property:** *series*

<AccumulationChartComponent
id='charts'>
<AccumulationSeriesDirective
bearFillColor='red'>
</AccumulationSeriesDirective>
</AccumulationChartComponent
>|

|Enable animation for series| **Property:** *enableAnimation*

var series =[{enableAnimation :
true}]
<EJ.Chart
 series={series}>
</EJ.Chart>| **Property:** *animation.enable*

<AccumulationChartComponent id='charts'>
<AccumulationSeriesDirective
animation={
enable={false}}>
</AccumulationSeriesDirective>
</AccumulationChartComponent>|

|Animation duration for series| **Property:** *animationDuration*

var series
=[{animationDuration: 1000}]
<EJ.Chart
 series={series}>
</EJ.Chart>| **Property:**
animation.duration

<AccumulationChartComponent
id='charts'>
<AccumulationSeriesDirective animation={ duration={1000}
}>
</AccumulationSeriesDirective>
</AccumulationChartComponent>|

| Animation delay for series | Not Applicable | **Property:** *animation.delay*

```
<br/><br/><AccumulationChartComponent id='charts'><br/><AccumulationSeriesDirective
animation={ delay={100}
}><br/></AccumulationSeriesDirective><br/></AccumulationChartComponent>|
```

| Border | **Property:** *border*

var series = [{ border: {color: 'red', width: 2}}]
<EJ.Chart

series={series}>
</EJ.Chart> | **Property:** *border*

<AccumulationChartComponent
id='charts'>
<AccumulationSeriesDirective border={ color: 'red', width:
2}>
</AccumulationSeriesDirective>
</AccumulationChartComponent>|

| DataLabel for series | **Property:** *dataLabel*

var series = [{ dataLabel: {}}]
<EJ.Chart

series={series}>
</EJ.Chart> | **Property:** *dataLabel*

<AccumulationChartComponent
id='charts'>
<AccumulationSeriesDirective dataLabel:
{}>
</AccumulationSeriesDirective>
</AccumulationChartComponent>|

| DataSource for series | **Property:** *dataSource*

var series = [{ dataSource: {} }]
<EJ.Chart

 series={series}>
</EJ.Chart> | **Property:** *dataSource*

<AccumulationChartComponent id='charts'>
<AccumulationSeriesDirective
dataSource: {}>
</AccumulationSeriesDirective>
</AccumulationChartComponent>|

| enableTooltip for series | **Property:** *tooltip.visible*

var series = [{ tooltip: { visible: true }
}]
<EJ.Chart
 series={series}>
</EJ.Chart> | **Property:** *maxRadius*

<AccumulationChartComponent id='charts'>
<AccumulationSeriesDirective
enableTooltip:
true>
</AccumulationSeriesDirective>
</AccumulationChartComponent>|

| start angle | **Property:** *startAngle*

var series = [{ startAngle: 80}]
<EJ.Chart

series={series}>
</EJ.Chart> | **Property:** *startAngle*

<AccumulationChartComponent
id='charts'>
<AccumulationSeriesDirective startAngle: 90
>
</AccumulationSeriesDirective>
</AccumulationChartComponent>|

| end angle | **Property:** *endAngle*

var series = [{ endAngle: 80}]
<EJ.Chart

series={series}>
</EJ.Chart> | **Property:** *endAngle*

<AccumulationChartComponent
id='charts'>
<AccumulationSeriesDirective endAngle:
90>
</AccumulationSeriesDirective>
</AccumulationChartComponent>|

| explode | **Property:** *explode*

var series = [{ explode: true}]
<EJ.Chart

series={series}>
</EJ.Chart> | **Property:** *explode*

<AccumulationChartComponent
id='charts'>
<AccumulationSeriesDirective explode:
true>
</AccumulationSeriesDirective>
</AccumulationChartComponent>|

| explodeAll | **Property:** *explodeAll*

var series = [{ explodeAll: true}]
<EJ.Chart

series={series}>
</EJ.Chart> | **Property:** *explodeAll*

<AccumulationChartComponent
id='charts'>
<AccumulationSeriesDirective
explodeAll=true>
</AccumulationSeriesDirective>
</AccumulationChartComponent>|

| explodeIndex | **Property:** *explodeIndex*

var series = [{ explodeIndex: 0}]
<EJ.Chart

series={series}>
</EJ.Chart> | **Property:** *explodeIndex*

<AccumulationChartComponent id='charts'>
<AccumulationSeriesDirective


```
explodeIndex={1}></AccumulationSeriesDirective></AccumulationChartComponent>
|
```

```
|explodeOffset| Property: explodeOffset <br/><br/>var series = [{ explodeOffset: '30%' }]<br/><EJ.Chart
<br/> series={series}> </EJ.Chart> | Property: explodeOffset
<br/><br/><AccumulationChartComponent id='charts'><br/><AccumulationSeriesDirective
explodeOffset='30%'></AccumulationSeriesDirective></AccumulationChartCompone
nt>|
```

```
|gapRatio| Property: gapRatio <br/><br/>var series = [{ gapRatio: 0.6 }]<br/><EJ.Chart <br/>
series={series}> </EJ.Chart> | Property: gapRatio <br/><br/><AccumulationChartComponent
id='charts'><br/><AccumulationSeriesDirective
gapRatio={0.6}></AccumulationSeriesDirective></AccumulationChartComponent> |
```

```
|gapWidth| Property: gapWidth <br/><br/>var series = [{ gapWidth: 0.6 }]<br/><EJ.Chart <br/>
series={series}> </EJ.Chart> | Not Applicable |
```

```
|inner radius of the accumulation chart| Property: innerRadius <br/><br/>var series = [{ innerRadius :
'30%' }]<br/><EJ.Chart <br/> series={series}> </EJ.Chart> | Property: innerRadius
<br/><br/><AccumulationChartComponent id='charts'><br/><AccumulationSeriesDirective
innerRadius='30%'></AccumulationSeriesDirective></AccumulationChartComponent
>|
```

```
|Legend shape of the series| Not applicable | Property: legendShape
<br/><br/><AccumulationChartComponent id='charts'><br/><AccumulationSeriesDirective
legendShape
='Rectangle'></AccumulationSeriesDirective></AccumulationChartComponent> |
```

```
|name of the series| Property: name <br/><br/>var series = [{ name : '30%' }]<br/><EJ.Chart <br/>
series={series}> </EJ.Chart> | Property: name <br/><br/><AccumulationChartComponent
id='charts'><br/><AccumulationSeriesDirective name=
'30%'></AccumulationSeriesDirective></AccumulationChartComponent> |
```

```
|neck height for funnel series| Property: neckHeight <br/><br/>var series = [{
neckHeight='30%' }]<br/><EJ.Chart <br/> series={series}> </EJ.Chart> | Property: neckHeight
<br/><br/><AccumulationChartComponent id='charts'><br/><AccumulationSeriesDirective
neckHeight='30%'></AccumulationSeriesDirective></AccumulationChartComponent>
|
```

```
|neck width for funnel series| Property: neckWidth <br/><br/>var series = [{neckWidth='30%'
}]<br/><EJ.Chart <br/> series={series}> </EJ.Chart> | Property: neckWidth
<br/><br/><AccumulationChartComponent id='charts'><br/><AccumulationSeriesDirective
neckWidth='30%'></AccumulationSeriesDirective></AccumulationChartComponent>
|
```

```
|opacity for series| Property: opacity <br/><br/>var series = [{ opacity : 0.4 }]<br/><EJ.Chart <br/>
series={series}> </EJ.Chart> | Property: opacity <br/><br/><AccumulationChartComponent
id='charts'><br/><AccumulationSeriesDirective opacity
=0.5></AccumulationSeriesDirective></AccumulationChartComponent> |
```

| palettes for series | **Property:** *palette*

var series = [{palette : []}]
<EJ.Chart
 series={series}>
</EJ.Chart> | **Property:** *palettes*

<AccumulationChartComponent id='charts'>
<AccumulationSeriesDirective palettes=[]>
</AccumulationSeriesDirective>
</AccumulationChartComponent> |

| point color mapping name for series | **Property:** *pointColorMappingName*

var series = [{pointColorMappingName : 'color'}]
<EJ.Chart
 series={series}>
</EJ.Chart> | **Property:** *pointColorMapping*

<AccumulationChartComponent id='charts'>
<AccumulationSeriesDirective pointColorMappingName='color'>
</AccumulationSeriesDirective>
</AccumulationChartComponent> |

| Mode of pyramid series | **Property:** *pyramidMode*

var series = [{pyramidMode : 'Surface'}]
<EJ.Chart
 series={series}>
</EJ.Chart> | **Property:** *pyramidMode*

<AccumulationChartComponent id='charts'>
<AccumulationSeriesDirective pyramidMode : 'Linear'>
</AccumulationSeriesDirective>
</AccumulationChartComponent> |

| query for datasource for series | **Property:** *query*

var series = [{query: ""}]
<EJ.Chart
 series={series}>
</EJ.Chart> | **Property:** *query*

<AccumulationChartComponent id='charts'>
<AccumulationSeriesDirective query= "">
</AccumulationSeriesDirective>
</AccumulationChartComponent> |

| Radius of Pie series | **Property:** *pieCoefficient*

var series = [{pieCoefficient : 0.5}]
<EJ.Chart
 series={series}>
</EJ.Chart> | **Property:** *radius*

<AccumulationChartComponent id='charts'>
<AccumulationSeriesDirective radius='50%'>
</AccumulationSeriesDirective>
</AccumulationChartComponent> |

| Selection Style of Accumulation chart | Not applicable | **Property:** *selectionStyle*

<AccumulationChartComponent id='charts'>
<AccumulationSeriesDirective selectionStyle="">
</AccumulationSeriesDirective>
</AccumulationChartComponent> |

| tooltip Mapping name | Not applicable | **Property:** *tooltipMappingName*

<AccumulationChartComponent id='charts'>
<AccumulationSeriesDirective tooltipMappingName="">
</AccumulationSeriesDirective>
</AccumulationChartComponent> |

| Type of series | **Property:** *type*

var series = [{type : 'Pie'}]
<EJ.Chart
 series={series}>
</EJ.Chart> | **Property:** *type*

<AccumulationChartComponent id='charts'>
<AccumulationSeriesDirective type='Pie'>
</AccumulationSeriesDirective>
</AccumulationChartComponent> |

| Name of the property in the datasource that contains x value for the series | **Property:** *xName*

var series = [{xName : 'x'}]
<EJ.Chart
 series={series}>
</EJ.Chart> | **Property:** *xName*

<AccumulationChartComponent id='charts'>
<AccumulationSeriesDirective xName='x'>
</AccumulationSeriesDirective>
</AccumulationChartComponent> |

| Name of the property in the datasource that contains x value for the series. | **Property:** *xName*

var series = [{xName : 'x'}]
<EJ.Chart
 series={series}>
</EJ.Chart> | **Property:** *xName*

<AccumulationChartComponent

```
id='charts'><br/><AccumulationSeriesDirective
xName='x'><br/></AccumulationSeriesDirective><br/></AccumulationChartComponent>|
```

| Specifies the name of the y-axis that has to be associated with this series. Add an axis instance with this name to axes collection. | **Property:** *yAxisName*

var series = [{yAxisName : 'secondaryYAxis'}]
<EJ.Chart
 series={series}>
</EJ.Chart> | **Property:** *yAxisName*

<AccumulationChartComponent id='charts'>
<AccumulationSeriesDirective yAxisName='secondaryYAxis'>
</AccumulationSeriesDirective>
</AccumulationChartComponent>|

| Name of the property in the datasource that contains y value for the series. | **Property:** *yName*

var series = [{yName : 'y'}]
<EJ.Chart
 series={series}>
</EJ.Chart> | **Property:** *yName*

<AccumulationChartComponent id='charts'>
<AccumulationSeriesDirective yName='y'>
</AccumulationSeriesDirective>
</AccumulationChartComponent>|

| Width of funnel series. | **Property:** *funnelWidth*

var series = [{funnelWidth : '100'}]
<EJ.Chart
 series={series}>
</EJ.Chart> | **Property:** *width*

<AccumulationChartComponent id='charts'>
<AccumulationSeriesDirective width='100'>
</AccumulationSeriesDirective>
</AccumulationChartComponent>|

| Grouping | Not Applicable | **Property:** *groupTo*

<AccumulationChartComponent id='charts'>
<AccumulationSeriesDirective groupTo='100'>
</AccumulationSeriesDirective>
</AccumulationChartComponent>|

| GroupMode | Not Applicable | **Property:** *groupMode*

<AccumulationChartComponent id='charts'>
<AccumulationSeriesDirective groupMode='Point'>
</AccumulationSeriesDirective>
</AccumulationChartComponent>|

DataLabel

<!-- markdownlint-disable MD033 -->

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

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

| DataLabelSettings of marker | **Property:** *marker.dataLabel*

var series = [{marker:{ dataLabel: { }}}]
<EJ.Chart
 series={series}>
</EJ.Chart> | **Property:** *marker.dataLabel*

<AccumulationChartComponent id='charts'>
<AccumulationSeriesDirective marker={ dataLabel: { }}>
</AccumulationSeriesDirective>
</AccumulationChartComponent>|

| border of dataLabel | **Property:** *marker.dataLabel.border*

var series = [{marker:{ dataLabel: { border: { } } } }]
<EJ.Chart
 series={series}>
</EJ.Chart> | **Property:** *dataLabel.border*

<AccumulationChartComponent id='charts'>
<AccumulationSeriesDirective marker={ dataLabel: { border: { } } }>
</AccumulationSeriesDirective>
</AccumulationChartComponent>|

| connector style for dataLabel connector line | **Property:** *marker.dataLabel.connectorLine*

var series = [{marker:{ dataLabel: { connectorLine: { } } } }]
<EJ.Chart
 series={series}>

```

</EJ.Chart>|Property: connectorStyle </></EJ.Chart><AccumulationChartComponent
id='charts'><br/><AccumulationSeriesDirective marker={ dataLabel: {connectorStyle: {}
}}></AccumulationSeriesDirective></></AccumulationChartComponent>|
| Fill for dataLabel| Property: marker.dataLabel.fill <br/><br/>var series =[{marker:{ dataLabel: { fill: 'red'
}}}]<br/><EJ.Chart <br/> series={series}> <br/></EJ.Chart>|Property: fill
<br/><br/><AccumulationChartComponent id='charts'><br/><AccumulationSeriesDirective
marker={ dataLabel: {fill: 'pink'
}}></AccumulationSeriesDirective></></AccumulationChartComponent>|
| font for dataLabel| Property: marker.dataLabel.font <br/><br/>var series =[{marker:{ dataLabel: { font:
}}}]<br/><EJ.Chart <br/> series={series}> <br/></EJ.Chart>|Property: font
<br/><br/><AccumulationChartComponent id='charts'><br/><AccumulationSeriesDirective
marker={ dataLabel: {font: { }
}}></AccumulationSeriesDirective></></AccumulationChartComponent>|
| position for dataLabel| Property: marker.dataLabel.position <br/><br/>var series =[{marker:{
dataLabel: { position: 'Inside'} } }]<br/><EJ.Chart <br/> series={series}> <br/></EJ.Chart>|Property:
position <br/><br/><AccumulationChartComponent
id='charts'><br/><AccumulationSeriesDirective marker={ dataLabel: {position: 'Outside'
}}></AccumulationSeriesDirective></></AccumulationChartComponent>|
| Rounded corner radius X| Not Applicable|Property: dataLabel.rx
<br/><br/><AccumulationChartComponent id='charts'><br/><AccumulationSeriesDirective
marker={ dataLabel: { rx: 10
}}></AccumulationSeriesDirective></></AccumulationChartComponent>|
| Rounded corner radius Y| Not Applicable|Property: dataLabel.ry
<br/><br/><AccumulationChartComponent id='charts'><br/><AccumulationSeriesDirective
marker={ dataLabel: { ry: 10
}}></AccumulationSeriesDirective></></AccumulationChartComponent>|
| HTML template in dataLabel| Property: dataLabel.template <br/><br/>var series =[{marker:{
dataLabel: { template: '<div>Chart</div>' } } }]<br/><EJ.Chart <br/> series={series}>
<br/></EJ.Chart>|Property: dataLabel.template <br/><br/><AccumulationChartComponent
id='charts'><br/><AccumulationSeriesDirective marker={ dataLabel: {template:
'<div>Chart</div>'
}}></AccumulationSeriesDirective></></AccumulationChartComponent>|

```

Legend

<!-- markdownlint-disable MD033 -->

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

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

```

| Default legend|Property: visible <br/><br/><EJ.Chart <br/> legend: { visible: true
}></EJ.Chart>|Property: visible<br/><br/>public legendSettings: LegendSettingsModel = { visible:
true };<br/><AccumulationChartComponent id='charts'
legendSettings={this.legendSettings};><br/></AccumulationChartComponent>|

```

| Legend height | **Property:** *size.height*

<EJ.Chart
 legend: { size : { height: 50 } }>
</EJ.Chart> | **Property:** *height*

public legendSettings: LegendSettingsModel = { height: '30' };
<AccumulationChartComponent id='charts' legendSettings={this.legendSettings};>
</AccumulationChartComponent> |

| Legend width | **Property:** *size.width*

<EJ.Chart
 legend: { size : { width: 50 } }>
</EJ.Chart> | **Property:** *width*

public legendSettings: LegendSettingsModel = { width: '30' };
<AccumulationChartComponent id='charts' legendSettings={this.legendSettings};>
</AccumulationChartComponent> |

| Legend location | **Property:** *size.location*

<EJ.Chart
 legend: { location: { x: 3, y: 45 } }>
</EJ.Chart> | **Property:** *location*

public legendSettings: LegendSettingsModel = { location: { x: 3, y: 45 } };
<AccumulationChartComponent id='charts' legendSettings={this.legendSettings};>
</AccumulationChartComponent> |

| Legend position | **Property:** *size.position*

<EJ.Chart
 legend: { position: 'top' } }>
</EJ.Chart> | **Property:** *position*

public legendSettings: LegendSettingsModel = { position: 'top' };
<AccumulationChartComponent id='charts' legendSettings={this.legendSettings};>
</AccumulationChartComponent> |

| Legend padding | Not applicable | **Property:** *padding*

public legendSettings: LegendSettingsModel = { padding: 8 };
<AccumulationChartComponent id='charts' legendSettings={this.legendSettings};>
</AccumulationChartComponent> |

| Legend alignment | **Property:** *alignment*

<EJ.Chart
 legend: { alignment: 'center' } }>
</EJ.Chart> | **Property:** *alignment*

public legendSettings: LegendSettingsModel = { alignment: 'center' };
<AccumulationChartComponent id='charts' legendSettings={this.legendSettings};>
</AccumulationChartComponent> |

| text style for legend | **Property:** *font*

<EJ.Chart
 legend: { font: { fontFamily: "" } }>
</EJ.Chart> | **Property:** *textStyle*

public legendSettings: LegendSettingsModel = { textStyle: { size: '12' } };
<AccumulationChartComponent id='charts' legendSettings={this.legendSettings};>
</AccumulationChartComponent> |

| shape height of legend | **Property:** *itemStyle.height*

<EJ.Chart
 legend: { itemStyle: { height: 20 } }>
</EJ.Chart> | **Property:** *shapeHeight*

public legendSettings: LegendSettingsModel = { shapeHeight: 20 };
<AccumulationChartComponent id='charts' legendSettings={this.legendSettings};>
</AccumulationChartComponent> |

| shape width of legend | **Property:** *itemStyle.width*

<EJ.Chart
 legend: { itemStyle: { width: 20 } }>
</EJ.Chart> | **Property:** *shapeWidth*

public legendSettings: LegendSettingsModel = { shapeWidth: 20 };
<AccumulationChartComponent id='charts' legendSettings={this.legendSettings};>
</AccumulationChartComponent> |

| shape border of legend | **Property:** *itemStyle.border*

<EJ.Chart
 legend: { itemStyle: { border: { width: 2 } } }>
</EJ.Chart> | Not Applicable |

| shape padding of legend | **Property:** *itemPadding*

<EJ.Chart
 legend: { itemPadding: 10 } }>
</EJ.Chart> | **Property:** *shapePadding*

public legendSettings:

```
LegendSettingsModel = { shapePadding: 20 };<br/><AccumulationChartComponent id='charts'
legendSettings={this.legendSettings};><br/></AccumulationChartComponent>|
```

```
| Background of legend | Property: background <br/><br/><EJ.Chart <br/> legend: { background:
'transparent' } }><br/></EJ.Chart> | Property: backgorund<br/><br/>public legendSettings:
LegendSettingsModel = { background: 'transparent' };<br/><AccumulationChartComponent
id='charts' legendSettings={this.legendSettings};><br/></AccumulationChartComponent>|
```

```
| Opacity of legend | Property: opacity <br/><br/><EJ.Chart <br/> legend: { opacity:
0.3}><br/></EJ.Chart> | Property: opacity<br/><br/>public legendSettings: LegendSettingsModel = {
opacity: 0.4 };<br/><AccumulationChartComponent id='charts'
legendSettings={this.legendSettings};><br/></AccumulationChartComponent>|
```

```
| Toggle visibility of series while legend click | Property: toggleSeriesVisibility <br/><br/><EJ.Chart <br/>
legend: { toggleSeriesVisibility: true}><br/></EJ.Chart> | Property: toggleVisibility<br/><br/>public
legendSettings: LegendSettingsModel = { toggleVisibility: true };<br/><AccumulationChartComponent
id='charts' legendSettings={this.legendSettings};><br/></AccumulationChartComponent>|
```

```
| Title for legend | Property: title <br/><br/><EJ.Chart <br/> legend: { title: { text:
'LegendTitle' } }><br/></EJ.Chart> | Not applicable|
```

```
| Text Overflow for legend | Property: textOverFlow <br/><br/><EJ.Chart <br/> legend={
textOverFlow: 'trim'}><br/></EJ.Chart> | Property: textStyle.textOverFlow<br/><br/>public
legendSettings: LegendSettingsModel = { text: { textOverFlow: 'trim' }
};<br/><AccumulationChartComponent id='charts'
legendSettings={this.legendSettings};><br/></AccumulationChartComponent>|
```

```
| Text width for legend while setting text overflow | Property: textWidth <br/><br/><EJ.Chart <br/>
legend={ textWidth: 20}><br/></EJ.Chart> | Not applicable|
```

```
| Scroll bar for legend | Property: enableScrollBar <br/><br/><EJ.Chart <br/> legend={ enableScrollBar:
true}><br/></EJ.Chart> | Not applicable|
```

```
| Row count for legend | Property: rowCount <br/><br/><EJ.Chart <br/> legend={ rowCount:
2}><br/></EJ.Chart> | Not applicable|
```

```
| Column count for legend | Property: columnCount <br/><br/><EJ.Chart <br/> legend={ columnCount:
2}><br/></EJ.Chart> | Not applicable|
```

```
| Color for legend items | Property: fill <br/><br/><EJ.Chart <br/> legend={ fill:
'#EEFFCC'}><br/></EJ.Chart> | Not applicable|
```

Methods

```
<!-- markdownlint-disable MD033 -->
```

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

```
| --- | --- | --- |
```

```
| animation for series | Property: animate() <br/><br/>var chartobj=
document.getElementById('chart');<br/><EJ.Chart <br/> id= chart>
<br/></EJ.Chart><br/>chartobj.animate();| Not applicable|
```

```
| Redraw for chart| Property: redraw() <br/><br/>var chartobj=
document.getElementById('chart');<br/><EJ.Chart <br/> id= chart>
<br/></EJ.Chart><br/>chartobj.redraw();| Property: refresh() <br/><br/>public loaded(args:
ILoadedEventArgs): void {<br/>function () =>
{<br/>args.chart.refresh();<br/><AccumulationChartComponent
id='charts'><br/></AccumulationChartComponent>|

| Export| Property: chart.export() <br/><br/>var chartObj =
$("#chartcontainer").ejChart("instance");<br/>chartObj.export("chartcontainer");<br/><EJ.Chart <br/>
id= chartcontainer><br/></EJ.Chart>| Property: chart.export() <br/><br/>public chartInstance:
ChartComponent;<br/>public clickHandler()
{<br/>this.chartInstance.export();<br/><AccumulationChartComponent
id='charts'><br/></AccumulationChartComponent>|

| Print| Property: chart.print() <br/><br/>var chartObj =
$("#chartcontainer").ejChart("instance");<br/>chartObj.print("chartcontainer");<br/><EJ.Chart <br/>
id= chartcontainer><br/></EJ.Chart>| Property: chart.print() <br/><br/>public chartInstance:
ChartComponent;<br/>public clickHandler()
{<br/>this.chartInstance.print();<br/><AccumulationChartComponent
id='charts'><br/></AccumulationChartComponent>|

| AddSeries| Not Applicable| Property: chart.addSeries() <br/><br/>public add()
{<br/>this.chartInstance.addSeries([{}]);<br/><AccumulationChartComponent
id='charts'><br/></AccumulationChartComponent>|

| RemoveSeries| Not Applicable| Property: chart.removeSeries() <br/><br/>public add()
{<br/>this.chartInstance.removeSeries([{}]);<br/><AccumulationChartComponent
id='charts'><br/></AccumulationChartComponent>|
```

Events

```
<!-- markdownlint-disable MD033 -->
```

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

```
| --- | --- | --- |
```

```
| Fires on annotation click| Property: annotationClick <br/><br/><EJ.Chart <br/> id= chartcontainer
annotationClick = {annotationClick}><br/></EJ.Chart><br/>function annotationClick({});| Not
applicable|
```

```
| Fires after animation| Property: animationComplete() <br/><br/><EJ.Chart <br/> id= chartcontainer
animationComplete = {animationComplete}><br/></EJ.Chart><br/>function animationComplete({
});| Property: animationComplete() <br/><br/><AccumulationChartComponent <br/> id='charts'
animationComplete={this.animationComplete.bind(this)}'><br/></AccumulationChartCompone
nt><br/>public animationComplete(args:IAAnimationCompleteEventArgs): void {};|
```

```
| Fires on axis label click| Property: axisLabelClick <br/><br/><EJ.Chart <br/> id= chartcontainer
axisLabelClick = {axisLabelClick}><br/></EJ.Chart><br/>function axisLabelClick({});| Not applicable|
```

```
| Fires before axis label render| Property: axisLabelRendering <br/><br/><EJ.Chart <br/> id=
chartcontainer axisLabelRendering = {axisLabelRendering}><br/></EJ.Chart><br/>function
```



```

axisLabelRendering(){ }; | Property: axisLabelRender() <br/><br/><AccumulationChartComponent
<br/> id='charts
axisLabelRender={this.axisLabelRender.bind(this)}><br/></AccumulationChartComponent><br/>
>public axisLabelRender(): void {}; |

| Fires on axis label mouseMove | Property: axisLabelMouseMove <br/><br/><EJ.Chart <br/> id=
chartcontainer axisLabelMouseMove = {axisLabelMouseMove}><br/></EJ.Chart><br/>function
axisLabelRendering(){ }; | Not applicable |

| Fires on axis label initialize | Property: axisLabelInitialize <br/><br/><EJ.Chart <br/> id=
chartcontainer axisLabelInitialize = {axisLabelInitialize}><br/></EJ.Chart><br/>function
axisLabelInitialize(){ }; | Not applicable |

| Fires before axis range calculation | Property: axesRangeCalculate <br/><br/><EJ.Chart <br/> id=
chartcontainer axesRangeCalculate = {axesRangeCalculate}><br/></EJ.Chart><br/>function
axesRangeCalculate(){ }; | Property: axisRangeCalculated <br/><br/><AccumulationChartComponent
<br/> id='charts
axisRangeCalculated={this.axisRangeCalculated.bind(this)}><br/></AccumulationChartCompone
nt><br/>public axisRangeCalculated(): void {}; |

| Fires on axis title rendering | Property: axisTitleRendering <br/><br/><EJ.Chart <br/> id=
chartcontainer axisTitleRendering = {axisTitleRendering}><br/></EJ.Chart><br/>function
axisTitleRendering(){ }; | Not applicable |

| Fires on after chart resize | Property: afterResize <br/><br/><EJ.Chart <br/> id= chartcontainer
afterResize = {afterResize}><br/></EJ.Chart><br/>function afterResize(){ }; | Not applicable |

| Fires on before chart resize | Property: beforeResize <br/><br/><EJ.Chart <br/> id= chartcontainer
beforeResize = {beforeResize}><br/></EJ.Chart><br/>function beforeResize(){ }; | Property: resized
<br/><br/><AccumulationChartComponent <br/> id='charts
resized={this.resized.bind(this)}><br/></AccumulationChartComponent><br/>public resized():
void {}; |

| Fires on chart click | Property: chartClick <br/><br/><EJ.Chart <br/> id= chartcontainer chartClick=
{chartClick}><br/></EJ.Chart><br/>function chartClick(){ }; | Property: chartMouseClicked
<br/><br/><AccumulationChartComponent <br/> id='charts
chartMouseClicked={this.chartMouseClicked.bind(this)}><br/></AccumulationChartComponent><br/>
>public chartMouseClicked(): void {}; |

| Fires on chart mouse move | Property: chartMouseMove <br/><br/><EJ.Chart <br/> id=
chartcontainer chartMouseMove= {chartMouseMove}><br/></EJ.Chart><br/>function
chartMouseMove(){ }; | Property: chartMouseMove <br/><br/><AccumulationChartComponent <br/>
id='charts
chartMouseMove={this.chartMouseMove.bind(this)}><br/></AccumulationChartComponent><b
r/>public chartMouseMove(): void {}; |

| Fires on chart mouse leave | Property: chartMouseLeave <br/><br/><EJ.Chart <br/> id=
chartcontainer chartMouseLeave= {chartMouseLeave}><br/></EJ.Chart><br/>function
chartMouseLeave(){ }; | Property: chartMouseLeave <br/><br/><AccumulationChartComponent <br/>
id='charts

```



```
chartMouseLeave={this.chartMouseLeave.bind(this)}><br/></AccumulationChartComponent><br/>public chartMouseLeave(): void {};
```

| Fires on before chart double click | **Property:** *chartDoubleClick*

<EJ.Chart
 id= chartcontainer chartDoubleClick= {chartDoubleClick}>
</EJ.Chart>
function chartDoubleClick(){ }; | Not applicable |

| Fires on chart mouse up | Not Applicable | **Property:** *chartmouseUp*

<AccumulationChartComponent
 id='charts chartmouseUp={this.chartmouseUp.bind(this)}>
</AccumulationChartComponent>
public chartmouseUp(): void {};

| Fires on chart mouse down | Not Applicable | **Property:** *chartmouseDown*

<AccumulationChartComponent
 id='charts chartmouseDown={this.chartmouseDown.bind(this)}>
</AccumulationChartComponent>
public chartmouseDown(): void {};

| Fires during the calculation of chart area bounds. You can use this event to customize the bounds of chart area | **Property:** *chartAreaBoundsCalculate*

<EJ.Chart
 id= chartcontainer chartAreaBoundsCalculate= {chartAreaBoundsCalculate}>
</EJ.Chart>
function chartAreaBoundsCalculate(){ }; | Not applicable |

| Fires when the dragging is started | **Property:** *dragStart*

<EJ.Chart
 id= chartcontainer dragStart= {dragStart}>
</EJ.Chart>
function dragStart(){ }; | Not applicable |

| Fires while dragging | **Property:** *dragging*

<EJ.Chart
 id= chartcontainer dragging= {dragging}>
</EJ.Chart>
function dragging(){ }; | Not applicable |

| Fires when the dragging is completed | **Property:** *dragEnd*

<EJ.Chart
 id= chartcontainer dragEnd= {dragEnd}>
</EJ.Chart>
function dragEnd(){ }; | **Property:** *dragComplete*

<AccumulationChartComponent
 id='charts dragComplete={this.dragComplete.bind(this)}>
</AccumulationChartComponent>
public dragComplete(): void {};

| Fires when chart is destroyed completely | **Property:** *destroy*

<EJ.Chart
 id= chartcontainer destroy= {destroy}>
</EJ.Chart>
function destroy(){ }; | Not applicable |

| Fires after chart is created. | **Property:** *create*

<EJ.Chart
 id= chartcontainer create= {create}>
</EJ.Chart>
function create(){ }; | **Property:** *loaded*

<AccumulationChartComponent
 id='charts loaded={this.loaded.bind(this)}>
</AccumulationChartComponent>
public loaded(): void {};

| Fires before rendering the data labels. | **Property:** *displayTextRendering*

<EJ.Chart
 id= chartcontainer displayTextRendering= {displayTextRendering}>
</EJ.Chart>
function displayTextRendering(){ }; | **Property:** *textRender*

<AccumulationChartComponent
 id='charts textRender={this.textRender.bind(this)}>
</AccumulationChartComponent>
public textRender(): void {};

| Fires, when error bar is rendering | **Property:** *errorBarRendering*

<EJ.Chart
 id= chartcontainer errorBarRendering= {errorBarRendering}>
</EJ.Chart>
function errorBarRendering(){ }; | Not applicable |

| Fires during the calculation of legend bounds | **Property:** *legendBoundsCalculate*

<EJ.Chart
 id= chartcontainer legendBoundsCalculate= {legendBoundsCalculate}>
</EJ.Chart>
function legendBoundsCalculate(){ }; | Not applicable |

| Fires on clicking the legend item. | **Property:** *legendItemClick*

<EJ.Chart
 id= chartcontainer legendItemClick= {legendItemClick}>
</EJ.Chart>
function legendItemClick(){ }; | Not applicable |

| Fires when moving mouse over legend item | **Property:** *legendItemMouseMove*

<EJ.Chart
 id= chartcontainer legendItemMouseMove= {legendItemMouseMove}>
</EJ.Chart>
function legendItemMouseMove(){ }; | Not applicable |

| Fires before rendering the legend item. | **Property:** *legendItemRendering*

<EJ.Chart
 id= chartcontainer legendItemRendering= {legendItemRendering}>
</EJ.Chart>
function legendItemRendering(){ }; | **Property:** *legendRender*

<AccumulationChartComponent
 id='charts' legendRender={this.legendRender.bind(this)}>
</AccumulationChartComponent>
public legendRender(): void {}; |

| Fires before loading the chart. | **Property:** *load*

<EJ.Chart
 id= chartcontainer load= {load}>
</EJ.Chart>
function load(){ }; | **Property:** *load*

<AccumulationChartComponent
 id='charts' load={this.load.bind(this)}>
</AccumulationChartComponent>
public load(): void {}; |

| Fires, when multi level labels are rendering. | **Property:** *multiLevelLabelRendering*

<EJ.Chart
 id= chartcontainer multiLevelLabelRendering= {multiLevelLabelRendering}>
</EJ.Chart>
function multiLevelLabelRendering(){ }; | **Property:** *axisMultiLabelRender*

<AccumulationChartComponent
 id='charts' axisMultiLabelRender={this.axisMultiLabelRender.bind(this)}>
</AccumulationChartComponent>
public axisMultiLabelRender(): void {}; |

| Fires on clicking a point in chart. | **Property:** *pointRegionClick*

<EJ.Chart
 id= 'chartcontainer' pointRegionClick= {pointRegionClick}>
</EJ.Chart>
function pointRegionClick(){ }; | **Property:** *pointClick*

<AccumulationChartComponent
 id='charts' pointClick={this.pointClick.bind(this)}>
</AccumulationChartComponent>
public pointClick(): void {}; |

| Fires when mouse is moved over a point. | **Property:** *pointRegionMouseMove*

<EJ.Chart
 id= 'chartcontainer' pointRegionMouseMove= {pointRegionMouseMove}>
</EJ.Chart>
function pointRegionMouseMove(){ }; | **Property:** *pointMove*

<AccumulationChartComponent
 id='charts'

```
pointMove={this.pointMove.bind(this)}><br/></AccumulationChartComponent><br/>public
pointMove(): void {};
```

| Fires before rendering chart. | **Property:** *preRender*

<EJ.Chart
 id= 'chartcontainer'
preRender= {preRender}>
</EJ.Chart>
function preRender(){ }; | Not applicable |

| Fires when point render. | Not Applicable | **Property:** *pointRender*

<AccumulationChartComponent
 id='charts'
pointRender={this.pointRender.bind(this)}>
</AccumulationChartComponent>
public
pointRender(): void {};

| Fires after selected the data in chart | **Property:** *rangeSelected*

<EJ.Chart
 id=
'chartcontainer' rangeSelected= {rangeSelected}>
</EJ.Chart>
function rangeSelected(){
}; | Not applicable |

| Fires after selecting a series. | **Property:** *seriesRegionClick*

<EJ.Chart
 id=
'chartcontainer' seriesRegionClick= {seriesRegionClick}>
</EJ.Chart>
function
seriesRegionClick(){ }; | Not applicable |

| Fires before rendering a series. | **Property:** *seriesRendering*

<EJ.Chart
 id=
'chartcontainer' seriesRendering= {seriesRendering}>
</EJ.Chart>
function
seriesRendering(){ }; | **Property:** *seriesRender*

<AccumulationChartComponent

id='charts'
seriesRender={this.seriesRender.bind(this)}>
</AccumulationChartComponent>
public
seriesRender(): void {};

| Fires before rendering the marker symbols. | **Property:** *symbolRendering*

<EJ.Chart

id= 'chartcontainer' symbolRendering= {symbolRendering}>
</EJ.Chart>
function
symbolRendering(){ }; | Not applicable |

| Fires before rendering the trendline | **Property:** *trendlineRendering*

<EJ.Chart
 id=
'chartcontainer' trendlineRendering= {trendlineRendering}>
</EJ.Chart>
function
trendlineRendering(){ }; | Not applicable |

| Fires before rendering the Chart title. | **Property:** *titleRendering*

<EJ.Chart
 id=
'chartcontainer' titleRendering= {titleRendering}>
</EJ.Chart>
function titleRendering(){
}; | Not applicable |

| Fires before rendering the Chart sub title. | **Property:** *subTitleRendering*

<EJ.Chart
 id=
'chartcontainer' subTitleRendering= {subTitleRendering}>
</EJ.Chart>
function
subTitleRendering(){ }; | Not applicable |

| Fires before rendering the tooltip. | **Property:** *tooltipInitialize*

<EJ.Chart
 id=
'chartcontainer' tooltipInitialize= {tooltipInitialize}>
</EJ.Chart>
function
tooltipInitialize(){ }; | **Property:** *tooltipRender*

<AccumulationChartComponent

id='charts'
tooltipRender={this.tooltipRender.bind(this)}>
</AccumulationChartComponent>
publi
c tooltipRender(): void {};

How To

Percentage tool tip in React Chart component

By using the [tooltipRender](#) event, you can show the percentage value for each point of pie series in tooltip.

To show the percentage value in pie tooltip, follow the given steps:

Step 1:

By using the [tooltipRender](#) event, you can get the `args.point.y` and `args.series.sumOfPoints` values. You can use these values to calculate the percentage value for each point of pie series. To display the percentage value in tooltip, use the `args.content` property.

The output will appear as follows,

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, Inject, AccumulationDataLabel,
AccumulationTooltip, PieSeries } from '@syncfusion/ej2-react-charts';
function App() {
  const data = [
    { 'x': 'Chrome', y: 37 }, { 'x': 'UC Browser', y: 17 },
    { 'x': 'iPhone', y: 19 }, { 'x': 'Others', y: 4, text: '4' },
    { 'x': 'Opera', y: 11 }
  ];
  const datalabel = { visible: true, position: 'Inside', name: 'text' };
  const tooltip = { enable: true };
  const tooltipRender = (args) => {
    let value = args.point.y / args.series.sumOfPoints * 100;
    args.text = args.point.x + ' ' + Math.ceil(value) + ' ' + '%';
  };
  return <AccumulationChartComponent id='charts' tooltip={tooltip}
title='Mobile Browser Statistics' tooltipRender={tooltipRender}>
  <Inject services={[AccumulationTooltip, PieSeries,
AccumulationDataLabel]}/>
  <AccumulationSeriesCollectionDirective>
    <AccumulationSeriesDirective dataSource={data} xName='x' yName='y'
radius='70%' dataLabel={datalabel}>
    </AccumulationSeriesDirective>
  </AccumulationSeriesCollectionDirective>
</AccumulationChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { AccumulationChartComponent, AccumulationSeriesCollectionDirective,
AccumulationSeriesDirective, IAccTooltipRenderEventArgs, Inject,
AccumulationDataLabel, AccumulationTooltip,
```

```

PieSeries, AccumulationDataLabelSettingsModel, TooltipSettingsModel } from
'@syncfusion/ej2-react-charts';
import{ EmitType } from '@syncfusion/ej2-base';
function App() {
  const data: any[] = [
    { 'x': 'Chrome', y: 37 }, { 'x': 'UC Browser', y: 17 },
    { 'x': 'iPhone', y: 19 }, { 'x': 'Others', y: 4, text: '4' },
    { 'x': 'Opera', y: 11 }
  ];
  const datalabel: AccumulationDataLabelSettingsModel = { visible: true,
position: 'Inside', name: 'text' };
  const tooltip: TooltipSettingsModel = { enable: true };
  const tooltipRender: EmitType<IAccTooltipRenderEventArgs> = (args:
IAccTooltipRenderEventArgs): void => {
    let value: any = args.point.y / args.series.sumOfPoints * 100;
    args.text = args.point.x + ' ' + Math.ceil(value) + ' ' + '%';
  };
  return <AccumulationChartComponent id='charts' tooltip={tooltip}
title='Mobile Browser Statistics'
  tooltipRender={tooltipRender}>
    <Inject services={[AccumulationTooltip, PieSeries,
AccumulationDataLabel]} />
    <AccumulationSeriesCollectionDirective>
      <AccumulationSeriesDirective dataSource={data} xName='x' yName='y'
radius='70%' dataLabel={datalabel}>
        </AccumulationSeriesDirective>
      </AccumulationSeriesCollectionDirective>
    </AccumulationChartComponent>
  </>
  export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

AppBar

Getting Started

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

Dependencies

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

```
`javascript
```

```
|-- @syncfusion/ej2-react-navigations
```

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

```
|-- @syncfusion/ej2-navigations
```

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

```
,
```

Setup your development environment

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](#) public registry.

To install **AppBar** component, use the following command

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

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

Adding Style sheet to the Application

Add AppBar component's styles as given below in `App.css`.

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

Add AppBar to the project

Now, you can create `AppBar` component in the application. Add `AppBar` component in `src/App.ts` file using the following code snippet.

```
`ts
import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
const App = () => {
  return (
    <div className='control-container'>
      <AppBarComponent colorMode="Primary">
        <ButtonComponent cssClass='e-inherit menu' iconCss='e-icons e-menu'></ButtonComponent>
        <span className="regular">React AppBar</span>
        <div className="e-appbar-spacer"></div>
        <ButtonComponent cssClass='e-inherit login'>FREE TRIAL</ButtonComponent>
      </AppBarComponent>
    </div>
  );
}
export default App;
`
```

Run the application

Run the application in the browser using the following command:

```
`
npm start
`
```

Size and color in React AppBar component

Size

The size of the AppBar can be set using the [mode](#) property. The available types of the AppBar are as follows:

- Regular AppBar
- Prominent AppBar
- Dense AppBar

Regular AppBar

This mode is the default one in which the AppBar is displayed with the default height.

INDEX.JSX

```
import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
    return (<div className='control-container'>
        <AppBarComponent colorMode="Primary">
            <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
menu"></ButtonComponent>
            <span className="regular">Regular AppBar</span>
            <div className="e-appbar-spacer"></div>
            <ButtonComponent cssClass="e-inherit">FREE TRIAL</ButtonComponent>
        </AppBarComponent>
    </div>);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);
```

INDEX.TSX

```
import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
    return (
        <div className='control-container'>
            <AppBarComponent colorMode="Primary">
                <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
menu"></ButtonComponent>
                <span className="regular">Regular AppBar</span>
                <div className="e-appbar-spacer"></div>
                <ButtonComponent cssClass="e-inherit">FREE TRIAL</ButtonComponent>
            </AppBarComponent>
        </div>
    );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);
```


Prominent AppBar

This height mode can be set to the AppBar by setting **Prominent** to the property `mode`. The prominent AppBar is displayed with a longer height and can be used for larger titles, images, or texts. It is also longer than the regular AppBar. In the following example, we have customized the prominent text using align-self and white-space CSS properties. You can change the prominent AppBar height if larger titles, images, or texts are used.

INDEX.JSX

```
import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
    return (<div className='control-container'>
        <AppBarComponent colorMode="Primary" mode="Prominent"
cssClass="prominent-appbar">
            <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
menu"></ButtonComponent>
            <span className="prominent">AppBar Component with Prominent
mode</span>
            <div className="e-appbar-spacer"></div>
            <ButtonComponent cssClass="e-inherit">FREE TRIAL</ButtonComponent>
        </AppBarComponent>
    </div>);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);
```

INDEX.TSX

```
import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
    return (
        <div className='control-container'>
            <AppBarComponent colorMode="Primary" mode="Prominent"
cssClass="prominent-appbar">
                <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
menu"></ButtonComponent>
                <span className="prominent">AppBar Component with Prominent
mode</span>
                <div className="e-appbar-spacer"></div>
                <ButtonComponent cssClass="e-inherit">FREE TRIAL</ButtonComponent>
            </AppBarComponent>
        </div>
    );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);
```

Dense AppBar

This height mode can be set to the AppBar by setting **Dense** to the property [mode](#). Dense AppBar is displayed with shorter height which is denser to accommodate all the AppBar content.

INDEX.JSX

```
import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
    return (<div className='control-container'>
        <AppBarComponent colorMode="Primary" mode="Dense">
            <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
menu"></ButtonComponent>
            <span className="dense">Dense AppBar</span>
            <div className="e-appbar-spacer"></div>
            <ButtonComponent cssClass="e-inherit">FREE TRIAL</ButtonComponent>
        </AppBarComponent>
    </div>);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);
```

INDEX.TSX

```
import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
    return (
        <div className='control-container'>
            <AppBarComponent colorMode="Primary" mode="Dense">
                <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
menu"></ButtonComponent>
                <span className="dense">Dense AppBar</span>
                <div className="e-appbar-spacer"></div>
                <ButtonComponent cssClass="e-inherit">FREE TRIAL</ButtonComponent>
            </AppBarComponent>
        </div>
    );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);
```

Color

The background and font colors can be set using the [colorMode](#) property. The available types of background color for the AppBar are as follows:

- Light AppBar
- Dark AppBar
- Primary AppBar
- Inherit AppBar

Light AppBar

This color mode is the default one in which the AppBar can be displayed with a light background and its corresponding font color.

INDEX.JSX

```
import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
    return (<div className='control-container'>
        <AppBarComponent>
            <a href="https://www.syncfusion.com/react-components"
target="_blank" rel="noopener" role="link" aria-label="Syncfusion react
components">
                <div className="syncfusion-logo"></div>
            </a>
            <div className="e-appbar-spacer"></div>
            <ButtonComponent isPrimary>FREE TRIAL</ButtonComponent>
        </AppBarComponent>
    </div>);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);
```

INDEX.TSX

```
import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
    return (
        <div className='control-container'>
            <AppBarComponent>
                <a href="https://www.syncfusion.com/react-components"
target="_blank" rel="noopener" role="link" aria-label="Syncfusion react
components">
                    <div className="syncfusion-logo"></div>
                </a>
                <div className="e-appbar-spacer"></div>
                <ButtonComponent isPrimary>FREE TRIAL</ButtonComponent>
            </AppBarComponent>
        </div>
    );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);
```

Dark AppBar

This color mode can be set to the AppBar by setting **Dark** to the property [colorMode](#). A dark AppBar can be displayed with a dark background and its corresponding font color.

INDEX.JSX

```
import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
    return (<div className='control-container'>
        <AppBarComponent colorMode="Dark">
            <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
menu"></ButtonComponent>
            <div className="e-appbar-spacer"></div>
            <ButtonComponent cssClass="e-inherit">FREE TRIAL</ButtonComponent>
        </AppBarComponent>
    </div>);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);
```

INDEX.TSX

```
import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
    return (
        <div className='control-container'>
            <AppBarComponent colorMode="Dark">
                <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
menu"></ButtonComponent>
                <div className="e-appbar-spacer"></div>
                <ButtonComponent cssClass="e-inherit">FREE TRIAL</ButtonComponent>
            </AppBarComponent>
        </div>
    );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);
```

Primary AppBar

This color mode can be set to the AppBar by setting **Primary** to the property [colorMode](#). The primary AppBar can be displayed with primary colors.

INDEX.JSX

```
import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
    return (<div className='control-container'>
        <AppBarComponent colorMode="Primary">
            <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
menu"></ButtonComponent>
            <div className="e-appbar-spacer"></div>
            <ButtonComponent cssClass="e-inherit">FREE TRIAL</ButtonComponent>
        </AppBarComponent>
    </div>);
}
```

```

        </AppBarComponent>
      </div>);
    }
    const root = ReactDOM.createRoot(document.getElementById('element'));
    root.render(<App />);

```

INDEX.TSX

```

import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
  return (
    <div className='control-container'>
      <AppBarComponent colorMode="Primary">
        <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-menu"></ButtonComponent>
      <div className="e-appbar-spacer"></div>
      <ButtonComponent cssClass="e-inherit">FREE TRIAL</ButtonComponent>
    </AppBarComponent>
    </div>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

Inherit AppBar

This color mode can be set to the AppBar by setting **Inherit** to the property [colorMode](#). The AppBar inherits the background and font color from its parent element.

INDEX.JSX

```

import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
  return (<div className='control-container'>
    <AppBarComponent colorMode="Inherit">
      <a href="https://www.syncfusion.com/react-components"
        target="_blank" rel="noopener" role="link" aria-label="Syncfusion react components">
        <div className="syncfusion-logo"></div>
      </a>
      <div className="e-appbar-spacer"></div>
      <ButtonComponent isPrimary>FREE TRIAL</ButtonComponent>
    </AppBarComponent>
    </div>);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

INDEX.TSX

```
import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
  return (
    <div className='control-container'>
      <AppBarComponent colorMode="Inherit">
        <a href="https://www.syncfusion.com/react-components"
target="_blank" rel="noopener" role="link" aria-label="Syncfusion react
components">
          <div className="syncfusion-logo"></div>
          </a>
          <div className="e-appbar-spacer"></div>
          <ButtonComponent isPrimary>FREE TRIAL</ButtonComponent>
        </AppBarComponent>
      </div>
    );
  }
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);
```

Accessibility in React AppBar component

The AppBar 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 AppBar 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

The AppBar component provides the focus element navigation based on its's tab key order.

Ensuring accessibility

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

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

See also

- [Accessibility in Syncfusion React components](#)

Position in React AppBar component

The position of the AppBar can be set using the [position](#) and [isSticky](#) property. The AppBar provides the following options for setting its position:

- Top AppBar
- Bottom AppBar
- Sticky AppBar

Top AppBar

The top AppBar is the default one in which it positions the AppBar at the top of the content.

INDEX.JSX

```
import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
```

```

    return (<div className='control-container'>
      <AppBarComponent colorMode="Primary">
        <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
menu"></ButtonComponent>
        <div className="e-appbar-spacer"></div>
        <ButtonComponent cssClass="e-inherit">FREE TRIAL</ButtonComponent>
      </AppBarComponent>
      <div className="appbar-content" style={{ fontSize: '12px' }}>
        <p>
          The AppBar also known as action bar or nav bar displays
          information and actions related to the current application screen. It is
          used to show branding, screen titles, navigation, and actions. The control
          supports height mode, color mode, positioning, and more.
        </p>
        <p>
          The AppBar control provides flexible ways to configure the look
          and feel of the bar to match your requirement.
        </p>
        <p>
          Developers can control the appearance and behaviors of the AppBar
          using a rich set of APIs.
        </p>
        <p>
          The AppBar component supports built-in themes such as Material,
          Bootstrap, Fabric (Office 365), Tailwind CSS, and high contrast. Users can
          customize these built-in themes or create new themes to achieve their
          desired look and feel by either simply overriding SASS variables or using
          our Theme Studio application.
        </p>
      </div>
    </div>);
  }
  const root = ReactDOM.createRoot(document.getElementById('element'));
  root.render(<App />);

```

INDEX.TSX

```

import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
  return (
    <div className='control-container'>
      <AppBarComponent colorMode="Primary">
        <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
menu"></ButtonComponent>
        <div className="e-appbar-spacer"></div>
        <ButtonComponent cssClass="e-inherit">FREE TRIAL</ButtonComponent>
      </AppBarComponent>
      <div className="appbar-content" style={{ fontSize: '12px' }}>
        <p>
          The AppBar also known as action bar or nav bar displays
          information and actions related to the current application screen. It is
          used to show branding, screen titles, navigation, and actions. The control
          supports height mode, color mode, positioning, and more.
        </p>

```



```

    </p>
    <p>
      The AppBar control provides flexible ways to configure the look
      and feel of the bar to match your requirement.
    </p>
    <p>
      Developers can control the appearance and behaviors of the AppBar
      using a rich set of APIs.
    </p>
    <p>
      The AppBar component supports built-in themes such as Material,
      Bootstrap, Fabric (Office 365), Tailwind CSS, and high contrast. Users can
      customize these built-in themes or create new themes to achieve their
      desired look and feel by either simply overriding SASS variables or using
      our Theme Studio application.
    </p>
  </div>
</div>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

Bottom AppBar

This position can be set to the AppBar by setting **Bottom** to the property [position](#). The bottom AppBar positions the AppBar at the bottom of the content.

INDEX.JSX

```

import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
  return (<div className='control-container'>
    <AppBarComponent colorMode="Primary" position="Bottom">
      <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
menu"></ButtonComponent>
      <div className="e-appbar-spacer"></div>
      <ButtonComponent cssClass="e-inherit">FREE TRIAL</ButtonComponent>
    </AppBarComponent>
    <div className="appbar-content" style={{ fontSize: '12px' }}>
      <p>
        The AppBar also known as action bar or nav bar displays
        information and actions related to the current application screen. It is
        used to show branding, screen titles, navigation, and actions. The control
        supports height mode, color mode, positioning, and more.
      </p>
      <p>
        The AppBar control provides flexible ways to configure the look
        and feel of the bar to match your requirement.
      </p>
      <p>
        Developers can control the appearance and behaviors of the AppBar
        using a rich set of APIs.
      </p>
    </div>
  </div>);
}

```

```

    <p>
      The AppBar component supports built-in themes such as Material,
      Bootstrap, Fabric (Office 365), Tailwind CSS, and high contrast. Users can
      customize these built-in themes or create new themes to achieve their
      desired look and feel by either simply overriding SASS variables or using
      our Theme Studio application.
    </p>
  </div>
</div>);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

INDEX.TSX

```

import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
  return (
    <div className='control-container'>
      <AppBarComponent colorMode="Primary" position="Bottom">
        <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
menu"></ButtonComponent>
        <div className="e-appbar-spacer"></div>
        <ButtonComponent cssClass="e-inherit">FREE TRIAL</ButtonComponent>
      </AppBarComponent>
      <div className="appbar-content" style={{ fontSize: '12px' }}>
        <p>
          The AppBar also known as action bar or nav bar displays
          information and actions related to the current application screen. It is
          used to show branding, screen titles, navigation, and actions. The control
          supports height mode, color mode, positioning, and more.
        </p>
        <p>
          The AppBar control provides flexible ways to configure the look
          and feel of the bar to match your requirement.
        </p>
        <p>
          Developers can control the appearance and behaviors of the AppBar
          using a rich set of APIs.
        </p>
        <p>
          The AppBar component supports built-in themes such as Material,
          Bootstrap, Fabric (Office 365), Tailwind CSS, and high contrast. Users can
          customize these built-in themes or create new themes to achieve their
          desired look and feel by either simply overriding SASS variables or using
          our Theme Studio application.
        </p>
      </div>
    </div>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

Sticky AppBar

This position can be set to the AppBar by setting `true` to the property `isSticky`. AppBar will be sticky while scrolling the AppBar content.

INDEX.JSX

```
import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
  return (<div className='control-container'>
    <AppBarComponent colorMode="Primary" isSticky>
      <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-menu"></ButtonComponent>
      <div className="e-appbar-spacer"></div>
      <ButtonComponent cssClass="e-inherit">FREE TRIAL</ButtonComponent>
    </AppBarComponent>
    <div className="appbar-content" style={{ fontSize: '12px' }}>
      <p>
        The AppBar also known as action bar or nav bar displays
        information and actions related to the current application screen. It is
        used to show branding, screen titles, navigation, and actions. The control
        supports height mode, color mode, positioning, and more.
      </p>
      <p>
        The AppBar control provides flexible ways to configure the look
        and feel of the bar to match your requirement.
      </p>
      <p>
        Developers can control the appearance and behaviors of the AppBar
        using a rich set of APIs.
      </p>
      <p>
        The AppBar component supports built-in themes such as Material,
        Bootstrap, Fabric (Office 365), Tailwind CSS, and high contrast. Users can
        customize these built-in themes or create new themes to achieve their
        desired look and feel by either simply overriding SASS variables or using
        our Theme Studio application.
      </p>
    </div>
  </div>);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);
```

INDEX.TSX

```
import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
  return (
```

```

<div className='control-container'>
  <AppBarComponent colorMode="Primary" isSticky>
    <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
menu"></ButtonComponent>
    <div className="e-appbar-spacer"></div>
    <ButtonComponent cssClass="e-inherit">FREE TRIAL</ButtonComponent>
  </AppBarComponent>
  <div className="appbar-content" style={{ fontSize: '12px' }}>
    <p>
      The AppBar also known as action bar or nav bar displays
      information and actions related to the current application screen. It is
      used to show branding, screen titles, navigation, and actions. The control
      supports height mode, color mode, positioning, and more.
    </p>
    <p>
      The AppBar control provides flexible ways to configure the look
      and feel of the bar to match your requirement.
    </p>
    <p>
      Developers can control the appearance and behaviors of the AppBar
      using a rich set of APIs.
    </p>
    <p>
      The AppBar component supports built-in themes such as Material,
      Bootstrap, Fabric (Office 365), Tailwind CSS, and high contrast. Users can
      customize these built-in themes or create new themes to achieve their
      desired look and feel by either simply overriding SASS variables or using
      our Theme Studio application.
    </p>
  </div>
</div>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

Design in React AppBar component

Spacer

Spacer is used to provide spacing between the AppBar contents which gives additional space to the content layout.

