Unveiling new capabilities of the Oracle RESTFUL Data Services

Marcel Ochsendorf

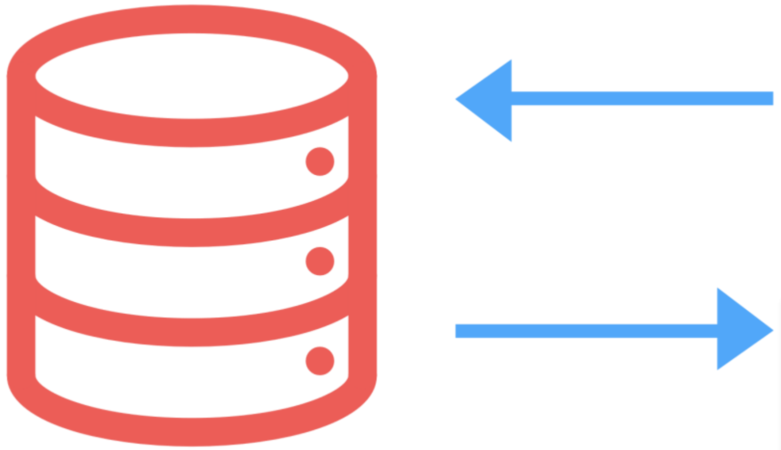
SUPERVISORS

Antonio Nappi

Artur Wiecek

CONTEXT | DATABASE

Database