The following example depicts the code to provide spacing between the home and pan buttons in the AppBar:

INDEX.JSX

```

import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
  return (<div className='control-container'>
    <AppBarComponent colorMode="Primary">
      <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
home"></ButtonComponent>

```

```

        <div className="e-appbar-spacer"></div>
        <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
cut"></ButtonComponent>
        <div className="e-appbar-spacer"></div>
        <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
pan"></ButtonComponent>
      </AppBarComponent>
    </div>);
  }
  const root = ReactDOM.createRoot(document.getElementById('element'));
  root.render(<App />);

```

INDEX.TSX

```

import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
  return (
    <div className='control-container'>
      <AppBarComponent colorMode="Primary">
        <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
home"></ButtonComponent>
        <div className="e-appbar-spacer"></div>
        <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
cut"></ButtonComponent>
        <div className="e-appbar-spacer"></div>
        <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
pan"></ButtonComponent>
      </AppBarComponent>
    </div>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

Separator

Separator shows a vertical line to visually group or separate the AppBar contents.

The following example depicts the code to provide a vertical line between a group of buttons in the AppBar.

INDEX.JSX

```

import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
  return (<div className='control-container'>
    <AppBarComponent colorMode="Primary">
      <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
cut"></ButtonComponent>
      <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
copy"></ButtonComponent>
    </AppBarComponent>
  </div>);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

```

    <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
paste"></ButtonComponent>
    <div className="e-appbar-separator"></div>
    <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
bold"></ButtonComponent>
    <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
underline"></ButtonComponent>
    <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
italic"></ButtonComponent>
    <div className="e-appbar-separator"></div>
    <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-align-
left"></ButtonComponent>
    <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-align-
right"></ButtonComponent>
    <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-align-
center"></ButtonComponent>
    <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
justify"></ButtonComponent>
  </AppBarComponent>
</div>);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

INDEX.TSX

```

import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
  return (
    <div className='control-container'>
      <AppBarComponent colorMode="Primary">
        <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
cut"></ButtonComponent>
        <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
copy"></ButtonComponent>
        <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
paste"></ButtonComponent>
        <div className="e-appbar-separator"></div>
        <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
bold"></ButtonComponent>
        <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
underline"></ButtonComponent>
        <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
italic"></ButtonComponent>
        <div className="e-appbar-separator"></div>
        <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-align-
left"></ButtonComponent>
        <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-align-
right"></ButtonComponent>
        <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-align-
center"></ButtonComponent>
        <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
justify"></ButtonComponent>

```

```

    </AppBarComponent>
  </div>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

Media Query

Media Query is used to adjusting the AppBar for different screen sizes. Resize the screen to observe the responsive layout of the AppBar.

INDEX.JSX

```

import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
  return (
    <div className='control-container'>
      <AppBarComponent colorMode="Primary">
        <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-menu"></ButtonComponent>
        <ButtonComponent cssClass="e-inherit">Home</ButtonComponent>
        <ButtonComponent cssClass="e-inherit">About</ButtonComponent>
        <ButtonComponent cssClass="e-inherit">Products</ButtonComponent>
        <ButtonComponent cssClass="e-inherit">Contacts</ButtonComponent>
        <div className="e-appbar-spacer"></div>
        <div className="e-appbar-separator"></div>
        <ButtonComponent cssClass="e-inherit">Login</ButtonComponent>
      </AppBarComponent>
    </div>);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

INDEX.TSX

```

import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
  return (
    <div className='control-container'>
      <AppBarComponent colorMode="Primary">
        <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-menu"></ButtonComponent>
        <ButtonComponent cssClass="e-inherit">Home</ButtonComponent>
        <ButtonComponent cssClass="e-inherit">About</ButtonComponent>
        <ButtonComponent cssClass="e-inherit">Products</ButtonComponent>
        <ButtonComponent cssClass="e-inherit">Contacts</ButtonComponent>
        <div className="e-appbar-spacer"></div>
        <div className="e-appbar-separator"></div>
        <ButtonComponent cssClass="e-inherit">Login</ButtonComponent>
      </AppBarComponent>
    </div>);
}

```

```

    </div>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

Designing AppBar with Menu

AppBar is rendered with a Menu component in its AppBar header area. Menu component's styles are inherited from the AppBar component using the `e-inherit` CSS class.

INDEX.JSX

```

import { AppBarComponent, MenuComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
  const companyMenuItems = [
    {
      text: 'Company',
      items: [
        { text: 'About Us' },
        { text: 'Customers' },
        { text: 'Blog' },
        { text: 'Careers' }
      ]
    }
  ];
  const productMenuItems = [
    {
      text: 'Products',
      items: [
        { text: 'Developer' },
        { text: 'Analytics' },
        { text: 'Reporting' },
        { text: 'Help Desk' }
      ]
    }
  ];
  const aboutMenuItems = [
    {
      text: 'About Us'
    }
  ];
  const carrerMenuItems = [
    {
      text: 'Carrers'
    }
  ];
  return (<div className='control-container'>
    <AppBarComponent colorMode="Primary">
      <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-menu"></ButtonComponent>
      <MenuComponent cssClass="e-inherit"
items={companyMenuItems}></MenuComponent>

```



```

        <MenuComponent cssClass="e-inherit"
items={productMenuItems}></MenuComponent>
        <MenuComponent cssClass="e-inherit"
items={aboutMenuItems}></MenuComponent>
        <MenuComponent cssClass="e-inherit"
items={carrerMenuItems}></MenuComponent>
        <div className="e-appbar-spacer"></div>
        <ButtonComponent cssClass="e-inherit">Login</ButtonComponent>
    </AppBarComponent>
</div>);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

INDEX.TSX

```

import { AppBarComponent, MenuComponent, MenuItemModel } from
"@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
    const companyMenuItems: MenuItemModel[] = [
        {
            text: 'Company',
            items: [
                { text: 'About Us' },
                { text: 'Customers' },
                { text: 'Blog' },
                { text: 'Careers' }
            ]
        }
    ];
    const productMenuItems: MenuItemModel[] = [
        {
            text: 'Products',
            items: [
                { text: 'Developer' },
                { text: 'Analytics' },
                { text: 'Reporting' },
                { text: 'Help Desk' }
            ]
        }
    ];
    const aboutMenuItems: MenuItemModel[] = [
        {
            text: 'About Us'
        }
    ];
    const carrerMenuItems: MenuItemModel[] = [
        {
            text: 'Carrers'
        }
    ];
    return (
        <div className='control-container'>

```

```

        <AppBarComponent colorMode="Primary">
          <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
menu"></ButtonComponent>
          <MenuComponent cssClass="e-inherit"
items={companyMenuItems}></MenuComponent>
          <MenuComponent cssClass="e-inherit"
items={productMenuItems}></MenuComponent>
          <MenuComponent cssClass="e-inherit"
items={aboutMenuItems}></MenuComponent>
          <MenuComponent cssClass="e-inherit"
items={carrerMenuItems}></MenuComponent>
          <div className="e-appbar-spacer"></div>
          <ButtonComponent cssClass="e-inherit">Login</ButtonComponent>
        </AppBarComponent>
      </div>
    );
  }
  const root = ReactDOM.createRoot(document.getElementById('element'));
  root.render(<App />);

```

Designing AppBar with Buttons

The AppBar is rendered with a Button and DropDownButton component in its AppBar header area.

Button and DropDownButton components' styles are inherited from the AppBar component using the `e-inherit` CSS class.

INDEX.JSX

```

import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { DropDownButtonComponent } from '@syncfusion/ej2-react-splitbuttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
  const productDropDownButtonItem = [
    { text: 'Developer' },
    { text: 'Analytics' },
    { text: 'Reporting' },
    { text: 'E-Signature' },
    { text: 'Help Desk' }
  ];
  return (<div className='control-container'>
    <AppBarComponent colorMode="Primary">
      <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
menu"></ButtonComponent>
      <DropDownButtonComponent cssClass="e-inherit"
items={productDropDownButtonItem}>Products</DropDownButtonComponent>
      <div className="e-appbar-spacer"></div>
      <ButtonComponent cssClass="e-inherit">Login</ButtonComponent>
    </AppBarComponent>
  </div>);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

INDEX.TSX

```
import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { DropDownButtonComponent, ItemModel } from '@syncfusion/ej2-react-splitbuttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
  const productDropDownButtonItem: ItemModel[] = [
    { text: 'Developer' },
    { text: 'Analytics' },
    { text: 'Reporting' },
    { text: 'E-Signature' },
    { text: 'Help Desk' }
  ];
  return (
    <div className='control-container'>
      <AppBarComponent colorMode="Primary">
        <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-menu"></ButtonComponent>
        <DropDownButtonComponent cssClass="e-inherit"
items={productDropDownButtonItem}>Products</DropDownButtonComponent>
        <div className="e-appbar-spacer"></div>
        <ButtonComponent cssClass="e-inherit">Login</ButtonComponent>
      </AppBarComponent>
    </div>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);
```

Designing AppBar with SideBar

The AppBar is rendered with the SideBar component below the AppBar. Click on the menu icon to expand/collapse the Sidebar. In the following sample, the `toggle` method has been used to show or hide the Sidebar on the AppBar button click.

INDEX.JSX

```
import * as React from "react";
import { useState, useRef } from "react";
import * as ReactDOM from "react-dom";
import { AppBarComponent, TreeViewComponent, SidebarComponent } from
"@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
const App = () => {
  const [isOpen, setIsOpen] = useState < boolean > (true);
  const sidebarRef = useRef < SidebarComponent > (null);
  const data = [
    {
      nodeId: '01', nodeText: 'Installation',
    },
    {
      nodeId: '02', nodeText: 'Deployment',
    },
  ],
```

```

    {
      nodeId: '03', nodeText: 'Quick Start',
    },
    {
      nodeId: '04', nodeText: 'Components',
      nodeChild: [
        { nodeId: '04-01', nodeText: 'Calendar' },
        { nodeId: '04-02', nodeText: 'DatePicker' },
        { nodeId: '04-03', nodeText: 'DateTimePicker' },
        { nodeId: '04-04', nodeText: 'DateRangePicker' },
        { nodeId: '04-05', nodeText: 'TimePicker' },
        { nodeId: '04-06', nodeText: 'SideBar' }
      ]
    }
  ];
  const width = '220px';
  const target = '.main-sidebar-content';
  const mediaQuery = '(min-width: 600px)';
  const fields = { dataSource: data, id: 'nodeId', text: 'nodeText', child:
'nodeChild' };
  const toggleSidebar = () => {
    setIsOpen(!isOpen);
  }
  return (
    <div className="control-container" id="responsive-wrapper">
      <div id="reswrapper">
        <div>
          <AppBarComponent>
            <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-menu"
onClick={toggleSidebar}></ButtonComponent>
            <div className="e-folder">
              <div className="e-folder-name">Navigation Pane</div>
            </div>
          </AppBarComponent>
        </div>
        <SidebarComponent id="sideTree" className="sidebar-treeview"
ref={sidebarRef} width={width} target={target} mediaQuery={mediaQuery}
isOpen={isOpen}>
          <div className="res-main-menu">
            <div className="table-content">
              <TextBoxComponent id="resSearch"
placeholder="Search..."></TextBoxComponent>
              <p className="main-menu-header">TABLE OF CONTENTS</p>
            </div>
            <div>
              <TreeViewComponent id='mainTree' cssClass="main-treeview"
fields={fields} expandOn='Click' />
            </div>
          </div>
        </SidebarComponent>
        <div className="main-sidebar-content" id="main-text">
          <div className="sidebar-content">
            <div className="sidebar-heading"> Responsive Sidebar with
Treeview</div>
            <p className="paragraph-content">
              This is a graphical aid for visualising and categorising the
site, in the style of an expandable and collapsable treeview component.

```

It auto-expands to display the node(s), **if** any, corresponding to the currently viewed title, highlighting that node(s) and its ancestors. Load-on-demand when expanding nodes **is** available **where** supported (most graphical browsers), falling back to a full-page reload. MediaWiki-supported caching, aside **from** squid, has been considered so that unnecessary re-downloads of content are avoided **where** possible. The complete expanded/collapsed state of the treeview persists across page views **in** most situations.

```

    </p>
  </div>
</div>
</div>
</div>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

INDEX.TSX

```

import * as React from "react";
import { useState, useRef } from "react";
import * as ReactDOM from "react-dom";
import { AppBarComponent, TreeViewComponent, SidebarComponent } from
"@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { TextBoxComponent } from '@syncfusion/ej2-react-inputs';
const App = () => {
  const [isOpen, setIsOpen] = useState<boolean>(true);
  const sidebarRef = useRef<SidebarComponent>(null);
  const data: { [key: string]: Object }[] = [
    {
      nodeId: '01', nodeText: 'Installation',
    },
    {
      nodeId: '02', nodeText: 'Deployment',
    },
    {
      nodeId: '03', nodeText: 'Quick Start',
    },
    {
      nodeId: '04', nodeText: 'Components',
      nodeChild: [
        { nodeId: '04-01', nodeText: 'Calendar' },
        { nodeId: '04-02', nodeText: 'DatePicker' },
        { nodeId: '04-03', nodeText: 'DateTimePicker' },
        { nodeId: '04-04', nodeText: 'DateRangePicker' },
        { nodeId: '04-05', nodeText: 'TimePicker' },
        { nodeId: '04-06', nodeText: 'SideBar' }
      ]
    }
  ];
  const width: string = '220px';
  const target: string = '.main-sidebar-content';
  const mediaQuery: string = '(min-width: 600px)';

```

```

const fields: object = { dataSource: data, id: 'nodeId', text: 'nodeText',
child: 'nodeChild' };
const toggleSidebar = () => {
  setIsOpen(!isOpen);
}
return (
  <div className="control-container" id="responsive-wrapper">
    <div id="reswrapper">
      <div>
        <AppBarComponent>
          <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-menu"
onClick={toggleSidebar}></ButtonComponent>
          <div className="e-folder">
            <div className="e-folder-name">Navigation Pane</div>
          </div>
        </AppBarComponent>
      </div>
      <SidebarComponent id="sideTree" className="sidebar-treeview"
ref={sidebarRef} width={width} target={target} mediaQuery={mediaQuery}
isOpen={isOpen}>
        <div className='res-main-menu'>
          <div className="table-content">
            <TextBoxComponent id="resSearch"
placeholder="Search..."></TextBoxComponent>
            <p className="main-menu-header">TABLE OF CONTENTS</p>
          </div>
          <div>
            <TreeViewComponent id='mainTree' cssClass="main-treeview"
fields={fields} expandOn='Click' />
          </div>
        </div>
      </SidebarComponent>
      <div className="main-sidebar-content" id="main-text">
        <div className="sidebar-content">
          <div className="sidebar-heading"> Responsive Sidebar with
Treeview</div>
          <p className="paragraph-content">
            This is a graphical aid for visualising and categorising the
site, in the style of an expandable and collapsable treeview component.
            It auto-expands to display the node(s), if any, corresponding
to the currently viewed title, highlighting that node(s)
            and its ancestors. Load-on-demand when expanding nodes is
available where supported (most graphical browsers),
            falling back to a full-page reload. MediaWiki-supported
caching, aside from squid, has been considered so that
            unnecessary re-downloads of content are avoided where
possible. The complete expanded/collapsed state of
            the treeview persists across page views in most situations.
          </p>
        </div>
      </div>
    </div>
  </div>
);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

INDEX.CSS

```
.e-appbar .e-folder {
    margin:0 5px;
}
#responsive-wrapper.control-section {
    padding: 0;
}
#responsive-wrapper .sidebar-treeview {
    z-index: 20 !important;
}
#responsive-wrapper .main-sidebar-content {
    height: 380px;
}
.sidebar-treeview .res-main-menu .main-menu-header {
    color: #656a70;
    padding: 15px 15px 15px 0;
    font-size: 14px;
    width: 13em;
    margin: 0;
}
#main-text .sidebar-heading {
    font-size: 16px;
}
.sidebar-treeview .table-content {
    padding: 20px 14px;
    height: 8em;
}
#main-text .sidebar-content .line {
    width: 100%;
    height: 1px;
    border-bottom: 1px dashed #ddd;
    margin: 40px 0;
}
#main-text .sidebar-content {
    padding: 15px;
    font-size: 14px;
}
#main-text .paragraph-content {
    padding: 15px 0;
    font-weight: normal;
    font-size: 14px;
}
.e-folder {
    text-align: center;
    font-weight: 500;
    font-size: 16px
}
.e-folder-name {
    margin-top: 1px;
}
.sidebar-treeview .e-treeview .e-icon-collapsible,
.sidebar-treeview .e-treeview .e-icon-expandable {
    float: right;
    margin: 3px;
```

```

}
.sidebar-treeview .e-treeview,
.sidebar-treeview .e-treeview .e-ul {
  padding: 0;
  margin: 0;
}
.sidebar-treeview .e-treeview .e-text-content {
  padding-left: 14px;
}
#responsive-wrapper .e-appbar {
  border-bottom: 1px solid #eaeaea0;
}
.sidebar-treeview {
  border-right: 1px solid #eaeaea0;
}

```

Style and appearance in React AppBar component

To modify the AppBar appearance, you need to override the default CSS of the AppBar component. Please find the list of CSS classes and their corresponding sections in the AppBar component. Also, you have an option to create your own custom theme for the controls using our [Theme Studio](#).

| CSS Class | Purpose of Class |

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

| .e-appbar | To customize the appbar. |

| .e-appbar.e-prominent | To customize the prominent appbar. |

| .e-appbar.e-dense | To customize the dense appbar. |

| .e-appbar.e-light | To customize the light appbar. |

| .e-appbar.e-dark | To customize the dark appbar. |

| .e-appbar.e-primary | To customize the dark appbar. |

| .e-appbar.e-inherit | To customize the inherit appbar. |

Note: You can change the prominent AppBar height if larger titles, images, or texts are used.

CssClass

CssClass is used for AppBar customization based on the custom class. In the example below, the AppBar background and color are customized using the [cssClass](#) property.

INDEX.JSX

```

import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
  return (<div className='control-container'>
    <AppBarComponent colorMode="Primary" cssClass="custom-appbar">
      <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
home"></ButtonComponent>
    </AppBarComponent>
  </div>);

```



```

}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

INDEX.TSX

```

import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
  return (
    <div className='control-container'>
      <AppBarComponent colorMode="Primary" cssClass="custom-appbar">
        <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-home"></ButtonComponent>
      </AppBarComponent>
    </div>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

HtmlAttributes

It can be used for additional inline attributes by specifying as inline attributes or by specifying htmlAttributes directive. In the code example below, the aria-label of the AppBar is customized by specifying as attributes.

INDEX.JSX

```

import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
  return (<div className='control-container'>
    <AppBarComponent colorMode="Primary" aria-label="appbar">
      <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-home"></ButtonComponent>
    </AppBarComponent>
  </div>);
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

INDEX.TSX

```

import { AppBarComponent } from "@syncfusion/ej2-react-navigations";
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
const App = () => {
  return (
    <div className='control-container'>
      <AppBarComponent colorMode="Primary" aria-label="appbar">

```

```

        <ButtonComponent cssClass="e-inherit" iconCss="e-icons e-
home"></ButtonComponent>
      </AppBarComponent>
    </div>
  );
}
const root = ReactDOM.createRoot(document.getElementById('element'));
root.render(<App />);

```

AutoComplete

Getting Started

This section explains how to create a simple [Link to the Video](#) component and configure its available functionalities in React.

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

Dependencies

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

```

`javascript
|-- @syncfusion/ej2-react-dropdowns
|-- @syncfusion/ej2-dropdowns
|-- @syncfusion/ej2-base
|-- @syncfusion/ej2-data
|-- @syncfusion/ej2-lists
|-- @syncfusion/ej2-inputs
|-- @syncfusion/ej2-navigations
|-- @syncfusion/ej2-popups
|-- @syncfusion/ej2-buttons
|-- @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

```

`bash

```

```
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](https://www.npmjs.com) public registry. You can choose the component that you want to install.

To install AutoComplete component, use the following command

```
`bash
```

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

```
,
```

Adding AutoComplete component

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

Add the below code in the `src/App.tsx` to initialize the AutoComplete.

[Class-component]

```
`ts
```

```
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
```

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

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

```
export default class App extends React.Component<{}, {}> {
```

```
  public render() {
```

```
    return (
```

```
      // specifies the tag for render the AutoComplete component
```

```
      <AutoCompleteComponent id='atcelement'></AutoCompleteComponent>
```

```
    );
```

```
  }
```

```
}
```

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

```
,
```

[Functional-component]

```
`ts
```

```
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
```

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

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

```
function App() {
  return (
    // specifies the tag for render the AutoComplete component
    <AutoCompleteComponent id='atcelement'></AutoCompleteComponent>
  );
}

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

Adding CSS reference

Import the AutoComplete component required CSS references as follows in `src/App.css`.

```
`css

/ import the AutoComplete dependency styles /

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

Binding data source

After initialization, populate the data using [dataSource](#) property. Here, an array of string values is passed to the AutoComplete component.

[Class-component]

```
`ts

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

export default class App extends React.Component<{}, {}> {
  // define the array of data
  private sportsData: string[] = ['Badminton', 'Basketball', 'Cricket', 'Football', 'Golf', 'Gymnastics', 'Hockey',
  'Rugby', 'Snooker', 'Tennis'];

  public render() {
    return (
      // specifies the tag for render the AutoComplete component
      <AutoCompleteComponent id="atcelement" dataSource={this.sportsData} />
    );
  }
}
```

```

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

```

[Functional-component]

```

`ts
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
  // define the array of data
  const sportsData: string[] = ['Badminton', 'Basketball', 'Cricket', 'Football', 'Golf', 'Gymnastics', 'Hockey',
  'Rugby', 'Snooker', 'Tennis'];
  return (
    // specifies the tag for render the AutoComplete component
    <AutoCompleteComponent id="atcelement" dataSource={sportsData} />
  );
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

Run the application

After completing the configuration required to render a basic AutoComplete, run the following command to display the output in your default browser.

```

npm start

```

[Class-component]

INDEX.JSX

```

import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
  // define the array of data
  sportsData = ['Badminton', 'Basketball', 'Cricket', 'Football', 'Golf',
  'Gymnastics', 'Hockey', 'Rugby', 'Snooker', 'Tennis'];
  render() {
    return (
      // specifies the tag for render the AutoComplete component
      <AutoCompleteComponent id="atcelement" dataSource={this.sportsData}
      placeholder="Find a game"/>);
  }
}

```

```

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

```

INDEX.TSX

```

import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}, {}> {
  // define the array of data
  private sportsData: string[] = ['Badminton', 'Basketball', 'Cricket',
  'Football', 'Golf', 'Gymnastics', 'Hockey', 'Rugby', 'Snooker', 'Tennis'];
  public render() {
    return (
      // specifies the tag for render the AutoComplete component
      <AutoCompleteComponent id="atcelement" dataSource={this.sportsData}
placeholder="Find a game" />
    );
  }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

[Functional-component]**INDEX.JSX**

```

import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
  // define the array of data
  const sportsData = ['Badminton', 'Basketball', 'Cricket', 'Football',
  'Golf', 'Gymnastics', 'Hockey', 'Rugby', 'Snooker', 'Tennis'];
  return (
    // specifies the tag for render the AutoComplete component
    <AutoCompleteComponent id="atcelement" dataSource={sportsData}
placeholder="Find a game"/>
  );
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.TSX

```

import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
  // define the array of data
  const sportsData: string[] = ['Badminton', 'Basketball', 'Cricket',
  'Football', 'Golf', 'Gymnastics', 'Hockey', 'Rugby', 'Snooker', 'Tennis'];
  return (
    // specifies the tag for render the AutoComplete component
    <AutoCompleteComponent id="atcelement" dataSource={sportsData}
placeholder="Find a game" />
  );
}

```

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

Configure the Popup List

By default, the width of the popup list automatically adjusts according to the AutoComplete input element's width and the height of the popup list has '300px'.

You can also customize the suggestion list height and width using [popupHeight](#) and [popupWidth](#) property respectively.

In the following sample, popup list's width and height are configured.

[Class-component]

INDEX.JSX

```
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
  // define the array of data
  sportsData = ['Badminton', 'Basketball', 'Cricket', 'Football', 'Golf',
  'Gymnastics', 'Hockey', 'Rugby', 'Snooker', 'Tennis'];
  render() {
    return (
      // specifies the tag for render the AutoComplete component
      <AutoCompleteComponent id="atcelement" dataSource={this.sportsData}
      popupHeight="250px" popupWidth="250px" placeholder="Find a game"/>);
  }
}
ReactDOM.render(<App />, document.getElementById('sample'));
```

INDEX.TSX

```
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}, {}> {
  // define the array of data
  private sportsData: string[] = ['Badminton', 'Basketball', 'Cricket',
  'Football', 'Golf', 'Gymnastics', 'Hockey', 'Rugby', 'Snooker', 'Tennis'];
  public render() {
    return (
      // specifies the tag for render the AutoComplete component
      <AutoCompleteComponent id="atcelement"
      dataSource={this.sportsData} popupHeight="250px" popupWidth="250px"
      placeholder="Find a game" />
    );
  }
}
ReactDOM.render(<App />, document.getElementById('sample'));
```

[Functional-component]

INDEX.JSX

```
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    // define the array of data
    const sportsData = ['Badminton', 'Basketball', 'Cricket', 'Football',
    'Golf', 'Gymnastics', 'Hockey', 'Rugby', 'Snooker', 'Tennis'];
    return (
        // specifies the tag for render the AutoComplete component
        <AutoCompleteComponent id="atcelement" dataSource={sportsData}
        popupHeight="250px" popupWidth="250px" placeholder="Find a game"/>);
    }
    ReactDOM.render(<App />, document.getElementById('sample'));
```

INDEX.TSX

```
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    // define the array of data
    const sportsData: string[] = ['Badminton', 'Basketball', 'Cricket',
    'Football', 'Golf', 'Gymnastics', 'Hockey', 'Rugby', 'Snooker', 'Tennis'];
    return (
        // specifies the tag for render the AutoComplete component
        <AutoCompleteComponent id="atcelement" dataSource={sportsData}
        popupHeight="250px" popupWidth="250px" placeholder="Find a game" />
        );
    }
    ReactDOM.render(<App />, document.getElementById('sample'));
```

See Also

- [How to bind the data](#)

Data binding in React Auto complete component

The AutoComplete loads the data either from local data sources or remote data services using the [dataSource](#) property .

It supports the data type of array or **DataManager**.

The AutoComplete also supports different kinds of data services such as OData, OData V4, and Web API, and data formats such as XML, JSON, and JSONP with the help of DataManager Adaptors.

Fields	Type	Description
value	number or string	Specifies the hidden data value mapped to each list item that should contain a unique value.
groupBy	string	Specifies the category under which the list item has to be grouped.
iconCss	string	Specifies the icon class of each list item.

When binding complex data to the AutoComplete, fields should be mapped correctly. Otherwise, the selected item remains undefined.

Binding local data

Local data can be represented in two ways as described below.

Array of string

The AutoComplete has support to load array of primitive data such as strings and numbers.

[Class-component]

INDEX.JSX

```
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
    // define the array of string
    sportsData = ['Badminton', 'Basketball', 'Cricket', 'Football', 'Golf',
    'Hockey', 'Snooker', 'Tennis'];
    render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="atcelement" dataSource={this.sportsData}
placeholder="Find a game"/>);
        }
    }
ReactDOM.render(<App />, document.getElementById('sample'));
```

INDEX.TSX

```
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}, {}> {
    // define the array of string
    private sportsData: string[] = ['Badminton', 'Basketball', 'Cricket',
    'Football', 'Golf', 'Hockey', 'Snooker', 'Tennis'];
    public render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="atcelement"
dataSource={this.sportsData} placeholder="Find a game" />
        );
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));
```

[Functional-component]

INDEX.JSX

```
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
```

```
// define the array of string
const sportsData = ['Badminton', 'Basketball', 'Cricket', 'Football',
'Golf', 'Hockey', 'Snooker', 'Tennis'];
return (
  // specifies the tag for render the AutoComplete component
  <AutoCompleteComponent id="atcelement" dataSource={sportsData}
placeholder="Find a game"/>);
}
ReactDOM.render(<App />, document.getElementById('sample'));
```

INDEX.TSX

```
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
  // define the array of string
  const sportsData: string[] = ['Badminton', 'Basketball', 'Cricket',
'Football', 'Golf', 'Hockey', 'Snooker', 'Tennis'];
  return (
    // specifies the tag for render the AutoComplete component
    <AutoCompleteComponent id="atcelement" dataSource={sportsData}
placeholder="Find a game" />
  );
}
ReactDOM.render(<App />, document.getElementById('sample'));
```

Array of object

The AutoComplete can generate its list items through an array of complex data. For this, the appropriate columns should be mapped to the [fields](#) property.

In the following example, **sports** column from complex data have been mapped to the **value** field.

[Class-component]

INDEX.JSX

```
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
  // define the JSON of data
  sportsData = [
    { id: 'Game1', game: 'Badminton' },
    { id: 'Game2', game: 'Basketball' },
    { id: 'Game3', game: 'Cricket' },
    { id: 'Game4', game: 'Football' },
    { id: 'Game5', game: 'Golf' },
    { id: 'Game6', game: 'Hockey' },
    { id: 'Game7', game: 'Rugby' },
    { id: 'Game8', game: 'Snooker' }
  ];
  // maps the appropriate column to fields property
  fields = { value: 'game' };
  render() {
    return (
```

```

        // specifies the tag for render the AutoComplete component
        <AutoCompleteComponent id="atcelement" dataSource={this.sportsData}
fields={this.fields} placeholder="Find a game"/>);
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.TSX

```

import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}, {}> {
    // define the JSON of data
    private sportsData: { [key: string]: Object }[] = [
        { id: 'Game1', game: 'Badminton' },
        { id: 'Game2', game: 'Basketball' },
        { id: 'Game3', game: 'Cricket' },
        { id: 'Game4', game: 'Football' },
        { id: 'Game5', game: 'Golf' },
        { id: 'Game6', game: 'Hockey' },
        { id: 'Game7', game: 'Rugby' },
        { id: 'Game8', game: 'Snooker' }
    ];
    // maps the appropriate column to fields property
    private fields: object = { value: 'game' };
    public render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="atcelement"
dataSource={this.sportsData} fields={this.fields} placeholder="Find a game"
/>
        );
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

[Functional-component]

INDEX.JSX

```

import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    // define the JSON of data
    const sportsData = [
        { id: 'Game1', game: 'Badminton' },
        { id: 'Game2', game: 'Basketball' },
        { id: 'Game3', game: 'Cricket' },
        { id: 'Game4', game: 'Football' },
        { id: 'Game5', game: 'Golf' },
        { id: 'Game6', game: 'Hockey' },
        { id: 'Game7', game: 'Rugby' },
        { id: 'Game8', game: 'Snooker' }
    ];
}

```

```
// maps the appropriate column to fields property
const fields = { value: 'game' };
return (
  // specifies the tag for render the AutoComplete component
  <AutoCompleteComponent id="atcelement" dataSource={sportsData}
  fields={fields} placeholder="Find a game"/>);
}
ReactDOM.render(<App />, document.getElementById('sample'));
```

INDEX.TSX

```
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
  // define the JSON of data
  const sportsData: { [key: string]: Object }[] = [
    { id: 'Game1', game: 'Badminton' },
    { id: 'Game2', game: 'Basketball' },
    { id: 'Game3', game: 'Cricket' },
    { id: 'Game4', game: 'Football' },
    { id: 'Game5', game: 'Golf' },
    { id: 'Game6', game: 'Hockey' },
    { id: 'Game7', game: 'Rugby' },
    { id: 'Game8', game: 'Snooker' }
  ];
  // maps the appropriate column to fields property
  const fields: object = { value: 'game' };
  return (
    // specifies the tag for render the AutoComplete component
    <AutoCompleteComponent id="atcelement" dataSource={sportsData}
    fields={fields} placeholder="Find a game" />
  );
}
ReactDOM.render(<App />, document.getElementById('sample'));
```

Array of Complex object

The AutoComplete can generate its list items through an array of complex data. For this, the appropriate columns should be mapped to the [fields](#) property.

In the following example, **Country.Name** column from complex data have been mapped to the **value** field.

[Class-component]

INDEX.JSX

```
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
  // define the JSON of data
  countriesData = [
    { Country: { Name: 'Australia' }, Code: { Id: 'AU' } },
    { Country: { Name: 'Bermuda' }, Code: { Id: 'BM' } },
    { Country: { Name: 'Canada' }, Code: { Id: 'CA' } },
  ];
}
```

```

    { Country: { Name: 'Cameroon' }, Code: { Id: 'CM' } },
    { Country: { Name: 'Denmark' }, Code: { Id: 'DK' } },
    { Country: { Name: 'France' }, Code: { Id: 'FR' } },
    { Country: { Name: 'Finland' }, Code: { Id: 'FI' } },
    { Country: { Name: 'Germany' }, Code: { Id: 'DE' } },
    { Country: { Name: 'Greenland' }, Code: { Id: 'GL' } },
    { Country: { Name: 'Hong Kong' }, Code: { Id: 'HK' } },
    { Country: { Name: 'India' }, Code: { Id: 'IN' } },
    { Country: { Name: 'Italy' }, Code: { Id: 'IT' } },
    { Country: { Name: 'Japan' }, Code: { Id: 'JP' } },
    { Country: { Name: 'Mexico' }, Code: { Id: 'MX' } },
    { Country: { Name: 'Norway' }, Code: { Id: 'NO' } },
    { Country: { Name: 'Poland' }, Code: { Id: 'PL' } },
    { Country: { Name: 'Switzerland' }, Code: { Id: 'CH' } },
    { Country: { Name: 'United Kingdom' }, Code: { Id: 'GB' } },
    { Country: { Name: 'United States' }, Code: { Id: 'US' } }
  ];
  // maps the appropriate column to fields property
  fields = { value: 'Country.Name' };
  render() {
    return (
      // specifies the tag for render the AutoComplete component
      <AutoCompleteComponent id="atcelement"
dataSource={this.countriesData} fields={this.fields} placeholder="Find a
country"/>);
    }
  }
ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.TSX

```

{% raw %}
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}, {}> {
  // define the JSON of data
  private countriesData: { [key: string]: Object }[] = [
    { Country: { Name: 'Australia' }, Code: { Id: 'AU' } },
    { Country: { Name: 'Bermuda' }, Code: { Id: 'BM' } },
    { Country: { Name: 'Canada' }, Code: { Id: 'CA' } },
    { Country: { Name: 'Cameroon' }, Code: { Id: 'CM' } },
    { Country: { Name: 'Denmark' }, Code: { Id: 'DK' } },
    { Country: { Name: 'France' }, Code: { Id: 'FR' } },
    { Country: { Name: 'Finland' }, Code: { Id: 'FI' } },
    { Country: { Name: 'Germany' }, Code: { Id: 'DE' } },
    { Country: { Name: 'Greenland' }, Code: { Id: 'GL' } },
    { Country: { Name: 'Hong Kong' }, Code: { Id: 'HK' } },
    { Country: { Name: 'India' }, Code: { Id: 'IN' } },
    { Country: { Name: 'Italy' }, Code: { Id: 'IT' } },
    { Country: { Name: 'Japan' }, Code: { Id: 'JP' } },
    { Country: { Name: 'Mexico' }, Code: { Id: 'MX' } },
    { Country: { Name: 'Norway' }, Code: { Id: 'NO' } },
    { Country: { Name: 'Poland' }, Code: { Id: 'PL' } },
    { Country: { Name: 'Switzerland' }, Code: { Id: 'CH' } },
    { Country: { Name: 'United Kingdom' }, Code: { Id: 'GB' } },

```

```

    { Country:{Name: 'United States'}, Code: { Id: 'US' } }
  ];
  // maps the appropriate column to fields property
  private fields: object = { value: 'Country.Name' };
  public render() {
    return (
      // specifies the tag for render the AutoComplete component
      <AutoCompleteComponent id="atcelement"
dataSource={this.countriesData} fields={this.fields} placeholder="Find a
country" />
    );
  }
}
ReactDOM.render(<App />, document.getElementById('sample'));
{% endraw %}

```

[Functional-component]

INDEX.JSX

```

import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
  // define the JSON of data
  const countriesData = [
    { Country: { Name: 'Australia' }, Code: { Id: 'AU' } },
    { Country: { Name: 'Bermuda' }, Code: { Id: 'BM' } },
    { Country: { Name: 'Canada' }, Code: { Id: 'CA' } },
    { Country: { Name: 'Cameroon' }, Code: { Id: 'CM' } },
    { Country: { Name: 'Denmark' }, Code: { Id: 'DK' } },
    { Country: { Name: 'France' }, Code: { Id: 'FR' } },
    { Country: { Name: 'Finland' }, Code: { Id: 'FI' } },
    { Country: { Name: 'Germany' }, Code: { Id: 'DE' } },
    { Country: { Name: 'Greenland' }, Code: { Id: 'GL' } },
    { Country: { Name: 'Hong Kong' }, Code: { Id: 'HK' } },
    { Country: { Name: 'India' }, Code: { Id: 'IN' } },
    { Country: { Name: 'Italy' }, Code: { Id: 'IT' } },
    { Country: { Name: 'Japan' }, Code: { Id: 'JP' } },
    { Country: { Name: 'Mexico' }, Code: { Id: 'MX' } },
    { Country: { Name: 'Norway' }, Code: { Id: 'NO' } },
    { Country: { Name: 'Poland' }, Code: { Id: 'PL' } },
    { Country: { Name: 'Switzerland' }, Code: { Id: 'CH' } },
    { Country: { Name: 'United Kingdom' }, Code: { Id: 'GB' } },
    { Country: { Name: 'United States' }, Code: { Id: 'US' } }
  ];
  // maps the appropriate column to fields property
  const fields = { value: 'Country.Name' };
  return (
    // specifies the tag for render the AutoComplete component
    <AutoCompleteComponent id="atcelement" dataSource={countriesData}
fields={fields} placeholder="Find a country"/>
  );
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.TSX

```
{% raw %}
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    // define the JSON of data
    const countriesData: { [key: string]: Object }[] = [
        { Country:{Name: 'Australia' }, Code: { Id: 'AU' } },
        { Country:{Name: 'Bermuda' }, Code: { Id: 'BM' } },
        { Country:{Name: 'Canada' }, Code: { Id: 'CA' } },
        { Country:{Name: 'Cameroon' }, Code: { Id: 'CM' } },
        { Country:{Name: 'Denmark' }, Code: { Id: 'DK' } },
        { Country:{Name: 'France' }, Code: { Id: 'FR' } },
        { Country:{Name: 'Finland' }, Code: { Id: 'FI' } },
        { Country:{Name: 'Germany' }, Code: { Id: 'DE' } },
        { Country:{Name: 'Greenland' }, Code: { Id: 'GL' } },
        { Country:{Name: 'Hong Kong' }, Code: { Id: 'HK' } },
        { Country:{Name: 'India' }, Code: { Id: 'IN' } },
        { Country:{Name: 'Italy' }, Code: { Id: 'IT' } },
        { Country:{Name: 'Japan' }, Code: { Id: 'JP' } },
        { Country:{Name: 'Mexico' }, Code: { Id: 'MX' } },
        { Country:{Name: 'Norway' }, Code: { Id: 'NO' } },
        { Country:{Name: 'Poland' }, Code: { Id: 'PL' } },
        { Country:{Name: 'Switzerland' }, Code: { Id: 'CH' } },
        { Country:{Name: 'United Kingdom' }, Code: { Id: 'GB' } },
        { Country:{Name: 'United States' }, Code: { Id: 'US' } }
    ];
    // maps the appropriate column to fields property
    const fields: object = { value: 'Country.Name' };
    return (
        // specifies the tag for render the AutoComplete component
        <AutoCompleteComponent id="atcelement" dataSource={countriesData}
        fields={fields} placeholder="Find a country" />
    );
}
ReactDOM.render(<App />, document.getElementById('sample'));
{% endraw %}
```

Binding remote data

The AutoComplete supports retrieval of data from remote data services with the help of **DataManager** component. The [Query](#) property is used to fetch data from the database and bind it to the AutoComplete.

The below sample displays the first 6 contacts from the **Customers** table of the **Northwind** data service.

[Class-component]**INDEX.JSX**

```
// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
```

```

import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
    // bind the DataManager instance to dataSource property
    customerData = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url: 'https://services.odata.org/V4/Northwind/Northwind.svc/'
    });
    // bind the Query instance to query property
    query = new Query().from('Customers').select(['ContactName',
'CustomerID']).take(6);
    // maps the appropriate column to fields property
    fields = { value: 'ContactName' };
    // sort the resulted items
    sortOrder = 'Ascending';
    render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="atcelement" query={this.query}
dataSource={this.customerData} fields={this.fields}
sortOrder={this.sortOrder} placeholder="Find a customer"/>);
        }
    }
ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.TSX

```

// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { SortOrder } from '@syncfusion/ej2-lists';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}, {}> {
    // bind the DataManager instance to dataSource property
    private customerData: DataManager = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url: 'https://services.odata.org/V4/Northwind/Northwind.svc/'
    });
    // bind the Query instance to query property
    private query: Query = new
Query().from('Customers').select(['ContactName', 'CustomerID']).take(6);
    // maps the appropriate column to fields property
    private fields: object = { value: 'ContactName' };
    // sort the resulted items
    private sortOrder: SortOrder = 'Ascending';
    public render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="atcelement" query={this.query}
dataSource={this.customerData}
fields={this.fields} sortOrder={this.sortOrder}
placeholder="Find a customer"/>
        );
    }
}

```



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

[Functional-component]

INDEX.JSX

```
// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    // bind the DataManager instance to dataSource property
    const customerData = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url: 'https://services.odata.org/V4/Northwind/Northwind.svc/'
    });
    // bind the Query instance to query property
    const query = new Query().from('Customers').select(['ContactName',
    'CustomerID']).take(6);
    // maps the appropriate column to fields property
    const fields = { value: 'ContactName' };
    // sort the resulted items
    const sortOrder = 'Ascending';
    return (
        // specifies the tag for render the AutoComplete component
        <AutoCompleteComponent id="atcelement" query={query}
        dataSource={customerData} fields={fields} sortOrder={sortOrder}
        placeholder="Find a customer"/>);
    }
ReactDOM.render(<App />, document.getElementById('sample'));
```

INDEX.TSX

```
// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { SortOrder } from '@syncfusion/ej2-lists';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    // bind the DataManager instance to dataSource property
    const customerData: DataManager = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url: 'https://services.odata.org/V4/Northwind/Northwind.svc/'
    });
    // bind the Query instance to query property
    const query: Query = new
    Query().from('Customers').select(['ContactName', 'CustomerID']).take(6);
    // maps the appropriate column to fields property
    const fields: object = { value: 'ContactName' };
    // sort the resulted items
    const sortOrder: SortOrder = 'Ascending';
```

```

    return (
      // specifies the tag for render the AutoComplete component
      <AutoCompleteComponent id="atcelement" query={query}
      dataSource={customerData}
        fields={fields} sortOrder={sortOrder} placeholder="Find a
customer"/>
    );
  }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

See Also

- [How to load data using template](#)
- [How to group the data using header](#)
- [How to filter the bound data](#)

Value binding in AutoComplete Component

Value binding in the AutoComplete control allows you to associate data values with each list item. This facilitates managing and retrieving selected values efficiently. The AutoComplete component provides flexibility in binding both primitive data types and complex objects.

Primitive Data Types

The AutoComplete control provides flexible binding capabilities for primitive data types like strings and numbers. You can effortlessly bind local primitive data arrays, fetch and bind data from remote sources, and even custom data binding to suit specific requirements. Bind the value of primitive data to the [value](#) property of the AutoComplete.

Primitive data types include:

- String
- Number
- Boolean
- Null

The following sample shows the example for preselect values for primitive data type

[Class-component]

INDEX.JSX

```

import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
  constructor(props) {
    super(props);
    this.records = ["Item 1", "Item 2", "Item 3", "Item 4", "Item 5",
"Item 6", "Item 7", "Item 8", "Item 9", "Item 10"];
  }
  value = "Item 5";
  fields = { value: 'text' };
  render() {
    return (

```

```

        // specifies the tag for render the AutoComplete component
        <AutoCompleteComponent id="datas" dataSource={this.records}
value={this.value} placeholder="e.g. Item 1" allowFiltering={true}
popupHeight="200px">
        </AutoCompleteComponent>);
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.TSX

```

import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}, {}> {
    // maps the appropriate column to fields property
    private fields: object = { value: 'text' };
    // define the array of string
    private records: string[] = [];
    private value: string = "Item 1";
    constructor(props) {
        super(props);
        this.records = ["Item 1", "Item 2", "Item 3", "Item 4", "Item 5", "Item
6", "Item 7", "Item 8", "Item 9", "Item 10"]
    }
    public render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="datas" dataSource={this.records}
placeholder="e.g. Item 1" allowFiltering={true} popupHeight="200px">
            </AutoCompleteComponent>
        );
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

Object Data Types

In the AutoComplete control, object binding allows you to bind to a dataset of objects. When [allowObjectBinding](#) is enabled, the value of the control will be an object of the same type as the selected item in the [value](#) property. This feature seamlessly binds arrays of objects, whether sourced locally, retrieved from remote endpoints, or customized to suit specific application needs.

The following sample shows the example for preselect values for object data type

[Link to the Video](#)

INDEX.JSX

```

import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
    constructor(props) {
        super(props);
        this.records = Array.from({ length: 150 }, (_, i) => ({

```

```

        id: 'id' + (i + 1),
        text: `Item ${i + 1}`,
    }));
    }
    value = { text: 'Item 5', value: 'id5' };
    fields = { value: 'text' };
    render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="datas" dataSource={this.records}
value={this.value} placeholder="e.g. Item 1" allowObjectBinding={true}
fields={this.fields} allowFiltering={true} popupHeight="200px">
            </AutoCompleteComponent>);
        )
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.TSX

```

import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}, {}> {
    // maps the appropriate column to fields property
    private fields: object = { value: 'text' };
    // define the array of string
    private records: { [key: string]: Object }[] = [];
    private value: { [key: string]: Object } = { id: 'id1', text: 'Item 1' };
    constructor(props) {
        super(props);
        this.records = Array.from({ length: 150 }, (_, i) => ({
            id: 'id' + (i + 1),
            text: `Item ${i + 1}`,
        }));
    }
    public render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="datas" dataSource={this.records}
placeholder="e.g. Item 1" allowObjectBinding={true} fields={this.fields}
allowFiltering={true} popupHeight="200px">
            </AutoCompleteComponent>
        );
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

Templates in React Auto complete component

The AutoComplete has been provided with several options to customize each list items, group title, header and footer elements.

To get started with React AutoComplete templates, you can check on this video:

Item template

The content of each list item within the AutoComplete can be customized with the help of [itemTemplate](#) property.

In the below sample, each list item is split into two columns to showcase relevant data's.

[Class-component]

INDEX.JSX

```
// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
    // bind the DataManager instance to dataSource property
    employeeData = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url: 'https://services.odata.org/V4/Northwind/Northwind.svc/'
    });
    // bind the Query instance to query property
    query = new Query().from('Employees').select(['FirstName', 'City', 'EmployeeID']).take(6);
    // maps the appropriate column to fields property
    fields = { value: 'FirstName' };
    // sort the resulted items
    sortOrder = 'Ascending';
    render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="atcelement" query={this.query}
            itemTemplate={this.itemTemplate} dataSource={this.employeeData}
            fields={this.fields} sortOrder={this.sortOrder} placeholder="Find an
            employee"/>);
        }
        // set the value to itemTemplate property
        itemTemplate(data) {
            return (<span><span className='name'>{data.FirstName}</span><span
            className='city'>{data.City}</span></span>);
        }
    }
    ReactDOM.render(<App />, document.getElementById('sample'));
```

INDEX.TSX

```
// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}, {}> {
    // bind the DataManager instance to dataSource property
    private employeeData: DataManager = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
```

```

        url: 'https://services.odata.org/V4/Northwind/Northwind.svc/'
    });
    // bind the Query instance to query property
    private query: Query = new
    Query().from('Employees').select(['FirstName', 'City',
    'EmployeeID']).take(6);
    // maps the appropriate column to fields property
    private fields: object = { value: 'FirstName' };
    // sort the resulted items
    private sortOrder: string = 'Ascending';
    public render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="atcelement" query={this.query}
            itemTemplate={this.itemTemplate} dataSource={this.employeeData}
            fields={this.fields} sortOrder={this.sortOrder} placeholder="Find an
            employee"/>
        );
    }
    // set the value to itemTemplate property
    private itemTemplate(data: any): JSX.Element {
        return (
            <span><span className='name'>{data.FirstName}</span><span className
            ='city'>{data.City}</span></span>
        );
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

[Functional-component]

INDEX.JSX

```

// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    // bind the DataManager instance to dataSource property
    const employeeData = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url: 'https://services.odata.org/V4/Northwind/Northwind.svc/'
    });
    // bind the Query instance to query property
    const query = new Query().from('Employees').select(['FirstName', 'City',
    'EmployeeID']).take(6);
    // maps the appropriate column to fields property
    const fields = { value: 'FirstName' };
    // sort the resulted items
    const sortOrder = 'Ascending';
    return (
        // specifies the tag for render the AutoComplete component

```

```

    <AutoCompleteComponent id="atcelement" query={query}
    itemTemplate={itemTemplate} dataSource={employeeData} fields={fields}
    sortOrder={sortOrder} placeholder="Find an employee"/>);
    // set the value to itemTemplate property
    function itemTemplate(data) {
        return (<span><span className='name'>{data.FirstName}</span><span
        className='city'>{data.City}</span></span>);
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.TSX

```

// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    // bind the DataManager instance to dataSource property
    const employeeData: DataManager = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url: 'https://services.odata.org/V4/Northwind/Northwind.svc/'
    });
    // bind the Query instance to query property
    const query: Query = new Query().from('Employees').select(['FirstName',
    'City', 'EmployeeID']).take(6);
    // maps the appropriate column to fields property
    const fields: object = { value: 'FirstName' };
    // sort the resulted items
    const sortOrder: string = 'Ascending';
    return (
        // specifies the tag for render the AutoComplete component
        <AutoCompleteComponent id="atcelement" query={query}
        itemTemplate={itemTemplate} dataSource={employeeData} fields={fields}
        sortOrder={sortOrder} placeholder="Find an employee"/>
    );
    // set the value to itemTemplate property
    function itemTemplate(data: any): JSX.Element {
        return (
            <span><span className='name'>{data.FirstName}</span><span className
            ='city'>{data.City}</span></span>
        );
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

Group template

The group header title under which appropriate sub-items are categorized can also be customize with the help of [groupTemplate](#) property. This template is common for both inline and floating group header template.

In the below sample, employees are grouped according to their city.

[Class-component]**INDEX.JSX**

```
// import DataManager related classes
import { DataManager, ODataV4Adaptor, Predicate, Query } from
 '@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
  // bind the DataManager instance to dataSource property
  employeeData = new DataManager({
    adaptor: new ODataV4Adaptor,
    crossDomain: true,
    url: 'https://services.odata.org/V4/Northwind/Northwind.svc/'
  });
  // form predicate to fetch the grouped data
  groupPredicate = new Predicate('City', 'equal', 'london').or('City',
  'equal', 'seattle');
  // bind the Query instance to query property
  query = new Query().from('Employees').select(['FirstName', 'City',
  'EmployeeID']).take(6).where(this.groupPredicate);
  // maps the appropriate column to fields property
  fields = { value: 'FirstName', groupBy: 'City' };
  // sort the resulted items
  sortOrder = 'Ascending';
  render() {
    return (
      // specifies the tag for render the AutoComplete component
      <AutoCompleteComponent id="atcelement" query={this.query}
groupTemplate={this.groupTemplate} dataSource={this.employeeData}
sortOrder={this.sortOrder} fields={this.fields} placeholder="Find an
employee"/>);
    }
    // set the value to groupTemplate
    groupTemplate(data) {
      return <strong>{data.City}</strong>;
    }
  }
  ReactDOM.render(<App />, document.getElementById('sample'));
```

INDEX.TSX

```
// import DataManager related classes
import { DataManager, ODataV4Adaptor, Predicate, Query } from
 '@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}, {}> {
  // bind the DataManager instance to dataSource property
  private employeeData: DataManager = new DataManager({
    adaptor: new ODataV4Adaptor,
    crossDomain: true,
    url: 'https://services.odata.org/V4/Northwind/Northwind.svc/'
  });
}
```



```

    // form predicate to fetch the grouped data
    private groupPredicate = new Predicate('City', 'equal',
'london').or('City', 'equal', 'seattle');
    // bind the Query instance to query property
    private query: Query = new Query().from('Employees').select(['FirstName',
'City', 'EmployeeID']).take(6).where(this.groupPredicate);
    // maps the appropriate column to fields property
    private fields: object = { value: 'FirstName', groupBy: 'City' };
    // sort the resulted items
    private sortOrder: string = 'Ascending';
    public render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="atcelement" query={this.query}
groupTemplate={this.groupTemplate} dataSource={this.employeeData}
sortOrder={this.sortOrder} fields={this.fields} placeholder="Find an
employee" />
        );
    }
    // set the value to groupTemplate
    private groupTemplate(data: any): JSX.Element {
        return (
            <strong>{data.City}</strong>
        );
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

[Functional-component]

INDEX.JSX

```

// import DataManager related classes
import { DataManager, ODataV4Adaptor, Predicate, Query } from
'@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    // bind the DataManager instance to dataSource property
    const employeeData = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url: 'https://services.odata.org/V4/Northwind/Northwind.svc/'
    });
    // form predicate to fetch the grouped data
    const groupPredicate = new Predicate('City', 'equal',
'london').or('City', 'equal', 'seattle');
    // bind the Query instance to query property
    const query = new Query().from('Employees').select(['FirstName', 'City',
'EmployeeID']).take(6).where(groupPredicate);
    // maps the appropriate column to fields property
    const fields = { value: 'FirstName', groupBy: 'City' };
    // sort the resulted items
    const sortOrder = 'Ascending';
    return (

```

```

    // specifies the tag for render the AutoComplete component
    <AutoCompleteComponent id="atcelement" query={query}
groupTemplate={groupTemplate} dataSource={employeeData}
sortOrder={sortOrder} fields={fields} placeholder="Find an employee"/>;
    // set the value to groupTemplate
    function groupTemplate(data) {
        return <strong>{data.City}</strong>;
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.TSX

```

// import DataManager related classes
import { DataManager, ODataV4Adaptor, Predicate, Query } from
 '@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    // bind the DataManager instance to dataSource property
    const employeeData: DataManager = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url: 'https://services.odata.org/V4/Northwind/Northwind.svc/'
    });
    // form predicate to fetch the grouped data
    const groupPredicate = new Predicate('City', 'equal', 'london').or('City',
    'equal', 'seattle');
    // bind the Query instance to query property
    const query: Query = new Query().from('Employees').select(['FirstName',
    'City', 'EmployeeID']).take(6).where(groupPredicate);
    // maps the appropriate column to fields property
    const fields: object = { value: 'FirstName', groupBy: 'City' };
    // sort the resulted items
    const sortOrder: string = 'Ascending';
    return (
        // specifies the tag for render the AutoComplete component
        <AutoCompleteComponent id="atcelement" query={query}
groupTemplate={groupTemplate} dataSource={employeeData}
sortOrder={sortOrder} fields={fields} placeholder="Find an employee" />
    );
    // set the value to groupTemplate
    function groupTemplate(data: any): JSX.Element {
        return (
            <strong>{data.City}</strong>
        );
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

Header template

The header element is shown statically at the top of the suggestion list items within the AutoComplete and any custom element can be placed as a header element using the [headerTemplate](#) property.

In the following sample, the list items and its headers are designed and displayed as two columns similar to multiple columns of the grid.

[Class-component]

INDEX.JSX

```
// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
    // bind the DataManager instance to dataSource property
    employeeData = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url: 'https://services.odata.org/V4/Northwind/Northwind.svc/',
    });
    // bind the Query instance to query property
    query = new Query().from('Employees').select(['FirstName', 'City', 'EmployeeID']).take(6);
    // maps the appropriate column to fields property
    fields = { value: 'FirstName' };
    // sort the resulted items
    sortOrder = 'Ascending';
    render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="atcelement" query={this.query}
            headerTemplate={this.headerTemplate} dataSource={this.employeeData}
            sortOrder={this.sortOrder} itemTemplate={this.itemTemplate}
            fields={this.fields} placeholder="Find an employee"/>);
        // set the value to header template
        headerTemplate(data) {
            return (<span className='head'><span
            className='name'>Name</span><span className='city'>City</span></span>);
        }
        // set the value to item template
        itemTemplate(data) {
            return (<span className='item'><span
            className='name'>{data.FirstName}</span><span
            className='city'>{data.City}</span></span>);
        }
    }
    ReactDOM.render(<App />, document.getElementById('sample'));
```

INDEX.TSX

```
// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}, {}> {
    // bind the DataManager instance to dataSource property
```

```

private employeeData: DataManager = new DataManager({
    adaptor: new ODataV4Adaptor,
    crossDomain: true,
    url: 'https://services.odata.org/V4/Northwind/Northwind.svc/',
});
// bind the Query instance to query property
private query: Query = new Query().from('Employees').select(['FirstName',
'City', 'EmployeeID']).take(6);
// maps the appropriate column to fields property
private fields: object = { value: 'FirstName' };
// sort the resulted items
private sortOrder: string = 'Ascending';
public render() {
    return (
        // specifies the tag for render the AutoComplete component
        <AutoCompleteComponent id="atcelement" query={this.query}
headerTemplate={this.headerTemplate} dataSource={this.employeeData}
sortOrder={this.sortOrder} itemTemplate={this.itemTemplate}
fields={this.fields} placeholder="Find an employee" />
    );
}
// set the value to header template
private headerTemplate(data: any): JSX.Element {
    return (
        <span className='head'><span className='name'>Name</span><span
className='city'>City</span></span>
    );
}
// set the value to item template
private itemTemplate(data: any): JSX.Element {
    return (
        <span className='item' ><span
className='name'>{data.FirstName}</span><span
className='city'>{data.City}</span></span>
    );
}
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

[Functional-component]

INDEX.JSX

```

// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    // bind the DataManager instance to dataSource property
    const employeeData = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url: 'https://services.odata.org/V4/Northwind/Northwind.svc/',
    });
    // bind the Query instance to query property

```

```

    const query = new Query().from('Employees').select(['FirstName', 'City',
'EmployeeID']).take(6);
    // maps the appropriate column to fields property
    const fields = { value: 'FirstName' };
    // sort the resulted items
    const sortOrder = 'Ascending';
    return (
    // specifies the tag for render the AutoComplete component
    <AutoCompleteComponent id="atcelement" query={query}
headerTemplate={headerTemplate} dataSource={employeeData}
sortOrder={sortOrder} itemTemplate={itemTemplate} fields={fields}
placeholder="Find an employee"/>);
    // set the value to header template
    function headerTemplate(data) {
        return (<span className='head'><span
className='name'>Name</span><span className='city'>City</span></span>);
    }
    // set the value to item template
    function itemTemplate(data) {
        return (<span className='item'><span
className='name'>{data.FirstName}</span><span
className='city'>{data.City}</span></span>);
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.TSX

```

// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    // bind the DataManager instance to dataSource property
    const employeeData: DataManager = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url: 'https://services.odata.org/V4/Northwind/Northwind.svc/',
    });
    // bind the Query instance to query property
    const query: Query = new Query().from('Employees').select(['FirstName',
'City', 'EmployeeID']).take(6);
    // maps the appropriate column to fields property
    const fields: object = { value: 'FirstName' };
    // sort the resulted items
    const sortOrder: string = 'Ascending';
    return (
    // specifies the tag for render the AutoComplete component
    <AutoCompleteComponent id="atcelement" query={query}
headerTemplate={headerTemplate} dataSource={employeeData}
sortOrder={sortOrder} itemTemplate={itemTemplate} fields={fields}
placeholder="Find an employee" />
    );
    // set the value to header template
    function headerTemplate(data: any): JSX.Element {

```

```

    return (
      <span className='head'><span className='name'>Name</span><span
className='city'>City</span></span>
    );
  }
  // set the value to item template
  function itemTemplate(data: any): JSX.Element {
    return (
      <span className='item' ><span
className='name'>{data.FirstName}</span><span
className='city'>{data.City}</span></span>
    );
  }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

Footer template

The AutoComplete has options to show a footer element at the bottom of the list items in the suggestion list. Here, you can place any custom element as a footer element using the [footerTemplate](#) property.

In the following sample, footer element displays the total number of list items present in the AutoComplete.

[Class-component]

INDEX.JSX

```

{% raw %}
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
  // define the array of data
  atcObject;
  sportsData = ['Badminton', 'Basketball', 'Cricket', 'Football', 'Golf',
'Gymnastics', 'Hockey', 'Rugby', 'Snooker', 'Tennis'];
  onOpen = (e) => {
    const count = this.atcObject.getItems().length;
    // set the value to footerTemplate property
    const ele = document.getElementsByClassName('foot')[0];
    ele.innerHTML = "Total list item: " + count;
  };
  render() {
    return (
      // specifies the tag for render the AutoComplete component
      <AutoCompleteComponent id="atcelement" ref={(autocomplete) => {
this.atcObject = autocomplete; }} footerTemplate={this.footerTemplate}
dataSource={this.sportsData} placeholder="Find a game" open={this.onOpen =
this.onOpen.bind(this)} />);
    }
    // set the footer template
    footerTemplate() {
      return (<span className='foot' />);
    }
  }
}

```

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

INDEX.TSX

```
{% raw %}
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}>, {}> {
    // define the array of data
    private atcObject: any;
    private sportsData: string[] = ['Badminton', 'Basketball', 'Cricket',
'Football', 'Golf', 'Gymnastics', 'Hockey', 'Rugby', 'Snooker', 'Tennis'];
    public onOpen = (e: any) => {
        const count = this.atcObject.getItems().length;
        // set the value to footerTemplate property
        const ele = document.getElementsByClassName('foot')[0];
        ele.innerHTML = "Total list item: " + count;
    };
    public render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="atcelement" ref={(autocomplete) => {
this.atcObject = autocomplete }} footerTemplate={this.footerTemplate}
dataSource={this.sportsData} placeholder="Find a game" open={ this.onOpen
=this.onOpen.bind(this) }/>
        );
    }
    // set the footer template
    private footerTemplate(): JSX.Element {
        return (
            <span className='foot' />
        );
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));
{% endraw %}
```

[Functional-component]

INDEX.JSX

```
{% raw %}
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    // define the array of data
    let atcObject;
    const sportsData = ['Badminton', 'Basketball', 'Cricket', 'Football',
'Golf', 'Gymnastics', 'Hockey', 'Rugby', 'Snooker', 'Tennis'];
    function onOpen(e) {
        const count = this.atcObject.getItems().length;
        // set the value to footerTemplate property
        const ele = document.getElementsByClassName('foot')[0];
    }
}
```

```

        ele.innerHTML = "Total list item: " + count;
    }
    ;
    return (
        // specifies the tag for render the AutoComplete component
        <AutoCompleteComponent id="atcelement" ref={ (autocomplete) => {
atcObject = autocomplete; }} footerTemplate={footerTemplate}
dataSource={sportsData} placeholder="Find a game" open={onOpen =
onOpen.bind(this) }/>);
        // set the footer template
        function footerTemplate() {
            return (<span className='foot' />);
        }
    )
    ReactDOM.render(<App />, document.getElementById('sample'));
    {% endraw %}

```

INDEX.TSX

```

{% raw %}
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    // define the array of data
    let atcObject: any;
    const sportsData: string[] = ['Badminton', 'Basketball', 'Cricket',
'Football', 'Golf', 'Gymnastics', 'Hockey', 'Rugby', 'Snooker', 'Tennis'];
    function onOpen(e: any){
        const count = this.atcObject.getItems().length;
        // set the value to footerTemplate property
        const ele = document.getElementsByClassName('foot')[0];
        ele.innerHTML = "Total list item: " + count;
    };
    return (
        // specifies the tag for render the AutoComplete component
        <AutoCompleteComponent id="atcelement" ref={ (autocomplete) =>
{atcObject = autocomplete }} footerTemplate={footerTemplate}
dataSource={sportsData} placeholder="Find a game" open={onOpen
=onOpen.bind(this) }/>
    );
    // set the footer template
    function footerTemplate(): JSX.Element {
        return (
            <span className='foot' />
        );
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));
{% endraw %}

```

No Records template

The AutoComplete is provided with support to custom design the suggestion list content when no data is found and no matches found on search with the help of

[noRecordsTemplate](#) property.

In the following sample, popup list content displays the notification of no data available.

[Class-component]

INDEX.JSX

```
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
    // define the array of data
    data = [];
    render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="atcelement"
noRecordsTemplate={this.noRecordsTemplate} dataSource={this.data} placeholder="Find an item"/>);
        }
        // set the value to noRecords template
        noRecordsTemplate() {
            return (<span className='norecord'> NO DATA AVAILABLE</span>);
        }
    }
    ReactDOM.render(<App />, document.getElementById('sample'));
```

INDEX.TSX

```
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}, {}> {
    // define the array of data
    private data: { [key: string]: Object }[] = [];
    public render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="atcelement"
noRecordsTemplate={this.noRecordsTemplate} dataSource={this.data} placeholder="Find an item"/>);
        }
        // set the value to noRecords template
        private noRecordsTemplate(): JSX.Element {
            return (
                <span className='norecord'> NO DATA AVAILABLE</span>
            );
        }
    }
    ReactDOM.render(<App />, document.getElementById('sample'));
```

[Functional-component]

INDEX.JSX

```
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    // define the array of data
    const data = [];
    return (
        // specifies the tag for render the AutoComplete component
        <AutoCompleteComponent id="atcelement"
noRecordsTemplate={noRecordsTemplate = noRecordsTemplate.bind(this)}
dataSource={data} placeholder="Find an item"/>);
    // set the value to noRecords template
    function noRecordsTemplate() {
        return (<span className='norecord'> NO DATA AVAILABLE</span>);
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));
```

INDEX.TSX

```
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    // define the array of data
    const data: { [key: string]: Object }[] = [];
    return (
        // specifies the tag for render the AutoComplete component
        <AutoCompleteComponent id="atcelement"
noRecordsTemplate={noRecordsTemplate = noRecordsTemplate.bind(this)}
dataSource={data} placeholder="Find an item"/>
    );
    // set the value to noRecords template
    function noRecordsTemplate(): JSX.Element {
        return (
            <span className='norecord'> NO DATA AVAILABLE</span>
        );
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));
```

Action Failure template

There is also an option to custom design the suggestion list content, when the data fetch request fails at the remote server. This can be achieved using the [actionFailureTemplate](#) property.

In the following sample, when the data fetch request fails, the AutoComplete displays the notification.

[Class-component]**INDEX.JSX**

```
// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
```

```

import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
    // bind the DataManager instance to dataSource property
    customerData = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url: 'https://services.odata.org/V4/Northwind/Northwind.svcs/'
    });
    // bind the Query instance to query property
    query = new Query().from('Customers').select(['ContactName',
'CustomerID']).take(6);
    // maps the appropriate column to fields property
    fields = { value: 'ContactName' };
    render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="atcelement" query={this.query}
actionFailureTemplate={this.template = this.template.bind(this)}
dataSource={this.customerData} fields={this.fields} placeholder="Find a
customer"/>);
        // set the value to action failure template
        template() {
            return (<span className='action-failure'> Data fetch get
fails</span>);
        }
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.TSX

```

// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}, {}> {
    // bind the DataManager instance to dataSource property
    private customerData: DataManager = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url: 'https://services.odata.org/V4/Northwind/Northwind.svcs/'
    });
    // bind the Query instance to query property
    private query: Query = new
Query().from('Customers').select(['ContactName', 'CustomerID']).take(6);
    // maps the appropriate column to fields property
    private fields: object = { value: 'ContactName' };
    public render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="atcelement" query={this.query}
actionFailureTemplate={this.template = this.template.bind(this)}
dataSource={this.customerData} fields={this.fields} placeholder="Find a
customer" />

```

```

    );
  }
  // set the value to action failure template
  private template(): JSX.Element {
    return (
      <span className='action-failure'> Data fetch get fails</span>
    );
  }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

[Functional-component]

INDEX.JSX

```

// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
  // bind the DataManager instance to dataSource property
  const customerData = new DataManager({
    adaptor: new ODataV4Adaptor,
    crossDomain: true,
    url: 'https://services.odata.org/V4/Northwind/Northwind.svc/'
  });
  // bind the Query instance to query property
  const query = new Query().from('Customers').select(['ContactName',
'CustomerID']).take(6);
  // maps the appropriate column to fields property
  const fields = { value: 'ContactName' };
  return (
    // specifies the tag for render the AutoComplete component
    <AutoCompleteComponent id="atcelement" query={query}
    actionFailureTemplate={template = template.bind(this)}
    dataSource={customerData} fields={fields} placeholder="Find a customer"/>);
    // set the value to action failure template
    function template() {
      return (<span className='action-failure'> Data fetch get
fails</span>);
    }
  }
ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.TSX

```

// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
  // bind the DataManager instance to dataSource property
  const customerData: DataManager = new DataManager({
    adaptor: new ODataV4Adaptor,

```

```

    crossDomain: true,
    url: 'https://services.odata.org/V4/Northwind/Northwind.svcs/'
  });
  // bind the Query instance to query property
  const query: Query = new Query().from('Customers').select(['ContactName',
  'CustomerID']).take(6);
  // maps the appropriate column to fields property
  const fields: object = { value: 'ContactName' };
  return (
    // specifies the tag for render the AutoComplete component
    <AutoCompleteComponent id="atcelement" query={query}
    actionFailureTemplate={template = template.bind(this)}
    dataSource={customerData} fields={fields} placeholder="Find a customer" />
  );
  // set the value to action failure template
  function template(): JSX.Element {
    return (
      <span className='action-failure'> Data fetch get fails</span>
    );
  }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

See Also

- [How to achieve filtering](#)
- [How to group the data using header](#)
- [How to show the list items with icon](#)

Virtualization in AutoComplete Component

AutoComplete virtualization is a technique used to efficiently render extensive lists of items while minimizing the impact on performance. This method is particularly advantageous when dealing with large datasets because it ensures that only a fixed number of DOM (Document Object Model) elements are created. When scrolling through the list, existing DOM elements are reused to display relevant data instead of generating new elements for each item. This recycling process is managed internally.

During virtual scrolling, the data retrieved from the data source depends on the popup height and the calculation of the list item height. Enabling the [enableVirtualization](#) option in a AutoComplete activates this virtualization technique.

When fetching data from the data source, the [actionBegin](#) event is triggered before data retrieval begins. Then, the [actionComplete](#) event is triggered once the data is successfully fetched.

When the enableVirtualization property is enabled, the `skip` and `take` properties provided by the user within the Query class at the initial state or during the `actionBegin` or `actionComplete` events will not be considered, since it is internally managed and calculated based on certain dimensions with respect to the popup height.

Binding local data

The AutoComplete can generate its list items through an array of complex data. For this, the appropriate columns should be mapped to the [fields](#) property. When using virtual scrolling, the list updates based on the scroll offset value, triggering a request to fetch more data from the server. As the data is being

fetches, the `actionBegin` event occurs before the data retrieval starts. Once the data retrieval is successful, the `actionComplete` event is triggered, indicating that the data fetch process is complete.

In the following example, `text` column from complex data have been mapped to the `value` field.

[Class-component]

INDEX.JSX

```
import { AutoCompleteComponent, Inject, VirtualScroll } from
 '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
  constructor(props) {
    super(props);
    this.records = Array.from({ length: 150 }, (_, i) => ({
      id: 'id' + (i + 1),
      text: `Item ${i + 1}`,
    }));
  }
  fields = { value: 'text' };
  render() {
    return (
      // specifies the tag for render the AutoComplete component
      <AutoCompleteComponent id="datas" dataSource={this.records}
placeholder="e.g. Item 1" enableVirtualization={true} fields={this.fields}
allowFiltering={true} popupHeight="200px">
        <Inject services={[VirtualScroll]} />
      </AutoCompleteComponent>);
  }
}
ReactDOM.render(<App />, document.getElementById('sample'));
```

INDEX.TSX

```
import { AutoCompleteComponent, Inject, VirtualScroll } from
 '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}, {}> {
  // maps the appropriate column to fields property
  private fields: object = { value: 'text' };
  // define the array of string
  private records: { [key: string]: Object }[] = [];
  constructor(props) {
    super(props);
    this.records = Array.from({ length: 150 }, (_, i) => ({
      id: 'id' + (i + 1),
      text: `Item ${i + 1}`,
    }));
  }
  public render() {
    return (
      // specifies the tag for render the AutoComplete component
```

```

        <AutoCompleteComponent id="datas" dataSource={this.records}
placeholder="e.g. Item 1" enableVirtualization={true} fields={this.fields}
allowFiltering={true} popupHeight="200px">
        <Inject services={[VirtualScroll]} />
    </AutoCompleteComponent>
    );
}
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

Binding remote data

The AutoComplete supports retrieval of data from remote data services with the help of **DataManager** component. When using remote data, it initially fetches all the data from the server, triggering the **actionBegin** and **actionComplete** events, and then stores the data locally. During virtual scrolling, additional data is retrieved from the locally stored data, triggering the **actionBegin** and **actionComplete** events at that time as well.

The following sample displays the OrderId from the **Orders** Data Service.

[Class-component]

INDEX.JSX

```

import { AutoCompleteComponent, Inject, VirtualScroll } from
 '@syncfusion/ej2-react-dropdowns';
import { DataManager, WebApiAdaptor, Query } from '@syncfusion/ej2-data';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
    customerData = new DataManager({
        url: 'https://services.syncfusion.com/react/production/api/Orders',
        adaptor: new WebApiAdaptor,
        crossDomain: true
    });
    customerField = { value: 'OrderID' };
    render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="datas" dataSource={this.customerData}
placeholder="OrderID" enableVirtualization={true}
fields={this.customerField} allowFiltering={true} popupHeight="200px">
                <Inject services={[VirtualScroll]} />
            </AutoCompleteComponent>);
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.TSX

```

import { AutoCompleteComponent, Inject, VirtualScroll } from
 '@syncfusion/ej2-react-dropdowns';
import { DataManager, WebApiAdaptor, Query } from '@syncfusion/ej2-data';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}, {}> {

```

```
// maps the appropriate column to fields property
private customerField: object = { value: 'OrderID' };

private customerData: DataManager = new DataManager({
  url: 'https://services.syncfusion.com/react/production/api/Orders',
  adaptor: new WebApiAdaptor,
  crossDomain: true
});

public render() {
  return (
    // specifies the tag for render the AutoComplete component
    <AutoCompleteComponent id="datas" dataSource={this.customerData}
    placeholder="OrderID" enableVirtualization={true}
    fields={this.customerField} allowFiltering={true} popupHeight="200px">
      <Inject services={[VirtualScroll]} />
    </AutoCompleteComponent>
  );
}

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

Grouping

The AutoComplete component supports grouping with Virtualization. It allows you to organize elements into groups based on different categories. Each item in the list can be classified using the [groupBy](#) field in the data table. After grouping, virtualization works similarly to local data binding, providing a seamless user experience. When the data source is bound to remote data, an initial request is made to retrieve all data for the purpose of grouping. Subsequently, the grouped data works in the same way as local data binding virtualization, enhancing performance and responsiveness.

The following sample shows the example for Grouping with Virtualization.

[Class-component]

INDEX.JSX

```
import { AutoCompleteComponent, Inject, VirtualScroll } from
'@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
  constructor(props) {
    super(props);
    for (let i = 1; i <= 150; i++) {
      let item = {};
      item.id = 'id' + i;
      item.text = `Item ${i}`;
      // Generate a random number between 1 and 4 to determine the
      group

      const randomGroup = Math.floor(Math.random() * 4) + 1;
      switch (randomGroup) {
        case 1:
          item.group = 'Group A';
          break;
        case 2:
          item.group = 'Group B';
          break;
```



```

        case 3:
            item.group = 'Group C';
            break;
        case 4:
            item.group = 'Group D';
            break;
        default:
            break;
    }
    this.records.push(item);
}
}
fields = { groupBy: 'group', value: 'text' };
render() {
    return (
        // specifies the tag for render the AutoComplete component
        <AutoCompleteComponent id="datas" dataSource={this.records}
placeholder="e.g. Item 1" enableVirtualization={true} fields={this.fields}
allowFiltering={true} popupHeight="200px">
            <Inject services={[VirtualScroll]} />
        </AutoCompleteComponent>);
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.TSX

```

import { AutoCompleteComponent, Inject, VirtualScroll } from
'@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}, {}> {
    // maps the appropriate column to fields property
    private fields: object = { groupBy: 'group', value: 'text' };
    // define the array of string
    private records: { [key: string]: Object }[] = [];
    constructor(props) {
        super(props);
        for (let i = 1; i <= 150; i++) {
            const item: { id: string, text: string, group: string } = {
                id: 'id' + i,
                text: `Item ${i}`,
                group: ''
            };
            // Generate a random number between 1 and 4 to determine the group
            const randomGroup = Math.floor(Math.random() * 4) + 1;
            switch (randomGroup) {
                case 1:
                    item.group = 'Group A';
                    break;
                case 2:
                    item.group = 'Group B';
                    break;
                case 3:
                    item.group = 'Group C';
                    break;
            }
            this.records.push(item);
        }
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

```

        case 4:
            item.group = 'Group D';
            break;
        default:
            break;
    }
    this.records.push(item);
}
}
}
public render() {
    return (
        // specifies the tag for render the AutoComplete component
        <AutoCompleteComponent id="datas" dataSource={this.records}
placeholder="e.g. Item 1" enableVirtualization={true} fields={this.fields}
allowFiltering={true} popupHeight="200px">
            <Inject services={[VirtualScroll]} />
        </AutoCompleteComponent>
    );
}
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

Grouping in React Auto complete component

The AutoComplete supports wrapping nested elements into a group based on different categories. The category of each list item can be mapped through the `groupBy` field in the data table. The group header is displayed both as inline and fixed headers. The fixed group header content is updated dynamically on scrolling the popup list with its category value.

In below sample, vegetables are grouped based on its category using `groupBy` field.

[Class-component]

INDEX.JSX

```

import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
    // define the data with category
    vegetableData = [
        { Vegetable: 'Cabbage', Category: 'Leafy and Salad', Id: 'item1' },
        { Vegetable: 'Spinach', Category: 'Leafy and Salad', Id: 'item2' },
        { Vegetable: 'Wheat grass', Category: 'Leafy and Salad', Id: 'item3' },
    ],
    { Vegetable: 'Yarrow', Category: 'Leafy and Salad', Id: 'item4' },
    { Vegetable: 'Pumpkins', Category: 'Leafy and Salad', Id: 'item5' },
    { Vegetable: 'Chickpea', Category: 'Beans', Id: 'item6' },
    { Vegetable: 'Green bean', Category: 'Beans', Id: 'item7' },
    { Vegetable: 'Horse gram', Category: 'Beans', Id: 'item8' }
    ];
    // map the groupBy field with Category column
    fields = { groupBy: 'Category', value: 'Vegetable' };
    render() {
        return (
            // specifies the tag for render the AutoComplete component

```

```

        <AutoCompleteComponent id="atcelement" fields={this.fields}
        dataSource={this.vegetableData} placeholder="Select a vegetable"/>);
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.TSX

```

import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}, {}> {
    // define the data with category
    private vegetableData: { [key: string]: Object }[] = [
        { Vegetable: 'Cabbage', Category: 'Leafy and Salad', Id: 'item1' },
        { Vegetable: 'Spinach', Category: 'Leafy and Salad', Id: 'item2' },
        { Vegetable: 'Wheat grass', Category: 'Leafy and Salad', Id: 'item3' },
    ],
    { Vegetable: 'Yarrow', Category: 'Leafy and Salad', Id: 'item4' },
    { Vegetable: 'Pumpkins', Category: 'Leafy and Salad', Id: 'item5' },
    { Vegetable: 'Chickpea', Category: 'Beans', Id: 'item6' },
    { Vegetable: 'Green bean', Category: 'Beans', Id: 'item7' },
    { Vegetable: 'Horse gram', Category: 'Beans', Id: 'item8' }
    ];
    // map the groupBy field with Category column
    private fields: object = { groupBy: 'Category', value: 'Vegetable' };
    public render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="atcelement" fields={this.fields}
            dataSource={this.vegetableData} placeholder="Select a vegetable" />
        );
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

[Functional-component]

INDEX.JSX

```

import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    // define the data with category
    let vegetableData = [
        { Vegetable: 'Cabbage', Category: 'Leafy and Salad', Id: 'item1' },
        { Vegetable: 'Spinach', Category: 'Leafy and Salad', Id: 'item2' },
        { Vegetable: 'Wheat grass', Category: 'Leafy and Salad', Id: 'item3' },
    ],
    { Vegetable: 'Yarrow', Category: 'Leafy and Salad', Id: 'item4' },
    { Vegetable: 'Pumpkins', Category: 'Leafy and Salad', Id: 'item5' },
    { Vegetable: 'Chickpea', Category: 'Beans', Id: 'item6' },
    { Vegetable: 'Green bean', Category: 'Beans', Id: 'item7' },
    { Vegetable: 'Horse gram', Category: 'Beans', Id: 'item8' }
    ];
}

```

```
// map the groupBy field with Category column
const fields = { groupBy: 'Category', value: 'Vegetable' };
return (
  // specifies the tag for render the AutoComplete component
  <AutoCompleteComponent id="atcelement" fields={fields}
  dataSource={vegetableData} placeholder="Select a vegetable"/>);
}
ReactDOM.render(<App />, document.getElementById('sample'));
```

INDEX.TSX

```
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
  // define the data with category
  let vegetableData: { [key: string]: Object }[] = [
    { Vegetable: 'Cabbage', Category: 'Leafy and Salad', Id: 'item1' },
    { Vegetable: 'Spinach', Category: 'Leafy and Salad', Id: 'item2' },
    { Vegetable: 'Wheat grass', Category: 'Leafy and Salad', Id: 'item3' },
  ],
  { Vegetable: 'Yarrow', Category: 'Leafy and Salad', Id: 'item4' },
  { Vegetable: 'Pumpkins', Category: 'Leafy and Salad', Id: 'item5' },
  { Vegetable: 'Chickpea', Category: 'Beans', Id: 'item6' },
  { Vegetable: 'Green bean', Category: 'Beans', Id: 'item7' },
  { Vegetable: 'Horse gram', Category: 'Beans', Id: 'item8' }
  ];
  // map the groupBy field with Category column
  const fields: object = { groupBy: 'Category', value: 'Vegetable' };
  return (
    // specifies the tag for render the AutoComplete component
    <AutoCompleteComponent id="atcelement" fields={fields}
    dataSource={vegetableData} placeholder="Select a vegetable" />
  );
}
ReactDOM.render(<App />, document.getElementById('sample'));
```

Customization

The grouping header is also provided with customization option. This allows custom designing using the `groupTemplate` property for both inline and fixed headers as referred here [groupTemplate](#) property for both inline and fixed header.

See Also

- [Group Template support to AutoComplete.](#)

Filtering in React Auto complete component

The AutoComplete has built-in support for filtering the data items when [allowFiltering](#) is enabled. The filter operation starts as soon as you start typing characters in the component.

Change the filter type

Determines on which filter type, the component needs to be considered on search action. The available [filterType](#) and its supported data types are

Filter Type	Description	Supported Types
-------------	-------------	-----------------

---	---	
-----	-----	--

StartsWith	Checks whether a value begins with the specified value.	String
------------	---	--------

EndsWith	Checks whether a value ends with specified value.	String
----------	---	--------

Contains	Checks whether a value contains with specified value.	String
----------	---	--------

The following examples, data filtering is done with **StartsWith** type.

[Class-component]

INDEX.JSX

```
// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
    // bind the DataManager instance to dataSource property
    searchData = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url: 'https://services.odata.org/V4/Northwind/Northwind.svc/'
    });
    // maps the appropriate column to fields property
    fields = { value: "ContactName" };
    // bind the Query instance to query property
    query = new Query().from('Suppliers').select(["SupplierID",
    "ContactName"]).take(10);
    // sort the resulted items
    sortOrder = 'Ascending';
    render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="atcelement" query={this.query}
            dataSource={this.searchData} fields={this.fields} placeholder="Find a
            customer" filterType="StartsWith" sortOrder={this.sortOrder}/>);
        }
    }
    ReactDOM.render(<App />, document.getElementById('sample'));
```

INDEX.TSX

```
// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { SortOrder } from '@syncfusion/ej2-lists';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}, {}> {
    // bind the DataManager instance to dataSource property
    private searchData: DataManager = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
```

```

        url: 'https://services.odata.org/V4/Northwind/Northwind.svc/'
    });
    // maps the appropriate column to fields property
    private fields: object = { value: "ContactName" };
    // bind the Query instance to query property
    private query: Query = new
    Query().from('Suppliers').select(["SupplierID", "ContactName"]).take(10);
    // sort the resulted items
    private sortOrder: SortOrder = 'Ascending';
    public render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="atcelement" query={this.query}
            dataSource={this.searchData} fields={this.fields} placeholder="Find a
            customer" filterType="StartsWith" sortOrder={this.sortOrder} />
        );
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

[Functional-component]

INDEX.JSX

```

// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    // bind the DataManager instance to dataSource property
    const searchData = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url: 'https://services.odata.org/V4/Northwind/Northwind.svc/'
    });
    // maps the appropriate column to fields property
    const fields = { value: "ContactName" };
    // bind the Query instance to query property
    const query = new Query().from('Suppliers').select(["SupplierID",
    "ContactName"]).take(10);
    // sort the resulted items
    const sortOrder = 'Ascending';
    return (
        // specifies the tag for render the AutoComplete component
        <AutoCompleteComponent id="atcelement" query={query}
        dataSource={searchData} fields={fields} placeholder="Find a customer"
        filterType="StartsWith" sortOrder={sortOrder}/>
    );
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.TSX

```

// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { SortOrder } from '@syncfusion/ej2-lists';

```

```

import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    // bind the DataManager instance to dataSource property
    const searchData: DataManager = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url: 'https://services.odata.org/V4/Northwind/Northwind.svc/'
    });
    // maps the appropriate column to fields property
    const fields: object = { value: "ContactName" };
    // bind the Query instance to query property
    const query: Query = new Query().from('Suppliers').select(["SupplierID",
    "ContactName"]).take(10);
    // sort the resulted items
    const sortOrder: SortOrder = 'Ascending';
    return (
        // specifies the tag for render the AutoComplete component
        <AutoCompleteComponent id="atcelement" query={query}
        dataSource={searchData} fields={fields} placeholder="Find a customer"
        filterType="StartsWith" sortOrder={sortOrder} />
    );
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

Filter item count

You can specify the filter suggestion item count through [suggestionCount](#) property of AutoComplete.

The following examples, to restrict the suggestion list item counts as 2.

[Class-component]

INDEX.JSX

```

// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
    // bind the DataManager instance to dataSource property
    customerData = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url: 'https://services.odata.org/V4/Northwind/Northwind.svc/'
    });
    // bind the Query instance to query property
    query = new Query().from('Customers').select(['ContactName',
    'CustomerID']);
    // maps the appropriate column to fields property
    fields = { value: 'ContactName' };
    // sort the resulted items
    sortOrder = 'Ascending';
    render() {
        return (
            // specifies the tag for render the AutoComplete component

```

```

        <AutoCompleteComponent id="atcelement" query={this.query}
        dataSource={this.customerData} fields={this.fields} placeholder="Find a
        customer" sortOrder={this.sortOrder} suggestionCount={2}
        filterType="StartsWith"/>;
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.TSX

```

// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { SortOrder } from '@syncfusion/ej2-lists';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}, {}> {
    // bind the DataManager instance to dataSource property
    private customerData: DataManager = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url: 'https://services.odata.org/V4/Northwind/Northwind.svc/'
    });
    // bind the Query instance to query property
    private query: Query = new
    Query().from('Customers').select(['ContactName', 'CustomerID']);
    // maps the appropriate column to fields property
    private fields: object = { value: 'ContactName' };
    // sort the resulted items
    private sortOrder: SortOrder = 'Ascending';
    public render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="atcelement" query={this.query}
            dataSource={this.customerData}
            fields={this.fields} placeholder="Find a customer"
            sortOrder={this.sortOrder} suggestionCount={2} filterType="StartsWith"/>
        );
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Syncfusion React AutoComplete</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/ej2-
base/styles/material.css" rel="stylesheet" />

```



```

<link href="https://cdn.syncfusion.com/ej2/20.3.56/ej2-react-
inputs/styles/material.css" rel="stylesheet" />
<link href="https://cdn.syncfusion.com/ej2/20.3.56/ej2-react-
dropdowns/styles/material.css" rel="stylesheet" />
<script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></sc
ript>
<script src="systemjs.config.js"></script>
<style>
    #loader {
        color: #008cff;
        height: 40px;
        left: 45%;
        position: absolute;
        top: 45%;
        width: 30%;
    }
</style>
</head>
<body>
    <div id='sample' style="margin: 20px auto 0; width:250px;">
        <div id='loader'>Loading....</div>
    </div>
</body>
</html>

```

[Functional-component]

INDEX.JSX

```

// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    // bind the DataManager instance to dataSource property
    const customerData = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url: 'https://services.odata.org/V4/Northwind/Northwind.svc/'
    });
    // bind the Query instance to query property
    const query = new Query().from('Customers').select(['ContactName',
    'CustomerID']);
    // maps the appropriate column to fields property
    const fields = { value: 'ContactName' };
    // sort the resulted items
    const sortOrder = 'Ascending';
    return (
        // specifies the tag for render the AutoComplete component
        <AutoCompleteComponent id="atcelement" query={query}
        dataSource={customerData} fields={fields} placeholder="Find a customer"
        sortOrder={sortOrder} suggestionCount={2} filterType="StartsWith"/>);
    }
ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.TSX

```

// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { SortOrder } from '@syncfusion/ej2-lists';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    // bind the DataManager instance to dataSource property
    const customerData: DataManager = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url: 'https://services.odata.org/V4/Northwind/Northwind.svc/'
    });
    // bind the Query instance to query property
    const query: Query = new
Query().from('Customers').select(['ContactName', 'CustomerID']);
    // maps the appropriate column to fields property
    const fields: object = { value: 'ContactName' };
    // sort the resulted items
    const sortOrder: SortOrder = 'Ascending';
    return (
        // specifies the tag for render the AutoComplete component
        <AutoCompleteComponent id="atcelement" query={query}
dataSource={customerData}
        fields={fields} placeholder="Find a customer"
sortOrder={sortOrder} suggestionCount={2} filterType= "StartsWith"/>
    );
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Syncfusion React AutoComplete</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/ej2-
base/styles/material.css" rel="stylesheet" />
    <link href="https://cdn.syncfusion.com/ej2/20.3.56/ej2-react-
inputs/styles/material.css" rel="stylesheet" />
    <link href="https://cdn.syncfusion.com/ej2/20.3.56/ej2-react-
dropdowns/styles/material.css" rel="stylesheet" />
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></sc
ript>
    <script src="systemjs.config.js"></script>
    <style>
        #loader {

```

```

        color: #008cff;
        height: 40px;
        left: 45%;
        position: absolute;
        top: 45%;
        width: 30%;
    }
</style>
</head>
<body>
    <div id='sample' style="margin: 20px auto 0; width:250px;">
        <div id='loader'>Loading....</div>
    </div>
</body>
</html>

```

Limit the minimum filter character

You can set the limit for the character count to filter the data on the AutoComplete. This can be done by set the [minLength](#) property to AutoComplete.

In the following example, the remote request does not fetch the search data, until the search key contains three characters.

[Class-component]

INDEX.JSX

```

import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
    // bind the DataManager instance to dataSource property
    searchData = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url:
'https://services.odata.org/V4/Northwind/Northwind.svc/Customers'
    });
    // maps the appropriate column to fields property
    fields = { value: 'ContactName' };
    // bind the Query instance to query property
    query = new Query().select(['ContactName', 'CustomerID']).take(10);
    // sort the resulted items
    sortOrder = 'Ascending';
    render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="atcelement" query={this.query}
dataSource={this.searchData} sortOrder={this.sortOrder}
filterType="StartsWith" fields={this.fields} placeholder="Find a customer"
minLength={3}/>);
        }
    }
ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.TSX

```

import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { SortOrder } from '@syncfusion/ej2-lists';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}>, {}> {
    // bind the DataManager instance to dataSource property
    private searchData: DataManager = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url:
'https://services.odata.org/V4/Northwind/Northwind.svc/Customers'
    });
    // maps the appropriate column to fields property
    private fields: object = { value: 'ContactName' };
    // bind the Query instance to query property
    private query: Query = new Query().select(['ContactName',
'CustomerID']).take(10);
    // sort the resulted items
    private sortOrder: SortOrder = 'Ascending';
    public render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="atcelement" query={this.query}
dataSource={this.searchData} sortOrder={this.sortOrder} filterType=
"StartsWith" fields={this.fields} placeholder="Find a customer"
minLength={3}/>
        );
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

[Functional-component]**INDEX.JSX**

```

import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    // bind the DataManager instance to dataSource property
    const searchData = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url:
'https://services.odata.org/V4/Northwind/Northwind.svc/Customers'
    });
    // maps the appropriate column to fields property
    const fields = { value: 'ContactName' };
    // bind the Query instance to query property
    const query = new Query().select(['ContactName',
'CustomerID']).take(10);
    // sort the resulted items
    const sortOrder = 'Ascending';

```

```

    return (
      // specifies the tag for render the AutoComplete component
      <AutoCompleteComponent id="atcelement" query={query}
      dataSource={searchData} sortOrder={sortOrder} filterType="StartsWith"
      fields={fields} placeholder="Find a customer" minLength={3}/>;
    )
  }
  ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.TSX

```

import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { SortOrder } from '@syncfusion/ej2-lists';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
  // bind the DataManager instance to dataSource property
  const searchData: DataManager = new DataManager({
    adaptor: new ODataV4Adaptor,
    crossDomain: true,
    url:
'https://services.odata.org/V4/Northwind/Northwind.svc/Customers'
  });
  // maps the appropriate column to fields property
  const fields: object = { value: 'ContactName' };
  // bind the Query instance to query property
  const query: Query = new Query().select(['ContactName',
'CustomerID']).take(10);
  // sort the resulted items
  const sortOrder: SortOrder = 'Ascending';
  return (
    // specifies the tag for render the AutoComplete component
    <AutoCompleteComponent id="atcelement" query={query}
    dataSource={searchData} sortOrder={sortOrder} filterType= "StartsWith"
    fields={fields} placeholder="Find a customer" minLength={3}/>
  );
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

Case sensitive filtering

Data items can be filtered either with or without case sensitivity using the DataManager. This can be done by set the [ignoreCase](#) property of AutoComplete.

The below sample depicts how to filter the data with case-sensitive.

[Class-component]

INDEX.JSX

```

import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
  // bind the DataManager instance to dataSource property
  searchData = ['ram', 'Ravi', 'suresh', 'Suresh'];
  // maps the appropriate column to fields property

```

```

    fields = { value: "ContactName" };
    // sort the resulted items
    sortOrder = 'Ascending';
    render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="atcelement" dataSource={this.searchData}
            filterType="StartsWith" sortOrder={this.sortOrder} fields={this.fields}
            placeholder="eg: Ravi" ignoreCase={false}/>);
        }
    }
    ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.TSX

```

import { SortOrder } from '@syncfusion/ej2-lists';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}, {}> {
    // bind the DataManager instance to dataSource property
    private searchData: string[] = ['ram', 'Ravi', 'suresh', 'Suresh'];
    // maps the appropriate column to fields property
    private fields: object = { value: "ContactName" };
    // sort the resulted items
    private sortOrder: SortOrder = 'Ascending';
    public render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="atcelement"
            dataSource={this.searchData} filterType= "StartsWith"
            sortOrder={this.sortOrder} fields={this.fields} placeholder="eg: Ravi"
            ignoreCase={false} />
        );
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

[Functional-component]

INDEX.JSX

```

import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    // bind the DataManager instance to dataSource property
    const searchData = ['ram', 'Ravi', 'suresh', 'Suresh'];
    // maps the appropriate column to fields property
    const fields = { value: "ContactName" };
    // sort the resulted items
    const sortOrder = 'Ascending';
    return (
        // specifies the tag for render the AutoComplete component

```

```

    <AutoCompleteComponent id="atcelement" dataSource={searchData}
    filterType="StartsWith" sortOrder={sortOrder} fields={fields}
    placeholder="eg: Ravi" ignoreCase={false}/>;
  }
  ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.TSX

```

import { SortOrder } from '@syncfusion/ej2-lists';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
  // bind the DataManager instance to dataSource property
  const searchData: string[] = ['ram', 'Ravi', 'suresh', 'Suresh'];
  // maps the appropriate column to fields property
  const fields: object = { value: "ContactName" };
  // sort the resulted items
  const sortOrder: SortOrder = 'Ascending';
  return (
    // specifies the tag for render the AutoComplete component
    <AutoCompleteComponent id="atcelement" dataSource={searchData}
    filterType= "StartsWith" sortOrder={sortOrder} fields={fields}
    placeholder="eg: Ravi" ignoreCase={false} />
  );
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

Diacritics Filtering

An AutoComplete supports diacritics filtering which will ignore the [diacritics](#) and makes it easier to filter the results in international characters lists when the [ignoreAccent](#) is enabled.

In the following sample, data with diacritics are bound as dataSource for AutoComplete.

[Class-component]

INDEX.JSX

```

import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
  diacriticsData = [
    'Aeróbics',
    'Aeróbics en Agua',
    'Aerografía',
    'Aerodelaje',
    'Águilas',
    'Ajedrez',
    'Ala Delta',
    'Álbumes de Música',
    'Alusivos',
    'Análisis de Escritura a Mano'
  ];
  render() {

```

```

        return (<AutoCompleteComponent id="diacritics" ignoreAccent={true}
dataSource={this.diacriticsData} placeholder="e.g: aero"/>);
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.TSX

```

import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}, {}> {
    private diacriticsData: string[] = [
        'Aeróbics',
        'Aeróbics en Agua',
        'Aerografía',
        'Aeromodelaje',
        'Águilas',
        'Ajedrez',
        'Ala Delta',
        'Álbumes de Música',
        'Alusivos',
        'Análisis de Escritura a Mano'];
    public render() {
        return (
            <AutoCompleteComponent id="diacritics" ignoreAccent={true}
dataSource={this.diacriticsData} placeholder="e.g: aero" />
        );
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

[Functional-component]

INDEX.JSX

```

import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    const diacriticsData = [
        'Aeróbics',
        'Aeróbics en Agua',
        'Aerografía',
        'Aeromodelaje',
        'Águilas',
        'Ajedrez',
        'Ala Delta',
        'Álbumes de Música',
        'Alusivos',
        'Análisis de Escritura a Mano'
    ];
    return (<AutoCompleteComponent id="diacritics" ignoreAccent={true}
dataSource={diacriticsData} placeholder="e.g: aero"/>);
}
ReactDOM.render(<App />, document.getElementById('sample'));

```


INDEX.TSX

```
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    const diacriticsData: string[] = [
        'Aeróbics',
        'Aeróbics en Agua',
        'Aerografía',
        'Aerodelaje',
        'Águilas',
        'Ajedrez',
        'Ala Delta',
        'Álbumes de Música',
        'Alusivos',
        'Análisis de Escritura a Mano'];
    return (
        <AutoCompleteComponent id="diacritics" ignoreAccent={true}
        dataSource={diacriticsData} placeholder="e.g: aero" />
    );
}
ReactDOM.render(<App />, document.getElementById('sample'));
```

See Also

- [How to achieve autofill while filtering](#)
- [How to group the data using header](#)
- [How to highlight the search data](#)

Localization in React Auto complete component

The Localization library allows you to localize the static text content of [noRecordsTemplate](#) and [actionFailureTemplate](#) properties according to the culture currently assigned to the AutoComplete.

| Locale key | en-US (default) |

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

| noRecordsTemplate | No Records Found |

| actionFailureTemplate | The Request Failed |

Loading translations

To load translation object to your application, use load function of the **L10n** class.

In the following sample, French culture is set to the AutoComplete and no data is loaded. Hence, the `noRecordsTemplate` property displays its text in French culture initially, and if the sample is run offline, then the `actionFailureTemplate` property displays its text appropriately.

[Class-component]

INDEX.JSX

```
// import L10n class for load function
import { L10n } from '@syncfusion/ej2-base';
// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
    // bind remotedata to showcase actionFailureTemplate in offline.
    customerData = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url:
'https://services.odata.org/V4/Northwind/Northwind.svc/Customers'
    });
    // maps the appropriate column to fields property
    fields = { value: 'ContactName' };
    // take 0 item to showcase noRecordsTemplate property.
    query = new Query().select(['ContactName', 'CustomerID']).take(0);
    // set locale culture to AutoComplete
    componentWillMount() {
        L10n.load({
            'fr-BE': {
                'dropdowns': {
                    'actionFailureTemplate': "Modèle d'échec d'action",
                    'noRecordsTemplate': "Aucun enregistrement trouvé",
                }
            }
        });
    }
    render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="atcelement" fields={this.fields}
            locale="fr-BE" query={this.query} dataSource={this.customerData}
            placeholder="Trouver un client"/>);
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));
```

INDEX.TSX

```
// import L10n class for load function
import { L10n } from '@syncfusion/ej2-base';
// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}>, {}> {
    // bind remotedata to showcase actionFailureTemplate in offline.
    private customerData: DataManager = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url: 'https://services.odata.org/V4/Northwind/Northwind.svc/Customers'
    });
};
```

```

// maps the appropriate column to fields property
private fields: object = { value: 'ContactName' };
// take 0 item to showcase noRecordsTemplate property.
private query: Query = new Query().select(['ContactName',
'CustomerID']).take(0);
// set locale culture to AutoComplete
public componentWillMount() {
    L10n.load({
        'fr-BE': {
            'dropdowns': {
                'actionFailureTemplate': "Modèle d'échec d'action",
                'noRecordsTemplate': "Aucun enregistrement trouvé",
            }
        }
    });
}
public render() {
    return (
        // specifies the tag for render the AutoComplete component
        <AutoCompleteComponent id="atcelement" fields={this.fields}
        locale="fr-BE" query={this.query} dataSource={this.customerData}
        placeholder="Trouver un client"/>
    );
}
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

[Functional-component]

INDEX.JSX

```

// import L10n class for load function
import { L10n } from '@syncfusion/ej2-base';
// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    // bind remotedata to showcase actionFailureTemplate in offline.
    const customerData = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url:
'https://services.odata.org/V4/Northwind/Northwind.svc/Customers'
    });
    // maps the appropriate column to fields property
    const fields = { value: 'ContactName' };
    // take 0 item to showcase noRecordsTemplate property.
    const query = new Query().select(['ContactName', 'CustomerID']).take(0);
    // set locale culture to AutoComplete
    React.useEffect(() => {
        L10n.load({
            'fr-BE': {
                'dropdowns': {
                    'actionFailureTemplate': "Modèle d'échec d'action",

```

```

        'noRecordsTemplate': "Aucun enregistrement trouvé",
    }
    });
}, []);
return (
    // specifies the tag for render the AutoComplete component
    <AutoCompleteComponent id="atcelement" fields={fields} locale="fr-BE"
    query={query} dataSource={customerData} placeholder="Trouver un client"/>);
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.TSX

```

// import L10n class for load function
import { L10n } from '@syncfusion/ej2-base';
// import DataManager related classes
import { DataManager, ODataV4Adaptor, Query } from '@syncfusion/ej2-data';
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    // bind remotedata to showcase actionFailureTemplate in offline.
    const customerData: DataManager = new DataManager({
        adaptor: new ODataV4Adaptor,
        crossDomain: true,
        url: 'https://services.odata.org/V4/Northwind/Northwind.svc/Customers'
    });
    // maps the appropriate column to fields property
    const fields: object = { value: 'ContactName' };
    // take 0 item to showcase noRecordsTemplate property.
    const query: Query = new Query().select(['ContactName',
    'CustomerID']).take(0);
    // set locale culture to AutoComplete
    React.useEffect(() => {
        L10n.load({
            'fr-BE': {
                'dropdowns': {
                    'actionFailureTemplate': "Modèle d'échec d'action",
                    'noRecordsTemplate': "Aucun enregistrement trouvé",
                }
            }
        });
    }, []);
    return (
        // specifies the tag for render the AutoComplete component
        <AutoCompleteComponent id="atcelement" fields={fields}
        locale="fr-BE" query={query} dataSource={customerData} placeholder="Trouver
        un client"/>
    );
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

See Also

- [Accessibility](#)
- [How to bind the data to the autocomplete](#)

Style in React Auto complete 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 wrapper element

Use the following CSS to customize the appearance of wrapper element.

```
`css
.e-ddl.e-input-group.e-control-wrapper .e-input {
font-size: 20px;
font-family: emoji;
color: #ab3243;
background: #32a5ab;
}
`
```

Customizing the dropdown icon's color

Use the following CSS to customize the dropdown icon's color.

```
`css
.e-ddl.e-input-group .e-input-group-icon,.e-ddl.e-input-group.e-control-wrapper .e-input-group-
icon:hover {
color: #bb233d;
font-size: 13px;
}
`
```

Customizing the focus color

Use the following CSS to customize the focusing color of input element.

```
`css
.e-ddl.e-input-group.e-control-wrapper.e-input-focus::before, .e-ddl.e-input-group.e-control-wrapper.e-
input-focus::after {
background: #c000ff;
}
`
```

Customizing the outline theme's focus color

Use the following CSS to customize the focusing color of outline theme.

```
`css
.e-outline.e-input-group.e-input-focus:hover:not(.e-success):not(.e-warning):not(.e-error):not(.e-disabled):not(.e-float-icon-left),.e-outline.e-input-group.e-input-focus.e-control-wrapper:hover:not(.e-success):not(.e-warning):not(.e-error):not(.e-disabled):not(.e-float-icon-left),.e-outline.e-input-group.e-input-focus:not(.e-success):not(.e-warning):not(.e-error):not(.e-disabled),.e-outline.e-input-group.e-control-wrapper.e-input-focus:not(.e-success):not(.e-warning):not(.e-error):not(.e-disabled) {
border-color: #b1bd15;
box-shadow: inset 1px 1px #b1bd15, inset -1px 0 #b1bd15, inset 0 -1px #b1bd15;
}
`
```

Customizing the disabled component's text color

Use the following CSS to customize the text color when the component is disabled.

```
`css
.e-input-group.e-control-wrapper .e-input[disabled] {
-webkit-text-fill-color: #0d9133;
}
`
```

Customizing the float label element's focusing color

Use the following CSS to customize the focusing color of float label element.

```
`css
.e-float-input.e-input-group:not(.e-float-icon-left) .e-float-line::before,.e-float-input.e-control-wrapper.e-input-group:not(.e-float-icon-left) .e-float-line::before,.e-float-input.e-input-group:not(.e-float-icon-left) .e-float-line::after,.e-float-input.e-control-wrapper.e-input-group:not(.e-float-icon-left) .e-float-line::after {
background-color: #2319b8;
}

.e-ddl.e-input-group.e-control-wrapper.e-float-input.e-input-focus .e-float-text.e-label-top, .e-float-input.e-control-wrapper:not(.e-error).e-input-focus input ~ label.e-float-text {
color: #2319b8;
}
`
```

Customizing the color of the placeholder text

Use the following CSS to customize the text color of placeholder.

```
`css
.e-ddl.e-input-group input.e-input::placeholder {
color: red;
}
```

```
}
,
```

Customizing the text selection color

Use the following CSS to customize the selection color of text and background.

```
`css
.e-ddl.e-input-group input.e-input::selection {
color: red;
background: yellow;
}
,
```

Customizing the background color of focus, hover, and active item's

Use the following CSS to customize the background color of focus, hover and active item's.

```
`css
.e-dropdownbase .e-list-item.e-item-focus, .e-dropdownbase .e-list-item.e-active, .e-dropdownbase .e-
list-item.e-active.e-hover, .e-dropdownbase .e-list-item.e-hover {
background-color: #1f9c99;
color: #2319b8;
}
,
```

Customizing the appearance of pop-up element

Use the following CSS to customize the appearance of popup element.

```
`css
.e-dropdownbase .e-list-item, .e-dropdownbase .e-list-item.e-item-focus {
background-color: #29c2b8;
color: #207cd9;
font-family: emoji;
min-height: 29px;
}
,
```

Adding mandatory asterisk to placeholder and float label

You can add a mandatory asterisk(*) to placeholder and float label using `.e-input-group.e-control-wrapper.e-float-input .e-float-text::after` class.

[Class-component]

INDEX.JSX

```
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
```

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
  // define the array of data
  sportsData = ['Badminton', 'Basketball', 'Cricket', 'Football', 'Golf',
    'Gymnastics', 'Hockey', 'Rugby', 'Snooker', 'Tennis'];
  render() {
    return (
      // specifies the tag for render the AutoComplete component
      <AutoCompleteComponent id="atcelement" dataSource={this.sportsData}
placeholder="Find a game" floatLabelType="auto"/>);
    }
  }
ReactDOM.render(<App />, document.getElementById('sample'));
```

INDEX.TSX

```
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}, {}> {
  // define the array of data
  private sportsData: string[] = ['Badminton', 'Basketball', 'Cricket',
    'Football', 'Golf', 'Gymnastics', 'Hockey', 'Rugby', 'Snooker', 'Tennis'];
  public render() {
    return (
      // specifies the tag for render the AutoComplete component
      <AutoCompleteComponent id="atcelement" dataSource={this.sportsData}
placeholder="Find a game" floatLabelType= "auto"/>
    );
  }
}
ReactDOM.render(<App />, document.getElementById('sample'));
```

[Functional-component]**INDEX.JSX**

```
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
  // define the array of data
  const sportsData = ['Badminton', 'Basketball', 'Cricket', 'Football',
    'Golf', 'Gymnastics', 'Hockey', 'Rugby', 'Snooker', 'Tennis'];
  return (
    // specifies the tag for render the AutoComplete component
    <AutoCompleteComponent id="atcelement" dataSource={sportsData}
placeholder="Find a game" floatLabelType="auto"/>);
}
ReactDOM.render(<App />, document.getElementById('sample'));
```

INDEX.TSX

```
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
```



```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
  // define the array of data
  const sportsData: string[] = ['Badminton', 'Basketball', 'Cricket',
  'Football', 'Golf', 'Gymnastics', 'Hockey', 'Rugby', 'Snooker', 'Tennis'];
  return (
    // specifies the tag for render the AutoComplete component
    <AutoCompleteComponent id="atcelement" dataSource={sportsData}
    placeholder="Find a game" floatLabelType= "auto"/>
  );
}
ReactDOM.render(<App />, document.getElementById('sample'));
```

Accessibility in React Auto complete component

The AutoComplete component has been designed, keeping in mind the WAI-ARIA specifications, and applies the [WAI-ARIA](#) roles, states, and properties along with [keyboard support](#). This component is characterized by complete keyboard interaction support and ARIA accessibility support that makes it easy for people who use assistive technologies(AT) or those who completely rely on keyboard navigation.

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

| Property | Functionalities |

| --- | --- |

| aria-haspopup | Indicates whether the AutoComplete input element has a suggestion list or not. |

| aria-expanded | Indicates whether the suggestion list has expanded or not. |

| aria-selected | Indicates the selected option from the list. |

| aria-readonly | Indicates the readonly state of the AutoComplete element. |

| aria-disabled | Indicates whether the AutoComplete component is in a disabled state or not. |

| aria-activedescendent | This attribute holds the ID of the active list item to focus its descendant child element. |

| aria-owns | This attribute contains the ID of the suggestion list to indicate popup as a child element. |

| aria-autocomplete | This attribute contains the 'both' to a list of options shows and the currently selected suggestion also shows inline. |

Keyboard Interaction

You can use the following key shortcuts to access the AutoComplete without interruptions.

| Keyboard shortcuts | Actions |

| --- | --- |

| Arrow Down | In popup hidden state, opens the suggestion list. In popup open state, selects the first item when no item selected else selects the item next to the currently selected item. |

| Arrow Up | In popup hidden state, opens the suggestion list. In popup open state, selects the last item when no item selected else selects the item previous to the currently selected one. |

| Page Down | Scrolls down to the next page and selects the first item when popup list opens. |

| Page Up | Scrolls up to previous page and select the first item when popup list open. |

| Enter | Selects the focused item and set to AutoComplete component. |

| Tab | Focuses on the next tab indexed element when the popup is closed. Otherwise, closes the popup list and remains the focus in component suppose if it is in an open state. |

| Shift + tab | Focuses the previous tab indexed element when the popup is closed. Otherwise, closes the popup list and remains the focus in component suppose if it is in an open state. |

| Alt + Down | Opens the popup list. |

| Alt + Up | In popup hidden state, opens the popup list. In popup open state, closes the popup list. |

| Esc(Escape) | Closes the popup list when it is in an open state then remove the selection. |

| Home | Cursor moves to before of first character in input. |

| End | Cursor moves to next of last character in input. |

In the below sample, focus the AutoComplete component using alt+t keys.

[Class-component]

INDEX.JSX

```
{% raw %}
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
  // defined the array of data
  gameList = [
    { Id: 'Game1', Game: 'Badminton' },
    { Id: 'Game2', Game: 'Basketball' },
    { Id: 'Game3', Game: 'Cricket' },
    { Id: 'Game4', Game: 'Football' },
    { Id: 'Game5', Game: 'Golf' },
    { Id: 'Game6', Game: 'Hockey' },
    { Id: 'Game7', Game: 'Rugby' },
    { Id: 'Game8', Game: 'Snooker' }
  ];
  // maps the appropriate column to fields property
  fields = { value: 'Game' };
  // instance of AutoComplete component
  AutoCompleteObj;
  componentDidMount() {
    const proxy = this;
    document.onkeyup = (e) => {
      if (e.altKey && e.keyCode === 84 /* t */) {
        // press alt+t to focus the control.
        proxy.AutoCompleteObj.inputElement.focus();
      }
    };
  }
  render() {
```

```

        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="atcelement" ref={(scope) => {
this.AutoCompleteObj = scope; }} fields={this.fields}
dataSource={this.gameList} placeholder="Find a game"/>);
        }
    }
    ReactDOM.render(<App />, document.getElementById('sample'));
    {% endraw %}

```

INDEX.TSX

```

{% raw %}
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}, {}> {
    // defined the array of data
    private gameList: { [key: string]: Object }[] = [
        { Id: 'Game1', Game: 'Badminton' },
        { Id: 'Game2', Game: 'Basketball' },
        { Id: 'Game3', Game: 'Cricket' },
        { Id: 'Game4', Game: 'Football' },
        { Id: 'Game5', Game: 'Golf' },
        { Id: 'Game6', Game: 'Hockey' },
        { Id: 'Game7', Game: 'Rugby' },
        { Id: 'Game8', Game: 'Snooker' }
    ];
    // maps the appropriate column to fields property
    private fields: object = { value: 'Game' };
    // instance of AutoComplete component
    private AutoCompleteObj: AutoCompleteComponent;
    public componentDidMount() {
        const proxy = this;
        document.onkeyup = (e) => {
            if (e.altKey && e.keyCode === 84 /* t */) {
                // press alt+t to focus the control.
                (proxy.AutoCompleteObj as any).inputElement.focus();
            }
        };
    }
    public render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="atcelement" ref={(scope) => {
(this.AutoCompleteObj as AutoCompleteComponent | null) = scope; }}
fields={this.fields} dataSource={this.gameList} placeholder="Find a game" />
        );
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));
{% endraw %}

```

[Functional-component]

INDEX.JSX

```
{% raw %}
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    // defined the array of data
    const gameList = [
        { Id: 'Game1', Game: 'Badminton' },
        { Id: 'Game2', Game: 'Basketball' },
        { Id: 'Game3', Game: 'Cricket' },
        { Id: 'Game4', Game: 'Football' },
        { Id: 'Game5', Game: 'Golf' },
        { Id: 'Game6', Game: 'Hockey' },
        { Id: 'Game7', Game: 'Rugby' },
        { Id: 'Game8', Game: 'Snooker' }
    ];
    // maps the appropriate column to fields property
    const fields = { value: 'Game' };
    // instance of AutoComplete component
    let AutoCompleteObj;
    React.useEffect(() => {
        const proxy = this;
        document.onkeyup = (e) => {
            if (e.altKey && e.keyCode === 84 /* t */) {
                // press alt+t to focus the control.
                proxy.AutoCompleteObj.inputElement.focus();
            }
        };
    }, []);
    return (
        // specifies the tag for render the AutoComplete component
        <AutoCompleteComponent id="atcelement" ref={(scope) => { AutoCompleteObj = scope; }} fields={fields} dataSource={gameList} placeholder="Find a game"/>
    );
}
ReactDOM.render(<App />, document.getElementById('sample'));
{% endraw %}
```

INDEX.TSX

```
{% raw %}
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    // defined the array of data
    const gameList: { [key: string]: Object }[] = [
        { Id: 'Game1', Game: 'Badminton' },
        { Id: 'Game2', Game: 'Basketball' },
        { Id: 'Game3', Game: 'Cricket' },
        { Id: 'Game4', Game: 'Football' },
        { Id: 'Game5', Game: 'Golf' },
        { Id: 'Game6', Game: 'Hockey' },
        { Id: 'Game7', Game: 'Rugby' },
    ];

```

```

    { Id: 'Game8', Game: 'Snooker' }
  ];
  // maps the appropriate column to fields property
  const fields: object = { value: 'Game' };
  // instance of AutoComplete component
  let AutoCompleteObj: AutoCompleteComponent;
  React.useEffect(() => {
    const proxy = this;
    document.onkeyup = (e) => {
      if (e.altKey && e.keyCode === 84 /* t */) {
        // press alt+t to focus the control.
        (proxy.AutoCompleteObj as any).inputElement.focus();
      }
    };
  }, []);
  return (
    // specifies the tag for render the AutoComplete component
    <AutoCompleteComponent id="atcelement" ref={ (scope) => {
      (AutoCompleteObj as AutoCompleteComponent | null) = scope; } }
      fields={fields} dataSource={gameList} placeholder="Find a game" />
  );
}
ReactDOM.render(<App />, document.getElementById('sample'));
{% enddraw %}

```

Ensuring accessibility

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

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

See also

- [Accessibility in Syncfusion React components](#)

How To

Autofill in React Auto complete component

The React AutoComplete supports the autofill behavior with the help of [autofill](#) property. Whenever you change the input value and press the down key, the AutoComplete will autocomplete your data by matching the typed character. Suppose, if no matches found then, AutoComplete doesn't suggest any item.

In the below sample, showcase that how to work `autofill` with AutoComplete.

INDEX.JSX

```

import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
  // define the JSON of data
  sportsData = [

```

```

        { Id: 'Game1', Game: 'Badminton' },
        { Id: 'Game2', Game: 'Basketball' },
        { Id: 'Game3', Game: 'Cricket' },
        { Id: 'Game4', Game: 'Football' },
        { Id: 'Game5', Game: 'Golf' },
        { Id: 'Game6', Game: 'Hockey' },
        { Id: 'Game7', Game: 'Rugby' },
        { Id: 'Game8', Game: 'Snooker' }
    ];
    // maps the appropriate column to fields property
    fields = { value: 'Game' };
    render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="atcelement" dataSource={this.sportsData}
fields={this.fields} placeholder="Find a game" autofill={true}/>);
        }
    }
    ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.TSX

```

import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}, {}> {
    // define the JSON of data
    private sportsData: { [key: string]: Object }[] = [
        { Id: 'Game1', Game: 'Badminton' },
        { Id: 'Game2', Game: 'Basketball' },
        { Id: 'Game3', Game: 'Cricket' },
        { Id: 'Game4', Game: 'Football' },
        { Id: 'Game5', Game: 'Golf' },
        { Id: 'Game6', Game: 'Hockey' },
        { Id: 'Game7', Game: 'Rugby' },
        { Id: 'Game8', Game: 'Snooker' }
    ];
    // maps the appropriate column to fields property
    private fields: object = { value: 'Game' };
    public render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="atcelement"
dataSource={this.sportsData} fields={this.fields} placeholder="Find a game"
autofill={true} />
        );
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));

```

Icon support in React Auto complete component

You can render **icons** to the list items by mapping the [iconCss](#) field. This `iconCss` field create a span in the list item with mapped class name to allow styling as per your need.

In the following sample, icon classes are mapped with `iconCss` field.

INDEX.JSX

```
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
    // define the array of data
    sortFormatData = [
        { Class: 'asc-sort', Type: 'Sort A to Z', Id: '1' },
        { Class: 'dsc-sort', Type: 'Sort Z to A ', Id: '2' },
        { Class: 'filter', Type: 'Filter', Id: '3' },
        { Class: 'clear', Type: 'Clear', Id: '4' }
    ];
    // map the icon column to iconCSS field.
    fields = { value: 'Type', iconCss: 'Class' };
    render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="" dataSource={this.sortFormatData}
            fields={this.fields} placeholder="Find a format"/>);
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));
```

INDEX.TSX

```
import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}, {}> {
    // define the array of data
    private sortFormatData: { [key: string]: Object }[] = [
        { Class: 'asc-sort', Type: 'Sort A to Z', Id: '1' },
        { Class: 'dsc-sort', Type: 'Sort Z to A ', Id: '2' },
        { Class: 'filter', Type: 'Filter', Id: '3' },
        { Class: 'clear', Type: 'Clear', Id: '4' }
    ];
    // map the icon column to iconCSS field.
    private fields: object = { value: 'Type', iconCss: 'Class' };
    public render() {
        return (
            // specifies the tag for render the AutoComplete component
            <AutoCompleteComponent id="" dataSource={this.sortFormatData}
            fields={this.fields} placeholder="Find a format" />
        );
    }
}
ReactDOM.render(<App />, document.getElementById('sample'));
```

Custom search in React Auto complete component

The AutoComplete has built-in support to highlight the searched characters on suggested list items when enabled the [highlight](#) property.

In the below sample, to customize the matched character in suggestion list by `e-highlight` class.

INDEX.JSX

```

import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component {
  // define the JSON of data
  sportsData = [
    { Id: 'Game1', Game: 'Badminton' },
    { Id: 'Game2', Game: 'Basketball' },
    { Id: 'Game3', Game: 'Cricket' },
    { Id: 'Game4', Game: 'Football' },
    { Id: 'Game5', Game: 'Golf' },
    { Id: 'Game6', Game: 'Hockey' },
    { Id: 'Game7', Game: 'Rugby' },
    { Id: 'Game8', Game: 'Snooker' }
  ];
  // maps the appropriate column to fields property
  fields = { value: 'Game' };
  render() {
    return (
      // specifies the tag for render the AutoComplete component
      <AutoCompleteComponent id="atcelement" dataSource={this.sportsData}
fields={this.fields} placeholder="Find a game" highlight={true}/>);
    }
  }
ReactDOM.render(<App />, document.getElementById('sample'));

```

INDEX.TSX

```

import { AutoCompleteComponent } from '@syncfusion/ej2-react-dropdowns';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
export default class App extends React.Component<{}, {}> {
  // define the JSON of data
  private sportsData: { [key: string]: Object }[] = [
    { Id: 'Game1', Game: 'Badminton' },
    { Id: 'Game2', Game: 'Basketball' },
    { Id: 'Game3', Game: 'Cricket' },
    { Id: 'Game4', Game: 'Football' },
    { Id: 'Game5', Game: 'Golf' },
    { Id: 'Game6', Game: 'Hockey' },
    { Id: 'Game7', Game: 'Rugby' },
    { Id: 'Game8', Game: 'Snooker' }
  ];
  // maps the appropriate column to fields property
  private fields: object = { value: 'Game' };
  public render() {
    return (
      // specifies the tag for render the AutoComplete component
      <AutoCompleteComponent id="atcelement"
dataSource={this.sportsData} fields={this.fields} placeholder="Find a game"
highlight={true} />
    );
  }
}

```

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

Ej1 api migration in React Auto complete component

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

MultiSelect concept is not present in EJ2-AutoComplete. If you want to use multiselection support in autocomplete, we suggest you to use MultiSelect component.

DataBinding

{% raw %}

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

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

| **Default** | **Property:** *datasource*
<EJ.AutoComplete id="default" dataSource = {carList}></EJ.AutoComplete> | **Property:** *dataSource*
<AutoCompleteComponent id="atcelement" ref={(scope) => { this.cmbObj = scope; }} dataSource={this.cityData} /> |

| **Fields for mapping** | **Property:** *fields*
<EJ.AutoComplete id="default" fields = {fields}></EJ.AutoComplete> | **Property:** *fields*
<AutoCompleteComponent id="atcelement" ref={(scope) => { this.cmbObj = scope; }} fields={this.fields} /> |

| **Query** | **Property:** *query*
<EJ.AutoComplete id="default" query = {query}></EJ.AutoComplete> | **Property:** *query*
<AutoCompleteComponent id="atcelement" ref={(scope) => { this.cmbObj = scope; }} query={this.query} /> |

| **Begin event** | **Event:** *actionBegin*
<EJ.AutoComplete id="default" actionBegin = {actionBegin}></EJ.AutoComplete> | **Event:** *actionBegin*
<AutoCompleteComponent id="atcelement" ref={(scope) => { this.cmbObj = scope; }} actionBegin={this.actionBegin.bind(this)} /> |

| **Complete event** | **Event:** *actionComplete*
<EJ.AutoComplete id="default" actionComplete = {actionComplete}></EJ.AutoComplete> | **Event:** *actionComplete*
<AutoCompleteComponent id="atcelement" ref={(scope) => { this.cmbObj = scope; }} actionComplete={this.actionComplete.bind(this)} /> |

| **Failure event** | **Event:** *actionFailure*
<EJ.AutoComplete id="default" actionFailure = {actionFailure}></EJ.AutoComplete> | **Event:** *actionFailure*
<AutoCompleteComponent id="atcelement" ref={(scope) => { this.cmbObj = scope; }} actionFailure={this.actionFailure.bind(this)} /> |

| **Success event** | **Event:** *actionSuccess*
<EJ.AutoComplete id="default" actionSuccess = {actionSuccess}></EJ.AutoComplete> | **Not Applicable** |

Filtering

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

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

| **Case sensitivity** | **Property:** *caseSensitiveSearch*
<EJ.AutoComplete id="default" caseSensitiveSearch = {true}></EJ.AutoComplete> | **Property:**

ignoreCase
<AutoCompleteComponent id="atcelement" ref={{scope}} => { this.cmbObj = scope; }} ignoreCase={true} />|

| **Accent effective search** | **Not applicable** | **Property:** *ignoreAccent*

<AutoCompleteComponent id="atcelement" ref={{scope}} => { this.cmbObj = scope; }} ignoreAccent={true} />|

| **Filtering Type** | **Property:** *filterType*
<EJ.AutoComplete id="default" filterType = {filterType} ></EJ.AutoComplete>| **Property:** *filterType*
<AutoCompleteComponent id="atcelement" ref={{scope}} => { this.cmbObj = scope; }} filterType={this.filterType} /> |

| **Autofill** | **Property:** *enableAutoFill*
<EJ.AutoComplete id="default" enableAutoFill = {true} ></EJ.AutoComplete>| **Property:** *autoFill*
<AutoCompleteComponent id="atcelement" ref={{scope}} => { this.cmbObj = scope; }} autoFill={true} />|

| **Highlight the search word** | **Property:** *highlightSearch*
<EJ.AutoComplete id="default" highlightSearch = {true} ></EJ.AutoComplete>| **Property:** *highlight*
<AutoCompleteComponent id="atcelement" ref={{scope}} => { this.cmbObj = scope; }} highlight={true} />|

| **No of items to be shown** | **Property:** *itemsCount*
<EJ.AutoComplete id="default" itemsCount = {itemsCount} ></EJ.AutoComplete>| **Property:** *suggestionCount*
<AutoCompleteComponent id="atcelement" ref={{scope}} => { this.cmbObj = scope; }} suggestionCount={this.suggestionCount} /> |

| **Minimum characters to enter** | **Property:** *minCharacter*
<EJ.AutoComplete id="default" minCharacter = {minCharacter} ></EJ.AutoComplete>| **Property:** *minLength*
<AutoCompleteComponent id="atcelement" ref={{scope}} => { this.cmbObj = scope; }} minLength={this.minLength} /> |

| **Search** | **Method:** *search*
<EJ.AutoComplete id="autocomplete" ></EJ.AutoComplete>

 \$(("#autocomplete").ejAutocomplete("search"));| **Not applicable** |

Placeholder

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

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

| **Watermark text** | **Property:** *watermarkText*
<EJ.AutoComplete id="default" watermarkText = {watermarkText} ></EJ.AutoComplete>| **Property:** *placeholder*
<AutoCompleteComponent id="atcelement" ref={{scope}} => { this.cmbObj = scope; }} placeholder={this.placeholder} />|

| **Floating of watermark text** | **Not applicable** | **Property:** *floatLabelType*
<AutoCompleteComponent id="atcelement" ref={{scope}} => { this.cmbObj = scope; }} floatLabelType={this.floatLabelType} />|

Popup

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

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

| **No records text** | **Property:** *emptyResultText*
 <EJ.AutoComplete id="default"
emptyResultText = {emptyResultText} ></EJ.AutoComplete> | **Property:** *noRecordsTemplate*

<AutoCompleteComponent id="atcelement" ref={{(scope) => { this.cmbObj = scope; }}
noRecordsTemplate={this.noRecordsTemplate} /> |

| **No records showing** | **Property:** *showEmptyResultText*
 <EJ.AutoComplete id="default"
showEmptyResultText = {true} ></EJ.AutoComplete> | **Not applicable** |

| **Popupbutton** | **Property:** *showPopupButton*
 <EJ.AutoComplete id="default"
showPopupButton = {showPopupButton} ></EJ.AutoComplete> | **Property:**
showPopupButton
 <AutoCompleteComponent id="atcelement" ref={{(scope) => {
this.cmbObj = scope; }} showPopupButton={this.showPopupButton} /> |

| **Clear button** | **Property:** *showResetIcon*
 <EJ.AutoComplete id="default" showResetIcon =
{showResetIcon} ></EJ.AutoComplete> | **Property:** *showClearButton*

<AutoCompleteComponent id="atcelement" ref={{(scope) => { this.cmbObj = scope; }}
showClearButton={this.showClearButton} /> |

| **Animation** | **Property:** *animateType*
 <EJ.AutoComplete id="default" animateType =
{animateType} ></EJ.AutoComplete> | **Not Applicable** |

| **Focusing the list item** | **Property:** *autoFocus*
 <EJ.AutoComplete id="default" autoFocus =
{true} ></EJ.AutoComplete> | **Not applicable** |

| **Delaying the popup open time** | **Property:** *delaySuggestionTimeout*
 <EJ.AutoComplete
id="default" delaySuggestionTimeout = {delaySuggestionTimeout} ></EJ.AutoComplete> | **Not
applicable** |

| **Popup text when there is no popup items** | **Property:** *emptyResultText*
 <EJ.AutoComplete
id="default" emptyResultText = {emptyResultText} ></EJ.AutoComplete>
[| https://ej2.syncfusion.com/react/demos/#/material/auto-complete/template |](https://ej2.syncfusion.com/react/demos/#/material/auto-complete/template)

| **Enable/disable the duplicate option** | **Property:** *enableDistinct*
 <EJ.AutoComplete
id="default" enableDistinct = {true} ></EJ.AutoComplete> | **Not applicable** |

| **Popup height** | **Property:** *popupHeight*
 <EJ.AutoComplete id="default" popupHeight =
{popupHeight} ></EJ.AutoComplete> | **Property:** *popupHeight*
 <AutoCompleteComponent
id="atcelement" ref={{(scope) => { this.cmbObj = scope; }} popupHeight={this.popupHeight} /> |

| **Popup Width** | **Property:** *popupWidth*
 <EJ.AutoComplete id="default" popupWidth =
{popupWidth} ></EJ.AutoComplete> | **Property:** *popupWidth*
 <AutoCompleteComponent
id="atcelement" ref={{(scope) => { this.cmbObj = scope; }} popupWidth={this.popupWidth} /> |

| **Open popup** | **Method:** *open*
<EJ.AutoComplete id="autocomplete"
></EJ.AutoComplete>
 \$(("#autocomplete").ejAutocomplete("open")); | **Method:**
showPopup
<AutoCompleteComponent id="atcelement" ref={{(scope) => { this.cmbObj =
scope; }} />

this.cmbObj.showPopup(); |

| **Close event** | **Event:** *close*
<EJ.AutoComplete id="autocomplete" close = "close"
></EJ.AutoComplete> | **Event:** *close*
 <AutoCompleteComponent id="atcelement"
ref={{(scope) => { this.cmbObj = scope; }} close={this.close.bind(this)} /> |

| **Open event** | **Event:** *open*
 <EJ.AutoComplete id="autocomplete" open="open"></EJ.AutoComplete> | **Event:** *open*
 <AutoCompleteComponent id="atcelement" ref={scope} => { this.cmbObj = scope; } } open={this.open.bind(this)} /> |

CSS

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

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

| **Default** | **Property:** *cssClass*
 <EJ.AutoComplete id="autocomplete" cssClass="cssClass"></EJ.AutoComplete> | **Property:** *cssClass*
 <AutoCompleteComponent id="atcelement" ref={scope} => { this.cmbObj = scope; } } cssClass={this.cssClass} /> |

| **Height** | **Property:** *height*
 <EJ.AutoComplete id="autocomplete" height="400px"></EJ.AutoComplete> | Achievable through the [cssClass](#) property. |

| **showRoundedCorner** | **Property:** *showRoundedCorner*
 <EJ.AutoComplete id="autocomplete" showRoundedCorner={true}></EJ.AutoComplete> | Achievable through the [cssClass](#) property. |

| **Width** | **Property:** *width*
 <EJ.AutoComplete id="autocomplete" width="300px"></EJ.AutoComplete> | **Property:** *width*
 <AutoCompleteComponent id="atcelement" ref={scope} => { this.cmbObj = scope; } } width={this.width} /> |

| **Visibility** | **Property:** *visible*
 <EJ.AutoComplete id="autocomplete" visible={true}></EJ.AutoComplete> | Achievable through the [cssClass](#) property. |

Grouping

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

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

| **Default** | **Property:** *fields*
 <EJ.AutoComplete id="default" fields = {fields}></EJ.AutoComplete> | **Property:** *fields*
 <AutoCompleteComponent id="atcelement" ref={scope} => { this.cmbObj = scope; } } fields={this.fields} /> |

Localization

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

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

| **Default** | **Property:** *Locale*
 <EJ.AutoComplete id="default" locale = {"fr-FE"}></EJ.AutoComplete> | **Property:** *Locale*
 <AutoCompleteComponent id="atcelement" ref={scope} => { this.cmbObj = scope; } } locale={this.locale} /> |

Template

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

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

| **Default** | **Property:** *template*
 <EJ.AutoComplete id="default" template = {template}></EJ.AutoComplete> | **Property:** *itemTemplate*
 <AutoCompleteComponent id="atcelement" ref={scope} => { this.cmbObj = scope; } } itemTemplate={this.itemTemplate} /> |

| **Group Template** | **Not Applicable** | **Property:** *groupTemplate*
<AutoCompleteComponent id="atcelement" ref={{(scope) => { this.cmbObj = scope; }}} groupTemplate={this.groupTemplate} />|

| **ValueTemplate** | **Not applicable** | **Property:** *valueTemplate*
<AutoCompleteComponent id="atcelement" ref={{(scope) => { this.cmbObj = scope; }}} valueTemplate={this.valueTemplate} />|

| **Header Template** | **Not applicable** | **Property:** *headerTemplate*
<AutoCompleteComponent id="atcelement" ref={{(scope) => { this.cmbObj = scope; }}} headerTemplate={this.headerTemplate} />|

| **FooterTemplate** | **Not applicable** | **Property:** *footerTemplate*
<AutoCompleteComponent id="atcelement" ref={{(scope) => { this.cmbObj = scope; }}} footerTemplate={this.footerTemplate} /> |

| **No records Template** | **Not applicable** | **Property:** *noRecordsTemplate*
<AutoCompleteComponent id="atcelement" ref={{(scope) => { this.cmbObj = scope; }}} noRecordsTemplate={this.noRecordsTemplate} /> |

| **Action failure Template** | **Not applicable** | **Property:** *actionFailureTemplate*
<AutoCompleteComponent id="atcelement" ref={{(scope) => { this.cmbObj = scope; }}} actionFailureTemplate={this.actionFailureTemplate} /> |

Sorting

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

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

| **Default** | **Property:** *allowSorting*
<EJ.AutoComplete id="default" allowSorting = {true} ></EJ.AutoComplete> | Achievable through [sortOrder](#) property |

| **Order of sorting** | **Property:** *sortOrder*
<EJ.AutoComplete id="default" sortOrder = {sortOrder} ></EJ.AutoComplete>| **Property:** *sortOrder*
<AutoCompleteComponent id="atcelement" ref={{(scope) => { this.cmbObj = scope; }}} sortOrder={this.sortOrder} /> |

Accessibility

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

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

| **RTL support** | **Property:** *enableRtl*
<EJ.AutoComplete id="default" enableRtl = {true} ></EJ.AutoComplete> | **Property:** *enableRtl*
<AutoCompleteComponent id="atcelement" ref={{(scope) => { this.cmbObj = scope; }}} enableRtl={this.enableRtl} />|

Selection

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

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

| **Selecting particular value** | **Property:** *selectValueByKey*
<EJ.AutoComplete id="default" selectValueByKey = {selectValueByKey} ></EJ.AutoComplete>| Achievable through [value](#) property. |

| **Selecting particular value** | **Property:** *value*
<EJ.AutoComplete id="default" value = {value}
></EJ.AutoComplete> | **Property:** *value*
<AutoCompleteComponent id="atcelement"
ref={{(scope) => { this.cmbObj = scope; }} value={this.value} /> |

| **Selecting particular text** | **Property:** *text*
<EJ.AutoComplete id="default" text = {text}
></EJ.AutoComplete> | **Not Applicable** |

| **Selecting particular value** | **Method:** *selectValueByKey*
<EJ.AutoComplete id="autocomplete"
enableRtl = {true} ></EJ.AutoComplete>
 \$(("#autocomplete").selectValueByKey("key")) |
Achievable through [value](#) property. |

| **Selecting particular text** | **Method:** *selectValueByText*
<EJ.AutoComplete id="autocomplete"
enableRtl = {true} ></EJ.AutoComplete>
 \$(("#autocomplete").selectValueByText("key")) | **Not applicable** |

| **Select event** | **Event:** *select*
<EJ.AutoComplete id="default" select = "select"
></EJ.AutoComplete> | **Event:** *select*
<AutoCompleteComponent id="atcelement"
ref={{(scope) => { this.cmbObj = scope; }} select={this.select.bind(this)} /> |

Miscellaneous

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

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

| **Enable/disable** | **Property:** *enabled*
<EJ.AutoComplete id="default" enabled = {true}
></EJ.AutoComplete> | **Property:** *enabled*
<AutoCompleteComponent id="atcelement"
ref={{(scope) => { this.cmbObj = scope; }} enabled={true} /> |

| **Enable persistence** | **Property:** *enablePersistence*
<EJ.AutoComplete id="default"
enablePersistence = {true} ></EJ.AutoComplete> | **Property:** *enablePersistence*
<AutoCompleteComponent id="atcelement" ref={{(scope) => { this.cmbObj = scope; }}
enablePersistence={true} /> |

| **Loading icon** | **Property:** *showLoadingIcon*
<EJ.AutoComplete id="default" showLoadingIcon
= {true} ></EJ.AutoComplete> | **By default, it is showing** |

| **Read only** | **Property:** *readOnly*
<EJ.AutoComplete id="default" readOnly = {true}
></EJ.AutoComplete> | **Property:** *readOnly*
<AutoCompleteComponent id="atcelement"
ref={{(scope) => { this.cmbObj = scope; }} readOnly={this.readOnly} /> |

| **Disable** | **Method:** *disable*
<EJ.AutoComplete
id="autocomplete"></EJ.AutoComplete>

\$(("#autocomplete").ejAutoComplete("disable")); | Achievable through [enabled](#) property. |

Common

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

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

| **Addition of new option watermark text** | **Property:** *addNewText*
<EJ.AutoComplete
id="default" addNewText = {addNewText} ></EJ.AutoComplete> | **Not applicable** |

| **Addition of new item** | **Property:** `allowAddNew` `
<EJ.AutoComplete id="default"`
`allowAddNew = {true} ></EJ.AutoComplete>` | **Property:** `allowCustom` `
`
`<AutoCompleteComponent id="atcelement" ref={{scope}} => { this.cmbObj = scope; }}`
`allowCustom={true} />` |

| **Reset the autocomplete** | **Property:** `showResetIcon` `
<EJ.AutoComplete id="default"`
`showResetIcon = {true} ></EJ.AutoComplete>` | **Property:** `showClearIcon` `
`
`<AutoCompleteComponent id="atcelement" ref={{scope}} => { this.cmbObj = scope; }}`
`showClearIcon={true} />` |

| **Destroy** | **Method:** `destroy` `
<EJ.AutoComplete id="autocomplete"`
`></EJ.AutoComplete>
 $("#autocomplete").ejAutoComplete("destroy");` | **Method:** `destroy`
`
<AutoCompleteComponent id="atcelement" ref={{scope}} => { this.cmbObj = scope; }}`
`/>

this.cmbObj.destroy();` |

| **Reset the autocomplete** | **Method:** `clearText` `
<EJ.AutoComplete id="autocomplete"`
`></EJ.AutoComplete>
 $("#autocomplete").ejAutoComplete("clearText");` | **Property:** `value`
`
<AutoCompleteComponent id="atcelement" ref={{scope}} => { this.cmbObj = scope; }}`
`value="" />` |

| **Multicolumn** | **Property:** `multiColumnSettings` `
 var autocompleteInstance = new`
`ej.AutoComplete($("#selectCar"),`
`{multiColumnSettings:{enable:true,showHeader:true,stringFormat:"{1}",searchColumnIndices[0`
`,1,2], columns:[{"field": "EmployeeID", "headerText": "EmployeeID"}, {"field": "FirstName",`
`"headerText": "FirstName"}, {"field": "City", "headerText": "City"}]}];` | **Not applicable** |

| **Hide the Autocomplete** | **Method:** `hide` `
<EJ.AutoComplete id="autocomplete"`
`></EJ.AutoComplete>
 $("#autocomplete").ejAutoComplete("hide");` | Achievable through
the [cssClass](#) property. |

| **Getting particular text** | **Method:** `getActiveText` `
<EJ.AutoComplete id="autocomplete"`
`></EJ.AutoComplete>
 $("#autocomplete").ejAutoComplete("getActiveText");` | **Not**
applicable |

| **Getting particular value** | **Method:** `getValue` `
<EJ.AutoComplete id="autocomplete"`
`></EJ.AutoComplete>
 $("#autocomplete").ejAutoComplete("getValue");` | Achievable
through [value](#) property. |

| **Change event** | **Event:** `change` `
<EJ.AutoComplete id="default" change = "change"`
`></EJ.AutoComplete>` | **Event:** `change` `
<AutoCompleteComponent id="atcelement"`
`ref={{scope}} => { this.cmbObj = scope; }}` `change={this.change.bind(this)} />` |

| **Create event** | **Event:** `create` `
<EJ.AutoComplete id="default" create = "create"`
`></EJ.AutoComplete>` | **Event:** `created` `
<AutoCompleteComponent id="atcelement"`
`ref={{scope}} => { this.cmbObj = scope; }}` `created={this.created.bind(this)} />` |

| **Destroy event** | **Event:** `destroy` `
<EJ.AutoComplete id="default" destroy = "destroy"`
`></EJ.AutoComplete>` | **Event:** `destroyed` `
<AutoCompleteComponent id="atcelement"`
`ref={{scope}} => { this.cmbObj = scope; }}` `destroyed={this.destroyed.bind(this)} />` |


```
| Focus out event | Event: focusOut<br/><EJ.AutoComplete id="default" focusOut = "focusOut"
></EJ.AutoComplete>| Event: blur <br/><AutoCompleteComponent id="atcelement"
ref={ (scope) => { this.cmbObj = scope; }} blur={this.blur.bind(this)} /> |
| Focus in event | Event : focusIn<br/><EJ.AutoComplete id="default" focusIn = "focusIn"
></EJ.AutoComplete> | Event: focus <br/><AutoCompleteComponent id="atcelement"
ref={ (scope) => { this.cmbObj = scope; }} focus={this.focus.bind(this)} /> |
{% endraw %}
```

Avatar

Getting Started

This section explains how to create a simple **Avatar** using Styles in the React Framework

Dependencies

The Avatar Component is pure CSS component so no specific dependencies to render the avatar.

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.

To install **Avatar** component, use the following command

```
`bash
```

```
npm install @syncfusion/ej2-layouts --save
```

Adding CSS Reference

The Avatar CSS files are available in the `ej2-layouts` 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-layouts/styles/material.css';
```

Adding a simple Avatar

Add the HTML `div` element with `e-avatar` class into your `App.tsx`.

```
`bash
<div className="e-avatar">AJ</div>
```

Run the application

Run the application in the browser using the following command:

```
`bash
npm start
```

Output will be as follows:

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
function ReactApp() {
    return (<div>
        <span className="e-avatar e-avatar-xlarge"></span>
        <span className="e-avatar e-avatar-large"></span>
        <span className="e-avatar"></span>
        <span className="e-avatar e-avatar-small"></span>
        <span className="e-avatar e-avatar-xsmall"></span>
    </div>);
}
export default ReactApp;
ReactDOM.render(<ReactApp />, document.getElementById("element"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
function ReactApp() {
    return (
```

```

    <div>
      <span className="e-avatar e-avatar-xlarge"></span>
      <span className="e-avatar e-avatar-large"></span>
      <span className="e-avatar"></span>
      <span className="e-avatar e-avatar-small"></span>
      <span className="e-avatar e-avatar-xsmall"></span>
    </div>
  );
}
export default ReactApp;
ReactDOM.render(<ReactApp />, document.getElementById("element"));

```

See Also

[Types of Avatar](#)

Types in React Avatar component

This section explains different types of avatar.

Avatar size

The Essential JS 2 Avatar has the following predefined sizes that can be used with the `.e-avatar` class to change the appearance of the avatar.

Class Name	Description
:-----:	:-----:
e-avatar-xlarge	Displays xlarge size avatar.
e-avatar-large	Displays apply large size avatar.
e-avatar	Displays apply default size avatar.
e-avatar-small	Displays apply small size avatar.
e-avatar-xsmall	Displays apply xsmall size avatar.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function ReactApp() {
  return (
    <div>
      <span className="e-avatar e-avatar-xlarge"></span>
      <span className="e-avatar e-avatar-large"></span>
      <span className="e-avatar"></span>
      <span className="e-avatar e-avatar-small"></span>
      <span className="e-avatar e-avatar-xsmall"></span>
    </div>
  );
}
export default ReactApp;
ReactDOM.render(<ReactApp />, document.getElementById("element"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function ReactApp() {

```

```

return (
  <div>
    <span className="e-avatar e-avatar-xlarge"></span>
    <span className="e-avatar e-avatar-large"></span>
    <span className="e-avatar"></span>
    <span className="e-avatar e-avatar-small"></span>
    <span className="e-avatar e-avatar-xsmall"></span>
  </div>
);
}
export default ReactApp;
ReactDOM.render(<ReactApp/>, document.getElementById("element"));

```

Avatar types

The types of Essential JS 2 avatar are:

- Default
- Circle

Default

The default style of the avatar is rectangular shape with rounded corners, which can be applied from adding the modifier

class `.e-avatar` to the target element.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function ReactApp() {
  return (
    <div>
      <span className="e-avatar e-avatar-xlarge">RT</span>
      <span className="e-avatar e-avatar-large">RT</span>
      <span className="e-avatar">RT</span>
      <span className="e-avatar e-avatar-small">RT</span>
      <span className="e-avatar e-avatar-xsmall">RT</span>
    </div>
  );
}
export default ReactApp;
ReactDOM.render(<ReactApp />, document.getElementById("element"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function ReactApp() {
  return (
    <div>
      <span className="e-avatar e-avatar-xlarge">RT</span>
      <span className="e-avatar e-avatar-large">RT</span>
      <span className="e-avatar">RT</span>
      <span className="e-avatar e-avatar-small">RT</span>
      <span className="e-avatar e-avatar-xsmall">RT</span>
    </div>
  );
}

```

```
);
}
export default ReactApp;
ReactDOM.render(<ReactApp/>, document.getElementById("element"));
```

Circle

The circle avatar style can be applied by adding the modifier class `e-avatar-circle` to the target element.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
function ReactApp() {
  return (
    <div>
      <span className="e-avatar e-avatar-xlarge e-avatar-circle">SJ</span>
      <span className="e-avatar e-avatar-large e-avatar-circle">SJ</span>
      <span className="e-avatar e-avatar-circle">SJ</span>
      <span className="e-avatar e-avatar-small e-avatar-circle">SJ</span>
      <span className="e-avatar e-avatar-xsmall e-avatar-circle">SJ</span>
    </div>
  );
}
export default ReactApp;
ReactDOM.render(<ReactApp />, document.getElementById("element"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
function ReactApp() {
  return (
    <div>
      <span className="e-avatar e-avatar-xlarge e-avatar-circle">SJ</span>
      <span className="e-avatar e-avatar-large e-avatar-circle">SJ</span>
      <span className="e-avatar e-avatar-circle">SJ</span>
      <span className="e-avatar e-avatar-small e-avatar-circle">SJ</span>
      <span className="e-avatar e-avatar-xsmall e-avatar-circle">SJ</span>
    </div>
  );
}
export default ReactApp;
ReactDOM.render(<ReactApp/>, document.getElementById("element"));
```

How To

Avatar customization in React Avatar component

Colour customization

The avatar comes with default background colour (grey). This can be easily customized to the desired colour by adding custom class or directly selecting the avatar class from the CSS.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
function ReactApp() {
```

```

    return (<div>
      <span className="e-avatar e-avatar-xlarge e-avatar-circle
green">AJ</span>
      <span className="e-avatar e-avatar-xlarge e-avatar-circle
violet">JK</span>
      <span className="e-avatar e-avatar-xlarge e-avatar-circle
rose">EL</span>
      <span className="e-avatar e-avatar-xlarge e-avatar-circle
blue">SR</span>
      <span className="e-avatar e-avatar-xlarge e-avatar-circle
red">PD</span>
    </div>);
  }
export default ReactApp;
ReactDOM.render(<ReactApp />, document.getElementById("element"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function ReactApp() {
  return (
    <div>
      <span className="e-avatar e-avatar-xlarge e-avatar-circle
green">AJ</span>
      <span className="e-avatar e-avatar-xlarge e-avatar-circle
violet">JK</span>
      <span className="e-avatar e-avatar-xlarge e-avatar-circle
rose">EL</span>
      <span className="e-avatar e-avatar-xlarge e-avatar-circle
blue">SR</span>
      <span className="e-avatar e-avatar-xlarge e-avatar-circle
red">PD</span>
    </div>
  );
}
export default ReactApp;
ReactDOM.render(<ReactApp/>, document.getElementById("element"));

```

Customize avatar sizes

Even though the avatar comes with five predefined sizes, sometimes it's not enough. So, the avatar is designed in such a way that the width and height will be relative to font-size. By changing the **font-size** of the avatar element, you can change the width and height automatically.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function ReactApp() {
  return (<div>
    <span className="e-avatar e-avatar-xlarge"></span>
    <span className="e-avatar e-avatar-large"></span>
    <span className="e-avatar"></span>
    <span className="e-avatar e-avatar-small"></span>
    <span className="e-avatar e-avatar-xsmall"></span>
  </div>);
}

```

```

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

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function ReactApp() {
    return (
        <div>
            <span className="e-avatar e-avatar-xlarge"></span>
            <span className="e-avatar e-avatar-large"></span>
            <span className="e-avatar"></span>
            <span className="e-avatar e-avatar-small"></span>
            <span className="e-avatar e-avatar-xsmall"></span>
        </div>
    );
}
export default ReactApp;
ReactDOM.render(<ReactApp/>, document.getElementById("element"));

```

Use various media in avatar

Avatars can be used with a wide variety of media formats like SVG, font-icons, images, letters, words, etc. Some of them are given below.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function ReactApp() {
    return (<div>
        <div className="sample_container avatar-types">
            <div className="avatar-block">
                /* Card Component */
                <div className="e-card e-avatar-showcase">
                    <div className="e-card-content">
                        /* XLarge Circle Avatar Component */
                        <div className="e-avatar e-avatar-xlarge e-avatar-circle">
                            
                        </div>
                    </div>
                    <div className="e-card-content">
                        <div>Image</div>
                    </div>
                </div>
            </div>
            <div className="avatar-block">
                /* Card Component */
                <div className="e-card e-avatar-showcase">
                    <div className="e-card-content">
                        /* XLarge Circle Avatar Component */

```

```

        <div className="e-avatar e-avatar-xlarge e-avatar-
circle">
            <div className="svg_icons chrome"></div>
        </div>
    </div>
    <div className="e-card-content">
        <div>SVG</div>
    </div>
</div>
<div className="avatar-block">
    { /* Card Component */ }
    <div className="e-card e-avatar-showcase">
        <div className="e-card-content">
            { /* XLarge Circle Avatar Component */ }
            <div className="e-avatar e-avatar-xlarge e-avatar-
circle">GR</div>
        </div>
        <div className="e-card-content">
            <div>Initial</div>
        </div>
    </div>
</div>
<div className="avatar-block">
    { /* Card Component */ }
    <div className="e-card e-avatar-showcase">
        <div className="e-card-content">
            { /* XLarge Circle Avatar Component */ }
            <div className="e-avatar e-avatar-xlarge e-avatar-
circle">
                <div className="e-people icons"></div>
            </div>
        </div>
        <div className="e-card-content">
            <div>Font Icon</div>
        </div>
    </div>
</div>
<div className="avatar-block">
    { /* Card Component */ }
    <div className="e-card e-avatar-showcase">
        <div className="e-card-content">
            { /* XLarge Circle Avatar Component */ }
            <div className="e-avatar e-avatar-xlarge e-avatar-
circle">User</div>
        </div>
        <div className="e-card-content">
            <div>Word</div>
        </div>
    </div>
</div>
<div className="avatar-block">
    { /* Card Component */ }
    <div className="e-card e-avatar-showcase">
        <div className="e-card-content">
            { /* XLarge Circle Avatar Component */ }

```



```

        <div className="e-avatar e-avatar-xlarge e-avatar-
circle custom">
            <div className="e-people icons"></div>
        </div>
    </div>
    <div className="e-card-content">
        <div>Custom</div>
    </div>
</div>
</div>
</div>
</div>);
}
export default ReactApp;
ReactDOM.render(<ReactApp />, document.getElementById("element"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function ReactApp() {
    return (
        <div>
            <div className="sample_container avatar-types">
                <div className="avatar-block">
                    /* Card Component */
                    <div className="e-card e-avatar-showcase">
                        <div className="e-card-content">
                            /* XLarge Circle Avatar Component */
                            <div className="e-avatar e-avatar-xlarge e-avatar-
circle">
                                
                                </div>
                            </div>
                            <div className="e-card-content">
                                <div>Image</div>
                            </div>
                        </div>
                    </div>
                    <div className="avatar-block">
                        /* Card Component */
                        <div className="e-card e-avatar-showcase">
                            <div className="e-card-content">
                                /* XLarge Circle Avatar Component */
                                <div className="e-avatar e-avatar-xlarge e-avatar-
circle">
                                    <div className="svg_icons chrome"></div>
                                </div>
                            </div>
                            <div className="e-card-content">
                                <div>SVG</div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>

```

```

        { /* Card Component */ }
        <div className="e-card e-avatar-showcase">
            <div className="e-card-content">
                { /* XLarge Circle Avatar Component */ }
                <div className="e-avatar e-avatar-xlarge e-avatar-
circle">GR</div>
            </div>
            <div className="e-card-content">
                <div>Initial</div>
            </div>
        </div>
    </div>
    <div className="avatar-block">
        { /* Card Component */ }
        <div className="e-card e-avatar-showcase">
            <div className="e-card-content">
                { /* XLarge Circle Avatar Component */ }
                <div className="e-avatar e-avatar-xlarge e-avatar-
circle">

                    <div className="e-people icons"></div>
                </div>
            </div>
            <div className="e-card-content">
                <div>Font Icon</div>
            </div>
        </div>
    </div>
    <div className="avatar-block">
        { /* Card Component */ }
        <div className="e-card e-avatar-showcase">
            <div className="e-card-content">
                { /* XLarge Circle Avatar Component */ }
                <div className="e-avatar e-avatar-xlarge e-avatar-
circle">User</div>
            </div>
            <div className="e-card-content">
                <div>Word</div>
            </div>
        </div>
    </div>
    <div className="avatar-block">
        { /* Card Component */ }
        <div className="e-card e-avatar-showcase">
            <div className="e-card-content">
                { /* XLarge Circle Avatar Component */ }
                <div className="e-avatar e-avatar-xlarge e-avatar-
circle custom">

                    <div className="e-people icons"></div>
                </div>
            </div>
            <div className="e-card-content">
                <div>Custom</div>
            </div>
        </div>
    </div>
</div>
</div>

```

```

    );
  }
  export default ReactApp;
  ReactDOM.render(<ReactApp/>, document.getElementById("element"));

```

Dynamic avatar rendering from datasource

We can render avatar component dynamically from a data-source. In this sample we have rendered the avatar component

using a data-source which contains the image source in different sizes dynamically. This is applicable also for data-source from the server or remote data or AJAX. We have also rendered the avatar using CSS property `background-image` and using image tag.

INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
function ReactApp() {
  let dataSource = [
    { id: '1', src:
'https://ej2.syncfusion.com/demos/src/grid/images/2.png', size: 'e-avatar-
xsmall' },
    { id: '2', src:
'https://ej2.syncfusion.com/demos/src/grid/images/2.png', size: 'e-avatar-
small' },
    { id: '3', src:
'https://ej2.syncfusion.com/demos/src/grid/images/2.png', size: 'e-avatar'
},
    { id: '4', src:
'https://ej2.syncfusion.com/demos/src/grid/images/2.png', size: 'e-avatar-
large' },
    { id: '5', src:
'https://ej2.syncfusion.com/demos/src/grid/images/2.png', size: 'e-avatar-
xlarge' }
  ];
  return (<div className='control-pane'>
    <div className="sample_container avatar-badge">
      <div className="avatar-block">
        <div className="e-card e-avatar-showcase">
          <div className="e-card-content">
            {dataSource.map(function (item) {
              return (<div key={item.id} className={`e-avatar e-avatar-circle
${item.size}` style={{ backgroundImage: `url(${item.src})` }}></div>);
            })}
          </div>
          <div className="e-card-content">
            <div>Using <code>background-image</code>
property</div>
          </div>
        </div>
      </div>
      <div className="circleAvatar avatar-block">
        <div className="e-card e-avatar-showcase">
          <div className="e-card-content">
            {dataSource.map(function (item) {

```

```

        return (<div key={item.id} className={`e-avatar e-avatar-circle
${item.size}`}>
            <img src={item.src}/>
        </div>);
    }}}
    </div>
    <div className="e-card-content">
        <div>Using <code>img</code> tag</div>
    </div>
    </div>
    </div>
    </div>
    </div>);
}
export default ReactApp;
ReactDOM.render(<ReactApp />, document.getElementById("element"));
{% endraw %}

```

INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
function ReactApp() {
    let dataSource: { [key: string]: string }[] = [
        { id: '1', src:
'https://ej2.syncfusion.com/demos/src/grid/images/2.png', size: 'e-avatar-
xsmall' },
        { id: '2', src:
'https://ej2.syncfusion.com/demos/src/grid/images/2.png', size: 'e-avatar-
small' },
        { id: '3', src:
'https://ej2.syncfusion.com/demos/src/grid/images/2.png', size: 'e-avatar'
},
        { id: '4', src:
'https://ej2.syncfusion.com/demos/src/grid/images/2.png', size: 'e-avatar-
large' },
        { id: '5', src:
'https://ej2.syncfusion.com/demos/src/grid/images/2.png', size: 'e-avatar-
xlarge' }
    ];
    return (
        <div className='control-pane'>
            <div className="sample_container avatar-badge">
                <div className="avatar-block">
                    <div className="e-card e-avatar-showcase">
                        <div className="e-card-content">
                            {dataSource.map(function (item) {
                                return (<div key={item.id} className={`e-
avatar e-avatar-circle ${item.size}`} style={{ backgroundImage:
'url(${item.src})` }}></div>)
                            })}
                        </div>
                        <div className="e-card-content">
                            <div>Using <code>background-image</code>
property</div>

```

```

        </div>
      </div>
    </div>
    <div className="circleAvatar avatar-block">
      <div className="e-card e-avatar-showcase">
        <div className="e-card-content">
          {dataSource.map(function (item) {
            return <div key={item.id} className={`e-
avatar e-avatar-circle ${item.size}` } >
              <img src={item.src} />
            </div>)
          )}}
        </div>
        <div className="e-card-content">
          <div>Using <code>img</code> tag</div>
        </div>
      </div>
    </div>
  </div>
</div>
);
}
export default ReactApp;
ReactDOM.render(<ReactApp/>, document.getElementById("element"));
{% endraw %}

```

Integrate avatar into listview in React Avatar component

Avatar is integrated into the listview to create contacts applications. The `xsmall` size avatar is used to match the size of the list item. Letters and images are also used as avatar content.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { ListViewComponent } from '@syncfusion/ej2-react-lists';
function ReactApp() {
  //Define an array of JSON data
  let data = [
    { id: 's_01', text: 'Robert', avatar: '', pic: 'pic04' },
    { id: 's_02', text: 'Nancy', avatar: 'N', pic: '' },
    { id: 's_05', text: 'John', pic: 'pic01', avatar: '' },
    { id: 's_03', text: 'Andrew', avatar: 'A', pic: '' },
    { id: 's_06', text: 'Margaret', avatar: 'M', pic: 'pic02' },
    { id: 's_07', text: 'Steven', pic: 'pic03', avatar: '' },
    { id: 's_08', text: 'Michael', pic: 'pic02', avatar: '' }
  ];
  function template(data) {
    let letterAvatar = <span className='e-avatar e-avatar-small e-
avatar-circle'>{data.avatar}</span>;
    let imageAvatar = <span className={` ${data.pic} e-avatar e-avatar-
small e-avatar-circle`}></span>;
    return <div className='listWrapper'>
      {data.avatar !== "" ? (letterAvatar) : (imageAvatar)}
      <span className='text'>{data.text}</span>
    </div>;
  }
}

```

```

    return (<div>
      <div className="sample_container listview">
        { /* ListView element */ }
        <ListViewComponent cssClass='letterAvatarList' dataSource={data}
headerTitle='Contacts' showHeader={true} sortOrder="Ascending"
template={template}></ListViewComponent>
      </div>
    </div>);
  }
export default ReactApp;
ReactDOM.render(<ReactApp />, document.getElementById("element"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { ListViewComponent } from '@syncfusion/ej2-react-lists';
function ReactApp() {
  //Define an array of JSON data
  let data: { [key: string]: Object }[] = [
    { id: 's_01', text: 'Robert', avatar: '', pic: 'pic04' },
    { id: 's_02', text: 'Nancy', avatar: 'N', pic: '' },
    { id: 's_05', text: 'John', pic: 'pic01', avatar: '' },
    { id: 's_03', text: 'Andrew', avatar: 'A', pic: '' },
    { id: 's_06', text: 'Margaret', avatar: 'M', pic: 'pic02' },
    { id: 's_07', text: 'Steven', pic: 'pic03', avatar: '' },
    { id: 's_08', text: 'Michael', pic: 'pic02', avatar: '' }
  ];
  function template(data: any): JSX.Element {
    let letterAvatar = <span className='e-avatar e-avatar-small e-
avatar-circle'>{data.avatar}</span>
    let imageAvatar = <span className={` ${data.pic} e-avatar e-avatar-
small e-avatar-circle`}></span>
    return (
      <div className='listWrapper'>
        {data.avatar !== "" ? (letterAvatar) : (imageAvatar)}
        <span className='text'>{data.text}</span>
      </div>
    );
  }
  return (
    <div>
      <div className="sample_container listview">
        { /* ListView element */ }
        <ListViewComponent cssClass='letterAvatarList' dataSource={data}
headerTitle='Contacts' showHeader={true}
sortOrder="Ascending" template={template as
any}></ListViewComponent>
      </div>
    </div>
  );
}
export default ReactApp;
ReactDOM.render(<ReactApp />, document.getElementById("element"));

```

Integrate avatar into badge in React Avatar component

The badge is dependent and supportive component, and it can be used with avatar to create a notification avatar.

The default avatar (.e-avatar) and circle avatar (.e-avatar-circle) have been used with notification badges (.e-badge-notification) in the following sample.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
function ReactApp() {
  return (
    <div>
      <div className="sample_container avatar-badge">
        <div className="avatar-block">
          { /* Card Component */ }
          <div className="e-card e-avatar-showcase">
            <div className="e-card-content">
              <div className="avatar-sub-block">
                { /* xSmall Avatar */ }
                <div className="e-avatar e-avatar-xsmall">
                  
                </div>
                { /* Notification Badge */ }
                <span className="e-badge e-badge-primary e-
badge-overlap e-badge-notification e-badge-circle">6</span>
              </div>
              <div className="avatar-sub-block">
                { /* Small Avatar */ }
                <div className="e-avatar e-avatar-small">
                  
                </div>
                { /* Notification Badge */ }
                <span className="e-badge e-badge-primary e-
badge-overlap e-badge-notification e-badge-circle">12</span>
              </div>
              <div className="avatar-sub-block">
                { /* Avatar */ }
                <div className="e-avatar">
                  
                </div>
                { /* Notification Badge */ }
                <span className="e-badge e-badge-primary e-
badge-overlap e-badge-notification">46</span>
              </div>
              <div className="avatar-sub-block">
                { /* Large Avatar */ }
                <div className="e-avatar e-avatar-large">
                  
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  );
}
```

```

        </div>
        { /* Notification Badge */}
        <span className="e-badge e-badge-primary e-
badge-overlap e-badge-notification">82</span>
    </div>
    <div className="avatar-sub-block">
        { /* xLarge Avatar */}
        <div className="e-avatar e-avatar-xlarge">
            
        </div>
        { /* Notification Badge */}
        <span className="e-badge e-badge-primary e-
badge-overlap e-badge-notification">99+</span>
    </div>
</div>
</div>
<div className="circleAvatar avatar-block">
    { /* Card Component */}
    <div className="e-card e-avatar-showcase">
        <div className="e-card-content">
            <div className="avatar-sub-block">
                { /* xSmall Circle Avatar */}
                <div className="e-avatar e-avatar-circle e-
avatar-xsmall">
                    
                </div>
                { /* Notification Badge */}
                <span className="e-badge e-badge-primary e-
badge-overlap e-badge-notification e-badge-circle">6</span>
            </div>
            <div className="avatar-sub-block">
                { /* Small Circle Avatar */}
                <div className="e-avatar e-avatar-circle e-
avatar-small">
                    
                </div>
                { /* Notification Badge */}
                <span className="e-badge e-badge-primary e-
badge-overlap e-badge-notification e-badge-circle">12</span>
            </div>
            <div className="avatar-sub-block">
                { /* Circle Avatar */}
                <div className="e-avatar e-avatar-circle">
                    
                </div>
                { /* Notification Badge */}
                <span className="e-badge e-badge-primary e-
badge-overlap e-badge-notification">46</span>
            </div>
        </div>
    </div>

```



```

        </div>
        <div className="avatar-sub-block">
            { /* Large Circle Avatar */ }
            <div className="e-avatar e-avatar-circle e-
avatar-large">
                
            </div>
            { /* Notification Badge */ }
            <span className="e-badge e-badge-primary e-
badge-overlap e-badge-notification">82</span>
        </div>
        <div className="avatar-sub-block">
            { /* xLarge Circle Avatar */ }
            <div className="e-avatar e-avatar-circle e-
avatar-xlarge">
                
            </div>
            { /* Notification Badge */ }
            <span className="e-badge e-badge-primary e-
badge-overlap e-badge-notification">99</span>
        </div>
    </div>
</div>
</div>
</div>
</div>);
}
export default ReactApp;
ReactDOM.render(<ReactApp />, document.getElementById("element"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function ReactApp() {
    return (
        <div>
            <div className="sample_container avatar-badge">
                <div className="avatar-block">
                    { /* Card Component */ }
                    <div className="e-card e-avatar-showcase">
                        <div className="e-card-content">
                            <div className="avatar-sub-block">
                                { /* xSmall Avatar */ }
                                <div className="e-avatar e-avatar-xsmall">
                                    
                                </div>
                                { /* Notification Badge */ }
                                <span className="e-badge e-badge-primary e-
badge-overlap e-badge-notification e-badge-circle">6</span>

```

```

        </div>
        <div className="avatar-sub-block">
            { /* Small Avatar */ }
            <div className="e-avatar e-avatar-small">
                
            </div>
            { /* Notification Badge */ }
            <span className="e-badge e-badge-primary e-
badge-overlap e-badge-notification e-badge-circle">12</span>
        </div>
        <div className="avatar-sub-block">
            { /* Avatar */ }
            <div className="e-avatar">
                
            </div>
            { /* Notification Badge */ }
            <span className="e-badge e-badge-primary e-
badge-overlap e-badge-notification">46</span>
        </div>
        <div className="avatar-sub-block">
            { /* Large Avatar */ }
            <div className="e-avatar e-avatar-large">
                
            </div>
            { /* Notification Badge */ }
            <span className="e-badge e-badge-primary e-
badge-overlap e-badge-notification">82</span>
        </div>
        <div className="avatar-sub-block">
            { /* xLarge Avatar */ }
            <div className="e-avatar e-avatar-xlarge">
                
            </div>
            { /* Notification Badge */ }
            <span className="e-badge e-badge-primary e-
badge-overlap e-badge-notification">99+</span>
        </div>
    </div>
</div>
<div className="circleAvatar avatar-block">
    { /* Card Component */ }
    <div className="e-card e-avatar-showcase">
        <div className="e-card-content">
            <div className="avatar-sub-block">
                { /* xSmall Circle Avatar */ }
                <div className="e-avatar e-avatar-circle e-
avatar-xsmall">

```

```


</div>
{ /* Notification Badge */}
<span className="e-badge e-badge-primary e-
badge-overlap e-badge-notification e-badge-circle">6</span>
</div>
<div className="avatar-sub-block">
{ /* Small Circle Avatar */}
<div className="e-avatar e-avatar-circle e-
avatar-small">

</div>
{ /* Notification Badge */}
<span className="e-badge e-badge-primary e-
badge-overlap e-badge-notification e-badge-circle">12</span>
</div>
<div className="avatar-sub-block">
{ /* Circle Avatar */}
<div className="e-avatar e-avatar-circle">

</div>
{ /* Notification Badge */}
<span className="e-badge e-badge-primary e-
badge-overlap e-badge-notification">46</span>
</div>
<div className="avatar-sub-block">
{ /* Large Circle Avatar */}
<div className="e-avatar e-avatar-circle e-
avatar-large">

</div>
{ /* Notification Badge */}
<span className="e-badge e-badge-primary e-
badge-overlap e-badge-notification">82</span>
</div>
<div className="avatar-sub-block">
{ /* xLarge Circle Avatar */}
<div className="e-avatar e-avatar-circle e-
avatar-xlarge">

</div>
{ /* Notification Badge */}
<span className="e-badge e-badge-primary e-
badge-overlap e-badge-notification">99+</span>
</div>
</div>
</div>

```

```
        </div>
      </div>
    </div>
  );
}
export default ReactApp;
ReactDOM.render(<ReactApp/>, document.getElementById("element"));
```

Badge

Getting Started

The following section explains how to create a simple Badge component using styles and Badge's basic usage.

Dependencies

Install the following required dependent package to render the `Badge` component.

```
`javascript
|-- @syncfusion/ej2-notifications
`
```

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.

```
`
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.

```
`
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.

To install `Badge` component, use the following command

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

Adding CSS Reference

Badge CSS files are available in the `ej2-notifications` package folder. Import the Badge 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-notifications/styles/material.css";
`
```

We can also use [CRG](#) to generate combined component styles.

Initialize the Badge Component

Add an HTML span element with `e-badge` class inside any wrapper element (`h1`) into your return method.

```
`ts
import * as React from 'react';
import * as ReactDOM from "react-dom";
import './App.css';
function App() {
  return (
    <h1>Badge Component <span className="e-badge e-badge-primary">New</span></h1>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
`
```

Run the application

Run the application in the browser using the following command:

```
`
npm start
`
```

Output will be as follows:

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (<h1>Badge Component <span className="e-badge e-badge-
primary">New</span></h1>);
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (
    <h1>Badge Component <span className="e-badge e-badge-
primary">New</span></h1>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
```

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Essential JS 2 React Badge Sample</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/ej2-
base/styles/material.css" rel="stylesheet" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/ej2-
notifications/styles/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>
  <link rel="stylesheet" href="index.css" />
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
  </style>
</head>
<body>
  <div id='element'>
    <div id='loader'>Loading....</div></div>
</body>
```

</html>

See Also

[Types of Badge](#)

Types in React Badge component

This section explains different styles and types of the badges.

Badge styles

The Essential JS 2 Badge has the following predefined styles that can be used with `.e-badge` class to change the appearance of a badge.

Class Name	Description
:-----	:-----
e-badge-primary	Represents a primary notification.
e-badge-secondary	Represents a secondary notification.
e-badge-success	Represents a positive notification.
e-badge-danger	Represents a negative notification.
e-badge-warning	Represents notification with caution.
e-badge-info	Represents an informative notification.
e-badge-light	Represents notification in light variant.
e-badge-dark	Represents notification in dark variant.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (
    <div className="sample_container">
      <div className="block">
        <div className="e-card e-badge-showcase">
          <div className="e-card-content">
            <div>
              <span className="e-badge e-badge-
primary">Primary</span>
            </div>
          </div>
          <div className="e-card-content">
            <div>
              <code>.e-badge-primary</code>
            </div>
          </div>
        </div>
      </div>
      <div className="block">
        <div className="e-card e-badge-showcase">
          <div className="e-card-content">
            <span className="e-badge e-badge-
secondary">Secondary</span>
          </div>
        </div>
      </div>
    </div>
  );
}
```

```
        <div className="e-card-content">
          <code>.e-badge-secondary</code>
        </div>
      </div>
    </div>
    <div className="block">
      <div className="e-card e-badge-showcase">
        <div className="e-card-content">
          <span className="e-badge e-badge-success">Success</span>
        </div>
        <div className="e-card-content">
          <code>.e-badge-success</code>
        </div>
      </div>
    </div>
    <div className="block">
      <div className="e-card e-badge-showcase">
        <div className="e-card-content">
          <span className="e-badge e-badge-danger">Danger</span>
        </div>
        <div className="e-card-content">
          <code>.e-badge-danger</code>
        </div>
      </div>
    </div>
    <div className="block">
      <div className="e-card e-badge-showcase">
        <div className="e-card-content">
          <span className="e-badge e-badge-warning">Warning</span>
        </div>
        <div className="e-card-content">
          <code>.e-badge-warning</code>
        </div>
      </div>
    </div>
    <div className="block">
      <div className="e-card e-badge-showcase">
        <div className="e-card-content">
          <span className="e-badge e-badge-info">Info</span>
        </div>
        <div className="e-card-content">
          <code>.e-badge-info</code>
        </div>
      </div>
    </div>
    <div className="block">
      <div className="e-card e-badge-showcase">
        <div className="e-card-content">
          <span className="e-badge e-badge-light">Light</span>
        </div>
        <div className="e-card-content">
          <code>.e-badge-light</code>
        </div>
      </div>
    </div>
    <div className="block">
      <div className="e-card e-badge-showcase">
```



```

        <div className="e-card-content">
          <span className="e-badge e-badge-dark">Dark</span>
        </div>
        <div className="e-card-content">
          <code>.e-badge-dark</code>
        </div>
      </div>
    </div>
  </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (
    <div className="sample_container">
      <div className="block">
        <div className="e-card e-badge-showcase">
          <div className="e-card-content">
            <div>
              <span className="e-badge e-badge-
primary">Primary</span>
            </div>
          </div>
          <div className="e-card-content">
            <div>
              <code>.e-badge-primary</code>
            </div>
          </div>
        </div>
      </div>
      <div className="block">
        <div className="e-card e-badge-showcase">
          <div className="e-card-content">
            <span className="e-badge e-badge-
secondary">Secondary</span>
          </div>
          <div className="e-card-content">
            <code>.e-badge-secondary</code>
          </div>
        </div>
      </div>
      <div className="block">
        <div className="e-card e-badge-showcase">
          <div className="e-card-content">
            <span className="e-badge e-badge-success">Success</span>
          </div>
          <div className="e-card-content">
            <code>.e-badge-success</code>
          </div>
        </div>
      </div>
    </div>
  );
}

```

```

    <div className="block">
      <div className="e-card e-badge-showcase">
        <div className="e-card-content">
          <span className="e-badge e-badge-danger">Danger</span>
        </div>
        <div className="e-card-content">
          <code>.e-badge-danger</code>
        </div>
      </div>
    </div>
    <div className="block">
      <div className="e-card e-badge-showcase">
        <div className="e-card-content">
          <span className="e-badge e-badge-warning">Warning</span>
        </div>
        <div className="e-card-content">
          <code>.e-badge-warning</code>
        </div>
      </div>
    </div>
    <div className="block">
      <div className="e-card e-badge-showcase">
        <div className="e-card-content">
          <span className="e-badge e-badge-info">Info</span>
        </div>
        <div className="e-card-content">
          <code>.e-badge-info</code>
        </div>
      </div>
    </div>
    <div className="block">
      <div className="e-card e-badge-showcase">
        <div className="e-card-content">
          <span className="e-badge e-badge-light">Light</span>
        </div>
        <div className="e-card-content">
          <code>.e-badge-light</code>
        </div>
      </div>
    </div>
    <div className="block">
      <div className="e-card e-badge-showcase">
        <div className="e-card-content">
          <span className="e-badge e-badge-dark">Dark</span>
        </div>
        <div className="e-card-content">
          <code>.e-badge-dark</code>
        </div>
      </div>
    </div>
  </div>
);
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));

```

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Essential JS 2 React Badge Sample</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/ej2-
base/styles/material.css" rel="stylesheet" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/ej2-
notifications/styles/material.css" rel="stylesheet" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/ej2-
layouts/styles/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>
  <link rel="stylesheet" href="index.css" />
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
  </style>
</head>
<body>
  <div id='element'>
    <div id='loader'>Loading....</div></div>
</body>
</html>
```

Badge types

The types of Essential JS 2 badges are as follows:

- Circle
- Pill
- Link
- Notification
- Overlap
- Dot
- Position

Circle

The circle badge style can be applied by adding the modifier class `.e-badge-circle` to the target element.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (
    <div>
      <div className="badge-block">
        <div className="skype svg_icons"></div>
        <span className="e-badge e-badge-success e-badge-overlap e-badge-notification e-badge-circle">18</span>
      </div>
      <div className="badge-block">
        <div className="twitter svg_icons"></div>
        <span className="e-badge e-badge-info e-badge-overlap e-badge-notification e-badge-circle">9</span>
      </div>
      <div className="badge-block">
        <div className="facebook svg_icons"></div>
        <span className="e-badge e-badge-info e-badge-overlap e-badge-notification e-badge-circle">2</span>
      </div>
      <div className="badge-block">
        <div className="firefox svg_icons"></div>
        <span className="e-badge e-badge-danger e-badge-overlap e-badge-notification e-badge-circle">35</span>
      </div>
    </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (
    <div>
      <div className="badge-block">
        <div className="skype svg_icons"></div>
        <span className="e-badge e-badge-success e-badge-overlap e-badge-notification e-badge-circle">18</span>
      </div>
      <div className="badge-block">
        <div className="twitter svg_icons"></div>
        <span className="e-badge e-badge-info e-badge-overlap e-badge-notification e-badge-circle">9</span>
      </div>
      <div className="badge-block">
        <div className="facebook svg_icons"></div>
        <span className="e-badge e-badge-info e-badge-overlap e-badge-notification e-badge-circle">2</span>
      </div>
      <div className="badge-block">
        <div className="firefox svg_icons"></div>
        <span className="e-badge e-badge-danger e-badge-overlap e-badge-notification e-badge-circle">35</span>
      </div>
    </div>
  );
}

```

```

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

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Essential JS 2 React Badge Sample</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/ej2-
base/styles/material.css" rel="stylesheet" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/ej2-
notifications/styles/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>
  <link rel="stylesheet" href="index.css" />
  <style>
    #loader {
      color: #008c8f;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
  </style>
</head>
<body>
  <div id='element'>
    <div id='loader'>Loading....</div></div>
</body>
</html>

```

Pill

The pill badge style can be applied by adding the modifier class `.e-badge-pill` to the target element.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (<h1>Badge Component <span className="e-badge e-badge-primary e-
badge-pill">New</span></h1>);
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));

```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (
    <h1>Badge Component <span className="e-badge e-badge-primary e-badge-pill">New</span></h1>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
```

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Essential JS 2 React Badge Sample</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/ej2-base/styles/material.css" rel="stylesheet" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/ej2-notifications/styles/material.css" rel="stylesheet" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></script>
  <script src="systemjs.config.js"></script>
  <link rel="stylesheet" href="index.css" />
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
  </style>
</head>
<body>
  <div id='element'>
    <div id='loader'>Loading....</div></div>
</body>
</html>
```

[Link](#)

When badge modifier classes are applied to the anchor tag, the badge's appearance will change from normal state to hover state on mouseover.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (<div className="badge-block">
    <a href="#" className="e-badge e-badge-primary">Link Badge</a>
  </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (
    <div className="badge-block">
      <a href="#" className="e-badge e-badge-primary">Link Badge</a>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
```

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Essential JS 2 React Badge Sample</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/ej2-base/styles/material.css" rel="stylesheet" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/ej2-notifications/styles/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>
  <link rel="stylesheet" href="index.css" />
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
  </style>
</head>
<body>
  <div id='element'>
```

```
<div id='loader'>Loading....</div></div>
</body>
</html>
```

Notification

The notification badge style can be applied by adding the modifier class `.e-badge-notification` to the target element. Notification badges are used when a content or a context needs special attention. While using the notification badge, set the parent element to `position: relative`.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (
    <div>
      <div className="badge-block">
        <div className="skype svg_icons"></div>
        <span className="e-badge e-badge-success e-badge-overlap e-badge-notification">99+</span>
      </div>
      <div className="badge-block">
        <div className="twitter svg_icons"></div>
        <span className="e-badge e-badge-info e-badge-overlap e-badge-notification">27</span>
      </div>
      <div className="badge-block">
        <div className="facebook svg_icons"></div>
        <span className="e-badge e-badge-info e-badge-overlap e-badge-notification">2</span>
      </div>
      <div className="badge-block">
        <div className="firefox svg_icons"></div>
        <span className="e-badge e-badge-danger e-badge-overlap e-badge-notification">35</span>
      </div>
    </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (
    <div>
      <div className="badge-block">
        <div className="skype svg_icons"></div>
        <span className="e-badge e-badge-success e-badge-overlap e-badge-notification">99+</span>
      </div>
      <div className="badge-block">
        <div className="twitter svg_icons"></div>
```



```

        <span className="e-badge e-badge-info e-badge-overlap e-badge-
notification">27</span>
      </div>
      <div className="badge-block">
        <div className="facebook svg_icons"></div>
        <span className="e-badge e-badge-info e-badge-overlap e-badge-
notification">2</span>
      </div>
      <div className="badge-block">
        <div className="firefox svg_icons"></div>
        <span className="e-badge e-badge-danger e-badge-overlap e-badge-
notification">35</span>
      </div>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Essential JS 2 React Badge Sample</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/ej2-
base/styles/material.css" rel="stylesheet" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/ej2-
notifications/styles/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>
  <link rel="stylesheet" href="index.css" />
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
  </style>
</head>
<body>
  <div id='element'>
    <div id='loader'>Loading....</div></div>
</body>
</html>

```

Dot

Dot can be applied by adding the modifier class `.e-badge-dot` to the target element. Dot badges are similar to notification badges, but in a minimalistic way. While using the dot badge, set the parent element to `position:relative`.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (
    <div>
      <div className="badge-block">
        <div className="skype svg_icons"></div>
        <span className="e-badge e-badge-success e-badge-overlap e-badge-dot"></span>
      </div>
      <div className="badge-block">
        <div className="twitter svg_icons"></div>
        <span className="e-badge e-badge-info e-badge-overlap e-badge-dot"></span>
      </div>
      <div className="badge-block">
        <div className="facebook svg_icons"></div>
        <span className="e-badge e-badge-info e-badge-overlap e-badge-dot"></span>
      </div>
      <div className="badge-block">
        <div className="firefox svg_icons"></div>
        <span className="e-badge e-badge-danger e-badge-overlap e-badge-dot"></span>
      </div>
    </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (
    <div>
      <div className="badge-block">
        <div className="skype svg_icons"></div>
        <span className="e-badge e-badge-success e-badge-overlap e-badge-dot"></span>
      </div>
      <div className="badge-block">
        <div className="twitter svg_icons"></div>
        <span className="e-badge e-badge-info e-badge-overlap e-badge-dot"></span>
      </div>
      <div className="badge-block">
        <div className="facebook svg_icons"></div>
```

```

        <span className="e-badge e-badge-info e-badge-overlap e-badge-
dot"></span>
      </div>
      <div className="badge-block">
        <div className="firefox svg_icons"></div>
        <span className="e-badge e-badge-danger e-badge-overlap e-badge-
dot"></span>
      </div>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Essential JS 2 React Badge Sample</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/ej2-
base/styles/material.css" rel="stylesheet" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/ej2-
notifications/styles/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>
  <link rel="stylesheet" href="index.css" />
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
  </style>
</head>
<body>
  <div id='element'>
    <div id='loader'>Loading....</div></div>
</body>
</html>

```

Overlap

The overlap badge can be used with notification or dot badge, which overlaps with the target element by adding the modifier class `.e-badge-overlap`. While using the overlap badge, set the parent element to `position: relative`.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (
        <div>
            <div className="badge-block">
                <div className="skype svg_icons"></div>
                <span className="e-badge e-badge-success e-badge-overlap e-badge-notification">99+</span>
            </div>
            <div className="badge-block">
                <div className="twitter svg_icons"></div>
                <span className="e-badge e-badge-info e-badge-overlap e-badge-notification">27</span>
            </div>
            <div className="badge-block">
                <div className="facebook svg_icons"></div>
                <span className="e-badge e-badge-info e-badge-overlap e-badge-notification">2</span>
            </div>
            <div className="badge-block">
                <div className="firefox svg_icons"></div>
                <span className="e-badge e-badge-danger e-badge-overlap e-badge-notification">35</span>
            </div>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (
        <div>
            <div className="badge-block">
                <div className="skype svg_icons"></div>
                <span className="e-badge e-badge-success e-badge-overlap e-badge-notification">99+</span>
            </div>
            <div className="badge-block">
                <div className="twitter svg_icons"></div>
                <span className="e-badge e-badge-info e-badge-overlap e-badge-notification">27</span>
            </div>
            <div className="badge-block">
                <div className="facebook svg_icons"></div>
                <span className="e-badge e-badge-info e-badge-overlap e-badge-notification">2</span>
            </div>
            <div className="badge-block">
                <div className="firefox svg_icons"></div>
            </div>
        </div>
    );
}

```

```

        <span className="e-badge e-badge-danger e-badge-overlap e-badge-notification">35</span>
      </div>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Essential JS 2 React Badge Sample</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/ej2-base/styles/material.css" rel="stylesheet" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/ej2-notifications/styles/material.css" rel="stylesheet" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></script>
  <script src="systemjs.config.js"></script>
  <link rel="stylesheet" href="index.css" />
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
  </style>
</head>
<body>
  <div id='element'>
    <div id='loader'>Loading....</div></div>
</body>
</html>

```

Position

The default position of the notification or dot badge is top. But, the position can be changed to bottom using the modifier class `.e-badge-bottom`. For example, the bottom class modifier is used with dot badge to display the status in the avatar as shown in the following sample.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {

```

```

    return (<div>
      <div className="badge-block">
        <div className="firefox svg_icons"></div>
        { /* Success Colored Bottom Dot Badge */ }
        <span className="e-badge e-badge-success e-badge-overlap e-badge-dot e-badge-bottom"></span>
      </div>
      <div className="badge-block">
        <div className="skype svg_icons"></div>
        { /* Info Colored Bottom Dot Badge */ }
        <span className="e-badge e-badge-info e-badge-overlap e-badge-dot e-badge-bottom"></span>
      </div>
      <div className="badge-block">
        <div className="facebook svg_icons"></div>
        { /* Info Colored Dot Badge */ }
        <span className="e-badge e-badge-info e-badge-overlap e-badge-dot"></span>
      </div>
      <div className="badge-block">
        <div className="twitter svg_icons"></div>
        { /* Danger Colored Dot Badge */ }
        <span className="e-badge e-badge-danger e-badge-overlap e-badge-dot e-badge-bottom"></span>
      </div>
    </div>);
  }
  export default App;
  ReactDOM.render(<App />, document.getElementById("element"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (
    <div>
      <div className="badge-block">
        <div className="firefox svg_icons"></div>
        { /* Success Colored Bottom Dot Badge */ }
        <span className="e-badge e-badge-success e-badge-overlap e-badge-dot e-badge-bottom"></span>
      </div>
      <div className="badge-block">
        <div className="skype svg_icons"></div>
        { /* Info Colored Bottom Dot Badge */ }
        <span className="e-badge e-badge-info e-badge-overlap e-badge-dot e-badge-bottom"></span>
      </div>
      <div className="badge-block">
        <div className="facebook svg_icons"></div>
        { /* Info Colored Dot Badge */ }
        <span className="e-badge e-badge-info e-badge-overlap e-badge-dot"></span>
      </div>
      <div className="badge-block">

```

```

        <div className="twitter svg_icons"></div>
        { /* Danger Colored Dot Badge */}
        <span className="e-badge e-badge-danger e-badge-overlap e-badge-
dot e-badge-bottom"></span>
      </div>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Essential JS 2 React Badge Sample</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/ej2-
base/styles/material.css" rel="stylesheet" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/ej2-
notifications/styles/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>
  <link rel="stylesheet" href="index.css" />
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
  </style>
</head>
<body>
  <div id='element'>
    <div id='loader'>Loading....</div></div>
</body>
</html>

```

How To

Badge customization in React Badge component

Colour customization

Even though badges come with 8 predefined colors, you can also customize the colour of the badge as desired.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (<div>
    <h1>Color Customization <span className="e-badge e-badge-primary e-
badge-pill green">New</span></h1>
    <h1>Color Customization <span className="e-badge e-badge-primary e-
badge-pill bue">New</span></h1>
    <h1>Color Customization <span className="e-badge e-badge-primary e-
badge-pill purple">New</span></h1>
    <h1>Color Customization <span className="e-badge e-badge-primary e-
badge-pill gradient">New</span></h1>
  </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (
    <div>
      <h1>Color Customization <span className="e-badge e-badge-primary e-
badge-pill green">New</span></h1>
      <h1>Color Customization <span className="e-badge e-badge-primary e-
badge-pill bue">New</span></h1>
      <h1>Color Customization <span className="e-badge e-badge-primary e-
badge-pill purple">New</span></h1>
      <h1>Color Customization <span className="e-badge e-badge-primary e-
badge-pill gradient">New</span></h1>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
```

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Essential JS 2 React Badge Sample</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/ej2-
base/styles/material.css" rel="stylesheet" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/ej2-
notifications/styles/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>
```



```

<link rel="stylesheet" href="index.css" />
<style>
  #loader {
    color: #008cff;
    height: 40px;
    left: 45%;
    position: absolute;
    top: 45%;
    width: 30%;
  }
</style>
</head>
<body>
  <div id='element'>
    <div id='loader'>Loading....</div></div>
</body>
</html>

```

Customize badge size

Badges are designed to change its size based on the content. To change the size of a badge, adjust the font size of the badge.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (
    <div>
      <h1>Badge Component <span className="e-badge e-badge-primary size_1">New</span></h1>
      <h1>Badge Component <span className="e-badge e-badge-primary size_2">New</span></h1>
      <h1>Badge Component <span className="e-badge e-badge-primary size_3">New</span></h1>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (
    <div>
      <h1>Badge Component <span className="e-badge e-badge-primary size_1">New</span></h1>
      <h1>Badge Component <span className="e-badge e-badge-primary size_2">New</span></h1>
      <h1>Badge Component <span className="e-badge e-badge-primary size_3">New</span></h1>
    </div>
  );
}

```

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

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Essential JS 2 React Badge Sample</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/ej2-
base/styles/material.css" rel="stylesheet" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/ej2-
notifications/styles/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>
  <link rel="stylesheet" href="index.css" />
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
  </style>
</head>
<body>
  <div id='element'>
    <div id='loader'>Loading....</div></div>
</body>
</html>
```

Custom position

Even though the badges support the conventional **top** and **bottom** positions, the position of the badges can be changed as desired. This can be done by adding a custom class to the badge element to override the default position applied from the source.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (
    <div>
      <div className="block">
        <div className="badge-block">
          <div className="firefox svg_icons"></div>
          { /* Warning Colored Notification Badge */ }
        </div>
      </div>
    </div>
  );
}
```

```

        <span className="e-badge e-badge-warning e-badge-
notification e-badge-overlap leftTop">99+</span>
      </div>
      <div className="badge-block">
        <div className="facebook svg_icons"></div>
        { /* Danger Colored Notification Badge */ }
        <span className="e-badge e-badge-danger e-badge-
notification e-badge-overlap leftTop">99+</span>
      </div>
      <div className="badge-block">
        <div className="skype svg_icons"></div>
        { /* Secondary Colored Notification Badge */ }
        <span className="e-badge e-badge-secondary e-badge-
notification e-badge-overlap leftTop">18</span>
      </div>
    </div>
    <div className="badge-block">
      <div className="badge-block">
        <div className="firefox svg_icons"></div>
        { /* Warning Colored Notification Badge */ }
        <span className="e-badge e-badge-warning e-badge-
notification e-badge-overlap leftBottom">99+</span>
      </div>
      <div className="badge-block">
        <div className="facebook svg_icons"></div>
        { /* Danger Colored Notification Badge */ }
        <span className="e-badge e-badge-danger e-badge-
notification e-badge-overlap leftBottom">99+</span>
      </div>
      <div className="badge-block">
        <div className="skype svg_icons"></div>
        { /* Secondary Colored Notification Badge */ }
        <span className="e-badge e-badge-secondary e-badge-
notification e-badge-overlap leftBottom">18</span>
      </div>
    </div>
  </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (
    <div>
      <div className="block">
        <div className="badge-block">
          <div className="firefox svg_icons"></div>
          { /* Warning Colored Notification Badge */ }
          <span className="e-badge e-badge-warning e-badge-
notification e-badge-overlap leftTop">99+</span>
        </div>
        <div className="badge-block">

```

```

        <div className="facebook svg_icons"></div>
        { /* Danger Colored Notification Badge */ }
        <span className="e-badge e-badge-danger e-badge-
notification e-badge-overlap leftTop">99+</span>
      </div>
      <div className="badge-block">
        <div className="skype svg_icons"></div>
        { /* Secondary Colored Notification Badge */ }
        <span className="e-badge e-badge-secondary e-badge-
notification e-badge-overlap leftTop">18</span>
      </div>
    </div>
    <div className="badge-block">
      <div className="badge-block">
        <div className="firefox svg_icons"></div>
        { /* Warning Colored Notification Badge */ }
        <span className="e-badge e-badge-warning e-badge-
notification e-badge-overlap leftBottom">99+</span>
      </div>
      <div className="badge-block">
        <div className="facebook svg_icons"></div>
        { /* Danger Colored Notification Badge */ }
        <span className="e-badge e-badge-danger e-badge-
notification e-badge-overlap leftBottom">99+</span>
      </div>
      <div className="badge-block">
        <div className="skype svg_icons"></div>
        { /* Secondary Colored Notification Badge */ }
        <span className="e-badge e-badge-secondary e-badge-
notification e-badge-overlap leftBottom">18</span>
      </div>
    </div>
  </div>
);
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Essential JS 2 React Badge Sample</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/ej2-
base/styles/material.css" rel="stylesheet" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/ej2-
notifications/styles/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>

```

```

<link rel="stylesheet" href="index.css" />
<style>
  #loader {
    color: #008cff;
    height: 40px;
    left: 45%;
    position: absolute;
    top: 45%;
    width: 30%;
  }
</style>
</head>
<body>
  <div id='element'>
    <div id='loader'>Loading....</div></div>
</body>
</html>

```

Integrate badge into listview in React Badge component

The badges can be integrated with the `listview` component to indicate new notification with colour based on priority.

In the following sample, `default` badges are used and there is no need to customize the badge size. The component will automatically adjust the size based on the container element.

INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { ListViewComponent } from '@syncfusion/ej2-react-lists';
function App() {
  // Datasource for listview, badge field is the class data for Badges
  let dataSource = [
    { id: 'p_01', text: 'Primary', messages: '3 New', badge: 'e-badge e-badge-primary', icons: 'primary', type: 'Primary' },
    { id: 'p_02', text: 'Social', messages: '27 New', badge: 'e-badge e-badge-secondary', icons: 'social', type: 'Primary' },
    { id: 'p_03', text: 'Promotions', messages: '7 New', badge: 'e-badge e-badge-success', icons: 'promotion', type: 'Primary' },
    { id: 'p_04', text: 'Updates', messages: '13 New', badge: 'e-badge e-badge-info', icons: 'updates', type: 'Primary' },
    { id: 'p_05', text: 'Starred', messages: '', badge: '', icons: 'starred', type: 'All Labels' },
    { id: 'p_06', text: 'Important', messages: '2 New', badge: 'e-badge e-badge-danger', icons: 'important', type: 'All Labels' },
    { id: 'p_07', text: 'Sent', messages: '', badge: '', icons: 'sent', type: 'All Labels' },
    { id: 'p_08', text: 'Outbox', messages: '', badge: '', icons: 'outbox', type: 'All Labels' },
    { id: 'p_09', text: 'Drafts', messages: '7 New', badge: 'e-badge e-badge-warning', icons: 'draft', type: 'All Labels' },
  ];
  // Map fields
  let fields = { groupBy: 'type' };

```

```

// Assign template
function template(data) {
    return (<div className='listWrapper' style={{ width: 'inherit',
height: 'inherit' }}>
        <span className={` ${data.icons} list_svg`} >&#160;</span>
        <span className='list_text'>{data.text}</span>
        <span className={` ${data.badge}`} style={{ float: 'right',
marginTop: '16px', fontSize: '12px' }}>{data.messages}</span>
    </div>);
}
function onActionComplete() {
    let list =
document.getElementById('lists').getElementsByClassName('e-list-group-
item')[0];
    list.style.display = 'none';
}
return (<div className="sample_container badge-list">
    { /* ListView element */ }
    <ListViewComponent id="lists" dataSource={dataSource}
fields={fields} headerTitle='Inbox' showHeader={true} template={template}
actionComplete={onActionComplete.bind(this)}></ListViewComponent>
    </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 { ListViewComponent } from '@syncfusion/ej2-react-lists';
function App() {
    // Datasource for listview, badge field is the class data for Badges
    let dataSource: { [key: string]: Object }[] = [
        { id: 'p_01', text: 'Primary', messages: '3 New', badge: 'e-badge e-
badge-primary', icons: 'primary', type: 'Primary' },
        { id: 'p_02', text: 'Social', messages: '27 New', badge: 'e-badge e-
badge-secondary', icons: 'social', type: 'Primary' },
        { id: 'p_03', text: 'Promotions', messages: '7 New', badge: 'e-badge
e-badge-success', icons: 'promotion', type: 'Primary' },
        { id: 'p_04', text: 'Updates', messages: '13 New', badge: 'e-badge e-
badge-info', icons: 'updates', type: 'Primary' },
        { id: 'p_05', text: 'Starred', messages: '', badge: '', icons:
'starred', type: 'All Labels' },
        { id: 'p_06', text: 'Important', messages: '2 New', badge: 'e-badge
e-badge-danger', icons: 'important', type: 'All Labels' },
        { id: 'p_07', text: 'Sent', messages: '', badge: '', icons: 'sent',
type: 'All Labels' },
        { id: 'p_08', text: 'Outbox', messages: '', badge: '', icons:
'outbox', type: 'All Labels' },
        { id: 'p_09', text: 'Drafts', messages: '7 New', badge: 'e-badge e-
badge-warning', icons: 'draft', type: 'All Labels' },
    ];
    // Map fields

```

```

let fields: object = { groupBy: 'type' };
// Assign template
function template(data: any): JSX.Element {
    return (
        <div className='listWrapper' style={{ width: 'inherit', height:
'inherit' }}>
            <span className={` ${data.icons} list_svg`} >&#160;</span>
            <span className='list_text'>{data.text}</span>
            <span className={` ${data.badge}`} style={{ float: 'right',
marginTop: '16px', fontSize: '12px' }}>{data.messages}</span>
        </div>
    );
}

function onActionComplete() {
    let list: HTMLElement =
document.getElementById('lists').getElementsByClassName('e-list-group-
item')[0] as HTMLElement;
    list.style.display = 'none';
}

return (
    <div className="sample_container badge-list">
        { /* Listview element */ }
        <ListViewComponent id="lists" dataSource={dataSource}
fields={fields} headerTitle='Inbox' showHeader={true} template={template as
any} actionComplete={onActionComplete.bind(this)} ></ListViewComponent>
    </div>
);
}

export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Essential JS 2 React Badge Sample</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/ej2-
base/styles/material.css" rel="stylesheet" />
    <link href="https://cdn.syncfusion.com/ej2/20.3.56/ej2-
notifications/styles/material.css" rel="stylesheet" />
    <link href="https://cdn.syncfusion.com/ej2/20.3.56/ej2-react-
lists/styles/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>
    <link rel="stylesheet" href="index.css" />
    <style>
        #loader {
            color: #008cff;

```

```

        height: 40px;
        left: 45%;
        position: absolute;
        top: 45%;
        width: 30%;
    }
</style>
</head>
<body>
    <div id='element'>
        <div id='loader'>Loading....</div></div>
</body>
</html>

```

Dynamic badge content in React Badge component

Badges in real-time needs to be updated dynamically based on the requirements. In this sample, using **React states**

the badges content will be updated dynamically. Click the increment button to change the badge value.

INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { ListViewComponent } from '@syncfusion/ej2-react-lists';
function BadgePortable(props) {
    return (<span className={props.BadgeType} style={{ float: 'right',
marginTop: '16px', fontSize: '12px' }}>{props.BadgeContent} New
        </span>);
}
function App(props) {
    // Datasource for listview, badge field is the class data for Badges
    let dataSource = [
        { id: 'p_01', text: 'Primary', badge: 'e-badge e-badge-primary',
icons: 'primary', type: 'Primary' },
        { id: 'p_02', text: 'Social', badge: 'e-badge e-badge-secondary',
icons: 'social', type: 'Primary' },
        { id: 'p_03', text: 'Promotions', badge: 'e-badge e-badge-success',
icons: 'promotion', type: 'Primary' },
        { id: 'p_04', text: 'Updates', badge: 'e-badge e-badge-info', icons:
'updates', type: 'Primary' },
        { id: 'p_05', text: 'Starred', badge: '', icons: 'starred', type:
'All Labels' },
        { id: 'p_06', text: 'Important', badge: 'e-badge e-badge-danger',
icons: 'important', type: 'All Labels' },
        { id: 'p_07', text: 'Sent', badge: '', icons: 'sent', type: 'All
Labels' },
        { id: 'p_08', text: 'Outbox', badge: '', icons: 'outbox', type: 'All
Labels' },
        { id: 'p_09', text: 'Drafts', badge: 'e-badge e-badge-warning',
icons: 'draft', type: 'All Labels' },
    ];
    // Map fields
    let fields = { groupBy: 'type' };
    let Values = {

```



```

        Primary: 3,
        Social: 27,
        Promotions: 7,
        Updates: 13,
        Drafts: 7,
        Important: 2
    };

    function listTemplate(data) {
        return (<div className='listWrapper' style={{ width: 'inherit',
height: 'inherit' }}>
            <span className={` ${data.icons} list_svg`} >&#160;</span>
            <span className='list_text'>{data.text}</span>
            {data.badge !== '' ?
                <BadgePortable BadgeContent={Values[data.text]}
BadgeType={data.badge}/> : ''}
            </div>);
    }

    function onClick() {
        let badgeElements =
Array.prototype.slice.call(document.getElementById('lists').getElementsByClas
sName('e-badge'));
        badgeElements.forEach((element) => {
            element.textContent = (Number(element.textContent.split(' ')[0]))
+ 1 + ' New';
        });
    }

    return (<div className="sample_container badge-list">
        /* Listview element */
        <ListViewComponent id="lists" dataSource={dataSource}
fields={fields} headerTitle='Inbox' showHeader={true}
template={listTemplate.bind(this)}></ListViewComponent>
        <p className='crossline'></p>
        <span className='incr_button'>
            <button className='e-btn e-primary'
onClick={onClick.bind(this)}>Increment Badge Count</button>
        </span>
    </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 { ListViewComponent } from '@syncfusion/ej2-react-lists';
interface IBadgeValuesProps {
    BadgeType: string;
    BadgeContent: string;
}
function BadgePortable(props: IBadgeValuesProps) {
    return (
        <span className={props.BadgeType}

```

```

        style={{ float: 'right', marginTop: '16px', fontSize: '12px'
    }}>{props.BadgeContent} New
    </span>
  );
}
interface IBadgeValues {
  Primary: number;
  Social: number;
  Promotions: number;
  Updates: number;
  Important: number;
  Drafts: number;
}
function App(props:IBadgeValues) {
  // DataSource for listview, badge field is the class data for Badges
  let dataSource: { [key: string]: Object }[] = [
    { id: 'p_01', text: 'Primary', badge: 'e-badge e-badge-primary',
icons: 'primary', type: 'Primary' },
    { id: 'p_02', text: 'Social', badge: 'e-badge e-badge-secondary',
icons: 'social', type: 'Primary' },
    { id: 'p_03', text: 'Promotions', badge: 'e-badge e-badge-success',
icons: 'promotion', type: 'Primary' },
    { id: 'p_04', text: 'Updates', badge: 'e-badge e-badge-info', icons:
'updates', type: 'Primary' },
    { id: 'p_05', text: 'Starred', badge: '', icons: 'starred', type:
'All Labels' },
    { id: 'p_06', text: 'Important', badge: 'e-badge e-badge-danger',
icons: 'important', type: 'All Labels' },
    { id: 'p_07', text: 'Sent', badge: '', icons: 'sent', type: 'All
Labels' },
    { id: 'p_08', text: 'Outbox', badge: '', icons: 'outbox', type: 'All
Labels' },
    { id: 'p_09', text: 'Drafts', badge: 'e-badge e-badge-warning',
icons: 'draft', type: 'All Labels' },
  ];
  // Map fields
  let fields: object = { groupBy: 'type' };
  let Values = {
    Primary: 3,
    Social: 27,
    Promotions: 7,
    Updates: 13,
    Drafts: 7,
    Important: 2
  };
  function listTemplate(data: any): JSX.Element {
    return (
      <div className='listWrapper' style={{ width: 'inherit', height:
'inherit' }}>
        <span className={` ${data.icons} list_svg`} >&#160;</span>
        <span className='list_text'>{data.text}</span>
        {
          data.badge !== '' ?
            <BadgePortable BadgeContent={Values[data.text]}
BadgeType={data.badge} /> : ''
        }
      </div>
    );
  }
}

```

```

    );
  }
  function onClick(): void {
    let badgeElements =
Array.prototype.slice.call(document.getElementById('lists').getElementsByClas
sName('e-badge'));
    badgeElements.forEach((element) => {
      element.textContent = (Number(element.textContent.split(' ')[0])) +
1 + ' New';
    })
  }
  return (
    <div className="sample_container badge-list">
      { /* Listview element */ }
      <ListViewComponent id="lists" dataSource={dataSource}
fields={fields} headerTitle='Inbox' showHeader={true}
template={listTemplate.bind(this) as any}></ListViewComponent>
      <p className='crossline'></p>
      <span className='incr_button'>
        <button className='e-btn e-primary'
onClick={onClick.bind(this)}>Increment Badge Count</button>
      </span>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById("element"));
{% endraw %}

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Essential JS 2 React Badge Sample</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/ej2-
base/styles/material.css" rel="stylesheet" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/ej2-
notifications/styles/material.css" rel="stylesheet" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/ej2-react-
lists/styles/material.css" rel="stylesheet" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/ej2-react-
buttons/styles/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>
  <link rel="stylesheet" href="index.css" />
  <style>
    #loader {
      color: #008cff;
      height: 40px;
    }
  </style>

```

```
        left: 45%;
        position: absolute;
        top: 45%;
        width: 30%;
    }
</style>
</head>
<body>
    <div id='element'>
        <div id='loader'>Loading....</div></div>
</body>
</html>
```

Barcode

<!-- markdownlint-disable MD036 -->

Getting Started

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

Dependencies

Below is the list of minimum dependencies required to use the barcode component.

```
`javascript
|-- @syncfusion/ej2-react-barcode-generator
|-- @syncfusion/ej2-base
|-- @syncfusion/ej2-data
|-- @syncfusion/ej2-barcode-generator
|-- @syncfusion/ej2-react-base
|-- @syncfusion/ej2-pdf-export
|-- @syncfusion/ej2-file-utils
|-- @syncfusion/ej2-compression
|-- @syncfusion/ej2-svg-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 setup basic `React` sample use following commands.

```

<div class='tsx'>
,

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

</div>
<div class='jsx'>
,

create-react-app quickstart
cd quickstart
,

</div>

```

- Install Syncfusion packages using below command.

```

,

npm install @syncfusion/ej2-react-barcodegenerator --save
,

```

Adding Barcode Generator control

You can start adding Essential JS 2 barcode-generator component to the application. To get started, add the barcode component in `app.ts` and `index.html` files using the following code.

Place the following barcode-generator code in the `app.ts`.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { BarcodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(<BarcodeGeneratorComponent id="barcode" width={"200px"}
height={"150px"} type='Codabar'
value='123456789'></BarcodeGeneratorComponent>,
document.getElementById("barcode"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { BarcodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(
  <BarcodeGeneratorComponent
    id="barcode"

```

```

        width={"200px"}
        height={"150px"}
        type='Codabar'
        value='123456789'
    ></BarcodeGeneratorComponent>,
    document.getElementById("barcode")
);

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Syncfusion React Chart-DataLabel</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" />
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
    <script src="systemjs.config.js"></script>
    <style>
        #loader {
            color: #008cff;
            height: 40px;
            left: 45%;
            position: absolute;
            top: 45%;
            width: 30%;
        }
        .barcodeStyle{
            height: 150px;
            width: 200px;
            padding-left: 40%;
            padding-top: 9%;
        }
    </style>
</head>
<body>
    <div id='barcode'class="barcodeStyle">
        <div id='loader'>Loading....</div>
    </div>
</body>
</html>

```

Adding QR Generator control

You can add the QR code in our barcode generator component.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { QRCodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-
generator';

```

```
ReactDOM.render(<QRCodeGeneratorComponent id="barcode" width={"200px"}
height={"150px"} value='Syncfusion'></QRCodeGeneratorComponent>,
document.getElementById("barcode"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { QRCodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(
  <QRCodeGeneratorComponent
    id="barcode"
    width={"200px"}
    height={"150px"}
    value='Syncfusion'
  ></QRCodeGeneratorComponent>,
  document.getElementById("barcode")
);
```

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Chart-DataLabel</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></script>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
    .barcodeStyle{
      height: 150px;
      width: 200px;
      padding-left: 40%;
      padding-top: 9%;
    }
  </style>
</head>
<body>
  <div id='barcode' class="barcodeStyle">
    <div id='loader'>Loading....</div>
  </div>
</body>
```

```
</html>
```

Adding Datamatrix Generator control

You can add the datamatrix code in our barcode generator component.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { DataMatrixGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(<DataMatrixGeneratorComponent id="barcode" width={"200px"} height={"150px"} value='Syncfusion'></DataMatrixGeneratorComponent>, document.getElementById("barcode"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { DataMatrixGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(
  <DataMatrixGeneratorComponent
    id="barcode"
    width={"200px"}
    height={"150px"}
    value='Syncfusion'
  ></DataMatrixGeneratorComponent>,
  document.getElementById("barcode")
);
```

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Chart-DataLabel</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></script>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
    .barcodeStyle{
```



```

        height: 150px;
        width: 200px;
        padding-left: 40%;
        padding-top: 9%;
    }
</style>
</head>
<body>
    <div id='barcode' class="barcodeStyle">
        <div id='loader'>Loading....</div>
    </div>
</body>
</html>

```

BarcodeGenerator in React Barcode component

Code39

The Code 39 character set includes the digits 0-9, the letters A-Z (upper case only), and the symbols: space, minus (-), plus (+), period (.), dollar sign (\$), slash (/), and percent (%). A special start / stop character is placed at the beginning and ending of each barcode. The barcode can be of any length; even more than 25 characters begin to push the bounds. Code 39 is the only type of barcode that does not require a checksum for common use.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { BarcodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(<BarcodeGeneratorComponent id="barcode" width={"200px"}
height={"150px"} type='Code39'
value='SYNCFUSION'></BarcodeGeneratorComponent>,
document.getElementById("barcode"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { BarcodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(
    <BarcodeGeneratorComponent
        id="barcode"
        width={"200px"}
        height={"150px"}
        type='Code39'
        value='SYNCFUSION'
    ></BarcodeGeneratorComponent>,
    document.getElementById("barcode")
);

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">

```

```

<head>
  <title>Syncfusion React Chart-DataLabel</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
    .barcodeStyle{
      height: 150px;
      width: 200px;
      padding-left: 40%;
      padding-top: 9%;
    }
  </style>
</head>
<body>
  <div id='barcode'class="barcodeStyle">
    <div id='loader'>Loading....</div>
  </div>
</body>
</html>

```

Code39 Extended

Code 39 Extended is an extended version of Code 39 that supports ASCII character set. In Code 39 Extended, you can also code 26 lower letters (a-z) and the special characters in the keyboard.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { BarcodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(<BarcodeGeneratorComponent id="barcode" width={"200px"}
height={"150px"} type='Code39Extension'
value='SYNCFUSION'></BarcodeGeneratorComponent>,
document.getElementById("barcode"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { BarcodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';

```

```
ReactDOM.render(
  <BarcodeGeneratorComponent
    id="barcode"
    width={"200px"}
    height={"150px"}
    type='Code39Extension'
    value='SYNCFUSION'
  ></BarcodeGeneratorComponent>,
  document.getElementById("barcode")
);
```

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>SynCFusion React Chart-DataLabel</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cFF;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
    .barcodeStyle{
      height: 150px;
      width: 200px;
      padding-left: 40%;
      padding-top: 9%;
    }
  </style>
</head>
<body>
  <div id='barcode' class="barcodeStyle">
    <div id='loader'>Loading....</div>
  </div>
</body>
</html>
```

Code 11

Code 11 is used primarily for labeling the telecommunication equipment's. The character set includes the digits 0 to 9, a dash (-), and a start / stop code.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { BarcodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(<BarcodeGeneratorComponent id="barcode" width={"200px"}
height={"150px"} type='Code11' value='112'></BarcodeGeneratorComponent>,
document.getElementById("barcode"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { BarcodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(
  <BarcodeGeneratorComponent
    id="barcode"
    width={"200px"}
    height={"150px"}
    type='Code11'
    value='112'
  ></BarcodeGeneratorComponent>,
  document.getElementById("barcode")
);
```

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Chart-DataLabel</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
    .barcodeStyle{
      height: 150px;
      width: 200px;
      padding-left: 40%;
      padding-top: 9%;
    }
  </style>
</head>
```

```
<body>
  <div id='barcode' class="barcodeStyle">
    <div id='loader'>Loading....</div>
  </div>
</body>
</html>
```

Codabar

Codabar is a variable length symbol that encodes the following 20 characters:

0123456789-\$.+ABCD

The characters, A, B, C and D are used as start and stop characters. Codabar is used in libraries, blood banks, the package delivery industry and a variety of other information processing applications.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { BarcodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(<BarcodeGeneratorComponent id="barcode" width={"200px"}
height={"150px"} type='Codabar'
value='123456789'></BarcodeGeneratorComponent>,
document.getElementById("barcode"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { BarcodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(
<BarcodeGeneratorComponent
  id="barcode"
  width={"200px"}
  height={"150px"}
  type='Codabar'
  value='123456789'
></BarcodeGeneratorComponent>,
document.getElementById("barcode")
);
```

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Chart-DataLabel</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" />
```

```

<script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
<script src="systemjs.config.js"></script>
<style>
  #loader {
    color: #008cff;
    height: 40px;
    left: 45%;
    position: absolute;
    top: 45%;
    width: 30%;
  }
  .barcodeStyle{
    height: 150px;
    width: 200px;
    padding-left: 40%;
    padding-top: 9%;
  }
</style>
</head>
<body>
  <div id='barcode' class="barcodeStyle">
    <div id='loader'>Loading....</div>
  </div>
</body>
</html>

```

Code 32

Code 32 is mainly used for coding pharmaceuticals, cosmetics and dietetics. It is often used to encode Italian Pharmacode that has the following structure:

- 'A' character (ASCII 65), that is not really encoded.
- 8 digits for Pharmacode (It generally begins with / and prefixed with 0).
- 1 digit for checksum module 10, that is automatically calculated by barcode.

The value to be encoded must be 8 digits Pharmacode (prefix it with '0' if necessary) and the 9th digit (the checksum) is automatically calculated by barcode.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { BarcodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(<BarcodeGeneratorComponent id="barcode" width={"200px"}
height={"150px"} type='Code32' value='01234567'></BarcodeGeneratorComponent>,
document.getElementById("barcode"));

```

INDEX.TSX

```

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

```

```
import { BarcodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(
  <BarcodeGeneratorComponent
    id="barcode"
    width={"200px"}
    height={"150px"}
    type='Code32'
    value='01234567'
  ></BarcodeGeneratorComponent>,
  document.getElementById("barcode")
);
```

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Chart-DataLabel</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></script>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
    .barcodeStyle{
      height: 150px;
      width: 200px;
      padding-left: 40%;
      padding-top: 9%;
    }
  </style>
</head>
<body>
  <div id='barcode' class="barcodeStyle">
    <div id='loader'>Loading....</div>
  </div>
</body>
</html>
```

Code 93

Code 93 is designed to complement and improve upon Code 39. It can represent the entire ASCII character set by using combinations of 2 characters. Code 93 is a continuous, variable-length symbology

and produces denser code. The Standard Mode (default implementation) can encode uppercase letters (A-Z), digits (0-9), and special characters like *, -, \$, %, (Space), ., /, and +.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { BarcodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(<BarcodeGeneratorComponent id="barcode" width={"200px"}
height={"150px"} type='Code93' value='01234567'></BarcodeGeneratorComponent>,
document.getElementById("barcode"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { BarcodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(
  <BarcodeGeneratorComponent
    id="barcode"
    width={"200px"}
    height={"150px"}
    type='Code93'
    value='01234567'
  ></BarcodeGeneratorComponent>,
  document.getElementById("barcode")
);
```

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Chart-DataLabel</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></script>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
    .barcodeStyle{
      height: 150px;
      width: 200px;
```



```

        padding-left: 40%;
        padding-top: 9%;
    }
</style>
</head>
<body>
    <div id='barcode' class="barcodeStyle">
        <div id='loader'>Loading....</div>
    </div>
</body>
</html>

```

Code 93 Extended

The Code 93 Extended Barcode symbology is continuous, variable length, and self-checking. It is based on Code 93 Barcode. The Extended Version can encode all 128 ASCII characters.

Code 128

Code 128 is a variable length, high density, alphanumeric, linear bar code symbology, capable of encoding the full 128-character ASCII character set and extended character sets. This symbology includes a checksum digit for verification and the barcode can also be verified character-by-character by verifying the parity of each data byte.

Code 128 Code Sets

- Code Set A (or Chars Set A) includes all of the standard upper case U.S. alphanumeric keyboard characters and punctuation characters along with the control characters, (namely, characters with ASCII values from 0 to 95 inclusive), and seven special characters.
- Code Set B (or Chars Set B) includes all of the standard upper case alphanumeric keyboard characters and punctuation characters along with the lower case alphabetic characters (namely, characters with ASCII values from 32 to 127 inclusive), and seven special characters.
- Code Set C (or Chars Set C) includes the set of 100 digit pairs from 00 to 99 inclusive along with three special characters. This allows numeric data to be encoded as two data digits per symbol character, at effectively twice the density of standard data.

Code 128 Special characters

The last seven characters of Code Sets A and B (character values 96 - 102) and the last three characters of Code Set C (character values 100 - 102) are special non-data characters with no ASCII character equivalents that have a particular significance to the Barcode reading device.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { BarcodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(<BarcodeGeneratorComponent id="barcode" width={"200px"}
height={"150px"} type='Code128'
value='SYNCFUSION'></BarcodeGeneratorComponent>,
document.getElementById("barcode"));

```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { BarcodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(
  <BarcodeGeneratorComponent
    id="barcode"
    width={"200px"}
    height={"150px"}
    type='Code128'
    value='SYNCFUSION'
  ></BarcodeGeneratorComponent>,
  document.getElementById("barcode")
);
```

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Chart-DataLabel</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></script>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
    .barcodeStyle{
      height: 150px;
      width: 200px;
      padding-left: 40%;
      padding-top: 9%;
    }
  </style>
</head>
<body>
  <div id='barcode'class="barcodeStyle">
    <div id='loader'>Loading....</div>
  </div>
</body>
</html>
```

Customizing the Barcode color

A page or printed media with barcode often appears colorful in the background and surrounding region with other contents. In such cases the barcode can also be customized to suit the needs. You can achieve this by using for forecolor property .

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { BarcodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(<BarcodeGeneratorComponent id="barcode" width={"200px"} height={"150px"} foreColor={"red"} type='Code128' value='SYNCFUSION'></BarcodeGeneratorComponent>, document.getElementById("barcode"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { BarcodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(
  <BarcodeGeneratorComponent
    id="barcode"
    width={"200px"}
    height={"150px"}
    foreColor={"red"}
    type='Code128'
    value='SYNCFUSION'
  ></BarcodeGeneratorComponent>,
  document.getElementById("barcode")
);
```

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Chart-DataLabel</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></script>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008c00;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
```

```

        width: 30%;
    }
    .barcodeStyle{
        height: 150px;
        width: 200px;
        padding-left: 40%;
        padding-top: 9%;
    }
</style>
</head>
<body>
    <div id='barcode' class="barcodeStyle">
        <div id='loader'>Loading....</div>
    </div>
</body>
</html>

```

Customizing the Barcode dimension

The dimension of the barcode can be changed using the height and width property of the barcodegenerator.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { BarcodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(<BarcodeGeneratorComponent id="barcode" width={"300px"}
height={"300px"} type='Code128'
value='SYNCFUSION'></BarcodeGeneratorComponent>,
document.getElementById("barcode"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { BarcodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(
<BarcodeGeneratorComponent
    id="barcode"
    width={"300px"}
    height={"300px"}
    type='Code128'
    value='SYNCFUSION'
></BarcodeGeneratorComponent>,
    document.getElementById("barcode")
);

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Syncfusion React Chart-DataLabel</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" />
<script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
<script src="systemjs.config.js"></script>
<style>
    #loader {
        color: #008cff;
        height: 40px;
        left: 45%;
        position: absolute;
        top: 45%;
        width: 30%;
    }
    .barcodeStyle{
        height: 150px;
        width: 200px;
        padding-left: 40%;
        padding-top: 9%;
    }
</style>
</head>
<body>
    <div id='barcode' class="barcodeStyle">
        <div id='loader'>Loading....</div>
    </div>
</body>
</html>

```

Customizing the text

In barcode generators you can customize the barcode text by using display text property .

INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { BarcodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(<BarcodeGeneratorComponent id="barcode" width={"300px"}
displayText={{ text: "text" }} height={"300px"} type='Code128'
value='SYNCFUSION'></BarcodeGeneratorComponent>,
document.getElementById("barcode"));
{% endraw %}

```

INDEX.TSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { BarcodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(

```

```

<BarcodeGeneratorComponent
  id="barcode"
  width={"300px"}
  displayText={{ text: "text" }}
  height={"300px"}
  type='Code128'
  value='SYNCFUSION'
></BarcodeGeneratorComponent>,
document.getElementById("barcode")
);
{% enddraw %}

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>SynCFusion React Chart-DataLabel</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
    .barcodeStyle{
      height: 150px;
      width: 200px;
      padding-left: 40%;
      padding-top: 9%;
    }
  </style>
</head>
<body>
  <div id='barcode' class="barcodeStyle">
    <div id='loader'>Loading....</div>
  </div>
</body>
</html>

```

Qrcodegenerator in React Barcode component

QR Code

A QR Code is a two-dimensional barcode that consists of a grid of dark and light dots or blocks that form a square. The data encoded in the barcode can be numeric, alphanumeric, or Shift Japanese Industrial

Standards (JIS8) characters. The QR Code uses version from 1 to 40. Version 1 measures 21 modules x 21 modules, Version 2 measures 25 modules x 25 modules, and so on. The number of modules increases in steps of 4 modules per side up to Version 40 that measures 177 modules x 177 modules. Each version has its own capacity. By default, the barcode control automatically set the version according to the length of the input text. The QR Barcodes are designed for industrial uses and also commonly used in consumer advertising.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { QRCodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(<QRCodeGeneratorComponent id="barcode" width={"200px"}
height={"150px"} value='SynCFusion'></QRCodeGeneratorComponent>,
document.getElementById("barcode"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { QRCodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(
  <QRCodeGeneratorComponent
    id="barcode"
    width={"200px"}
    height={"150px"}
    value='SynCFusion'
  ></QRCodeGeneratorComponent>,
  document.getElementById("barcode")
);
```

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>SynCFusion React Chart-DataLabel</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
```

```

    }
    .barcodeStyle{
        height: 150px;
        width: 200px;
        padding-left: 40%;
        padding-top: 9%;
    }
</style>
</head>
<body>
    <div id='barcode'class="barcodeStyle">
        <div id='loader'>Loading....</div>
    </div>
</body>
</html>

```

Customizing the Barcode color

A page or printed media with barcode often appears colorful in the background and surrounding region with other contents. In such cases the barcode can also be customized to suit the needs. You can achieve this by using for forecolor property .

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { QRCodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(<QRCodeGeneratorComponent id="barcode" width={"200px"}
height={"150px"} foreColor={"red"}
value='Syncfusion'></QRCodeGeneratorComponent>,
document.getElementById("barcode"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { QRCodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(
    <QRCodeGeneratorComponent
        id="barcode"
        width={"200px"}
        height={"150px"}
        foreColor={"red"}
        value='Syncfusion'
    ></QRCodeGeneratorComponent>,
    document.getElementById("barcode")
);

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Syncfusion React Chart-DataLabel</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" />
<script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
<script src="systemjs.config.js"></script>
<style>
    #loader {
        color: #008cff;
        height: 40px;
        left: 45%;
        position: absolute;
        top: 45%;
        width: 30%;
    }
    .barcodeStyle{
        height: 150px;
        width: 200px;
        padding-left: 40%;
        padding-top: 9%;
    }
</style>
</head>
<body>
    <div id='barcode' class="barcodeStyle">
        <div id='loader'>Loading....</div>
    </div>
</body>
</html>

```

Customizing the Barcode dimension

The dimension of the barcode can be changed using the height and width properties of the barcodegenerator.

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { QRCodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-
generator';
ReactDOM.render(<QRCodeGeneratorComponent id="barcode" width={"100px"}
height={"100px"} value='Syncfusion'></QRCodeGeneratorComponent>,
document.getElementById("barcode"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { QRCodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-
generator';
ReactDOM.render(
    <QRCodeGeneratorComponent
        id="barcode"

```

```

        width={"100px"}
        height={"100px"}
        value='Syncfusion'
    </QRCodeGeneratorComponent>,
    document.getElementById("barcode")
);

```

INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Syncfusion React Chart-DataLabel</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" />
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
    <script src="systemjs.config.js"></script>
    <style>
        #loader {
            color: #008cff;
            height: 40px;
            left: 45%;
            position: absolute;
            top: 45%;
            width: 30%;
        }
        .barcodeStyle{
            height: 150px;
            width: 200px;
            padding-left: 40%;
            padding-top: 9%;
        }
    </style>
</head>
<body>
    <div id='barcode'class="barcodeStyle">
        <div id='loader'>Loading....</div>
    </div>
</body>
</html>

```

Customizing the text

In barcode generators You can customize the barcode text by using display text property .

INDEX.JSX

```

{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { QRCodeGeneratorComponent } from '@syncfusion/ej2-react-barcode-
generator';

```

```
ReactDOM.render(<QRCodeGeneratorComponent id="barcode" width={"200px"}
height={"150px"} displayText={{ text: "text" }}
value='SynCFusion'></QRCodeGeneratorComponent>,
document.getElementById("barcode"));
{% endraw %}
```

INDEX.TSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { QRCodeGeneratorComponent } from '@synCFusion/ej2-react-barcode-
generator';
ReactDOM.render(
  <QRCodeGeneratorComponent
    id="barcode"
    width={"200px"}
    height={"150px"}
    displayText={{ text: "text" }}
    value='SynCFusion'
  ></QRCodeGeneratorComponent>,
  document.getElementById("barcode")
);
{% endraw %}
```

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>SynCFusion React Chart-DataLabel</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cFF;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
    .barcodeStyle{
      height: 150px;
      width: 200px;
      padding-left: 40%;
      padding-top: 9%;
    }
  </style>
</head>
```

```
<body>
  <div id='barcode' class="barcodeStyle">
    <div id='loader'>Loading....</div>
  </div>
</body>
</html>
```

Datamatrixgenerator in React Barcode component

Data Matrix

DataMatrix Barcode is a two dimensional barcode that consists of a grid of dark and light dots or blocks forming square or rectangular symbol. The data encoded in the barcode can either be numbers or alphanumeric. They are widely used in printed media such as labels and letters. You can read it easily with the help of a barcode reader and mobile phones.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { DataMatrixGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(<DataMatrixGeneratorComponent id="barcode" width={"200px"} height={"150px"} value='Syncfusion'></DataMatrixGeneratorComponent>,
document.getElementById("barcode"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { DataMatrixGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(
  <DataMatrixGeneratorComponent
    id="barcode"
    width={"200px"}
    height={"150px"}
    value='Syncfusion'
  ></DataMatrixGeneratorComponent>,
  document.getElementById("barcode")
);
```

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Chart-DataLabel</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></script>
  <script src="systemjs.config.js"></script>
```

```

<style>
  #loader {
    color: #008cff;
    height: 40px;
    left: 45%;
    position: absolute;
    top: 45%;
    width: 30%;
  }
  .barcodeStyle{
    height: 150px;
    width: 200px;
    padding-left: 40%;
    padding-top: 9%;
  }
</style>
</head>
<body>
  <div id='barcode' class="barcodeStyle">
    <div id='loader'>Loading....</div>
  </div>
</body>
</html>

```

Customizing the Barcode color

A page or printed media with barcode often appears colorful in the background and surrounding region with other contents. In such cases the barcode can also be customized to suit the needs. You can achieve this by using the forecolor property .

INDEX.JSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { DataMatrixGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(<DataMatrixGeneratorComponent id="barcode" width={"200px"}
height={"150px"} foreColor={"red"}
value='Syncfusion'></DataMatrixGeneratorComponent>,
document.getElementById("barcode"));

```

INDEX.TSX

```

import * as React from "react";
import * as ReactDOM from "react-dom";
import { DataMatrixGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(
  <DataMatrixGeneratorComponent
    id="barcode"
    width={"200px"}
    height={"150px"}
    foreColor={"red"}
    value='Syncfusion'
  ></DataMatrixGeneratorComponent>,
  document.getElementById("barcode")
)

```

```
);
```

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Chart-DataLabel</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
    .barcodeStyle{
      height: 150px;
      width: 200px;
      padding-left: 40%;
      padding-top: 9%;
    }
  </style>
</head>
<body>
  <div id='barcode'class="barcodeStyle">
    <div id='loader'>Loading....</div>
  </div>
</body>
</html>
```

Customizing the Barcode dimension

The dimension of the barcode can be changed using the height and width property of the barcodegenerator.

INDEX.JSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { DataMatrixGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(<DataMatrixGeneratorComponent id="barcode" width={"100px"}
height={"100px"} value='Syncfusion'></DataMatrixGeneratorComponent>,
document.getElementById("barcode"));
```

INDEX.TSX

```
import * as React from "react";
import * as ReactDOM from "react-dom";
import { DataMatrixGeneratorComponent } from '@syncfusion/ej2-react-
barcode-generator';
ReactDOM.render(
  <DataMatrixGeneratorComponent
    id="barcode"
    width={"100px"}
    height={"100px"}
    value='Syncfusion'
  ></DataMatrixGeneratorComponent>,
  document.getElementById("barcode")
);
```

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Chart-DataLabel</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
      width: 30%;
    }
    .barcodeStyle{
      height: 150px;
      width: 200px;
      padding-left: 40%;
      padding-top: 9%;
    }
  </style>
</head>
<body>
  <div id='barcode'class="barcodeStyle">
    <div id='loader'>Loading....</div>
  </div>
</body>
</html>
```

Customizing the text

In barcode generators you can customize the barcode text by using the display text property .

INDEX.JSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { DataMatrixGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(<DataMatrixGeneratorComponent id="barcode" width={"200px"}
height={"150px"} displayText={{ text: "text" }}
value='Syncfusion'></DataMatrixGeneratorComponent>,
document.getElementById("barcode"));
{% endraw %}
```

INDEX.TSX

```
{% raw %}
import * as React from "react";
import * as ReactDOM from "react-dom";
import { DataMatrixGeneratorComponent } from '@syncfusion/ej2-react-barcode-generator';
ReactDOM.render(
  <DataMatrixGeneratorComponent
    id="barcode"
    width={"200px"}
    height={"150px"}
    displayText={{ text: "text" }}
    value='Syncfusion'
  ></DataMatrixGeneratorComponent>,
  document.getElementById("barcode")
);
{% endraw %}
```

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Chart-DataLabel</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" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></script>
  <script src="systemjs.config.js"></script>
  <style>
    #loader {
      color: #008cff;
      height: 40px;
      left: 45%;
      position: absolute;
      top: 45%;
```



```

        width: 30%;
    }
    .barcodeStyle{
        height: 150px;
        width: 200px;
        padding-left: 40%;
        padding-top: 9%;
    }
</style>
</head>
<body>
    <div id='barcode' class="barcodeStyle">
        <div id='loader'>Loading....</div>
    </div>
</body>
</html>

```

Export in React Barcode component

Export barcode as an image and base64 string is common for barcode,QRcode and datamatrix. The following code samples explain how to export the barcode as an image and base64 string.

Export

Barcode provides the support to export its content as an image in the specified image type and downloads it in the browser.

The following code example shows how to export the barcode as an image

```

`ts
let barcodeInstance;
ReactDOM.render(
  <BarcodeGeneratorComponent
    id="barcode"
    ref={barcode => (barcodeInstance = barcode)}
    width={"200px"}
    height={"150px"}
    type='Code39'
    value='SYNCFUSION'
  </BarcodeGeneratorComponent>,
  document.getElementById("barcode")
);
let filename = 'Export';
barcodeInstance.exportImage(filename,'JPG');
`

```

The filename specifies the name of the file to be downloaded

[Export As Base64String](#)

Barcode provides support to export its content as an image in the specified image type and returns it as base64 string.

The following code example illustrates how to export the barcode as a base64 string

```
`ts
let barcodeInstance;

ReactDOM.render(
  <BarcodeGeneratorComponent
    id="barcode"
    ref={barcode => (barcodeInstance = barcode)}
    width={"200px"}
    height={"150px"}
    type='Code39'
    value='SYNCFUSION'
  </BarcodeGeneratorComponent>,
  document.getElementById("barcode")
);

base64string();

async function base64string() {
  let base64 = await barcodeInstance.exportAsBase64Image('JPG');
  console.log(base64);
};
`
```

Note:

Format is to specify the type or format of the exported file. You can export the barcode to the following formats:

- * JPG.

- * PNG.

Breadcrumbs

[Getting Started](#)

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

[Dependencies](#)

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

```
`javascript
|-- @syncfusion/ej2-react-navigations
|-- @syncfusion/ej2-react-base
|-- @syncfusion/ej2-navigations
|-- @syncfusion/ej2-base
|-- @syncfusion/ej2-data
|-- @syncfusion/ej2-inputs
|-- @syncfusion/ej2-lists
|-- @syncfusion/ej2-popups
|-- @syncfusion/ej2-buttons
`,`
```

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

```
<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 **Breadcrumb** component, use the following command

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

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

Adding Style sheet to the Application

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

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

Add Breadcrumb to the project

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

```
`ts
import { BreadcrumbComponent, BreadcrumbItemDirective, BreadcrumbItemsDirective } from
 '@syncfusion/ej2-react-navigations';
import * as React from 'react';
function App() {
  return (<BreadcrumbComponent enableNavigation={false} > </BreadcrumbComponent>);
}
export default App;
`

`ts
import { BreadcrumbComponent } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
function App() {
  return (<BreadcrumbComponent enableNavigation={false}> </BreadcrumbComponent>);
}
export default App;
`
```

Run the application

Run the application in the browser using the following command:

```
`
```

npm start

The following example shows a basic Breadcrumb component.

INDEX.JSX

```
import { enableRipple } from '@syncfusion/ej2-base';
import { BreadcrumbComponent } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
function App() {
    return (<BreadcrumbComponent enableNavigation={false}>
</BreadcrumbComponent>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

INDEX.TSX

```
import { enableRipple } from '@syncfusion/ej2-base';
import { BreadcrumbComponent } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
function App() {
    return (
        <BreadcrumbComponent enableNavigation={false}>
</BreadcrumbComponent>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

Add Items to the Breadcrumb Component

Use `items` property to bind items for Breadcrumb component. The below example demonstrates the basic rendering of Breadcrumb with items support.

INDEX.JSX

```
import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    return (<BreadcrumbComponent enableNavigation={false}>
        <BreadcrumbItemsDirective>
            <BreadcrumbItemDirective iconCss="e-icons e-home"
url="https://ej2.syncfusion.com/home/react.html#platform"/>
            <BreadcrumbItemDirective text="Components"
url="https://ej2.syncfusion.com/react/demos/#/material/grid/overview"/>
            <BreadcrumbItemDirective text="Navigations"
url="https://ej2.syncfusion.com/react/demos/#/material/menu/default"/>
        </BreadcrumbItemsDirective>
    </BreadcrumbComponent>);
}
```

```

        <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb/default"/>
    </BreadcrumbItemsDirective>
</BreadcrumbComponent>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

INDEX.TSX

```

import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    return (
        <BreadcrumbComponent enableNavigation={false}>
            <BreadcrumbItemsDirective>
                <BreadcrumbItemDirective iconCss="e-icons e-home"
url="https://ej2.syncfusion.com/home/react.html#platform" />
                <BreadcrumbItemDirective text="Components"
url="https://ej2.syncfusion.com/react/demos/#/material/grid/overview/" />
                <BreadcrumbItemDirective text="Navigations"
url="https://ej2.syncfusion.com/react/demos/#/material/menu/default" />
                <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb/default" />
            </BreadcrumbItemsDirective>
        </BreadcrumbComponent>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

Enable or Disable Navigation

This feature enables or disables the item navigation. By default, the navigation will be enabled when setting `Url` property. To prevent breadcrumb item navigation, set `enableNavigation` property as `false` in `Breadcrumb`. The below example shows enabling and disabling the navigation of `Breadcrumb` items.

INDEX.JSX

```

import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    return (<div id="breadcrumb-control">
        <div><b>EnableNavigation - false</b></div><br />
        <BreadcrumbComponent enableNavigation={false}>
            <BreadcrumbItemsDirective>
                <BreadcrumbItemDirective iconCss="e-icons e-home"
url="https://ej2.syncfusion.com/home/react.html#platform"/>
                <BreadcrumbItemDirective text="Components"
url="https://ej2.syncfusion.com/react/demos/#/material/grid/overview/" />
                <BreadcrumbItemDirective text="Navigations"
url="https://ej2.syncfusion.com/react/demos/#/material/menu/default"/>

```

```

        <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb/default"/>
    </BreadcrumbItemsDirective>
</BreadcrumbComponent>
<br />
<br />
<div><b>EnableNavigation - true</b></div><br />
<BreadcrumbComponent enableNavigation={true}>
    <BreadcrumbItemsDirective>
        <BreadcrumbItemDirective iconCss="e-icons e-home"
url="https://ej2.syncfusion.com/home/react.html#platform"/>
        <BreadcrumbItemDirective text="Components"
url="https://ej2.syncfusion.com/react/demos/#/material/grid/overview"/>
        <BreadcrumbItemDirective text="Navigations"
url="https://ej2.syncfusion.com/react/demos/#/material/menu/default"/>
        <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb/default"/>
    </BreadcrumbItemsDirective>
</BreadcrumbComponent>
</div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

INDEX.TSX

```

import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    return (
        <div id="breadcrumb-control">
            <div><b>EnableNavigation - false</b></div><br />
            <BreadcrumbComponent enableNavigation={false}>
                <BreadcrumbItemsDirective>
                    <BreadcrumbItemDirective iconCss="e-icons e-home"
url="https://ej2.syncfusion.com/home/react.html#platform" />
                    <BreadcrumbItemDirective text="Components"
url="https://ej2.syncfusion.com/react/demos/#/material/grid/overview/" />
                    <BreadcrumbItemDirective text="Navigations"
url="https://ej2.syncfusion.com/react/demos/#/material/menu/default" />
                    <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb/default" />
                </BreadcrumbItemsDirective>
            </BreadcrumbComponent>
            <br />
            <br/>
            <div><b>EnableNavigation - true</b></div><br />
            <BreadcrumbComponent enableNavigation={true}>
                <BreadcrumbItemsDirective>
                    <BreadcrumbItemDirective iconCss="e-icons e-home"
url="https://ej2.syncfusion.com/home/react.html#platform" />
                    <BreadcrumbItemDirective text="Components"
url="https://ej2.syncfusion.com/react/demos/#/material/grid/overview/" />

```

```

        <BreadcrumbItemDirective text="Navigations"
url="https://ej2.syncfusion.com/react/demos/#/material/menu/default" />
        <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb/default" />
    </BreadcrumbItemsDirective>
</BreadcrumbComponent>
</div>
);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

Data binding in React Breadcrumb component

Items as tag directive

The React Breadcrumb contains **BreadcrumbItemsDirective** and **BreadcrumbItemDirective** tags to render items for the component. To bind items, use **BreadcrumbItemsDirective** tag and **BreadcrumbItemDirective** tag to bind properties for breadcrumb items.

INDEX.JSX

```

import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    return (<BreadcrumbComponent enableNavigation={false}>
        <BreadcrumbItemsDirective>
            <BreadcrumbItemDirective iconCss="e-icons e-home"
url="https://ej2.syncfusion.com/home/react.html#platform"/>
            <BreadcrumbItemDirective text="Components"
url="https://ej2.syncfusion.com/react/demos/#/material/grid/overview"/>
            <BreadcrumbItemDirective text="Navigations"
url="https://ej2.syncfusion.com/react/demos/#/material/menu/default"/>
            <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb/default"/>
        </BreadcrumbItemsDirective>
    </BreadcrumbComponent>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

INDEX.TSX

```

import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    return (
        <BreadcrumbComponent enableNavigation={false}>
            <BreadcrumbItemsDirective>
                <BreadcrumbItemDirective iconCss="e-icons e-home"
url="https://ej2.syncfusion.com/home/react.html#platform" />

```



```

        <BreadcrumbItemDirective text="Components"
url="https://ej2.syncfusion.com/react/demos/#/material/grid/overview/" />
        <BreadcrumbItemDirective text="Navigations"
url="https://ej2.syncfusion.com/react/demos/#/material/menu/default" />
        <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb/default" />
    </BreadcrumbItemsDirective>
</BreadcrumbComponent>
);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

Items based on current URL

The breadcrumb items can be generated from the current URL of the page, if the [url](#) property is not provided or when the user does not specify the breadcrumb items using the `BreadcrumbItemDirective` tag. The following example shows the breadcrumb items that are generated based on the current URL.

INDEX.JSX

```

import { BreadcrumbComponent } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    return (<BreadcrumbComponent
enableNavigation={false}></BreadcrumbComponent>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

INDEX.TSX

```

import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    return (
        <BreadcrumbComponent enableNavigation={false}></BreadcrumbComponent>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

This sample is hosted in different location, so the Breadcrumb Component is rendered with different location instead of the actual location.

Static URL

The breadcrumb items can be generated by providing the [url](#) property in the component, if the user does not specify the breadcrumb items using the `BreadcrumbItemDirective` tag. The following example shows the breadcrumb items generated from the provided url in the component.

INDEX.JSX

```
import { BreadcrumbComponent } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    return (<BreadcrumbComponent enableNavigation={false}
url="https://ej2.syncfusion.com/demos/breadcrumb/bind-to-
location"></BreadcrumbComponent>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

INDEX.TSX

```
import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    return (
        <BreadcrumbComponent enableNavigation={false}
url="https://ej2.syncfusion.com/demos/breadcrumb/bind-to-
location"></BreadcrumbComponent>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

Customize text when generated items using Url

The breadcrumb items text can be customized by using the [beforeItemRender](#) event. In the following example, `bind-to-location` text was customized as `location`.

INDEX.JSX

```
import { BreadcrumbComponent } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    function beforeItemRenderHandler(args) {
        if (args.item.text === 'bind-to-location') {
            args.item.text = 'location';
        }
    }
    return (<BreadcrumbComponent enableActiveItemNavigation={true}
url="https://ej2.syncfusion.com/demos/breadcrumb/bind-to-location"
beforeItemRender={beforeItemRenderHandler}>
        </BreadcrumbComponent>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

INDEX.TSX

```
import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
```

```
import { BreadcrumbBeforeItemRenderEventArgs } from '@syncfusion/ej2-
navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    function beforeItemRenderHandler(args:
BreadcrumbBeforeItemRenderEventArgs): void {
        if (args.item.text === 'bind-to-location') {
            args.item.text = 'location';
        }
    }
    return (
        <BreadcrumbComponent enableActiveItemNavigation={true}
url="https://ej2.syncfusion.com/demos/breadcrumb/bind-to-location"
beforeItemRender={beforeItemRenderHandler}>
        </BreadcrumbComponent>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

Icons in React Breadcrumb component

The Breadcrumb component contains an icon/image to provide a visual representation of an item.

Loading Icons in Breadcrumb Item

To load the icon on the breadcrumb item, set the [iconCss](#) property.

Breadcrumb with font icons

To place the font icon on the breadcrumb 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 item.

INDEX.JSX

```
import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    return (<BreadcrumbComponent enableNavigation={false}>
        <BreadcrumbItemsDirective>
            <BreadcrumbItemDirective iconCss="e-icons e-home"
url="https://ej2.syncfusion.com/home/react.html#platform"/>
            <BreadcrumbItemDirective text="Components"
url="https://ej2.syncfusion.com/react/demos/#/material/grid/overview"/>
            <BreadcrumbItemDirective text="Navigations"
url="https://ej2.syncfusion.com/react/demos/#/material/menu/default"/>
            <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb/default"/>
        </BreadcrumbItemsDirective>
    </BreadcrumbComponent>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

INDEX.TSX

```
import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    return (
        <BreadcrumbComponent enableNavigation={false}>
            <BreadcrumbItemsDirective>
                <BreadcrumbItemDirective iconCss="e-icons e-home"
url="https://ej2.syncfusion.com/home/react.html#platform" />
                <BreadcrumbItemDirective text="Components"
url="https://ej2.syncfusion.com/react/demos/#/material/grid/overview/" />
                <BreadcrumbItemDirective text="Navigations"
url="https://ej2.syncfusion.com/react/demos/#/material/menu/default" />
                <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb/default" />
            </BreadcrumbItemsDirective>
        </BreadcrumbComponent>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

Breadcrumb with image

In the Breadcrumb component, images can be added for the items using the [iconCss](#) property. In the following example, an image was added to the breadcrumb item by using the iconCss class `e-image-home` and specifying height and width.

INDEX.JSX

```
import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    return (<BreadcrumbComponent enableNavigation={false}>
        <BreadcrumbItemsDirective>
            <BreadcrumbItemDirective iconCss="e-image-home"
url="https://ej2.syncfusion.com/home/react.html#platform"/>
            <BreadcrumbItemDirective text="Components"
url="https://ej2.syncfusion.com/react/demos/#/material/grid/overview/" />
            <BreadcrumbItemDirective text="Navigations"
url="https://ej2.syncfusion.com/react/demos/#/material/menu/default"/>
            <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb/default"/>
        </BreadcrumbItemsDirective>
    </BreadcrumbComponent>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

INDEX.TSX

```
import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
```

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
  return (
    <BreadcrumbComponent enableNavigation={false}>
      <BreadcrumbItemsDirective>
        <BreadcrumbItemDirective iconCss="e-image-home"
url="https://ej2.syncfusion.com/home/react.html#platform" />
        <BreadcrumbItemDirective text="Components"
url="https://ej2.syncfusion.com/react/demos/#/material/grid/overview/" />
        <BreadcrumbItemDirective text="Navigations"
url="https://ej2.syncfusion.com/react/demos/#/material/menu/default" />
        <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb/default" />
      </BreadcrumbItemsDirective>
    </BreadcrumbComponent>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

Breadcrumb with SVG image

In the Breadcrumb component, SVG image can be added for the items using the [iconCss](#) property. In the following example, SVG image was added to the breadcrumb item by using the iconCss class `e-svg-home` and specifying height and width.

INDEX.JSX

```
import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
  return (<BreadcrumbComponent enableNavigation={false}>
    <BreadcrumbItemsDirective>
      <BreadcrumbItemDirective iconCss="e-svg-home"
url="https://ej2.syncfusion.com/home/react.html#platform"/>
      <BreadcrumbItemDirective text="Components"
url="https://ej2.syncfusion.com/react/demos/#/material/grid/overview/" />
      <BreadcrumbItemDirective text="Navigations"
url="https://ej2.syncfusion.com/react/demos/#/material/menu/default/" />
      <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb/default"/>
    </BreadcrumbItemsDirective>
  </BreadcrumbComponent>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

INDEX.TSX

```
import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
```

```
function App() {
  return (
    <BreadcrumbComponent enableNavigation={false}>
      <BreadcrumbItemsDirective>
        <BreadcrumbItemDirective iconCss="e-svg-home"
url="https://ej2.syncfusion.com/home/react.html#platform" />
        <BreadcrumbItemDirective text="Components"
url="https://ej2.syncfusion.com/react/demos/#/material/grid/overview/" />
        <BreadcrumbItemDirective text="Navigations"
url="https://ej2.syncfusion.com/react/demos/#/material/menu/default" />
        <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb/default" />
      </BreadcrumbItemsDirective>
    </BreadcrumbComponent>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

Icon Position

By default, the icon is positioned to the left side of the item in the Breadcrumb component. If you need to add the icon right to the breadcrumb item, add the `e-icon-right` class to the required item. In the following example, the `e-icon-right` class was added to the breadcrumb items using the [beforeItemRender](#) event.

INDEX.JSX

```
import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
  function beforeItemRenderHandler(args) {
    if (args.element) {
      args.element.classList.add('e-icon-right');
    }
  }
  return (<div id="breadcrumb-control">
    <div class='header'><b>IconPosition- left</b></div><br />
    <BreadcrumbComponent>
      <BreadcrumbItemsDirective>
        <BreadcrumbItemDirective text="Program Files" iconCss="e-
bicons e-folder"></BreadcrumbItemDirective>
        <BreadcrumbItemDirective text="Services" iconCss="e-bicons e-
folder"></BreadcrumbItemDirective>
        <BreadcrumbItemDirective text="Config.json" iconCss="e-bicons
e-file"></BreadcrumbItemDirective>
      </BreadcrumbItemsDirective>
    </BreadcrumbComponent>
    <br />
    <br />
    <div class='header'><b>IconPosition- Right</b></div><br />
    <BreadcrumbComponent beforeItemRender={beforeItemRenderHandler}>
      <BreadcrumbItemsDirective>
```

```

        <BreadcrumbItemDirective text="Program Files" iconCss="e-
bicons e-folder"></BreadcrumbItemDirective>
        <BreadcrumbItemDirective text="Services" iconCss="e-bicons e-
folder"></BreadcrumbItemDirective>
        <BreadcrumbItemDirective text="Config.json" iconCss="e-bicons
e-file"></BreadcrumbItemDirective>
    </BreadcrumbItemsDirective>
</BreadcrumbComponent></div>;
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

INDEX.TSX

```

import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import { BreadcrumbBeforeItemRenderEventArgs } from '@syncfusion/ej2-
navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    function beforeItemRenderHandler(args:
BreadcrumbBeforeItemRenderEventArgs): void {
        if (args.element) {
            args.element.classList.add('e-icon-right');
        }
    }
    return (<div id="breadcrumb-control">
        <div class='header'><b>IconPosition- left</b></div><br />
        <BreadcrumbComponent>
            <BreadcrumbItemsDirective>
                <BreadcrumbItemDirective text="Program Files" iconCss="e-
bicons e-folder"></BreadcrumbItemDirective>
                <BreadcrumbItemDirective text="Services" iconCss="e-bicons e-
folder"></BreadcrumbItemDirective>
                <BreadcrumbItemDirective text="Config.json" iconCss="e-bicons
e-file"></BreadcrumbItemDirective>
            </BreadcrumbItemsDirective>
        </BreadcrumbComponent>
        <br />
        <br/>
        <div class='header'><b>IconPosition- Right</b></div><br />
        <BreadcrumbComponent beforeItemRender={beforeItemRenderHandler}>
            <BreadcrumbItemsDirective>
                <BreadcrumbItemDirective text="Program Files" iconCss="e-
bicons e-folder"></BreadcrumbItemDirective>
                <BreadcrumbItemDirective text="Services" iconCss="e-bicons e-
folder"></BreadcrumbItemDirective>
                <BreadcrumbItemDirective text="Config.json" iconCss="e-bicons
e-file"></BreadcrumbItemDirective>
            </BreadcrumbItemsDirective>
        </BreadcrumbComponent></div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

Icons only

To display only icons for the items, add icons using the [iconCss](#) property. In the following example, breadcrumb items were demonstrated with only icons by providing the [iconCss](#) property.

INDEX.JSX

```
import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    return (<BreadcrumbComponent>
        <BreadcrumbItemsDirective>
            <BreadcrumbItemDirective iconCss="e-icons e-
home"></BreadcrumbItemDirective>
            <BreadcrumbItemDirective iconCss="e-bicons e-
folder"></BreadcrumbItemDirective>
            <BreadcrumbItemDirective iconCss="e-bicons e-
file"></BreadcrumbItemDirective>
        </BreadcrumbItemsDirective>
    </BreadcrumbComponent>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

INDEX.TSX

```
import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    return (
        <BreadcrumbComponent>
            <BreadcrumbItemsDirective>
                <BreadcrumbItemDirective iconCss="e-icons e-
home"></BreadcrumbItemDirective>
                <BreadcrumbItemDirective iconCss="e-bicons e-
folder"></BreadcrumbItemDirective>
                <BreadcrumbItemDirective iconCss="e-bicons e-
file"></BreadcrumbItemDirective>
            </BreadcrumbItemsDirective>
        </BreadcrumbComponent>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

Show icon only for first item

To show icon only for the first item in the Breadcrumb component, add icons to the first item using the [iconCss](#) property. In the following example, the icon was provided only for the first item by setting the [iconCss](#) property.

INDEX.JSX

```
import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    return (<BreadcrumbComponent enableNavigation={false}>
        <BreadcrumbItemsDirective>
            <BreadcrumbItemDirective iconCss="e-icons e-home"
url="https://ej2.syncfusion.com/home/react.html#platform"/>
            <BreadcrumbItemDirective text="Components"
url="https://ej2.syncfusion.com/react/demos/#/material/grid/overview/" />
            <BreadcrumbItemDirective text="Navigations"
url="https://ej2.syncfusion.com/react/demos/#/material/menu/default"/>
            <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb/default"/>
        </BreadcrumbItemsDirective>
    </BreadcrumbComponent>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

INDEX.TSX

```
import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    return (
        <BreadcrumbComponent enableNavigation={false}>
            <BreadcrumbItemsDirective>
                <BreadcrumbItemDirective iconCss="e-icons e-home"
url="https://ej2.syncfusion.com/home/react.html#platform" />
                <BreadcrumbItemDirective text="Components"
url="https://ej2.syncfusion.com/react/demos/#/material/grid/overview/" />
                <BreadcrumbItemDirective text="Navigations"
url="https://ej2.syncfusion.com/react/demos/#/material/menu/default" />
                <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb/default" />
            </BreadcrumbItemsDirective>
        </BreadcrumbComponent>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

Navigation in React Breadcrumb component

The breadcrumb item navigates to the path while clicking the item. To enable navigation, [url](#) property was bound to the items.

URL

In the Breadcrumb component, the item represents the url. The breadcrumb items can be provided with either relative or absolute URL.

Relative URL

The breadcrumb items with relative URL contain only the path but do not locate the path or server. The following example represents the breadcrumb items with relative url.

INDEX.JSX

```
import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    return (<BreadcrumbComponent enableNavigation={false}>
        <BreadcrumbItemsDirective>
            <BreadcrumbItemDirective text="Home"
url=".."></BreadcrumbItemDirective>
            <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb"></BreadcrumbItemDirective>
            <BreadcrumbItemDirective text="Default"
url="./breadcrumb/default-functionalities"></BreadcrumbItemDirective>
            <BreadcrumbItemDirective text="Icons"
url="./breadcrumb/icons"></BreadcrumbItemDirective>
            <BreadcrumbItemDirective text="Navigation"
url="./breadcrumb/navigation"></BreadcrumbItemDirective>
            <BreadcrumbItemDirective text="Overflow"
url="./breadcrumb/overflow"></BreadcrumbItemDirective>
        </BreadcrumbItemsDirective>
    </BreadcrumbComponent>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

INDEX.TSX

```
import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    return (
        <BreadcrumbComponent enableNavigation={false}>
            <BreadcrumbItemsDirective>
                <BreadcrumbItemDirective text="Home"
url=".."></BreadcrumbItemDirective>
                <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb"></BreadcrumbItemDirective>
                <BreadcrumbItemDirective text="Default"
url="./breadcrumb/default-functionalities"></BreadcrumbItemDirective>
                <BreadcrumbItemDirective text="Icons"
url="./breadcrumb/icons"></BreadcrumbItemDirective>
                <BreadcrumbItemDirective text="Navigation"
url="./breadcrumb/navigation"></BreadcrumbItemDirective>
                <BreadcrumbItemDirective text="Overflow"
url="./breadcrumb/overflow"></BreadcrumbItemDirective>
            </BreadcrumbItemsDirective>
        </BreadcrumbComponent>
    );
};
```

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

Absolute URL

The breadcrumb items with Absolute URL contain the path and locate to the resource if the static url is bound to the breadcrumb item. The following example represents the breadcrumb items with static url.

INDEX.JSX

```
import { BreadcrumbComponent, BreadcrumbItemDirective,  
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';  
import * as React from 'react';  
import * as ReactDOM from 'react-dom';  
function App() {  
    return (<BreadcrumbComponent enableNavigation={false}>  
        <BreadcrumbItemsDirective>  
            <BreadcrumbItemDirective text="Home"  
url="https://ej2.syncfusion.com/documentation/breadcrumb/introduction"/>  
            <BreadcrumbItemDirective text="Getting"  
url="https://ej2.syncfusion.com/documentation/breadcrumb/getting-started"/>  
            <BreadcrumbItemDirective text="Icons"  
url="https://ej2.syncfusion.com/documentation/breadcrumb/icons"/>  
            <BreadcrumbItemDirective text="Navigations"  
url="https://ej2.syncfusion.com/documentation/breadcrumb/navigation"/>  
            <BreadcrumbItemDirective text="Overflow"  
url="https://ej2.syncfusion.com/documentation/breadcrumb/overflow"/>  
        </BreadcrumbItemsDirective>  
    </BreadcrumbComponent>);  
}  
export default App;  
ReactDOM.render(<App />, document.getElementById('element'));
```

INDEX.TSX

```
import { BreadcrumbComponent, BreadcrumbItemDirective,  
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';  
import * as React from 'react';  
import * as ReactDOM from 'react-dom';  
function App() {  
    return (  
        <BreadcrumbComponent enableNavigation={false}>  
            <BreadcrumbItemsDirective>  
                <BreadcrumbItemDirective text="Home"  
url="https://ej2.syncfusion.com/documentation/breadcrumb/introduction" />  
                <BreadcrumbItemDirective text="Getting"  
url="https://ej2.syncfusion.com/documentation/breadcrumb/getting-started" />  
                <BreadcrumbItemDirective text="Icons"  
url="https://ej2.syncfusion.com/documentation/breadcrumb/icons" />  
                <BreadcrumbItemDirective text="Navigations"  
url="https://ej2.syncfusion.com/documentation/breadcrumb/navigation" />  
                <BreadcrumbItemDirective text="Overflow"  
url="https://ej2.syncfusion.com/documentation/breadcrumb/overflow" />  
            </BreadcrumbItemsDirective>  
        </BreadcrumbComponent>  
    );  
}
```

```

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

```

Enable navigation for last Breadcrumb item

The feature enables the last item of the Breadcrumb component by setting the [enableActiveItemNavigation](#) property to true. In the following example, the last item of the Breadcrumb was enabled.

INDEX.JSX

```

import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
  return (<BreadcrumbComponent enableNavigation={false}
enableActiveItemNavigation={true}>
    <BreadcrumbItemsDirective>
    <BreadcrumbItemDirective text="Home"
url="https://ej2.syncfusion.com/documentation/breadcrumb/introduction"/>
    <BreadcrumbItemDirective text="Getting"
url="https://ej2.syncfusion.com/documentation/breadcrumb/getting-started"/>
    <BreadcrumbItemDirective text="Icons"
url="https://ej2.syncfusion.com/documentation/breadcrumb/icons"/>
    <BreadcrumbItemDirective text="Navigations"
url="https://ej2.syncfusion.com/documentation/breadcrumb/navigation"/>
    <BreadcrumbItemDirective text="Overflow"
url="https://ej2.syncfusion.com/documentation/breadcrumb/overflow"/>
    </BreadcrumbItemsDirective>
    </BreadcrumbComponent>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

INDEX.TSX

```

import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
  return (
    <BreadcrumbComponent enableNavigation={false}
enableActiveItemNavigation={true}>
    <BreadcrumbItemsDirective>
    <BreadcrumbItemDirective text="Home"
url="https://ej2.syncfusion.com/documentation/breadcrumb/introduction" />
    <BreadcrumbItemDirective text="Getting"
url="https://ej2.syncfusion.com/documentation/breadcrumb/getting-started" />
    <BreadcrumbItemDirective text="Icons"
url="https://ej2.syncfusion.com/documentation/breadcrumb/icons" />
    <BreadcrumbItemDirective text="Navigations"
url="https://ej2.syncfusion.com/documentation/breadcrumb/navigation" />

```

```

        <BreadcrumbItemDirective text="Overflow"
url="https://ej2.syncfusion.com/documentation/breadcrumb/overflow" />
      </BreadcrumbItemsDirective>
    </BreadcrumbComponent>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

Open URL in new page or tab

To open the breadcrumb item in a new page or tab, set the target property of the required item url to blank in the Breadcrumb component. In the following example, the target property of items url was set to blank by using the [beforeItemRender](#) event which locates to the path in the new tab.

INDEX.JSX

```

import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
  function beforeItemRenderHandler(args) {
    if (args.element.children[0]) {
      args.element.children[0].target = "_blank";
    }
  }
  return (<BreadcrumbComponent enableActiveItemNavigation={true}
beforeItemRender={beforeItemRenderHandler}>
    <BreadcrumbItemsDirective>
      <BreadcrumbItemDirective text="Home"
url="https://ej2.syncfusion.com/documentation/breadcrumb/introduction"/>
      <BreadcrumbItemDirective text="Getting"
url="https://ej2.syncfusion.com/documentation/breadcrumb/getting-started"/>
      <BreadcrumbItemDirective text="Icons"
url="https://ej2.syncfusion.com/documentation/breadcrumb/icons"/>
      <BreadcrumbItemDirective text="Navigations"
url="https://ej2.syncfusion.com/documentation/breadcrumb/navigation"/>
      <BreadcrumbItemDirective text="Overflow"
url="https://ej2.syncfusion.com/documentation/breadcrumb/overflow"/>
    </BreadcrumbItemsDirective>
  </BreadcrumbComponent>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

INDEX.TSX

```

import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import { BreadcrumbBeforeItemRenderEventArgs } from '@syncfusion/ej2-
navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {

```

```

function beforeItemRenderHandler(args:
BreadcrumbBeforeItemRenderEventArgs): void {
    if (args.element.children[0]) {
        args.element.children[0].target = "_blank";
    }
}

return (
    <BreadcrumbComponent enableActiveItemNavigation={true}
beforeItemRender={beforeItemRenderHandler}>
        <BreadcrumbItemsDirective>
            <BreadcrumbItemDirective text="Home"
url="https://ej2.syncfusion.com/documentation/breadcrumb/introduction" />
            <BreadcrumbItemDirective text="Getting"
url="https://ej2.syncfusion.com/documentation/breadcrumb/getting-started" />
            <BreadcrumbItemDirective text="Icons"
url="https://ej2.syncfusion.com/documentation/breadcrumb/icons" />
            <BreadcrumbItemDirective text="Navigations"
url="https://ej2.syncfusion.com/documentation/breadcrumb/navigation" />
            <BreadcrumbItemDirective text="Overflow"
url="https://ej2.syncfusion.com/documentation/breadcrumb/overflow" />
        </BreadcrumbItemsDirective>
    </BreadcrumbComponent>
);
}

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

```

Template in React Breadcrumb component

The Breadcrumb component provides a way to customize the items using [itemTemplate](#) and the separators using [separatorTemplate](#) properties.

Item Template

In the following example, Shopping Cart details are used as breadcrumb Items and the items are customized by the chips component using [itemTemplate](#).

INDEX.JSX

```

import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import { ChipListComponent, ChipsDirective, ChipDirective } from
 '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';

function App() {
    function chipTemplate(data) {
        return (<ChipListComponent>
            <ChipsDirective>
                <ChipDirective text={data.text}></ChipDirective>
            </ChipsDirective>
        </ChipListComponent>);
    }

    return (<BreadcrumbComponent cssClass="e-breadcrumb-chips"
itemTemplate={chipTemplate}>
        <BreadcrumbItemsDirective>
            <BreadcrumbItemDirective
text="Cart"></BreadcrumbItemDirective>

```

```

        <BreadcrumbItemDirective
text="Billing"></BreadcrumbItemDirective>
        <BreadcrumbItemDirective
text="Shipping"></BreadcrumbItemDirective>
        <BreadcrumbItemDirective
text="Payment"></BreadcrumbItemDirective>
    </BreadcrumbItemsDirective>
</BreadcrumbComponent>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

INDEX.TSX

```

import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import { ChipListComponent, ChipsDirective, ChipDirective } from
 '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    function chipTemplate(data: any): JSX.Element {
        return (
            <ChipListComponent>
                <ChipsDirective>
                    <ChipDirective text={data.text}></ChipDirective>
                </ChipsDirective>
            </ChipListComponent>
        );
    }
    return (
        <BreadcrumbComponent cssClass="e-breadcrumb-chips"
itemTemplate={chipTemplate}>
            <BreadcrumbItemsDirective>
                <BreadcrumbItemDirective
text="Cart"></BreadcrumbItemDirective>
                <BreadcrumbItemDirective
text="Billing"></BreadcrumbItemDirective>
                <BreadcrumbItemDirective
text="Shipping"></BreadcrumbItemDirective>
                <BreadcrumbItemDirective
text="Payment"></BreadcrumbItemDirective>
            </BreadcrumbItemsDirective>
        </BreadcrumbComponent>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

Separator Template

In the following example, the separators are customized with icons using [separatorTemplate](#).

INDEX.JSX

```

import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';

```

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
  function arrowSeparatorTemplate() {
    return (<span className="e-icons e-arrow"></span>);
  }
  return (<BreadcrumbComponent separatorTemplate={arrowSeparatorTemplate}>
    <BreadcrumbItemsDirective>
      <BreadcrumbItemDirective
text="Cart"></BreadcrumbItemDirective>
      <BreadcrumbItemDirective
text="Billing"></BreadcrumbItemDirective>
      <BreadcrumbItemDirective
text="Shipping"></BreadcrumbItemDirective>
      <BreadcrumbItemDirective
text="Payment"></BreadcrumbItemDirective>
    </BreadcrumbItemsDirective>
  </BreadcrumbComponent>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

INDEX.TSX

```
import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
  function arrowSeparatorTemplate(): JSX.Element {
    return (
      <span className="e-icons e-arrow"></span>
    );
  }
  return (
    <BreadcrumbComponent separatorTemplate={arrowSeparatorTemplate}>
      <BreadcrumbItemsDirective>
        <BreadcrumbItemDirective
text="Cart"></BreadcrumbItemDirective>
        <BreadcrumbItemDirective
text="Billing"></BreadcrumbItemDirective>
        <BreadcrumbItemDirective
text="Shipping"></BreadcrumbItemDirective>
        <BreadcrumbItemDirective
text="Payment"></BreadcrumbItemDirective>
      </BreadcrumbItemsDirective>
    </BreadcrumbComponent>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

Customize Specific Item Template

The specific breadcrumb item can be customizable using `itemTemplate` with conditional rendering. In the following example, added the span element only for the `breadcrumb` text in breadcrumb item.

INDEX.JSX

```
{% raw %}
import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    function specificItemTemplate(data) {
        return (<div>
            {data.text == "Breadcrumb" ? (<span><span className="e-
searchfor-text"><span style={{ marginRight: "5px" }}>Search for:</span>
                <a className="e-breadcrumb-text" href={data.url}
onClick={() => { return false; }}>{data.text}</a></span></span>) : (<a
className="e-breadcrumb-text" href={data.url} onClick={() => { return false;
}}>{data.text}</a>)}
            </div>);
    }
    return (<BreadcrumbComponent itemTemplate={specificItemTemplate}
cssClass="e-specific-item-template" enableNavigation={false}>
        <BreadcrumbItemsDirective>
            <BreadcrumbItemDirective text="Home"
url="https://ej2.syncfusion.com/home/react.html#platform"/>
            <BreadcrumbItemDirective text="Components"
url="https://ej2.syncfusion.com/react/demos/#/material/grid/overview"/>
            <BreadcrumbItemDirective text="Navigations"
url="https://ej2.syncfusion.com/react/demos/#/material/menu/default"/>
            <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb/default"/>
        </BreadcrumbItemsDirective>
    </BreadcrumbComponent>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
{% endraw %}
```

INDEX.TSX

```
{% raw %}
import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    function specificItemTemplate(data: any): JSX.Element {
        return (
            <div>
                {data.text == "Breadcrumb" ? (
                    <span><span className="e-searchfor-text"><span style={{
marginRight: "5px" }}>Search for:</span>
                        <a className="e-breadcrumb-text" href={data.url}
onClick={() => { return false; }}>{data.text}</a></span></span>
                    ) : (
                        <a className="e-breadcrumb-text" href={data.url}
onClick={() => { return false; }}>{data.text}</a>
                    )}
            </div>
        );
    }
}
```

```

        </div>
    );
}
return (
    <BreadcrumbComponent itemTemplate={specificItemTemplate} cssClass="e-
specific-item-template" enableNavigation={false}>
        <BreadcrumbItemsDirective>
            <BreadcrumbItemDirective text="Home"
url="https://ej2.syncfusion.com/home/react.html#platform" />
            <BreadcrumbItemDirective text="Components"
url="https://ej2.syncfusion.com/react/demos/#/material/grid/overview/" />
            <BreadcrumbItemDirective text="Navigations"
url="https://ej2.syncfusion.com/react/demos/#/material/menu/default" />
            <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb/default" />
        </BreadcrumbItemsDirective>
    </BreadcrumbComponent>
);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
{% enddraw %}

```

Overflow mode in React Breadcrumb component

In the Breadcrumb component, [maxItems](#) and [overflowMode](#) properties were used to limit the number of breadcrumb items to be displayed.

In the following example, the [maxItems](#) is set as 3 with overflowMode as Default. To prevent breadcrumb item navigation, the [enableNavigation](#) property has been set to false in the Breadcrumb component.

INDEX.JSX

```

import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    function breadcrumbTemplate() {
        return (<span className="e-bicons e-arrow"></span>);
    }
    return (<BreadcrumbComponent maxItems={3} enableNavigation={false}
separatorTemplate={breadcrumbTemplate}>
        <BreadcrumbItemsDirective>
            <BreadcrumbItemDirective text="Home"
url=".."></BreadcrumbItemDirective>
            <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb"></BreadcrumbItemDirective>
            <BreadcrumbItemDirective text="Default"
url="./breadcrumb/default-functionalities"></BreadcrumbItemDirective>
            <BreadcrumbItemDirective text="Icons"
url="./breadcrumb/icons"></BreadcrumbItemDirective>
            <BreadcrumbItemDirective text="Navigation"
url="./breadcrumb/navigation"></BreadcrumbItemDirective>
            <BreadcrumbItemDirective text="Overflow"
url="./breadcrumb/overflow"></BreadcrumbItemDirective>

```

```

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

```

INDEX.TSX

```

import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
  function breadcrumbTemplate(): JSX.Element {
    return (
      <span className="e-bicons e-arrow"></span>
    );
  }
  return (
    <BreadcrumbComponent maxItems={3} enableNavigation={false}
separatorTemplate={breadcrumbTemplate}>
      <BreadcrumbItemsDirective>
        <BreadcrumbItemDirective text="Home"
url=".."></BreadcrumbItemDirective>
        <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb"></BreadcrumbItemDirective>
        <BreadcrumbItemDirective text="Default"
url="./breadcrumb/default-functionalities"></BreadcrumbItemDirective>
        <BreadcrumbItemDirective text="Icons"
url="./breadcrumb/icons"></BreadcrumbItemDirective>
        <BreadcrumbItemDirective text="Navigation"
url="./breadcrumb/navigation"></BreadcrumbItemDirective>
        <BreadcrumbItemDirective text="Overflow"
url="./breadcrumb/overflow"></BreadcrumbItemDirective>
      </BreadcrumbItemsDirective>
    </BreadcrumbComponent>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

The following overflow modes are available in the Breadcrumb component.

- Collapsed
- Menu
- Wrap
- Scroll
- Hidden
- None

Collapsed

Collapsed mode shows the first and last Breadcrumb items and hides the remaining items with a collapsed icon. When the collapsed icon is clicked, all items become visible and navigable.

INDEX.JSX

```
import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    function breadcrumbTemplate() {
        return (<span className="e-bicons e-arrow"></span>);
    }
    return (<BreadcrumbComponent maxItems={3} enableNavigation={false}
overflowMode='Collapsed' separatorTemplate={breadcrumbTemplate}>
        <BreadcrumbItemsDirective>
            <BreadcrumbItemDirective text="Home"
url=".."></BreadcrumbItemDirective>
            <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb"></BreadcrumbItemDirective>
            <BreadcrumbItemDirective text="Default"
url="./breadcrumb/default-functionalities"></BreadcrumbItemDirective>
            <BreadcrumbItemDirective text="Icons"
url="./breadcrumb/icons"></BreadcrumbItemDirective>
            <BreadcrumbItemDirective text="Navigation"
url="./breadcrumb/navigation"></BreadcrumbItemDirective>
            <BreadcrumbItemDirective text="Overflow"
url="./breadcrumb/overflow"></BreadcrumbItemDirective>
        </BreadcrumbItemsDirective>
    </BreadcrumbComponent>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

INDEX.TSX

```
import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    function breadcrumbTemplate(): JSX.Element {
        return (
            <span className="e-bicons e-arrow"></span>
        );
    }
    return (
        <BreadcrumbComponent maxItems={3} enableNavigation={false}
overflowMode='Collapsed' separatorTemplate={breadcrumbTemplate}>
            <BreadcrumbItemsDirective>
                <BreadcrumbItemDirective text="Home"
url=".."></BreadcrumbItemDirective>
                <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb"></BreadcrumbItemDirective>
                <BreadcrumbItemDirective text="Default"
url="./breadcrumb/default-functionalities"></BreadcrumbItemDirective>
                <BreadcrumbItemDirective text="Icons"
url="./breadcrumb/icons"></BreadcrumbItemDirective>
            </BreadcrumbItemsDirective>
        </BreadcrumbComponent>
    );
}
```

```

        <BreadcrumbItemDirective text="Navigation"
url="./breadcrumb/navigation"></BreadcrumbItemDirective>
        <BreadcrumbItemDirective text="Overflow"
url="./breadcrumb/overflow"></BreadcrumbItemDirective>
    </BreadcrumbItemsDirective>
</BreadcrumbComponent>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

Menu

Menu mode shows the number of Breadcrumb items that can be accommodated within the container space and creates a submenu with the remaining items.

INDEX.JSX

```

import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    function breadcrumbTemplate() {
        return (<span className="e-bicons e-arrow"></span>);
    }
    return (<BreadcrumbComponent maxItems={3} enableNavigation={false}
overflowMode='Menu' separatorTemplate={breadcrumbTemplate}>
        <BreadcrumbItemsDirective>
            <BreadcrumbItemDirective text="Home"
url=".."></BreadcrumbItemDirective>
            <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb"></BreadcrumbItemDirective>
            <BreadcrumbItemDirective text="Default"
url="./breadcrumb/default-functionalities"></BreadcrumbItemDirective>
            <BreadcrumbItemDirective text="Icons"
url="./breadcrumb/icons"></BreadcrumbItemDirective>
            <BreadcrumbItemDirective text="Navigation"
url="./breadcrumb/navigation"></BreadcrumbItemDirective>
            <BreadcrumbItemDirective text="Overflow"
url="./breadcrumb/overflow"></BreadcrumbItemDirective>
        </BreadcrumbItemsDirective>
    </BreadcrumbComponent>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

INDEX.TSX

```

import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    function breadcrumbTemplate(): JSX.Element {
        return (

```

```

        <span className="e-bicons e-arrow"></span>
      );
    }
    return (
      <BreadcrumbComponent maxItems={3} enableNavigation={false}
        overflowMode='Menu' separatorTemplate={breadcrumbTemplate}>
        <BreadcrumbItemsDirective>
          <BreadcrumbItemDirective text="Home"
            url=".."></BreadcrumbItemDirective>
          <BreadcrumbItemDirective text="Breadcrumb"
            url="./breadcrumb"></BreadcrumbItemDirective>
          <BreadcrumbItemDirective text="Default"
            url="./breadcrumb/default-functionalities"></BreadcrumbItemDirective>
          <BreadcrumbItemDirective text="Icons"
            url="./breadcrumb/icons"></BreadcrumbItemDirective>
          <BreadcrumbItemDirective text="Navigation"
            url="./breadcrumb/navigation"></BreadcrumbItemDirective>
          <BreadcrumbItemDirective text="Overflow"
            url="./breadcrumb/overflow"></BreadcrumbItemDirective>
        </BreadcrumbItemsDirective>
      </BreadcrumbComponent>
    );
  }
  export default App;
  ReactDOM.render(<App />, document.getElementById('element'));

```

Wrap

Wrap mode wraps the items to multiple lines when the Breadcrumb's width exceeds the container space.

INDEX.JSX

```

import { BreadcrumbComponent, BreadcrumbItemDirective,
  BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
  function breadcrumbTemplate() {
    return <span className="e-bicons e-arrow"></span>;
  }
  return (
    <BreadcrumbComponent maxItems={3} enableNavigation={false}
      overflowMode='Wrap' separatorTemplate={breadcrumbTemplate}>
      <BreadcrumbItemsDirective>
        <BreadcrumbItemDirective text="Home"
          url=".."></BreadcrumbItemDirective>
        <BreadcrumbItemDirective text="Breadcrumb"
          url="./breadcrumb"></BreadcrumbItemDirective>
        <BreadcrumbItemDirective text="Default"
          url="./breadcrumb/default-functionalities"></BreadcrumbItemDirective>
        <BreadcrumbItemDirective text="Icons"
          url="./breadcrumb/icons"></BreadcrumbItemDirective>
        <BreadcrumbItemDirective text="Navigation"
          url="./breadcrumb/navigation"></BreadcrumbItemDirective>
        <BreadcrumbItemDirective text="Overflow"
          url="./breadcrumb/overflow"></BreadcrumbItemDirective>
      </BreadcrumbItemsDirective>
    )
  );
}

```

```

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

```

INDEX.TSX

```

import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    function breadcrumbTemplate(): JSX.Element {
        return (
            <span className="e-bicons e-arrow"></span>
        );
    }
    return (
        <BreadcrumbComponent maxItems={3} enableNavigation={false}
overflowMode='Wrap' separatorTemplate={breadcrumbTemplate}>
            <BreadcrumbItemsDirective>
                <BreadcrumbItemDirective text="Home"
url=".."></BreadcrumbItemDirective>
                <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb"></BreadcrumbItemDirective>
                <BreadcrumbItemDirective text="Default"
url="./breadcrumb/default-functionalities"></BreadcrumbItemDirective>
                <BreadcrumbItemDirective text="Icons"
url="./breadcrumb/icons"></BreadcrumbItemDirective>
                <BreadcrumbItemDirective text="Navigation"
url="./breadcrumb/navigation"></BreadcrumbItemDirective>
                <BreadcrumbItemDirective text="Overflow"
url="./breadcrumb/overflow"></BreadcrumbItemDirective>
            </BreadcrumbItemsDirective>
        </BreadcrumbComponent>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

Scroll

Scroll mode shows an HTML scroll bar when the Breadcrumb's width exceeds the container space.

INDEX.JSX

```

import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    function breadcrumbTemplate() {
        return (<span className="e-bicons e-arrow"></span>);
    }
    return (<BreadcrumbComponent maxItems={3} enableNavigation={false}
overflowMode='Scroll' separatorTemplate={breadcrumbTemplate}>
        <BreadcrumbItemsDirective>

```

```

        <BreadcrumbItemDirective text="Home"
url=".."></BreadcrumbItemDirective>
        <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb"></BreadcrumbItemDirective>
        <BreadcrumbItemDirective text="Default"
url="./breadcrumb/default-functionalities"></BreadcrumbItemDirective>
        <BreadcrumbItemDirective text="Icons"
url="./breadcrumb/icons"></BreadcrumbItemDirective>
        <BreadcrumbItemDirective text="Navigation"
url="./breadcrumb/navigation"></BreadcrumbItemDirective>
        <BreadcrumbItemDirective text="Overflow"
url="./breadcrumb/overflow"></BreadcrumbItemDirective>
    </BreadcrumbItemsDirective>
</BreadcrumbComponent>;
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

INDEX.TSX

```

import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    function breadcrumbTemplate(): JSX.Element {
        return (
            <span className="e-bicons e-arrow"></span>
        );
    }
    return (
        <BreadcrumbComponent maxItems={3} enableNavigation={false}
overflowMode='Scroll' separatorTemplate={breadcrumbTemplate}>
            <BreadcrumbItemsDirective>
                <BreadcrumbItemDirective text="Home"
url=".."></BreadcrumbItemDirective>
                <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb"></BreadcrumbItemDirective>
                <BreadcrumbItemDirective text="Default"
url="./breadcrumb/default-functionalities"></BreadcrumbItemDirective>
                <BreadcrumbItemDirective text="Icons"
url="./breadcrumb/icons"></BreadcrumbItemDirective>
                <BreadcrumbItemDirective text="Navigation"
url="./breadcrumb/navigation"></BreadcrumbItemDirective>
                <BreadcrumbItemDirective text="Overflow"
url="./breadcrumb/overflow"></BreadcrumbItemDirective>
            </BreadcrumbItemsDirective>
        </BreadcrumbComponent>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```


Hidden

Hidden mode shows the maximum number of items possible in the container space and hides the remaining items. Clicking on a previous item will make the hidden item visible.

INDEX.JSX

```
import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    function breadcrumbTemplate() {
        return (<span className="e-bicons e-arrow"></span>);
    }
    return (<BreadcrumbComponent maxItems={3} enableNavigation={false}
overflowMode='Hidden' separatorTemplate={breadcrumbTemplate}>
        <BreadcrumbItemsDirective>
            <BreadcrumbItemDirective text="Home"
url=".."></BreadcrumbItemDirective>
            <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb"></BreadcrumbItemDirective>
            <BreadcrumbItemDirective text="Default"
url="./breadcrumb/default-functionalities"></BreadcrumbItemDirective>
            <BreadcrumbItemDirective text="Icons"
url="./breadcrumb/icons"></BreadcrumbItemDirective>
            <BreadcrumbItemDirective text="Navigation"
url="./breadcrumb/navigation"></BreadcrumbItemDirective>
            <BreadcrumbItemDirective text="Overflow"
url="./breadcrumb/overflow"></BreadcrumbItemDirective>
        </BreadcrumbItemsDirective>
    </BreadcrumbComponent>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));
```

INDEX.TSX

```
import { BreadcrumbComponent, BreadcrumbItemDirective,
BreadcrumbItemsDirective } from '@syncfusion/ej2-react-navigations';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
function App() {
    function breadcrumbTemplate(): JSX.Element {
        return (
            <span className="e-bicons e-arrow"></span>
        );
    }
    return (
        <BreadcrumbComponent maxItems={3} enableNavigation={false}
overflowMode='Hidden' separatorTemplate={breadcrumbTemplate}>
            <BreadcrumbItemsDirective>
                <BreadcrumbItemDirective text="Home"
url=".."></BreadcrumbItemDirective>
                <BreadcrumbItemDirective text="Breadcrumb"
url="./breadcrumb"></BreadcrumbItemDirective>
            </BreadcrumbItemsDirective>
        </BreadcrumbComponent>
    );
}
```

```

        <BreadcrumbItemDirective text="Default"
url="./breadcrumb/default-functionalities"></BreadcrumbItemDirective>
        <BreadcrumbItemDirective text="Icons"
url="./breadcrumb/icons"></BreadcrumbItemDirective>
        <BreadcrumbItemDirective text="Navigation"
url="./breadcrumb/navigation"></BreadcrumbItemDirective>
        <BreadcrumbItemDirective text="Overflow"
url="./breadcrumb/overflow"></BreadcrumbItemDirective>
    </BreadcrumbItemsDirective>
</BreadcrumbComponent>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('element'));

```

None

None mode shows all the items on a single line.

Accessibility in React Breadcrumb component

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

| Attributes | Purpose |

| --- | --- |

| `aria-label` | Indicates the breadcrumb item text. |

| `aria-disabled` | Indicates the state of breadcrumb item whether it is disabled. |

Keyboard interaction

The Breadcrumb 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 Breadcrumb component.

| Press | To do this |

| --- | --- |

| Tab | Navigate to the next item and also next item in the popup of menu type overflow. |

| Shift + Tab | Navigate to the previous item also previous item in the popup of menu type overflow. |

| Enter key in normal mode | Select the breadcrumb item. |

| Enter key in normal mode | To open the popup of menu type overflow mode when you press enter on collapsed button and It will expand the items of collapsed type overflow mode when you press enter on collapsed button. |

Ensuring accessibility

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

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

See also

- [Accessibility in Syncfusion React components](#)

Bullet Chart

<!-- markdownlint-disable MD036 -->

Getting Started

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

Dependencies

Below is the list of minimum dependencies required to use the Bullet 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
`,`
```

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 Bullet Chart to the Project

Now, you can start adding Bullet Chart component in the application.

For getting started, add the Bullet Chart component in `src/App.tsx` file using following code.

INDEX.JSX

```
import { BulletChartComponent } from "@syncfusion/ej2-react-charts";
import * as React from "react";
function App() {
  return <BulletChartComponent />;
}
export default App;
```

INDEX.TSX

```
import { BulletChartComponent } from "@syncfusion/ej2-react-charts";
import * as React from "react";
function App() {
  return <BulletChartComponent />;
}
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 Bullet Chart.

INDEX.JSX

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

```
function App() {
    return <BulletChartComponent id="bulletChart"/>;
}
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

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

Module Injection

Bullet Chart are segregated into individual feature-wise modules. In order to use a particular feature, you need to inject its feature service in the AppModule. Please find the relevant feature service name and description as follows.

- **BulletTooltip** - Inject this module in to **services** to use tooltip feature.

These modules should be injected to the **services** section as follows,

INDEX.JSX

```
import { BulletChartComponent, BulletTooltip } from "@syncfusion/ej2-react-charts";
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (
        <BulletChartComponent id="bulletChart">
            <Inject services={[BulletTooltip]} />
        </BulletChartComponent>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

INDEX.TSX

```
import { BulletChartComponent, BulletTooltip } from "@syncfusion/ej2-react-charts";
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id="bulletChart">
        <Inject services={[BulletTooltip]} />
    </BulletChartComponent>);
}
export default App;
```

```
ReactDOM.render(<App />, document.getElementById("charts"));
```

Bullet Chart With Data

This section explains how to plot local data to the Bullet Chart.

INDEX.JSX

```
const data = [
  { value: 100, target: 80 },
  { value: 200, target: 180 },
  { value: 300, target: 280 },
  { value: 400, target: 380 },
  { value: 500, target: 480 }
];
```

INDEX.TSX

```
const data: any[] = [
  { value: 100, target: 80 },
  { value: 200, target: 180 },
  { value: 300, target: 280 },
  { value: 400, target: 380 },
  { value: 500, target: 480 }
];
```

Now assign the local data to `dataSource` property. `value` and `target` values should be mapped with `valueField` and `targetField` respectively.

INDEX.JSX

```
{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data = [
    { value: 100, target: 80 },
    { value: 200, target: 180 },
    { value: 300, target: 280 },
    { value: 400, target: 380 },
    { value: 500, target: 480 },
  ];
  return (<BulletChartComponent id='Revenue' style={{ textAlign: "center"
}} animation={{ enable: false }} valueField='value' targetField='target'
minimum={0} maximum={300} interval={50} dataSource={data}>
    </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

INDEX.TSX

```
{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  const data: any[] = [
    { value: 100, target: 80 },
    { value: 200, target: 180 },
    { value: 300, target: 280 },
    { value: 400, target: 380 },
    { value: 500, target: 480 },
  ];
  return (
    <BulletChartComponent id='Revenue'
      style={{ textAlign: "center" }}
      animation={{ enable: false }}
      valueField='value'
      targetField='target'
      minimum={0}
      maximum={300}
      interval={50}
      dataSource={data}>
    </BulletChartComponent>
  );
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

Add Bullet Chart Title

You can add a title using `title` property to the Bullet Chart to provide quick information to the user about the data plotted in the Bullet Chart.

INDEX.JSX

```
{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (
    <BulletChartComponent id='title' style={{ textAlign: "center" }}
      animation={{ enable: false }} valueField='value' targetField='target'
      title='Revenue' minimum={0} maximum={300} interval={50} dataSource={[{ value:
      270, target: 250 }]}>
    </BulletChartComponent>
  );
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

INDEX.TSX

```
{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
```



```
function App() {
  return (<BulletChartComponent id='title'
    style={{ textAlign: "center" }}
    animation={{ enable: false }}
    valueField='value'
    targetField='target'
    title='Revenue'
    minimum={0}
    maximum={300}
    interval={50}
    dataSource=[[{value: 270, target: 250}]]>
    </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

Ranges

You can add a range using **BulletRangeCollectionDirective** and **BulletRangeDirective** directives of the Bullet Chart.

INDEX.JSX

```
{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (<BulletChartComponent id='ranges' style={{ textAlign: "center" }}
    animation={{ enable: false }} valueField='value' targetField='target'
    title='Revenue' minimum={0} maximum={300} interval={50} dataSource=[[{ value:
    270, target: 250 }]]>
    <BulletRangeCollectionDirective>
      <BulletRangeDirective end={100}
        color='red'></BulletRangeDirective>
      <BulletRangeDirective end={200}
        color='blue'></BulletRangeDirective>
      <BulletRangeDirective end={300}
        color='green'></BulletRangeDirective>
    </BulletRangeCollectionDirective>
  </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

INDEX.TSX

```
{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
```

```
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (
        <BulletChartComponent id='ranges'
            style={{ textAlign: "center" }}
            animation={{ enable: false }}
            valueField='value'
            targetField='target'
            title='Revenue'
            minimum={0}
            maximum={300}
            interval={50}
            dataSource={[{value: 270, target: 250}]}>
            <BulletRangeCollectionDirective>
                <BulletRangeDirective end={100} color='red'
            </BulletRangeDirective>
                <BulletRangeDirective end={200}
            color='blue'></BulletRangeDirective>
                <BulletRangeDirective end={300}
            color='green'></BulletRangeDirective>
            </BulletRangeCollectionDirective>
        </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

Tooltip

You can use tooltip for the Bullet Chart by setting the `enable` property to true in `tooltip` object and by injecting the `BulletTooltip` module into the services.

INDEX.JSX

```
{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletTooltip } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (
        <BulletChartComponent id='tooltip' style={{ textAlign: "center"
        }} animation={{ enable: false }} tooltip={{ enable: true }}
        valueField='value' targetField='target' title='Revenue' minimum={0}
        maximum={300} interval={50} dataSource={[{ value: 270, target: 250 }]}>
            <Inject services={[BulletTooltip]} />
        </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

INDEX.TSX

```
{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
```

```
import { BulletTooltip } from '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (
        <BulletChartComponent id='tooltip'
            style={{ textAlign: "center" }}
            animation={{ enable: false }}
            tooltip={{ enable: true }}
            valueField='value'
            targetField='target'
            title='Revenue'
            minimum={0}
            maximum={300}
            interval={50}
            dataSource={[{ value: 270, target: 250 }]}>
            <Inject services={ [BulletTooltip] }/>
        </BulletChartComponent>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

Bullet chart dimensions in React Bullet chart component

Size for Container

The size of the Bullet Chart is determined by the container size, and it can be changed inline or via CSS as following.

```
<div id="charts" style="width:650px; height:350px"></div>
```

INDEX.JSX

```
{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (
        <BulletChartComponent id='ranges' animation={{ enable: false }}
        width="80%" height="90%" valueField='value' targetField='target'
        title='Revenue' minimum={0} maximum={30} interval={5} dataSource={[{ value:
        23, target: 22 }]}>
            <BulletRangeCollectionDirective>
                <BulletRangeDirective end={20}
                    color='red'></BulletRangeDirective>
                <BulletRangeDirective end={25}
                    color='blue'></BulletRangeDirective>
                <BulletRangeDirective end={30}
                    color='green'></BulletRangeDirective>
            </BulletRangeCollectionDirective>
        </BulletChartComponent>
    );
}
```

```
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

INDEX.TSX

```
{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}
        width="80%"
        height="90%"
        valueField='value'
        targetField='target'
        title='Revenue'
        minimum={0}
        maximum={30}
        interval={5}
        dataSource=[[{value: 23, target: 22}]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={20} color='red'
></BulletRangeDirective>
            <BulletRangeDirective end={25}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={30}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

Size for Bullet Chart

The [width](#) and [height](#) properties are used to adjust the size of the Bullet Chart.

INDEX.JSX

```
{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
width="80%" height="90%" valueField='value' targetField='target'
title='Revenue' minimum={0} maximum={30} interval={5} dataSource=[[{ value:
23, target: 22 }]]>
        <BulletRangeCollectionDirective>
```

```

        <BulletRangeDirective end={20}
color='red'></BulletRangeDirective>
        <BulletRangeDirective end={25}
color='blue'></BulletRangeDirective>
        <BulletRangeDirective end={30}
color='green'></BulletRangeDirective>
    </BulletRangeCollectionDirective>
</BulletChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% enddraw %}

```

INDEX.TSX

```

{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}
        width="80%"
        height="90%"
        valueField='value'
        targetField='target'
        title='Revenue'
        minimum={0}
        maximum={30}
        interval={5}
        dataSource=[[{value: 23, target: 22}]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={20} color='red'
></BulletRangeDirective>
            <BulletRangeDirective end={25}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={30}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% enddraw %}

```

Pixel

Can set the size of the Bullet Chart in pixels as shown below.

INDEX.JSX

```

{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';

```

```

import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
width="600px" height="100px" valueField='value' targetField='target'
title='Revenue' minimum={0} maximum={30} interval={5} dataSource=[{ value:
23, target: 22 }]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={20}
color='red'></BulletRangeDirective>
            <BulletRangeDirective end={25}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={30}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% enddraw %}

```

INDEX.TSX

```

{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}
        width="600px"
        height="100px"
        valueField='value'
        targetField='target'
        title='Revenue'
        minimum={0}
        maximum={30}
        interval={5}
        dataSource=[{value: 23, target: 22}]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={20} color='red'
></BulletRangeDirective>
            <BulletRangeDirective end={25}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={30}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));

```

```
{% endraw %}
```

Percentage

By setting a value in percentage, the Bullet Chart gets its dimension with respect to its container. For example, when the height is **50%**, the Bullet Chart renders to half of the container's height.

INDEX.JSX

```
{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
width="80%" height="90%" valueField='value' targetField='target'
title='Revenue' minimum={0} maximum={30} interval={5} dataSource=[{ value:
23, target: 22 }]}>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={20}
color='red'></BulletRangeDirective>
            <BulletRangeDirective end={25}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={30}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

INDEX.TSX

```
{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}
        width="80%"
        height="90%"
        valueField='value'
        targetField='target'
        title='Revenue'
        minimum={0}
        maximum={30}
        interval={5}
        dataSource=[{value: 23, target: 22}]}>
        <BulletRangeCollectionDirective>
```

```

                                <BulletRangeDirective end={20} color='red'
></BulletRangeDirective>
                                <BulletRangeDirective end={25}
color='blue'></BulletRangeDirective>
                                <BulletRangeDirective end={30}
color='green'></BulletRangeDirective>
                                </BulletRangeCollectionDirective>
                                </BulletChartComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

If the size is not specified, the Bullet Chart will be rendered with a height of **126px** and a width of the window.

Axis customization in React Bullet chart component

MajorTickLines and MinorTickLines Customization

The following properties can be used to customize [majorTicklines](#) and [minorTicklines](#).

- **width** - Specifies the width of ticklines.
- **height** - Specifies the height of ticklines.
- **color** - Specifies the color of ticklines.
- **useRangeColor** - Specifies the color of ticklines and represents the color from corresponding range colors.

INDEX.JSX

```

{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
valueField='value' targetField='target' title='Revenue' majorTickLines={{
color: 'blue', width: 5 }} minorTickLines={{ width: 4, color: 'red' }}
minimum={0} maximum={30} interval={5} dataSource=[[{ value: 23, target: 22
}]]>
                                <BulletRangeCollectionDirective>
                                <BulletRangeDirective end={20}
color='red'></BulletRangeDirective>
                                <BulletRangeDirective end={25}
color='blue'></BulletRangeDirective>
                                <BulletRangeDirective end={30}
color='green'></BulletRangeDirective>
                                </BulletRangeCollectionDirective>
                                </BulletChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```


INDEX.TSX

```
{% raw %}
import { BulletChartComponent, Inject} from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective} from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}
        valueField='value'
        targetField='target'
        title='Revenue'
        majorTickLines={{ color: 'blue', width: 5}}
        minorTickLines={{ width: 4, color: 'red'}}
        minimum={0}
        maximum={30}
        interval={5}
        dataSource=[[{value: 23, target: 22}]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={20} color='red'
></BulletRangeDirective>
            <BulletRangeDirective end={25}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={30}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

Tick Placement

The major and the minor ticks can be placed **inside** or **outside** the ranges using the [tickPosition](#) property.

INDEX.JSX

```
{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
valueField='value' targetField='target' title='Revenue' tickPosition='inside'
minimum={0} maximum={30} interval={5} dataSource=[[{ value: 23, target: 22
}]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={20}
color='red'></BulletRangeDirective>
{% endraw %}
```

```

        <BulletRangeDirective end={25}
color='blue'></BulletRangeDirective>
        <BulletRangeDirective end={30}
color='green'></BulletRangeDirective>
    </BulletRangeCollectionDirective>
</BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

INDEX.TSX

```

{% raw %}
import { BulletChartComponent, Inject} from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective} from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}
        valueField='value'
        targetField='target'
        title='Revenue'
        tickPosition='inside'
        minimum={0}
        maximum={30}
        interval={5}
        dataSource=[[{value: 23, target: 22}]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={20} color='red'
></BulletRangeDirective>
            <BulletRangeDirective end={25}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={30}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

Label Format

Axis numeric labels can be formatted by using the [labelFormat](#) property. Axis labels support all globalize formats.

INDEX.JSX

```

{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';

```

```

import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
valueField='value' targetField='target' title='Revenue' labelFormat='c'
minimum={0} maximum={30} interval={5} dataSource=[[{ value: 23, target: 22
}]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={20}
color='red'></BulletRangeDirective>
            <BulletRangeDirective end={25}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={30}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

INDEX.TSX

```

{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}
        valueField='value'
        targetField='target'
        title='Revenue'
        labelFormat='c'
        minimum={0}
        maximum={30}
        interval={5}
        dataSource=[[{value: 23, target: 22}]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={20} color='red'
></BulletRangeDirective>
            <BulletRangeDirective end={25}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={30}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

The following table describes the result of applying some commonly used formats to numeric axis labels.

<!-- markdownlint-disable MD033 -->

Label Value	Label Format property value	Result	Description
1000	n1	1000.0	The Number is rounded to 1 decimal place
1000	n2	1000.00	The Number is rounded to 2 decimal place
1000	n3	1000.000	The Number is rounded to 3 decimal place
0.01	p1	1.0%	The Number is converted to percentage with 1 decimal place
0.01	p2	1.00%	The Number is converted to percentage with 2 decimal place
0.01	p3	1.000%	The Number is converted to percentage with 3 decimal place
1000	c1	\$1000.0	The Currency symbol is appended to number and number is rounded to 1 decimal place
1000	c2	\$1000.00	The Currency symbol is appended to number and number is rounded to 2 decimal place

GroupingSeparator

To separate the groups of thousands, set the [enableGroupSeparator](#) property to **true**

INDEX.JSX

```
{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
valueField='value' targetField='target' title='Sales Rate'
enableGroupSeparator={true} minimum={0} maximum={2500} interval={250}
dataSource=[[{ value: 1500, target: 1300 }]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={500}
color='red'></BulletRangeDirective>
            <BulletRangeDirective end={1500}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={2500}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
```

```
{% enddraw %}
```

INDEX.TSX

```
{% raw %}
import { BulletChartComponent, Inject} from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective} from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}
        valueField='value'
        targetField='target'
        title='Sales Rate'
        enableGroupSeparator={true}
        minimum={0}
        maximum={2500}
        interval={250}
        dataSource=[[{value: 1500, target: 1300}]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={500} color='red'
></BulletRangeDirective>
            <BulletRangeDirective end={1500}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={2500}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

Custom Label Format

Using the [labelFormat](#) property, axis labels can be specified with a custom defined format in addition to the axis value. The label format uses a placeholder such as `${value}K`, which represents the axis label.

INDEX.JSX

```
{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
valueField='value' targetField='target' title='Sales Rate'
labelFormat='${value}K' minimum={0} maximum={2500} interval={250}
dataSource=[[{ value: 1500, target: 1300 }]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={500}
color='red'></BulletRangeDirective>

```

```

        <BulletRangeDirective end={1500}
color='blue'></BulletRangeDirective>
        <BulletRangeDirective end={2500}
color='green'></BulletRangeDirective>
    </BulletRangeCollectionDirective>
</BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

INDEX.TSX

```

{% raw %}
import { BulletChartComponent, Inject} from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective} from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}
        valueField='value'
        targetField='target'
        title='Sales Rate'
        labelFormat='${value}K'
        minimum={0}
        maximum={2500}
        interval={250}
        dataSource=[[{value: 1500, target: 1300}]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={500} color='red'
></BulletRangeDirective>
            <BulletRangeDirective end={1500}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={2500}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

Label Placement

Label can be placed **Inside** or **Outside** of the ranges using the [labelPosition](#) property.

INDEX.JSX

```

{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";

```

```
function App() {
  return (<BulletChartComponent id='ranges' animation={{ enable: false }}
    valueField='value' targetField='target' title='Revenue'
    labelPosition='inside' minimum={0} maximum={30} interval={5} dataSource=[{
      value: 23, target: 22 }])>
    <BulletRangeCollectionDirective>
      <BulletRangeDirective end={20}
        color='red'></BulletRangeDirective>
      <BulletRangeDirective end={25}
        color='blue'></BulletRangeDirective>
      <BulletRangeDirective end={30}
        color='green'></BulletRangeDirective>
    </BulletRangeCollectionDirective>
  </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

INDEX.TSX

```
{% raw %}
import { BulletChartComponent, Inject} from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective} from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (<BulletChartComponent id='ranges'
    animation={{ enable: false }}
    valueField='value'
    targetField='target'
    title='Revenue'
    labelPosition='inside'
    minimum={0}
    maximum={30}
    interval={5}
    dataSource=[{value: 23, target: 22}])>
    <BulletRangeCollectionDirective>
      <BulletRangeDirective end={20} color='red'
    ></BulletRangeDirective>
      <BulletRangeDirective end={25}
        color='blue'></BulletRangeDirective>
      <BulletRangeDirective end={30}
        color='green'></BulletRangeDirective>
    </BulletRangeCollectionDirective>
  </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

Opposed Position

To place an axis opposite to its original position, set the [opposedPosition](#) property to **true**.

INDEX.JSX

```
{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
valueField='value' targetField='target' title='Revenue' majorTickLines={{
color: 'blue', width: 5 }} minorTickLines={{ width: 4, color: 'red' }}
opposedPosition={true} minimum={0} maximum={30} interval={5} dataSource=[{
value: 23, target: 22 }]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={20}
color='red'></BulletRangeDirective>
            <BulletRangeDirective end={25}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={30}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

INDEX.TSX

```
{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}
        valueField='value'
        targetField='target'
        title='Revenue'
        majorTickLines={{ color: 'blue', width: 5 }}
        minorTickLines={{ width: 4, color: 'red' }}
        opposedPosition={true}
        minimum={0}
        maximum={30}
        interval={5}
        dataSource=[{value: 23, target: 22}]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={20} color='red'
></BulletRangeDirective>
            <BulletRangeDirective end={25}
color='blue'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```



```

                                <BulletRangeDirective end={30}
color='green'></BulletRangeDirective>
                                </BulletRangeCollectionDirective>
                                </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

Category Label

The Bullet Chart supports X-axis label by specifying the property from the data source to the [categoryField](#). It helps to understand the input data in a more efficient way.

INDEX.JSX

```

{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
valueField='value' targetField='target' title='Sales Rate'
categoryField='category' minimum={0} maximum={2500} interval={250}
dataSource=[[{ value: 1500, target: 1300, category: 'Product A' }]]>
                                <BulletRangeCollectionDirective>
                                <BulletRangeDirective end={500}
color='red'></BulletRangeDirective>
                                <BulletRangeDirective end={1500}
color='blue'></BulletRangeDirective>
                                <BulletRangeDirective end={2500}
color='green'></BulletRangeDirective>
                                </BulletRangeCollectionDirective>
                                </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

INDEX.TSX

```

{% raw %}
import { BulletChartComponent, Inject} from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective} from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
                                animation={{ enable: false }}
                                valueField='value'
                                targetField='target'
                                title='Sales Rate'

```

```

        categoryField='category'
        minimum={0}
        maximum={2500}
        interval={250}
        dataSource=[[{value: 1500, target: 1300, category:
'Product A'}]]}>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={500} color='red'
></BulletRangeDirective>
            <BulletRangeDirective end={1500}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={2500}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

Category Label Customization

The label color, opacity, font size, font family, font weight, and font style can be customized by using the [categoryLabelStyle](#) setting for category and the [labelStyle](#) setting for axis label. The [useRangeColor](#) property specifies the color of the axis label and represents the color from the corresponding range colors.

INDEX.JSX

```

{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
'@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
valueField='value' targetField='target' title='Sales Rate'
categoryField='category' categoryLabelStyle={{ color: 'Orange' }} minimum={0}
maximum={2500} interval={250} dataSource=[[{ value: 1500, target: 1300,
category: 'Product A' }]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={500}
color='red'></BulletRangeDirective>
            <BulletRangeDirective end={1500}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={2500}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

INDEX.TSX

```
{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}
        valueField='value'
        targetField='target'
        title='Sales Rate'
        categoryField='category'
        categoryLabelStyle={{ color:'Orange' }}
        minimum={0}
        maximum={2500}
        interval={250}
        dataSource=[[{value: 1500, target: 1300, category:
'Product A'}]]>>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={500} color='red'
></BulletRangeDirective>
            <BulletRangeDirective end={1500}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={2500}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

Data binding in React Bullet chart component

The [dataSource](#) property accepts a collection of values as input that helps to display measures, and compares them to a target bar. To display the actual and target bar, specify the property from the datasource into the [valueField](#) and [targetField](#) respectively.

INDEX.JSX

```
{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
height="400" valueField='value' targetField='comparativeMeasureValue'
categoryField='category' title='Profit in %' minimum={0} maximum={20}
interval={5} dataSource=[[{ value: 5, comparativeMeasureValue: 7.5, category:
'2001' },
{ value: 7, comparativeMeasureValue: 5, category: '2002' },
{ value: 10, comparativeMeasureValue: 6, category: '2003' },
{% endraw %}
```

```

        { value: 5, comparativeMeasureValue: 8, category: '2004' },
        { value: 12, comparativeMeasureValue: 5, category: '2005' },
        { value: 8, comparativeMeasureValue: 6, category: '2006' } ] ] } >
        <BulletRangeCollectionDirective>
            <BulletRangeDirective
end={5}></BulletRangeDirective>
            <BulletRangeDirective
end={15}></BulletRangeDirective>
            <BulletRangeDirective
end={20}></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

INDEX.TSX

```

{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}
        height="400"
        valueField='value'
        targetField='comparativeMeasureValue'
        categoryField='category'
        title='Profit in %'
        minimum={0}
        maximum={20}
        interval={5}
        dataSource=[[{value: 5, comparativeMeasureValue: 7.5,
category: '2001'},
                    {value: 7, comparativeMeasureValue: 5, category:
'2002'},
                    {value: 10, comparativeMeasureValue: 6, category:
'2003'},
                    {value: 5, comparativeMeasureValue: 8, category:
'2004'},
                    {value: 12, comparativeMeasureValue: 5, category:
'2005'},
                    {value: 8, comparativeMeasureValue: 6, category:
'2006' } ] ] >
            <BulletRangeCollectionDirective>
                <BulletRangeDirective end={5}
></BulletRangeDirective>
                <BulletRangeDirective
end={15}></BulletRangeDirective>
                <BulletRangeDirective
end={20}></BulletRangeDirective>

```

```

        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

Ranges in React Bullet chart component

Ranges represent the quality of a specific range such as **Good**, **Bad** and **Satisfactory** in the Bullet Chart scale. The ending point of a qualitative range is specified in the [end](#) property in [ranges](#). The [minimum](#) value of a quantitative scale is considered the starting point of the first range or the previous range end point.

Color Customization

Enhance the readability of ranges with color and opacity. It can be applied using the [color](#) and [opacity](#) properties in [ranges](#).

INDEX.JSX

```

{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
height="400" valueField='value' targetField='target' categoryField='category'
categoryLabelStyle={{ color: 'red', size: '13', fontWeight: 'bold' }}
title='ales Rate' minimum={0} maximum={100} interval={10} dataSource=[[{
value: 55, target: 7.5, category: 'Year 1' },
{ value: 7, target: 5, category: 'Year 2' },
{ value: 10, target: 6, category: 'Year 3' }]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={35} color='darkred'
opacity={0.5}></BulletRangeDirective>
            <BulletRangeDirective end={50} color='red'
opacity={1}></BulletRangeDirective>
            <BulletRangeDirective end={75} color='blue'
opacity={0.7}></BulletRangeDirective>
            <BulletRangeDirective end={90} color='lightgreen'
opacity={1}></BulletRangeDirective>
            <BulletRangeDirective end={100} color='green'
opacity={1}></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

INDEX.TSX

```

{% raw %}

```

```

import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}
        height="400"
        valueField='value'
        targetField='target'
        categoryField='category'
        categoryLabelStyle={{color:'red', size:'13',
fontWeight:'bold'}}
        title='ales Rate'
        minimum={0}
        maximum={100}
        interval={10}
        dataSource=[[{value: 55, target: 7.5, category: 'Year 1'},
        {value: 7, target: 5, category: 'Year 2'},
        {value: 10, target: 6, category: 'Year 3'}]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={35} color='darkred'
opacity={0.5}></BulletRangeDirective>
            <BulletRangeDirective end={50} color='red'
opacity={1}></BulletRangeDirective>
            <BulletRangeDirective end={75} color='blue'
opacity={0.7}></BulletRangeDirective>
            <BulletRangeDirective end={90} color='lightgreen'
opacity={1}></BulletRangeDirective>
            <BulletRangeDirective end={100} color='green'
opacity={1}></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

Value bar in React Bullet chart component

To display the primary data or the current value of the data being measured known as the **Feature Measure** that should be encoded as a bar. This is called as the **Actual Bar** or the **Feature Bar** in the Bullet Chart, and to display the actual bar the [valueField](#) should be mapped to the appropriate field from the data source.

INDEX.JSX

```

{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {

```

```

    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
valueField='value' targetField='target' title='Sales Rate' minimum={0}
maximum={100} interval={20} dataSource=[[{ value: 55, target: 75 }]]>
      <BulletRangeCollectionDirective>
        <BulletRangeDirective
end={35}></BulletRangeDirective>
        <BulletRangeDirective
end={50}></BulletRangeDirective>
        <BulletRangeDirective
end={100}></BulletRangeDirective>
      </BulletRangeCollectionDirective>
    </BulletChartComponent>);
  }
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

INDEX.TSX

```

{% raw %}
import { BulletChartComponent, Inject} from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective} from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (<BulletChartComponent id='ranges'
    animation={{ enable: false }}
    valueField='value'
    targetField='target'
    title='Sales Rate'
    minimum={0}
    maximum={100}
    interval={20}
    dataSource=[[{value: 55, target: 75}]]>
      <BulletRangeCollectionDirective>
        <BulletRangeDirective end={35}
></BulletRangeDirective>
        <BulletRangeDirective end={50}
></BulletRangeDirective>
        <BulletRangeDirective end={100}
></BulletRangeDirective>
      </BulletRangeCollectionDirective>
    </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

Types of actual bar

The shape of the actual bar can be customized using the [type](#) property of the Bullet Chart. The actual bar contains **Rect** and **Dot** shapes. By default, the actual bar shape is Rect.

INDEX.JSX

```
{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
valueField='value' targetField='target' title='Sales Rate' type='Dot'
minimum={0} maximum={100} interval={20} dataSource=[[{ value: 55, target: 75
}]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective
end={35}></BulletRangeDirective>
            <BulletRangeDirective
end={50}></BulletRangeDirective>
            <BulletRangeDirective
end={100}></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

INDEX.TSX

```
{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}
        valueField='value'
        targetField='target'
        title='Sales Rate'
        type='Dot'
        minimum={0}
        maximum={100}
        interval={20}
        dataSource=[[{value: 55, target: 75}]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={35}
></BulletRangeDirective>
            <BulletRangeDirective end={50}
></BulletRangeDirective>
            <BulletRangeDirective end={100}
></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
};
export default App;
```



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

Actual bar customization

Border customization

Using the [valueBorder](#) property of the bullet chart, you can customize the border [color](#) and [width](#) of the actual bar.

INDEX.JSX

```
{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
valueField='value' targetField='target' title='Sales Rate' valueBorder={{
color: 'red', width: 3 }} minimum={0} maximum={100} interval={20}
dataSource=[[{ value: 55, target: 75 }]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective
end={35}></BulletRangeDirective>
            <BulletRangeDirective
end={50}></BulletRangeDirective>
            <BulletRangeDirective
end={100}></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

INDEX.TSX

```
{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}
        valueField='value'
        targetField='target'
        title='Sales Rate'
        valueBorder={{color: 'red', width: 3}}
        minimum={0}
        maximum={100}
        interval={20}
        dataSource=[[{value: 55, target: 75}]]>
    </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

```

        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={35}>
        </BulletRangeDirective>
            <BulletRangeDirective end={50}>
        </BulletRangeDirective>
            <BulletRangeDirective end={100}>
        </BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

Fill color and height customization

Customize the fill color and height of the actual bar using the [valueFill](#) and [valueHeight](#) properties of the bullet chart. Also, you can bind the color for the actual bar from [dataSource](#) for the bullet chart using [valueFill](#) property.

INDEX.JSX

```

{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
valueField='value' targetField='target' title='Sales Rate' valueFill='color'
valueHeight={15} minimum={0} maximum={100} interval={20} dataSource=[{
value: 55, target: 75, color: 'blue' }]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={35}></BulletRangeDirective>
            <BulletRangeDirective end={50}></BulletRangeDirective>
            <BulletRangeDirective end={100}></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

INDEX.TSX

```

{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}

```

```

    valueField='value'
    targetField='target'
    title='Sales Rate'
    valueFill='color'
    valueHeight={15}
    minimum={0}
    maximum={100}
    interval={20}
    dataSource=[[{ value: 55, target: 75, color: 'blue' }]]>
    <BulletRangeCollectionDirective>
      <BulletRangeDirective end={35} ></BulletRangeDirective>
      <BulletRangeDirective end={50} ></BulletRangeDirective>
      <BulletRangeDirective end={100} ></BulletRangeDirective>
    </BulletRangeCollectionDirective>
  </BulletChartComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

Comparative bar in React Bullet chart component

The line marker that runs perpendicular to the orientation of the graph is known as the **Comparative Measure** and it is used as a target marker to compare against the feature measure value. This is also called as the **Target Bar** in the Bullet Chart. To display the target bar, the [targetField](#) should be mapped to the appropriate field from the datasource.

INDEX.JSX

```

{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (<BulletChartComponent id='ranges' animation={{ enable: false }}
    valueField='value' targetField='target' title='Sales Rate' minimum={0}
    maximum={100} interval={20} dataSource=[[{ value: 55, target: 75 }]]>
      <BulletRangeCollectionDirective>
        <BulletRangeDirective
end={35}></BulletRangeDirective>
        <BulletRangeDirective
end={50}></BulletRangeDirective>
        <BulletRangeDirective
end={100}></BulletRangeDirective>
      </BulletRangeCollectionDirective>
    </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

INDEX.TSX

```
{% raw %}
import { BulletChartComponent, Inject} from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective} from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}
        valueField='value'
        targetField='target'
        title='Sales Rate'
        minimum={0}
        maximum={100}
        interval={20}
        dataSource=[[{value: 55, target: 75}]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={35}
></BulletRangeDirective>
            <BulletRangeDirective end={50}
></BulletRangeDirective>
            <BulletRangeDirective end={100}
></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

Types of target bar

The shape of the target bar can be customized using the [targetTypes](#) property and it supports **Circle**, **Cross**, and **Rect** shapes. The default type of the target bar is **Rect**.

INDEX.JSX

```
{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
valueField='value' targetField='target' title='Sales Rate'
targetTypes={['Circle']} minimum={0} maximum={100} interval={20}
dataSource=[[{ value: 55, target: 75 }]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective
end={35}></BulletRangeDirective>
            <BulletRangeDirective
end={50}></BulletRangeDirective>
            <BulletRangeDirective
end={100}></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
};
```

```

}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

INDEX.TSX

```

{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (
        <BulletChartComponent id='ranges'
            animation={{ enable: false }}
            valueField='value'
            targetField='target'
            title='Sales Rate'
            targetTypes={['Circle']}
            minimum={0}
            maximum={100}
            interval={20}
            dataSource={[{value: 55, target: 75}]}>
            <BulletRangeCollectionDirective>
                <BulletRangeDirective end={35}>
            </BulletRangeDirective>
                <BulletRangeDirective end={50}>
            </BulletRangeDirective>
                <BulletRangeDirective end={100}>
            </BulletRangeDirective>
            </BulletRangeCollectionDirective>
        </BulletChartComponent>
    );
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

Target bar customization

The following properties can be used to customize the target bar. Also, you can bind the color for the target bar from [dataSource](#) for the bullet chart.

- [targetColor](#) - Specifies the fill color of Target Bar.
- [targetWidth](#) - Specifies the width of Target Bar.

INDEX.JSX

```

{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";

```

```
function App() {
  return (<BulletChartComponent id='ranges' animation={{ enable: false }}
valueField='value' targetField='target' title='Sales Rate'
targetColor='color' targetWidth={15} minimum={0} maximum={100} interval={20}
dataSource=[[{ value: 55, target: 75, color: 'red' }]]>
    <BulletRangeCollectionDirective>
      <BulletRangeDirective end={35}></BulletRangeDirective>
      <BulletRangeDirective end={50}></BulletRangeDirective>
      <BulletRangeDirective end={100}></BulletRangeDirective>
    </BulletRangeCollectionDirective>
  </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

INDEX.TSX

```
{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (<BulletChartComponent id='ranges'
    animation={{ enable: false }}
    valueField='value'
    targetField='target'
    title='Sales Rate'
    targetColor='color'
    targetWidth={15}
    minimum={0}
    maximum={100}
    interval={20}
    dataSource=[[{ value: 55, target: 75, color: 'red' }]]>
    <BulletRangeCollectionDirective>
      <BulletRangeDirective end={35} ></BulletRangeDirective>
      <BulletRangeDirective end={50} ></BulletRangeDirective>
      <BulletRangeDirective end={100} ></BulletRangeDirective>
    </BulletRangeCollectionDirective>
  </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

Title in React Bullet chart component**Title**

The title of the Bullet Chart displays the information about the data plotted by specifying it in the [title](#) property.

INDEX.JSX

```
{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
valueField='value' targetField='target' title='Sales Rate' minimum={0}
maximum={100} interval={20} dataSource=[[{ value: 55, target: 75 }]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={35}
color='red'></BulletRangeDirective>
            <BulletRangeDirective end={50}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={100}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

INDEX.TSX

```
{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}
        valueField='value'
        targetField='target'
        title='Sales Rate'
        minimum={0}
        maximum={100}
        interval={20}
        dataSource=[[{value: 55, target: 75}]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={35} color='red'
></BulletRangeDirective>
            <BulletRangeDirective end={50}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={100}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

Subtitle

To show additional information about the data plotted, the Bullet Chart can also be given a subtitle using the [subtitle](#) property.

INDEX.JSX

```
{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
valueField='value' targetField='target' title='Sales Rate in dollars'
subtitle='(in dollars $)' minimum={0} maximum={100} interval={20}
dataSource=[{ value: 55, target: 45 }]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={35}
color='red'></BulletRangeDirective>
            <BulletRangeDirective end={50}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={100}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

INDEX.TSX

```
{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}
        valueField='value'
        targetField='target'
        title='Sales Rate in dollars'
        subtitle='(in dollars $)'
        minimum={0}
        maximum={100}
        interval={20}
        dataSource=[{value: 55, target: 45}]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={35} color='red'
></BulletRangeDirective>
```



```

                                <BulletRangeDirective end={50}
color='blue'></BulletRangeDirective>
                                <BulletRangeDirective end={100}
color='green'></BulletRangeDirective>
                                </BulletRangeCollectionDirective>
                                </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

Title and SubTitle Position

The title and the subtitle positions can be customized using the [titlePosition](#) property. Possible positions are **Left**, **Right**, **Top**, and **Bottom**.

Position as Left

By setting the [titlePosition](#) to **Left**, you can display the title and subtitle at the left side of the Bullet Chart.

INDEX.JSX

```

{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
valueField='value' targetField='target' title='Sales Rate in dollars'
subtitle='(in dollars $)' titlePosition='Left' labelFormat='${value}'
minimum={0} maximum={100} interval={20} dataSource=[{ value: 55, target: 45
}]]>
                                <BulletRangeCollectionDirective>
                                <BulletRangeDirective end={35}
color='red'></BulletRangeDirective>
                                <BulletRangeDirective end={50}
color='blue'></BulletRangeDirective>
                                <BulletRangeDirective end={100}
color='green'></BulletRangeDirective>
                                </BulletRangeCollectionDirective>
                                </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

INDEX.TSX

```

{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";

```

```
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}
        valueField='value'
        targetField='target'
        title='Sales Rate in dollars'
        subtitle='(in dollars $)'
        titlePosition='Left'
        labelFormat='${value}'
        minimum={0}
        maximum={100}
        interval={20}
        dataSource=[[{value: 55, target: 45}]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={35} color='red'
></BulletRangeDirective>
            <BulletRangeDirective end={50}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={100}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

Position as Right

By setting the [titlePosition](#) to **Right**, you can display the title and subtitle at the right side of the Bullet Chart.

INDEX.JSX

```
{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
valueField='value' targetField='target' title='Sales Rate in dollars'
subtitle='(in dollars $)' titlePosition='Right' labelFormat='${value}'
minimum={0} maximum={100} interval={20} dataSource=[[{ value: 55, target: 45
}]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={35}
color='red'></BulletRangeDirective>
            <BulletRangeDirective end={50}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={100}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
}
```

```
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

INDEX.TSX

```
{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}
        valueField='value'
        targetField='target'
        title='Sales Rate in dollars'
        subtitle='(in dollars $)'
        titlePosition='Right'
        labelFormat='${value}'
        minimum={0}
        maximum={100}
        interval={20}
        dataSource=[[{value: 55, target: 45}]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={35} color='red'
></BulletRangeDirective>
            <BulletRangeDirective end={50}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={100}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

Position as Top

By setting the [titlePosition](#) to **Top**, you can display the title and subtitle at the top of the Bullet Chart. The default title and subtitle positions of the Bullet Chart is **Top**.

INDEX.JSX

```
{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
valueField='value' targetField='target' title='Sales Rate in dollars'

```

```

subtitle='(in dollars $)' titlePosition='Top' labelFormat='${value}'
minimum={0} maximum={100} interval={20} dataSource=[[{ value: 55, target: 45
}]]>
    <BulletRangeCollectionDirective>
        <BulletRangeDirective end={35}
color='red'></BulletRangeDirective>
        <BulletRangeDirective end={50}
color='blue'></BulletRangeDirective>
        <BulletRangeDirective end={100}
color='green'></BulletRangeDirective>
    </BulletRangeCollectionDirective>
</BulletChartComponent>;
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

INDEX.TSX

```

{% raw %}
import { BulletChartComponent, Inject} from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective} from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}
        valueField='value'
        targetField='target'
        title='Sales Rate in dollars'
        subtitle='(in dollars $)'
        titlePosition='Top'
        labelFormat='${value}'
        minimum={0}
        maximum={100}
        interval={20}
        dataSource=[[{value: 55, target: 45}]]>
            <BulletRangeCollectionDirective>
                <BulletRangeDirective end={35} color='red'
></BulletRangeDirective>
                <BulletRangeDirective end={50}
color='blue'></BulletRangeDirective>
                <BulletRangeDirective end={100}
color='green'></BulletRangeDirective>
            </BulletRangeCollectionDirective>
        </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

Position as Bottom

By setting the [titlePosition](#) to **Bottom**, you can display the title and subtitle at the bottom of the Bullet Chart.

INDEX.JSX

```
{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
valueField='value' targetField='target' title='Sales Rate in dollars'
subtitle='(in dollars $)' titlePosition='Bottom' labelFormat='${value}'
minimum={0} maximum={100} interval={20} dataSource=[[{ value: 55, target: 45
}]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={35}
color='red'></BulletRangeDirective>
            <BulletRangeDirective end={50}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={100}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

INDEX.TSX

```
{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}
        valueField='value'
        targetField='target'
        title='Sales Rate in dollars'
        subtitle='(in dollars $)'
        titlePosition='Bottom'
        labelFormat='${value}'
        minimum={0}
        maximum={100}
        interval={20}
        dataSource=[[{value: 55, target: 45}]]>
        <BulletRangeCollectionDirective>
```

```

                                <BulletRangeDirective end={35} color='red'
></BulletRangeDirective>
                                <BulletRangeDirective end={50}
color='blue'></BulletRangeDirective>
                                <BulletRangeDirective end={100}
color='green'></BulletRangeDirective>
                                </BulletRangeCollectionDirective>
                                </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

Title Customization

The title color, opacity, font size, font family, font weight, and font style can be customized using the [titleStyle](#) property.

INDEX.JSX

```

{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
valueField='value' targetField='target' title='Sales Rate in dollars'
subtitle='(in dollars $)' labelFormat='${value}' minimum={0} maximum={100}
interval={20} titleStyle={{ size: '22px', color: 'red', fontFamily:
'cursive', fontWeight: 'Bold' }} dataSource=[[{ value: 55, target: 45 }]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={35}
color='red'></BulletRangeDirective>
            <BulletRangeDirective end={50}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={100}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

INDEX.TSX

```

{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {

```

```

    return (<BulletChartComponent id='ranges'
      animation={{ enable: false }}
      valueField='value'
      targetField='target'
      title='Sales Rate in dollars'
      subtitle='(in dollars $)'
      labelFormat='${value}'
      minimum={0}
      maximum={100}
      interval={20}
      titleStyle={{size: '22px', color:'red',
fontFamily:'cursive', fontWeight:'Bold'}}
      dataSource=[[{value: 55, target: 45}]]>
        <BulletRangeCollectionDirective>
          <BulletRangeDirective end={35} color='red'
></BulletRangeDirective>
          <BulletRangeDirective end={50}
color='blue'></BulletRangeDirective>
          <BulletRangeDirective end={100}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
      </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

SubTitle Customization

The sub-title color, opacity, font size, font family, font weight, and font style can be customized using the [subtitleStyle](#) property.

INDEX.JSX

```

{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (<BulletChartComponent id='ranges' animation={{ enable: false }}
valueField='value' targetField='target' title='Sales Rate in dollars'
subtitle='(in dollars $)' labelFormat='${value}' minimum={0} maximum={100}
interval={20} subtitleStyle={{ size: '22px', color: 'red', fontFamily:
'cursive', fontWeight: 'Bold' }} dataSource=[[{ value: 55, target: 45 }]]>
    <BulletRangeCollectionDirective>
      <BulletRangeDirective end={35}
color='red'></BulletRangeDirective>
      <BulletRangeDirective end={50}
color='blue'></BulletRangeDirective>
      <BulletRangeDirective end={100}
color='green'></BulletRangeDirective>
    </BulletRangeCollectionDirective>
  </BulletChartComponent>);
}
;

```

```
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

INDEX.TSX

```
{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
  return (<BulletChartComponent id='ranges'
    animation={{ enable: false }}
    valueField='value'
    targetField='target'
    title='Sales Rate in dollars'
    subtitle='(in dollars $)'
    labelFormat='${value}'
    minimum={0}
    maximum={100}
    interval={20}
    subtitleStyle={{ size: '22px', color:'red',
fontFamily:'cursive', fontWeight:'Bold'}}
    dataSource=[[{value: 55, target: 45}]]>
    <BulletRangeCollectionDirective>
      <BulletRangeDirective end={35} color='red'
></BulletRangeDirective>
      <BulletRangeDirective end={50}
color='blue'></BulletRangeDirective>
      <BulletRangeDirective end={100}
color='green'></BulletRangeDirective>
    </BulletRangeCollectionDirective>
  </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

Customization in React Bullet chart component

Orientation

The Bullet Chart can be rendered in different orientations such as **Horizontal** or **Vertical** via the [orientation](#) property. By default, the Bullet Chart is rendered in the **Horizontal** orientation.

INDEX.JSX

```
{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
```



```

    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
valueField='value' targetField='target' title='Sales Rate in dollars'
width='20%' orientation='vertical' minimum={0} maximum={100} interval={20}
dataSource=[[{ value: 55, target: 45 }]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective
end={35}></BulletRangeDirective>
            <BulletRangeDirective
end={50}></BulletRangeDirective>
            <BulletRangeDirective
end={100}></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

INDEX.TSX

```

{% raw %}
import { BulletChartComponent, Inject} from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective} from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}
        valueField='value'
        targetField='target'
        title='Sales Rate in dollars'
        width='20%'
        orientation='vertical'
        minimum={0}
        maximum={100}
        interval={20}
        dataSource=[[{value: 55, target: 45}]]>
            <BulletRangeCollectionDirective>
                <BulletRangeDirective
end={35}></BulletRangeDirective>
                <BulletRangeDirective
end={50}></BulletRangeDirective>
                <BulletRangeDirective
end={100}></BulletRangeDirective>
            </BulletRangeCollectionDirective>
        </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

Right-to-left (RTL)

The Bullet Chart supports the right-to-left rendering that can be enabled by setting the [enableRtl](#) property to **true**.

INDEX.JSX

```
{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
valueField='value' targetField='target' enableRtl={true} minimum={0}
maximum={2000} interval={200} dataSource=[[{ value: 1500, target: 1000 }]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={500}
color='red'></BulletRangeDirective>
            <BulletRangeDirective end={1500}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={2000}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

INDEX.TSX

```
{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}
        valueField='value'
        targetField='target'
        enableRtl={true}
        minimum={0}
        maximum={2000}
        interval={200}
        dataSource=[[{value: 1500, target: 1000}]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={500} color='red'
></BulletRangeDirective>
            <BulletRangeDirective end={1500}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={2000}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

```

        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

Animation

The actual and the target bar supports the linear animation via the [animation](#) setting. The speed and the delay are controlled using the `duration` and `delay` properties respectively.

INDEX.JSX

```

{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: true }}
valueField='value' targetField='target' minimum={0} maximum={2000}
interval={200} dataSource=[[{ value: 1500, target: 1000 }]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={500}
color='red'></BulletRangeDirective>
            <BulletRangeDirective end={1500}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={2000}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

INDEX.TSX

```

{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: true }}
        valueField='value'
        targetField='target'
        minimum={0}
        maximum={2000}
        interval={200}
        dataSource=[[{value: 1500, target: 1000}]]>

```

```

        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={500} color='red'
></BulletRangeDirective>
            <BulletRangeDirective end={1500}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={2000}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

Theme

The Bullet Chart supports different type of themes via the [theme](#) property.

INDEX.JSX

```

{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
valueField='value' targetField='target' title='Profit in %'
theme='HighContrast' minimum={0} maximum={100} interval={10} dataSource=[[{
value: 50, target: 45 }]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={15}
color='red'></BulletRangeDirective>
            <BulletRangeDirective end={50}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={100}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

INDEX.TSX

```

{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}

```

```

        valueField='value'
        targetField='target'
        title='Profit in %'
        theme='HighContrast'
        minimum={0}
        maximum={100}
        interval={10}
        dataSource=[[{value: 50, target: 45}]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={15} color='red'
></BulletRangeDirective>
            <BulletRangeDirective end={50}
color='blue'></BulletRangeDirective>
            <BulletRangeDirective end={100}
color='green'></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% enddraw %}

```

Data label in React Bullet chart component

Data Labels are used to identify the value of actual bar in the Bullet Chart component. The Data Labels will be shown by specifying the [dataLabel](#) setting's [enable](#) property to **true**.

INDEX.JSX

```

{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
height="400" valueField='value' targetField='comparativeMeasureValue'
categoryField='category' title='Profit in %' minimum={0} maximum={20}
interval={5} dataLabel={{ enable: true }} dataSource=[{ value: 5,
comparativeMeasureValue: 7.5, category: '2001' },
    { value: 7, comparativeMeasureValue: 5, category: '2002' },
    { value: 10, comparativeMeasureValue: 6, category: '2003' },
    { value: 5, comparativeMeasureValue: 8, category: '2004' },
    { value: 12, comparativeMeasureValue: 5, category: '2005' },
    { value: 8, comparativeMeasureValue: 6, category: '2006' } ]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective
end={5}></BulletRangeDirective>
            <BulletRangeDirective
end={15}></BulletRangeDirective>
            <BulletRangeDirective
end={20}></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>;
}
;

```

```
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

INDEX.TSX

```
{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}
        height="400"
        valueField='value'
        targetField='comparativeMeasureValue'
        categoryField='category'
        title='Profit in %'
        minimum={0}
        maximum={20}
        interval={5}
        dataLabel={{enable: true}}
        dataSource=[[{value: 5, comparativeMeasureValue: 7.5,
category: '2001'},
                    {value: 7, comparativeMeasureValue: 5, category:
'2002'},
                    {value: 10, comparativeMeasureValue: 6, category:
'2003'},
                    {value: 5, comparativeMeasureValue: 8, category:
'2004'},
                    {value: 12, comparativeMeasureValue: 5, category:
'2005'},
                    {value: 8, comparativeMeasureValue: 6, category:
'2006'}]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={5}
></BulletRangeDirective>
            <BulletRangeDirective
end={15}></BulletRangeDirective>
            <BulletRangeDirective
end={20}></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

Data Label Customization

Data Labels color, opacity, font size, font family, font weight, and font style can be customized using the [labelStyle](#).

INDEX.JSX

```
{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
height="400" valueField='value' targetField='comparativeMeasureValue'
categoryField='category' title='Profit in %' minimum={0} maximum={20}
interval={5} dataLabel={{ enable: true, labelStyle: { color: 'yellow', size:
'20' } }} dataSource=[{ value: 5, comparativeMeasureValue: 7.5, category:
'2001' },
    { value: 7, comparativeMeasureValue: 5, category: '2002' },
    { value: 10, comparativeMeasureValue: 6, category: '2003' },
    { value: 5, comparativeMeasureValue: 8, category: '2004' },
    { value: 12, comparativeMeasureValue: 5, category: '2005' },
    { value: 8, comparativeMeasureValue: 6, category: '2006' } ]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective
end={5}></BulletRangeDirective>
            <BulletRangeDirective
end={15}></BulletRangeDirective>
            <BulletRangeDirective
end={20}></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}
```

INDEX.TSX

```
{% raw %}
import { BulletChartComponent, Inject } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}
        height="400"
        valueField='value'
        targetField='comparativeMeasureValue'
        categoryField='category'
        title='Profit in %'
        minimum={0}
        maximum={20}
        interval={5}
        dataLabel={{enable: true, labelStyle:
{color:'yellow', size: '20'}}}}
        dataSource=[{value: 5, comparativeMeasureValue: 7.5,
category: '2001'},
```

```

        {value: 7, comparativeMeasureValue: 5, category:
'2002'},
        {value: 10, comparativeMeasureValue: 6, category:
'2003'},
        {value: 5, comparativeMeasureValue: 8, category:
'2004'},
        {value: 12, comparativeMeasureValue: 5, category:
'2005'},
        {value: 8, comparativeMeasureValue: 6, category:
'2006' }]]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={5}
></BulletRangeDirective>
            <BulletRangeDirective
end={15}></BulletRangeDirective>
            <BulletRangeDirective
end={20}></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>;
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% endraw %}

```

Tool tip in React Bullet chart component

When the mouse is hovered over a bar in the Bullet Chart, the tooltip displays important summary about the actual and the target bar values.

Default Tooltip

The tooltip is not visible by default. To make it visible, set the [enable](#) property in the [tooltip](#) to **true**.

INDEX.JSX

```

{% raw %}
import { BulletChartComponent } from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective } from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges' animation={{ enable: false }}
height="400" valueField='value' targetField='comparativeMeasureValue'
categoryField='category' title='Profit in %' minimum={0} maximum={20}
interval={5} tooltip={{ enable: true }} dataSource=[[{ value: 5,
comparativeMeasureValue: 7.5, category: '2001' },
        { value: 7, comparativeMeasureValue: 5, category: '2002' },
        { value: 10, comparativeMeasureValue: 6, category: '2003' },
        { value: 5, comparativeMeasureValue: 8, category: '2004' },
        { value: 12, comparativeMeasureValue: 5, category: '2005' },
        { value: 8, comparativeMeasureValue: 6, category: '2006' },]]>
            <BulletRangeCollectionDirective>
                <BulletRangeDirective
end={5}></BulletRangeDirective>
                <BulletRangeDirective
end={15}></BulletRangeDirective>
            </BulletRangeCollectionDirective>
        </BulletChartComponent>);
}
ReactDOM.render(App(), document.getElementById("charts"));
{% endraw %}

```



```

                                <BulletRangeDirective
end={20}></BulletRangeDirective>
                                </BulletRangeCollectionDirective>
                                </BulletChartComponent>);
}
;
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% enddraw %}

```

INDEX.TSX

```

{% raw %}
import { BulletChartComponent, Inject} from '@syncfusion/ej2-react-charts';
import { BulletRangeCollectionDirective, BulletRangeDirective} from
 '@syncfusion/ej2-react-charts';
import * as React from "react";
import * as ReactDOM from "react-dom";
function App() {
    return (<BulletChartComponent id='ranges'
        animation={{ enable: false }}
        height="400"
        valueField='value'
        targetField='comparativeMeasureValue'
        categoryField='category'
        title='Profit in %'
        minimum={0}
        maximum={20}
        interval={5}
        tooltip={{enable: true}}
        dataSource=[[{value: 5, comparativeMeasureValue: 7.5,
category: '2001'},
                    {value: 7, comparativeMeasureValue: 5, category:
'2002'},
                    {value: 10, comparativeMeasureValue: 6, category:
'2003'},
                    {value: 5, comparativeMeasureValue: 8, category:
'2004'},
                    {value: 12, comparativeMeasureValue: 5, category:
'2005'},
                    {value: 8, comparativeMeasureValue: 6, category:
'2006'}],]>
        <BulletRangeCollectionDirective>
            <BulletRangeDirective end={5}
></BulletRangeDirective>
            <BulletRangeDirective
end={15}></BulletRangeDirective>
            <BulletRangeDirective
end={20}></BulletRangeDirective>
        </BulletRangeCollectionDirective>
    </BulletChartComponent>);
};
export default App;
ReactDOM.render(<App />, document.getElementById("charts"));
{% enddraw %}

```

Tooltip Template

Any HTML elements can be displayed in the tooltip by using the [template](#) property of the [tooltip](#). You can use the `${target}` and `${value}` as place holders in the HTML element to display the value and target values from the data source of corresponding data point.

Customize the Appearance of 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.

Tooltip Customization

The following properties can be used to customize the Bullet Chart tooltip.

- [fill](#) - Specifies the color of tooltip.
- [border](#) - Specifies the tooltip border color and width.
- [textStyle](#) - Specifies the tooltip font family, font style, font weight, color and size.

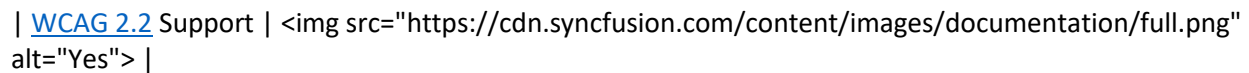
Accessibility in React Bullet chart component

The Bullet 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 Bullet chart component is outlined below.

| Accessibility Criteria | Compatibility |

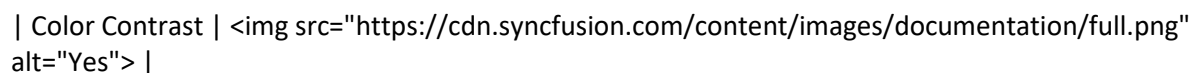
| -- | -- |

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

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

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

| Right-To-Left Support |  alt="Yes" > |

| 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>
```

WAI-ARIA attributes

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

- `img (role)`
- `button (role)`
- `aria-label (attribute)`
- `aria-pressed (attribute)`

Keyboard interaction

The Bullet 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 Bullet chart component.

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

| --- | --- |

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

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

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

Ensuring accessibility

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

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

See also

- [Accessibility in Syncfusion React components](#)

ButtonGroup

Getting Started

This section explains how to create a simple ButtonGroup, and configure its available functionalities in React by using the React quickstart application.

Dependencies

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

```
`javascript
|-- @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.

```
`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 ButtonGroup component, use the following command

```
`bash
npm install @syncfusion/ej2-splitbuttons --save
`
```

Adding CSS Reference

Import the ButtonGroup 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";
@import "../node_modules/@syncfusion/ej2-popups/styles/material.css";
@import "../node_modules/@syncfusion/ej2-splitbuttons/styles/material.css";
`
```

Adding ButtonGroup component to the Application

To include the ButtonGroup component in your application, add a HTML div tag with class name as `e-btn-group` and to render button

as react component, [Button dependencies](#) should be added.

Then import the `ButtonComponent` from `ej2-react-buttons` package and add button elements that should be group inside the div in `App.tsx`.

Add the ButtonGroup component in application as shown in below code example.

```
`ts
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import './App.css';
// Import the Button.
// To render ButtonGroup.
function App() {
  return (
    <div className='e-btn-group' style={{marginTop: '150px'}}>
    <ButtonComponent>HTML</ButtonComponent>
    <ButtonComponent>CSS</ButtonComponent>
    <ButtonComponent>Javascript</ButtonComponent>
    </div>
  );
}
export default App;
`
```

Run the application

Run the application in the browser by using the following command:

```
npm start
```

The following example shows a basic ButtonGroup component.

APP.JSX

```
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render ButtonGroup.
function App() {
    return (
        <div>
            <div className='e-btn-group'>
                <ButtonComponent>HTML</ButtonComponent>
                <ButtonComponent>CSS</ButtonComponent>
                <ButtonComponent>Javascript</ButtonComponent>
            </div>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));
```

APP.TSX

```
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render ButtonGroup.
function App() {
    return (
        <div>
            <div className='e-btn-group'>
                <ButtonComponent>HTML</ButtonComponent>
                <ButtonComponent>CSS</ButtonComponent>
                <ButtonComponent>Javascript</ButtonComponent>
            </div>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));
```

See Also

- [Initialize ButtonGroup using util function](#)

Types and styles in React Button group component

This section explains about different types and styles of ButtonGroup.

ButtonGroup types

Outline ButtonGroup

An Outline ButtonGroup has a border with transparent background. To create Outline ButtonGroup, `e-outline` class should be added to the target element and to the button element using `cssClass` property.

The following sample illustrates how to achieve an Outline ButtonGroup,

APP.JSX

```
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render outline ButtonGroup.
function App() {
    return (
        <div>
            <div className='e-btn-group e-outline'>
                <ButtonComponent cssClass='e-outline'>HTML</ButtonComponent>
                <ButtonComponent cssClass='e-outline'>CSS</ButtonComponent>
                <ButtonComponent cssClass='e-outline'>Javascript</ButtonComponent>
            </div>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));
```

APP.TSX

```
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render outline ButtonGroup.
function App() {
    return (
        <div>
            <div className='e-btn-group e-outline'>
                <ButtonComponent cssClass='e-outline'>HTML</ButtonComponent>
                <ButtonComponent cssClass='e-outline'>CSS</ButtonComponent>
                <ButtonComponent cssClass='e-outline'>Javascript</ButtonComponent>
            </div>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));
```

ButtonGroup does not have support for **flat** and **round** types.

ButtonGroup styles

The Essential JS 2 ButtonGroup has the following predefined styles. This can be achieved by adding corresponding class name in each button elements using **cssClass** property.

Class	Description
-----	-----
e-primary	Used to represent a primary action.
e-success	Used to represent a positive action.
e-info	Used to represent an informative action.
e-warning	Used to represent an action with caution.

| e-danger | Used to represent a negative action. |

The following example illustrates how to achieve predefined styles in ButtonGroup.

APP.JSX

```
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render ButtonGroup with different styles.
function App() {
    return (
        <div>
            <div className='e-btn-group'>
                <ButtonComponent cssClass='e-info'>View</ButtonComponent>
                <ButtonComponent>Edit</ButtonComponent>
                <ButtonComponent cssClass='e-danger'>Delete</ButtonComponent>
            </div>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));
```

APP.TSX

```
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render ButtonGroup with different styles.
function App() {
    return (
        <div>
            <div className='e-btn-group'>
                <ButtonComponent cssClass='e-info'>View</ButtonComponent>
                <ButtonComponent>Edit</ButtonComponent>
                <ButtonComponent cssClass='e-danger'>Delete</ButtonComponent>
            </div>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));
```

Predefined ButtonGroup styles provide only the visual indication. So, ButtonGroup content should define the ButtonGroup style for the users of assistive technologies such as screen readers.

See Also

- [ButtonGroup with icons](#)
- [Create ButtonGroup with rounded corner](#)

Selection and nesting in React Button group component

Selection

Single selection

ButtonGroup supports radio type selection in which only one button can be selected. This can be achieved by adding input element

along with `id` attribute with its corresponding label along with `htmlFor` attribute inside the target element. In this ButtonGroup,

the type of the input element should be `radio` and `e-btn` is added to the `label` element.

The following example illustrates the single selection behavior in ButtonGroup.

APP.JSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render radio type ButtonGroup.
function App() {
  return (
    <div>
      <div className='e-btn-group'>
        <input type="radio" id="radioleft" name="align" value="left"/>
        <label className="e-btn" htmlFor="radioleft">Left</label>
        <input type="radio" id="radiomiddle" name="align" value="middle"/>
        <label className="e-btn" htmlFor="radiomiddle">Center</label>
        <input type="radio" id="radiatoright" name="align" value="right"/>
        <label className="e-btn" htmlFor="radiatoright">Right</label>
      </div>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));
```

APP.TSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render radio type ButtonGroup.
function App() {
  return (
    <div>
      <div className='e-btn-group'>
        <input type="radio" id="radioleft" name="align" value="left" />
        <label className="e-btn" htmlFor="radioleft">Left</label>
        <input type="radio" id="radiomiddle" name="align" value="middle" />
        <label className="e-btn" htmlFor="radiomiddle">Center</label>
        <input type="radio" id="radiatoright" name="align" value="right"/>
        <label className="e-btn" htmlFor="radiatoright">Right</label>
      </div>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));
```

Multiple selection

ButtonGroup supports checkbox type selection in which multiple button can be selected. This can be achieved by adding input element along with `id` attribute with its corresponding label along with `htmlFor` attribute inside the target element. In this ButtonGroup, the type of the input element should be `checkbox` and `e-btn` is added to the `label` element.

The following example illustrates the multiple selection behavior in ButtonGroup.

APP.JSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render checkbox type ButtonGroup.
function App() {
  return (
    <div>
      <div className='e-btn-group'>
        <input type="checkbox" id="checkbold" name="font" value="bold"/>
        <label className="e-btn" htmlFor="checkbold">Bold</label>
        <input type="checkbox" id="checkitalic" name="font"
value="italic"/>
        <label className="e-btn" htmlFor="checkitalic">Italic</label>
        <input type="checkbox" id="checkline" name="font"
value="underline"/>
        <label className="e-btn" htmlFor="checkline">Underline</label>
      </div>
    </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));
```

APP.TSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render checkbox type ButtonGroup.
function App() {
  return (
    <div>
      <div className='e-btn-group'>
        <input type="checkbox" id="checkbold" name="font" value="bold"/>
        <label className="e-btn" htmlFor="checkbold">Bold</label>
        <input type="checkbox" id="checkitalic" name="font" value="italic"
/>
        <label className="e-btn" htmlFor="checkitalic">Italic</label>
        <input type="checkbox" id="checkline" name="font"
value="underline"/>
        <label className="e-btn" htmlFor="checkline">Underline</label>
      </div>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));
```

Nesting

Nesting with other components can be possible in ButtonGroup. The following components can be nested in ButtonGroup,

- DropDownButton
- SplitButton

For nesting support, [SplitButton dependencies](#) should be configured and it should be added in system.config.js.

DropDownButton

To initialize DropDownButton component, refer [DropDownButton Getting Started documentation](#).

In the following example, DropDownButton component is added in `src/App.tsx` by importing it from `ej2-react-splitbuttons`.

APP.JSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { DropDownButtonComponent } from '@syncfusion/ej2-react-splitbuttons';
// To render ButtonGroup with DropDownButton nesting.
function App() {
  let items = [
    {
      text: 'Learn SQL'
    },
    {
      text: 'Learn PHP'
    },
    {
      text: 'Learn Bootstrap'
    }
  ];
  return (
    <div>
      <div className='e-btn-group'>
        <ButtonComponent>HTML</ButtonComponent>
        <ButtonComponent>CSS</ButtonComponent>
        <ButtonComponent>Javascript</ButtonComponent>
        <DropDownButtonComponent id="dropdownnelement" items={items}> More
      </DropDownButtonComponent>
      </div>
    </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));
```

APP.TSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { DropDownButtonComponent, ItemModel } from '@syncfusion/ej2-react-splitbuttons';
```

```
// To render ButtonGroup with DropDownButton nesting.
function App() {
  let items: ItemModel[] = [
    {
      text: 'Learn SQL'
    },
    {
      text: 'Learn PHP'
    },
    {
      text: 'Learn Bootstrap'
    }
  ];
  return (
    <div>
      <div className='e-btn-group'>
        <ButtonComponent>HTML</ButtonComponent>
        <ButtonComponent>CSS</ButtonComponent>
        <ButtonComponent>Javascript</ButtonComponent>
        <DropDownButtonComponent id="dropdownnelement" items={items}> More
      </DropDownButtonComponent>
      </div>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));
```

SplitButton

To initialize SplitButton component, refer [SplitButton Getting Started documentation](#).

In the following example, SplitButton component is added in `src/App.tsx` by importing it from `ej2-react-splitbuttons`.

APP.JSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { SplitButtonComponent } from '@syncfusion/ej2-react-splitbuttons';
// To render ButtonGroup with SplitButton nesting.
function App() {
  let items = [
    {
      text: 'Paste'
    },
    {
      text: 'Paste Text'
    },
    {
      text: 'Paste Special'
    }
  ];
  return (
    <div>
      <div className='e-btn-group'>
        <ButtonComponent>Cut</ButtonComponent>
        <ButtonComponent>Copy</ButtonComponent>

```

```

        <SplitButtonComponent id="splitbuttonelement" items={items}> Paste
    </SplitButtonComponent>
    </div>
    </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));

```

APP.TSX

```

import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { SplitButtonComponent, ItemModel } from '@syncfusion/ej2-react-splitbuttons';
// To render ButtonGroup with SplitButton nesting.
function App() {
    let items: ItemModel[] = [
        {
            text: 'Paste'
        },
        {
            text: 'Paste Text'
        },
        {
            text: 'Paste Special'
        }
    ];
    return (
        <div>
            <div className='e-btn-group'>
                <ButtonComponent>Cut</ButtonComponent>
                <ButtonComponent>Copy</ButtonComponent>
                <SplitButtonComponent id="splitbuttonelement" items={items}> Paste
            </SplitButtonComponent>
            </div>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));

```

See Also

- [Show ButtonGroup selected state on initial render](#)

Accessibility in React ButtonGroup component

The ButtonGroup 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 ButtonGroup 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

The ButtonGroup 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 ButtonGroup component.

Normal behavior

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

| --- | --- |

| **Tab** | **Focuses the next button in the ButtonGroup.** |

| Enter/Space | Activates the focussed button in the ButtonGroup. |

Checkbox behavior

| Press | To do this |

| --- | --- |

| Tab | Focuses the next button in the ButtonGroup. |

| Space | Activates the focussed button in the ButtonGroup. |

Radiobutton behavior

| Press | To do this |

| --- | --- |

| Tab | Focuses the active button in the ButtonGroup. |

| Right | Activates next/previous button in the ButtonGroup. |

Ensuring accessibility

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

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

See also

- [Accessibility in Syncfusion React components](#)

Style and appearance in React Button group component

To modify the ButtonGroup appearance, you need to override the default CSS of ButtonGroup component. Please find the list of CSS classes and its corresponding section in ButtonGroup. Also, you have an option to create your own custom theme for the controls using our [Theme Studio](#).

CSS Class | Purpose of Class

|.e-btn|To customize the ButtonGroup.

|.e-btn: hover|To customize the ButtonGroup on hover.

|.e-btn: focus|To customize the ButtonGroup on focus.

|.e-btn: active|To customize the ButtonGroup on active.

How To

Create buttongroup with icons in React Button group component

ButtonGroup with icons can be achieved by using `iconCss` property of the Button component.

The following example illustrates how to create ButtonGroup with icons.

APP.JSX

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

```
// To render ButtonGroup with icons.
function App() {
  return (<div>
    <div className='e-btn-group'>
      <ButtonComponent iconCss='e-icons e-left-icon'>HTML</ButtonComponent>
      <ButtonComponent iconCss='e-icons e-middle-
icon'>CSS</ButtonComponent>
      <ButtonComponent iconCss='e-icons e-right-
icon'>Javascript</ButtonComponent>
    </div>
  </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));
```

APP.TSX

```
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render ButtonGroup with icons.
function App() {
  return (
    <div>
      <div className='e-btn-group'>
        <ButtonComponent iconCss='e-icons e-left-icon'>HTML</ButtonComponent>
        <ButtonComponent iconCss='e-icons e-middle-
icon'>CSS</ButtonComponent>
        <ButtonComponent iconCss='e-icons e-right-
icon'>Javascript</ButtonComponent>
      </div>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));
```

Create buttongroup with rounded corner in React Button group component

The ButtonGroup with rounded corner has round edges on both side. In the ButtonGroup with rounded corner, **e-round-corner** class is to be

added to the target element.

The following example illustrates how to create ButtonGroup with rounded corner.

APP.JSX

```
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render ButtonGroup with rounded corner.
function App() {
  return (<div>
    <div className='e-btn-group e-round-corner'>
      <ButtonComponent>HTML</ButtonComponent>
      <ButtonComponent>CSS</ButtonComponent>
    </div>
  </div>);
}
```



```

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

```

APP.TSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render ButtonGroup with rounded corner.
function App() {
  return (
    <div>
      <div className='e-btn-group e-round-corner'>
        <ButtonComponent>HTML</ButtonComponent>
        <ButtonComponent>CSS</ButtonComponent>
        <ButtonComponent>Javascript</ButtonComponent>
      </div>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));

```

Orientation in React Button group component

ButtonGroup can be arranged in a vertical and horizontal orientation. By default, it is horizontally aligned.

Vertical Orientation

ButtonGroup can be aligned vertically by using the built-in CSS class **e-vertical** to the target element.

The following example illustrates how to achieve vertical orientation in ButtonGroup.

APP.JSX

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render ButtonGroup with vertical orientation.
function App() {
  return (
    <div>
      <div className='e-btn-group e-vertical'>
        <ButtonComponent>HTML</ButtonComponent>
        <ButtonComponent>CSS</ButtonComponent>
        <ButtonComponent>Javascript</ButtonComponent>
      </div>
    </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));

```

APP.TSX

```
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render ButtonGroup with vertical orientation.
function App() {
  return (
    <div>
      <div className='e-btn-group e-vertical'>
        <ButtonComponent>HTML</ButtonComponent>
        <ButtonComponent>CSS</ButtonComponent>
        <ButtonComponent>Javascript</ButtonComponent>
      </div>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));
```

ButtonGroup does not support SplitButton component nesting in a vertical orientation.

Disable in React Button group component

Particular button

To disable a particular button in a ButtonGroup, [disabled](#) attribute should be added to corresponding button element.

Whole ButtonGroup

To disable whole ButtonGroup, [disabled](#) attribute should be added to all the button elements.

The following example illustrates how to disable the particular and the whole ButtonGroup.

APP.JSX

```
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To disable single button/whole ButtonGroup.
function App() {
  return (<div>
    <div className='e-btn-group'>
      <ButtonComponent>HTML</ButtonComponent>
      <ButtonComponent disabled={true}>CSS</ButtonComponent>
      <ButtonComponent>Javascript</ButtonComponent>
    </div>
    <div className='e-btn-group'>
      <ButtonComponent disabled={true}>HTML</ButtonComponent>
      <ButtonComponent disabled={true}>CSS</ButtonComponent>
      <ButtonComponent disabled={true}>Javascript</ButtonComponent>
    </div>
  </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));
```

APP.TSX

```
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To disable single button/whole ButtonGroup.
function App() {
  return (
    <div>
      <div className='e-btn-group'>
        <ButtonComponent>HTML</ButtonComponent>
        <ButtonComponent disabled = {true}>CSS</ButtonComponent>
        <ButtonComponent>Javascript</ButtonComponent>
      </div>
      <div className='e-btn-group'>
        <ButtonComponent disabled = {true}>HTML</ButtonComponent>
        <ButtonComponent disabled = {true}>CSS</ButtonComponent>
        <ButtonComponent disabled = {true}>Javascript</ButtonComponent>
      </div>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));
```

To disable radio/checkbox type ButtonGroup, the `disabled` attribute should be added to the particular input element.

Enable ripple in React Button group component

Ripple can be enabled by importing `enableRipple` method from `ej2-base` and set `enableRipple` as `true`.

The following example illustrates how to enable ripple for ButtonGroup.

APP.JSX

```
import { enableRipple } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To enable ripple in ButtonGroup.
function App() {
  return (<div>
    <div className='e-btn-group'>
      <ButtonComponent>HTML</ButtonComponent>
      <ButtonComponent>CSS</ButtonComponent>
      <ButtonComponent>Javascript</ButtonComponent>
    </div>
  </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));
```

APP.TSX

```
import { enableRipple } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
```

```
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To enable ripple in ButtonGroup.
function App() {
  return (
    <div>
      <div className='e-btn-group'>
        <ButtonComponent>HTML</ButtonComponent>
        <ButtonComponent>CSS</ButtonComponent>
        <ButtonComponent>Javascript</ButtonComponent>
      </div>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));
```

Enable rtl in React Button group component

ButtonGroup supports RTL functionality. This can be achieved by adding `e-rtl` class to the target element.

The following example illustrates how to create ButtonGroup with RTL support.

APP.JSX

```
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render ButtonGroup with RTL.
function App() {
  return (<div>
    <div className='e-btn-group e-rtl'>
      <ButtonComponent>HTML</ButtonComponent>
      <ButtonComponent>CSS</ButtonComponent>
      <ButtonComponent>Javascript</ButtonComponent>
    </div>
  </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));
```

APP.TSX

```
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render ButtonGroup with RTL.
function App() {
  return (
    <div>
      <div className='e-btn-group e-rtl'>
        <ButtonComponent>HTML</ButtonComponent>
        <ButtonComponent>CSS</ButtonComponent>
        <ButtonComponent>Javascript</ButtonComponent>
      </div>
    </div>
  );
}
```

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

Form submit in React Button group component

The name attribute of the input element is used to group the radio/checkbox type ButtonGroup. When the radio/checkbox type are grouped in the form, the checked items value attribute will be posted to the server on form submit that can be retrieved through the name. The disabled radio/checkbox type value will not be sent to the server on form submit.

In the following code snippet, the radio type ButtonGroup is explained with male value as checked. Now, the value that is in checked state will be sent on form submit.

APP.JSX

```
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render ButtonGroup in form.
function App() {
  function componentDidMount() {
    document.getElementById('female').checked = true;
  }
  return (
    <div>
      <form>
        <div className='e-btn-group'>
          <input type="radio" id="male" name="gender" value="male"/>
          <label className="e-btn" htmlFor="male">Male</label>
          <input type="radio" id="female" name="gender" value="female"/>
          <label className="e-btn" htmlFor="female">Female</label>
          <input type="radio" id="transgender" name="gender"
value="transgender"/>
          <label className="e-btn" htmlFor="transgender">Transgender</label>
        </div>
        <ButtonComponent isPrimary={true}>Submit</ButtonComponent>
      </form>
    </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));
```

APP.TSX

```
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render ButtonGroup in form.
function App() {
  function componentDidMount(): void {
    (document.getElementById('female') as HTMLInputElement).checked = true;
  }
  return (
    <div>
      <form>
```

```

    <div className='e-btn-group'>
      <input type="radio" id="male" name="gender" value="male"/>
      <label className="e-btn" htmlFor="male">Male</label>
      <input type="radio" id="female" name="gender" value="female"/>
      <label className="e-btn" htmlFor="female">Female</label>
      <input type="radio" id="transgender" name="gender"
value="transgender"/>
      <label className="e-btn" htmlFor="transgender">Transgender</label>
    </div>
    <ButtonComponent isPrimary={true}>Submit</ButtonComponent>
  </form>
</div>
);
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));

```

Initialize buttongroup using util function in React Button group component

Though, it is a CSS component for easy initialization of ButtonGroup `createButtonGroup` util function can be used.

To use `createButtonGroup` util function, [SplitButton dependencies](#) should be configured and it should be added in `system.config.js`.

Using `createButtonGroup` method, the Button options, selector, and cssClass is passed and the corresponding classes is added to the elements.

Create basic ButtonGroup

To create a basic ButtonGroup, the target element along with the button elements should be created and `createButtonGroup` is to be imported from `ej2-splitbuttons`.

For radio type ButtonGroup

To create a radio type ButtonGroup, the target element along with the input elements should be created with type `radio`.

For checkbox type ButtonGroup

Checkbox type ButtonGroup creation is similar to radio type ButtonGroup, instead the type of the input elements should be `checkbox`.

The following example illustrates how to create ButtonGroup using `createButtonGroup` function for basic, checkbox, and radio type behaviors.

APP.JSX

```

import { createButtonGroup } from '@syncfusion/ej2-splitbuttons';
import { useEffect } from 'react';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render ButtonGroup using `util` function.
function App() {
  useEffect(() => {
    createButtonGroup('#basic', {
      buttons: [
        { content: 'HTML' },
        { content: 'CSS' },

```

```

        { content: 'Javascript' }
      ]
    });
    createButtonGroup('#checkbox', {
      buttons: [
        { content: 'Bold' },
        { content: 'Italic' },
        { content: 'Undeline' }
      ]
    });
    createButtonGroup('#radio', {
      buttons: [
        { content: 'Left' },
        { content: 'Center' },
        { content: 'Right' }
      ]
    });
  }, []);
  return (<div>
    <h5>Normal behavior</h5>
    <div id='basic'>
      <button />
      <button />
      <button />
    </div>
    <h5>Checkbox type behavior</h5>
    <div id='checkbox'>
      <input type="checkbox" id="checkbold" name="font" value='bold' />
      <input type="checkbox" id="checkitalic" name="font" value='italic' />
      <input type="checkbox" id="checkunderline" name="font"
value='underline' />
    </div>
    <h5>Radiobutton type behavior</h5>
    <div id='radio'>
      <input type="radio" id="radioleft" name="align" value='left' />
      <input type="radio" id="radiomiddle" name="align" value='middle' />
      <input type="radio" id="radiatoright" name="align" value='right' />
    </div>
  </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));

```

APP.TSX

```

import { createButtonGroup } from '@syncfusion/ej2-splitbuttons';
import { useEffect } from 'react';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render ButtonGroup using `util` function.
function App() {
  useEffect(() => {
    createButtonGroup('#basic', {
      buttons: [
        { content: 'HTML' },
        { content: 'CSS' },

```

```

        { content: 'Javascript' }
      ]
    });
    createButtonGroup('#checkbox', {
      buttons: [
        { content: 'Bold' },
        { content: 'Italic' },
        { content: 'Undeline' }
      ]
    });
    createButtonGroup('#radio', {
      buttons: [
        { content: 'Left' },
        { content: 'Center' },
        { content: 'Right' }
      ]
    });
  }, []);

  return (
    <div>
      <h5>Normal behavior</h5>
      <div id='basic'>
        <button/>
        <button/>
        <button/>
      </div>
      <h5>Checkbox type behavior</h5>
      <div id='checkbox'>
        <input type="checkbox" id="checkbold" name="font" value='bold' />
        <input type="checkbox" id="checkitalic" name="font" value='italic' />
        <input type="checkbox" id="checkunderline" name="font"
value='underline' />
      </div>
      <h5>Radiobutton type behavior</h5>
      <div id='radio'>
        <input type="radio" id="radioleft" name="align" value='left' />
        <input type="radio" id="radiomiddle" name="align" value='middle' />
        <input type="radio" id="radiatoright" name="align" value='right' />
      </div>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));

```

If null value is passed in button options, then that particular button will be skipped from processing in `createButtonGroup` util function.

Show `buttongroup` selected state on initial render in React Button group component

To show selected state on initial render, [Link to the Video](#) property should be added to the corresponding input element.

The following example illustrates how to show selected state on initial render.

APP.JSX


```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render checkbox type ButtonGroup.
function App() {
  return (<div>
    <div className='e-btn-group'>
      <input type="checkbox" id="checkbold" name="font" value="bold"
checked={true}/>
      <label className="e-btn" htmlFor="checkbold">Bold</label>
      <input type="checkbox" id="checkitalic" name="font"
value="italic"/>
      <label className="e-btn" htmlFor="checkitalic">Italic</label>
      <input type="checkbox" id="checkline" name="font"
value="underline"/>
      <label className="e-btn" htmlFor="checkline">Underline</label>
    </div>
  </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));
```

APP.TSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
// To render checkbox type ButtonGroup.
function App() {
  return (
    <div>
      <div className='e-btn-group'>
        <input type="checkbox" id="checkbold" name="font" value="bold"
checked={true}/>
        <label className="e-btn" htmlFor="checkbold">Bold</label>
        <input type="checkbox" id="checkitalic" name="font" value="italic"
/>
        <label className="e-btn" htmlFor="checkitalic">Italic</label>
        <input type="checkbox" id="checkline" name="font"
value="underline"/>
        <label className="e-btn" htmlFor="checkline">Underline</label>
      </div>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('buttongroup'));
```

Button

Getting Started

This section explains how to create a simple Button and to configure the Button component.

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

Dependencies

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

```
`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](https://www.npmjs.com) public registry.

To install Button component, use the following command

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

Adding CSS Reference

Import the Button 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 Button component to the Application

To include the Button component in your application import the **ButtonComponent** from **ej2-react-buttons** package in **App.tsx**.

Add the Button component in application as shown in below code example.

```
`ts  
// Import the Button.  
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';  
import * as React from 'react';  
import './App.css';  
// To render Button.  
function App() {  
  return ( <div style={{marginTop: '150px'}}>  
    <ButtonComponent>Button</ButtonComponent>  
  </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 Button component.

APP.JSX

```
import { enableRipple } from '@syncfusion/ej2-base';  
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';  
import * as React from 'react';  
import * as ReactDOM from 'react-dom';  
enableRipple(true);  
// To render Button.  
function App() {  
  return (<ButtonComponent>Button</ButtonComponent>);  
}  
export default App;  
ReactDOM.render(<App />, document.getElementById('button'));
```

APP.TSX

```
import { enableRipple } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To render Button.
function App() {
  return (
    <ButtonComponent>Button</ButtonComponent>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
```

You can refer to our [React Button](#) feature tour page for its groundbreaking feature representations. You can also explore our [React Button example](#) that shows how to render the Button in React.

See Also

- [Types of Button](#)

Types and styles in React Button component

This section explains the different styles and types of Buttons.

Button styles

The Essential JS 2 Button has the following predefined styles that can be defined using [cssClass](#) property.

Class	Description
-----	-----
e-primary	Used to represent a primary action.
e-success	Used to represent a positive action.
e-info	Used to represent an informative action.
e-warning	Used to represent an action with caution.
e-danger	Used to represent a negative action.
e-link	Changes the appearance of the Button like a hyperlink.

APP.JSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { enableRipple } from '@syncfusion/ej2-base';
enableRipple(true);
function App() {
  return (
    <div>
      { /* Primary Button - Used to represent a primary action. */ }
      <ButtonComponent cssClass='e-primary'>Primary</ButtonComponent>
    </div>
  );
}
```

```

    { /* Success Button - Used to represent a positive action. */ }
    <ButtonComponent cssClass='e-success'>Success</ButtonComponent>
    { /* Info Button - Used to represent an informative action. */ }
    <ButtonComponent cssClass='e-info'>Info</ButtonComponent>
    { /* Warning Button - Used to represent an action with caution.*/ }
    <ButtonComponent cssClass='e-warning'>Warning</ButtonComponent>
    { /* Danger Button - Used to represent a negative action.*/ }
    <ButtonComponent cssClass='e-danger'>Danger</ButtonComponent>
    { /* Link Button - Changes the appearance of the Button like a
hyperlink.*/ }
    <ButtonComponent cssClass='e-link'>Link</ButtonComponent>
  </div>;
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

APP.TSX

```

import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { enableRipple } from '@syncfusion/ej2-base';
enableRipple(true);
function App() {
  return (
    <div>
      { /* Primary Button - Used to represent a primary action. */ }
      <ButtonComponent cssClass='e-primary'>Primary</ButtonComponent>
      { /* Success Button - Used to represent a positive action. */ }
      <ButtonComponent cssClass='e-success'>Success</ButtonComponent>
      { /* Info Button - Used to represent an informative action. */ }
      <ButtonComponent cssClass='e-info'>Info</ButtonComponent>
      { /* Warning Button - Used to represent an action with caution.*/ }
      <ButtonComponent cssClass='e-warning'>Warning</ButtonComponent>
      { /* Danger Button - Used to represent a negative action.*/ }
      <ButtonComponent cssClass='e-danger'>Danger</ButtonComponent>
      { /* Link Button - Changes the appearance of the Button like a
hyperlink.*/ }
      <ButtonComponent cssClass='e-link'>Link</ButtonComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

Predefined Button styles provide only the visual indication. So, Button content should define the Button style for the users of assistive technologies such as screen readers.

Primary action button can also be achieved by setting `isPrimary` property as `true`.

Button types

The types of Essential JS 2 Button are as follows:

- Basic types

- Flat Button
- Outline Button
- Round Button
- Toggle Button

Basic types

The basic Button types are explained below.

Type	Description
-----	-----
Button	Defines a click Button.
Submit	This Button submits the form data to the server.
Reset	This Button resets all the controls of the form elements to their initial values.

APP.JSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { enableRipple } from '@syncfusion/ej2-base';
enableRipple(true);
function App() {
    return (<form>
        { /* Submit.*/ }
        <ButtonComponent type='submit'>Submit</ButtonComponent>
        { /* Reset.*/ }
        <ButtonComponent type='reset'>Reset</ButtonComponent>
    </form>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
```

APP.TSX

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { enableRipple } from '@syncfusion/ej2-base';
enableRipple(true);
function App() {
    return (
        <form>
            { /* Submit.*/ }
            <ButtonComponent type='submit'>Submit</ButtonComponent>
            { /* Reset.*/ }
            <ButtonComponent type='reset'>Reset</ButtonComponent>
        </form>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
```

Flat Button

The Flat Button is styled with no background color. To create a flat Button, set the [cssClass](#) property to `e-flat`.

Outline Button

An outline Button has border with transparent background. To create an outline Button, set the [cssClass](#) property as `e-outline`.

Round Button

A round Button is shaped like a circle. Usually, it contains an icon representing its action. To create a round Button, set the [cssClass](#) property to `e-round`.

APP.JSX

```
import { enableRipple } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
function App() {
    return (
        <div>
            /* Flat Button. */
            <ButtonComponent cssClass='e-flat'>Flat</ButtonComponent>
            /* Outline Button. */
            <ButtonComponent cssClass='e-outline'>Outline</ButtonComponent>
            /* Round Button - Icon can be loaded by setting "e-icons e-plus-
            icon" in "iconCss" property.
            Use "e-icons" class name to load Essential JS 2 built-in icons.
            Use "e-plus-icon" class name to load plus icon, that you can refer in
            "styles.css". */
            <ButtonComponent cssClass='e-round' iconCss='e-icons e-plus-icon'
            isPrimary={true}/>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
```

APP.TSX

```
import { enableRipple } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
function App() {
    return (
        <div>
            { /* Flat Button. */ }
            <ButtonComponent cssClass='e-flat'>Flat</ButtonComponent>
            { /* Outline Button. */ }
            <ButtonComponent cssClass='e-outline'>Outline</ButtonComponent>
            { /* Round Button - Icon can be loaded by setting "e-icons e-plus-
            icon" in "iconCss" property.
            Use "e-icons" class name to load Essential JS 2 built-in icons.
            Use "e-plus-icon" class name to load plus icon, that you can refer
            in "styles.css". */ }
        </div>
    );
}
```

```

        <ButtonComponent cssClass='e-round' iconCss='e-icons e-plus-icon'
isPrimary={true}/>
      </div>
    );
  }
  export default App;
  ReactDOM.render(<App />, document.getElementById('button'));

```

Toggle Button

A toggle Button allows you to change between the two states. The Button is active in toggled state and can be recognized through the `e-active` class. The functionality of the toggle Button is handled by click event. To create a toggle Button, set the `isToggle` property to `true`. In the following code snippet, the toggle Button text changes to play/pause based on the state of the Button with the use of click event.

APP.JSX

```

import { enableRipple } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { useState } from "react";
import * as React from "react";
import * as ReactDOM from "react-dom";
enableRipple(true);
function App() {
  const [state, setState] = useState({
    content: 'Play',
    iconCss: 'e-btn-sb-icon e-play-icon'
  });
  // Click Event.
  function btnClick() {
    if (state.content === "Play") {
      setState({ content: 'Pause', iconCss: 'e-btn-sb-icon e-pause-
icon' });
    }
    else {
      setState({ content: 'Play', iconCss: 'e-btn-sb-icon e-play-icon'
});
    }
  }
  // Button is active in toggled state.
  return (<ButtonComponent cssClass='e-flat' iconCss={state.iconCss}
content={state.content} isToggle={true} onClick={btnClick}/>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

APP.TSX

```

import { enableRipple } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { useState } from "react";
import * as React from "react";
import * as ReactDOM from "react-dom";
enableRipple(true);
function App() {
  const [state, setState] = useState({

```



```

        content: 'Play' ,
        iconCss: 'e-btn-sb-icon e-play-icon'
    });
    // Click Event.
    function btnClick(): void {
        if (state.content === "Play") {
            setState({ content: 'Pause', iconCss: 'e-btn-sb-icon e-pause-
icon' });
        }
        else {
            setState({ content: 'Play', iconCss: 'e-btn-sb-icon e-play-icon'
});
        }
    }
    // Button is active in toggled state.
    return (
        <ButtonComponent cssClass='e-flat' iconCss={state.iconCss} content=
{state.content} isToggle={true} onClick={ btnClick } />
    )
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

Change Button type

To change the default Button to flat Button, set the [cssClass](#) property to `e-flat` and text content of the Button is set using [content](#) property.

APP.JSX

```

import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { enableRipple } from '@syncfusion/ej2-base';
enableRipple(true);
//To change the Button type.
function App() {
    return (<ButtonComponent cssClass='e-flat'
content='Button'></ButtonComponent>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

APP.TSX

```

import * as React from 'react';
import * as ReactDOM from 'react-dom';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import { enableRipple } from '@syncfusion/ej2-base';
enableRipple(true);
//To change the Button type.
function App() {
    return (
        <ButtonComponent cssClass='e-flat' content='Button'></ButtonComponent>
    );
}

```

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

Icons

Button with font icons

The Button can have an icon to provide the visual representation of the action. To place the icon on a Button, set the [iconCss](#) property with the required icon CSS. By default, the icon is positioned to the left side of the Button. You can customize the icon's position by

using the [iconPosition](#) property.

APP.JSX

```
import { enableRipple } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
function App() {
    return (<div>
        { /* To position the icon to the left of the text on a Button. */ }
        <ButtonComponent iconCss='e-btn-sb-icon e-prev-
icon'>Previous</ButtonComponent>
        { /* To position the icon to the right of the text on a Button. */ }
        <ButtonComponent iconCss='e-btn-sb-icon e-stop-icon'
iconPosition='Right'>Stop</ButtonComponent>
    </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
```

APP.TSX

```
import { enableRipple } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
function App() {
    return (
        <div>
            { /* To position the icon to the left of the text on a Button. */ }
            <ButtonComponent iconCss='e-btn-sb-icon e-prev-
icon'>Previous</ButtonComponent>
            { /* To position the icon to the right of the text on a Button. */ }
        <ButtonComponent iconCss='e-btn-sb-icon e-stop-icon'
iconPosition='Right'>Stop</ButtonComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
```

Button with SVG image

SVG image can be added to the Button using [iconCss](#) property.

In the following example, SVG image is added using the iconCss class `e-search-icon` by setting `height` and `width`.

APP.JSX

```
import { enableRipple } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
function App() {
    return (<div>
        <ButtonComponent iconCss='e-search-icon' />
    </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
```

APP.TSX

```
import { enableRipple } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
function App() {
    return (
        <div>
            <ButtonComponent iconCss='e-search-icon' />
        </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 Button using the [iconCss](#) property.

Button size

The two types of Button sizes are default and small. To change the size of the default Button to small Button, set the [cssClass](#) property to `e-small`.

APP.JSX

```
import { enableRipple } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
function App() {
    return (<div>
        /* Small Button. */
    </div>);
}
```

```

        <ButtonComponent cssClass='e-small'>Small</ButtonComponent>
        { /* Normal Button. */ }
        <ButtonComponent>Normal</ButtonComponent>
    </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

APP.TSX

```

import { enableRipple } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
function App() {
    return (
        <div>
            { /* Small Button. */ }
            <ButtonComponent cssClass='e-small'>Small</ButtonComponent>
            { /* Normal Button. */ }
            <ButtonComponent>Normal</ButtonComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

See Also

- [Customize Button appearance](#)
- [How to create block button](#)
- [How to create repeat button](#)

Style and appearance in React Button component

To modify the Button appearance, you need to override the default CSS of Button component. Please find the list of CSS classes and its corresponding section in Button component. Also, you have an option to create your own custom theme for the controls using our [Theme Studio](#).

CSS Class | Purpose of Class

- |.e-btn|To customize the button.
- |.e-btn:hover|To customize the button on hover.
- |.e-btn:focus|To customize the button on focus.
- |.e-btn:active|To customize the button on active.

Accessibility in React Button component

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

| Attributes | Purpose |

| --- | --- |

| **aria-label** | Provides an accessible name for the icon only button. |

Keyboard interaction

The Button 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 Button component.

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

| --- | --- |

| **Space** | When the button has focus, pressing the space key changes the state of the button. |

Ensuring accessibility

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

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

See also

- [Accessibility in Syncfusion React components](#)

How To

Add link to a button in React Button component

The appearance of the Button can be changed like a link by **e-link** class using [cssClass](#) property and link navigation can be handled in Button click.

In the following example, link is added in Button click by using **window.open()** method.

APP.JSX

```
import { enableRipple } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
function App() {
  // Click Event.
  function btnClick() {
    window.open("https://www.google.com");
  }
  return (
    <div>
      <ButtonComponent cssClass='e-link' onClick={btnClick}>Go to
      google</ButtonComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
```

APP.TSX

```
import { enableRipple } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
```

```
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
function App() {
  // Click Event.
  function btnClick (): void {
    window.open("https://www.google.com");
  }
  return (
    <div>
      <ButtonComponent cssClass='e-link' onClick={btnClick}>Go to
google</ButtonComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
```

Create a block button in React Button component

You can customize a Button into a Block Button that will span the entire width of its parent element. To create a Block Button, set the [cssClass](#) property to `e-block`.

APP.JSX

```
import { enableRipple } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
function App() {
  return <div>
    { /* Block Button. */ }
    <ButtonComponent cssClass='e-block'>Block Button</ButtonComponent>
    { /* Primary Block Button. */ }
    <ButtonComponent cssClass='e-block' isPrimary={true}>Block
Button</ButtonComponent>
    { /* Success Block Button. */ }
    <ButtonComponent cssClass='e-block e-success'>Block
Button</ButtonComponent>
  </div>;
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
```

APP.TSX

```
import { enableRipple } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
function App() {
  return (
    <div>
      { /* Block Button. */ }
    )
}
```

```

        <ButtonComponent cssClass='e-block'>Block Button</ButtonComponent>
        {/* Primary Block Button. */}
        <ButtonComponent cssClass='e-block' isPrimary={true}>Block
Button</ButtonComponent>
        {/* Success Block Button. */}
        <ButtonComponent cssClass='e-block e-success'>Block
Button</ButtonComponent>
    </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

Customize button appearance in React Button component

You can customize the appearance of the Button by using the Cascading Style Sheets (CSS). Define the CSS according to your requirement, and assign the class name to the [cssClass](#) property. In the following code snippet the background color, text color, height, width, and sharp corner of the Button can be customized through the `e-custom` class for all states (hover, focus, and active).

APP.JSX

```

import { enableRipple } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To customize Button appearance.
// Refer the "e-custom" class details in "styles.css".
function App() {
    return <ButtonComponent cssClass='e-custom'>Custom</ButtonComponent>;
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

APP.TSX

```

import { enableRipple } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
// To customize Button appearance.
// Refer the "e-custom" class details in "styles.css".
function App() {
    return (
        <ButtonComponent cssClass='e-custom'>Custom</ButtonComponent>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

Customize input and anchor elements in React Button component

You can customize the appearance of the input and anchor elements using predefined styles through the class property. In the following code snippet, the input element is customized as a link Button by

setting the `e-btn e-link` class, and the anchor element is customized as a primary Button by setting the `e-btn e-primary` class.

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Syncfusion React Button</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/ej2-
base/styles/material.css" rel="stylesheet" />
  <link href="https://cdn.syncfusion.com/ej2/20.3.56/ej2-
buttons/styles/material.css" rel="stylesheet" />
  <script
src="https://cdn.jsdelivr.cloudflare.com/ajax/libs/systemjs/0.19.38/system.js"></scr
ipt>
  <script src="systemjs.config.js"></script>
</head>
<body>
  <div>
    <input type="button" id="inputbtn" value="Input Button"
class="e-btn e-link">
  </div><br>
  <div>
    <a id="anchorbtn" class="e-btn e-primary" href="#">Google</a>
  </div>
</body>
</html>
```

Repeat button in React Button component

The Repeat button is a type of Button in that the click event is triggered at regular time interval from the pressed state till the released state.

The following example explains about how to achieve Repeat Button in mouse and touch events.

APP.JSX

```
{% raw %}
import { enableRipple, EventHandler } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
enableRipple(true);
function App() {
  let btn;
  let timeout = null;
  function onCreated() {
    const btnElem = btn.element;
    btnElem.addEventListener("mousedown", mouseDownHandler);
    btnElem.addEventListener("mouseup", mouseUpHandler);
    btnElem.addEventListener("touchstart", mouseDownHandler);
    btnElem.addEventListener("touchend", mouseUpHandler);
  }
}
```

```

    }
    function mouseDownHandler() {
        timeout = setInterval(() => processEvents(), 200);
    }
    function onClick() {
        document.getElementById('eventlog').innerHTML = '';
    }
    function mouseUpHandler() {
        clearInterval(timeout);
    }
    function clickEventHandler() {
        EventHandler.trigger(document.getElementById('btn'), "click");
        const span = document.createElement('span');
        span.innerHTML = 'Button click event triggered.<hr>';
        const log = document.getElementById('eventlog');
        log.insertBefore(span, log.firstChild);
    }
    return (<div className='container'>
        <div className='btncontainer'>
            <ButtonComponent id='btn' ref={ (scope) => { btn = scope; }}
created={onCreated} content='Button' onClick={clickEventHandler}/>
        </div>
        <div className='event'>
            <table className='table' title='Event Trace'>
            <tbody>
            <tr>
            <th>Event Trace</th>
            </tr>
            <tr>
            <td>
                <div className="eventarea">
                    <span id="eventlog"/>
                </div>
            </td>
            </tr>
            <tr>
            <td>
                <div className="evtbtn">
                    <ButtonComponent id="clr-btn" content='Clear'
onClick={onClick}/>
                </div>
            </td>
            </tr>
            </tbody>
            </table>
        </div>
        </div>);
    function processEvents() {
        const span = document.createElement('span');
        span.innerHTML = 'Button click event triggered.<hr>';
        const log = document.getElementById('eventlog');
        log.insertBefore(span, log.firstChild);
    }
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}

```

APP.TSX

```

{% raw %}
import { enableRipple, EventHandler } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from "react";
import * as ReactDOM from "react-dom";
enableRipple(true);
function App() {
    let btn: ButtonComponent;
    let timeout: any = null;
    function onCreated(): void {
        const btnElem = btn.element as HTMLElement;
        btnElem.addEventListener("mousedown", mouseDownHandler);
        btnElem.addEventListener("mouseup", mouseUpHandler);
        btnElem.addEventListener("touchstart", mouseDownHandler);
        btnElem.addEventListener("touchend", mouseUpHandler);
    }
    function mouseDownHandler(): void {
        timeout = setInterval(() => processEvents(), 200);
    }
    function onClick(): void {
        (document.getElementById('eventlog') as HTMLElement).innerHTML = '';
    }
    function mouseUpHandler(): void {
        clearInterval(timeout);
    }
    function clickEventHandler(): void {
        EventHandler.trigger((document.getElementById('btn') as HTMLElement),
"click");
        const span = document.createElement('span');
        span.innerHTML = 'Button click event triggered.<hr>';
        const log: HTMLElement = document.getElementById('eventlog') as
HTMLElement;
        log.insertBefore(span, log.firstChild);
    }
    return (
        <div className='container'>
            <div className='btncontainer'>
                <ButtonComponent id='btn' ref={(scope) => { btn = scope as
ButtonComponent; }} created={onCreated} content='Button' onClick =
{clickEventHandler} />
            </div>
            <div className='event'>
                <table className='table' title ='Event Trace'>
                    <tbody>
                        <tr>
                            <th>Event Trace</th>
                        </tr>
                        <tr>
                            <td>
                                <div className="eventarea">
                                    <span id="eventlog" />
                                </div>
                            </td>
                        </tr>
                    </tbody>
                </table>
            </div>
        </div>
    );
}

```

```

        </tr>
        <tr>
            <td>
                <div className="evtbtn">
                    <ButtonComponent id="clr-btn" content='Clear' onClick =
{onClick} />
                </div>
            </td>
        </tr>
    </tbody>
</table>
</div>
</div>
)
function processEvents(): void {
    const span = document.createElement('span');
    span.innerHTML = 'Button click event triggered.<hr>';
    const log = document.getElementById('eventlog') as HTMLElement;
    log.insertBefore(span, log.firstChild);
}
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% enddraw %}

```

Right to left in React Button component

Button 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 Button component.

APP.JSX

```

import { enableRipple } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
function App() {
    return <div>
        <ButtonComponent enableRtl={true} iconCss='e-btn-icons e-setting-
icon'>Settings</ButtonComponent>
    </div>;
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

APP.TSX

```

import { enableRipple } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
function App() {
    return (
        <div>

```

```

        <ButtonComponent enableRtl={true} iconCss='e-btn-icons e-setting-
        icon'>Settings</ButtonComponent>
      </div>
    );
  }
  export default App;
  ReactDOM.render(<App />, document.getElementById('button'));

```

Set the disabled state in React Button component

Button component can be enabled/disabled by giving [disabled](#) property. To disable Button component, the `disabled` property can be set as `true`.

The following example demonstrates Button in `disabled` state.

APP.JSX

```

import { enableRipple } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
function App() {
  return (
    <div>
      <ButtonComponent disabled={true}>Disabled</ButtonComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

APP.TSX

```

import { enableRipple } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
function App() {
  return (
    <div>
      <ButtonComponent disabled={true}>Disabled</ButtonComponent>
    </div>
  );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));

```

Tooltip for button in React Button component

Tooltip can be shown on Button hover and it can be achieved by setting `title` attribute.

The following snippets illustrates how to show tooltip on Button hover.

APP.JSX

```

{% raw %}
import { enableRipple } from '@syncfusion/ej2-base';

```

```

import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
function App() {
    let btn;
    function onCreated() {
        btn.element.setAttribute('title', 'Primary Button');
    }
    return (<div>
        <ButtonComponent id='btn' ref={(scope) => { btn = scope; }}
        created={onCreated} isPrimary={true}>Button</ButtonComponent>
    </div>);
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}

```

APP.TSX

```

{% raw %}
import { enableRipple } from '@syncfusion/ej2-base';
import { ButtonComponent } from '@syncfusion/ej2-react-buttons';
import * as React from 'react';
import * as ReactDOM from 'react-dom';
enableRipple(true);
function App() {
    let btn: ButtonComponent;
    function onCreated(): void {
        (btn.element as HTMLElement).setAttribute('title', 'Primary Button');
    }
    return (
        <div>
            <ButtonComponent id='btn' ref={(scope) => { btn = scope as
ButtonComponent; }} created={onCreated}
isPrimary={true}>Button</ButtonComponent>
        </div>
    );
}
export default App;
ReactDOM.render(<App />, document.getElementById('button'));
{% endraw %}

```

[Ej1 api migration in React Button component](#)

This article describes the API migration process of Button 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 button | **Property:** *text* `

 <EJ.Button id="btn" text="Button"></EJ.Button>` | **Property:** *content* `

 <ButtonComponent id="btn" content="Button"></ButtonComponent>` |

| Specifies the content type of the button | **Property:** *ContentType* `

 <EJ.Button id="btn" text="Save" contentType="textandimage" prefixIcon="e-icon e-save"></EJ.Button>` | Not applicable |

| Specifies icon for the button | **Property:** *prefixIcon* `

 <EJ.Button id="btn" contentType="imageonly" prefixIcon="e-icon e-save"></EJ.Button>` | **Property:** *iconCss* `

 <ButtonComponent id="btn" iconCss="e-icons e-save"></ButtonComponent>` |

| Positioning icon in the button | **Property:** *imagePosition* `

 <EJ.Button id="btn" contentType="textandimage" text="Save" prefixIcon="e-icon e-save" imagePosition="imageright"></EJ.Button>` | **Property:** *iconPosition* `

 <ButtonComponent id="btn" content="Save" iconCss="e-icons e-save" iconPosition="Right"></ButtonComponent>` |

| Specifies secondary icon for the button | **Property:** *suffixIcon* `

 <EJ.Button id="btn" text="FileSearch" contentType="imagetextimage" prefixIcon="e-icon e-search" suffixIcon="e-icon e-file-html"></EJ.Button>` | Not applicable |

| Adding custom class | **Property:** *cssClass* `

 <EJ.Button id="btn" text="Button" cssClass="custom-class"></EJ.Button>` | **Property:** *cssClass* `

 <ButtonComponent id="btn" cssClass="custom-class" content="Button"></ButtonComponent>` |

| Specifies the size of the button | **Property:** *size* `

 <EJ.Button id="btn" text="Button" size="small"></EJ.Button>` | **Property:** *cssClass* `

 <ButtonComponent id="btn" cssClass="e-small" content="Button"></ButtonComponent>` |

| Triggers click event repeatedly while pressing the button | **Property:** *repeatButton* `

 <EJ.Button id="btn" text="Click" repeatButton={true}></EJ.Button>` | Not applicable |

| Sets time interval between two consecutive click event on the repeat button | **Property:** *timeInterval* `

 <EJ.Button id="btn" text="Click" repeatButton={true} timeInterval="100"></EJ.Button>` | Not applicable |

| Specifies the type of the button | **Property:** *type* `

 <EJ.Button id="btn" text="Submit" buttonType="submit"></EJ.Button>` | Not applicable |

| Changes normal button into rounded corner | **Property:** *showRoundedCorner* `

 <EJ.Button id="btn" text="Button" showRoundedCorner={true}></EJ.Button>` | Not applicable |

| Specifies the width of the button | **Property:** *width* `

 <EJ.Button id="btn" text="Button" width="150px"></EJ.Button>` | Not applicable |

| Specifies the height of the button | **Property:** *height* `

 <EJ.Button id="btn" text="Button" height="50px"></EJ.Button>` | Not applicable |

| Adds HTML attributes to the button | **Property:** *htmlAttributes* `

 var attributes = { disabled: "disabled" };
 <EJ.Button id="btn" text="Button" htmlAttributes={attributes}></EJ.Button>` | Not applicable |

| Allows the appearance of the Button to be enhanced and visually appealing | Not applicable |

Property: *isPrimary*

 <ButtonComponent id="btn" isPrimary={true} content="Button"></ButtonComponent> |

| Makes the button toggle from normal to active state | **Property:** *isToggle*

 Not applicable |

Property: *isToggle*

 <ButtonComponent id="btn" isToggle={true} content="Play"></ButtonComponent> |

| Specifies the disabled state of the button | **Property:** *enabled*

 <EJ.Button id="btn" text="Button" enabled={false}></EJ.Button> | **Property:** *disabled*

 <ButtonComponent id="btn" disabled={true} content="Button"></ButtonComponent> |

| Enable or disable rendering component in right to left direction | **Property:** *enableRTL*

 <EJ.Button id="btn" contentType="textandimage" text="Save" prefixIcon="e-icon e-save" enableRTL={true}></EJ.Button> | **Property:** *enableRtl*

 <ButtonComponent id="btn" enableRtl={true} content="Save" iconCss="e-icons e-save"></ButtonComponent> |

Methods

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

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

| Destroys the control | **Methods:** *destroy*

 <EJ.Button id="btn" text="Button"></EJ.Button>
 var btnObj = \$('#btn').ejButton('instance');
btnObj.destroy(); | **Methods:** *destroy*

 <ButtonComponent id="btn" content="Button" ref={{scope}} => {this.btnObj = scope}></ButtonComponent>
 constructor(props: {}) {
 this.btnObj.destroy();
 } |

| Disables the button | **Methods:** *disable*

 <EJ.Button id="btn" text="Button"></EJ.Button>
 var btnObj = \$('#btn').ejButton('instance');
btnObj.disable(); | Not applicable |

| Enables the button | **Methods:** *enable*

 <EJ.Button id="btn" text="Button"></EJ.Button>
 var btnObj = \$('#btn').ejButton('instance');
btnObj.enable(); | Not applicable |

Events

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

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

| Triggers while clicking the button | **Events:** *click*

 <EJ.Button id="btn" text="Button" click={btnClick}></EJ.Button>
function btnClick(args) {
/ code block */
} | Not applicable |

| Triggers once the component rendering is completed | **Events:** *create*

 <EJ.Button id="btn" text="Button" click={create}></EJ.Button>
function create(args) {
/ code block /
} | **Events:** *created*

 <ButtonComponent id="btn" content="Button" created={this.created.bind(this)}></ButtonComponent>
 created() {
/ code block /
} |

| Triggers once the component is destroyed | **Events:** *destroy*

 <EJ.Button id="btn" text="Button" destroy={destroy}></EJ.Button>
function destroy(args) {
/ code block */
} | Not applicable |

{% endraw %}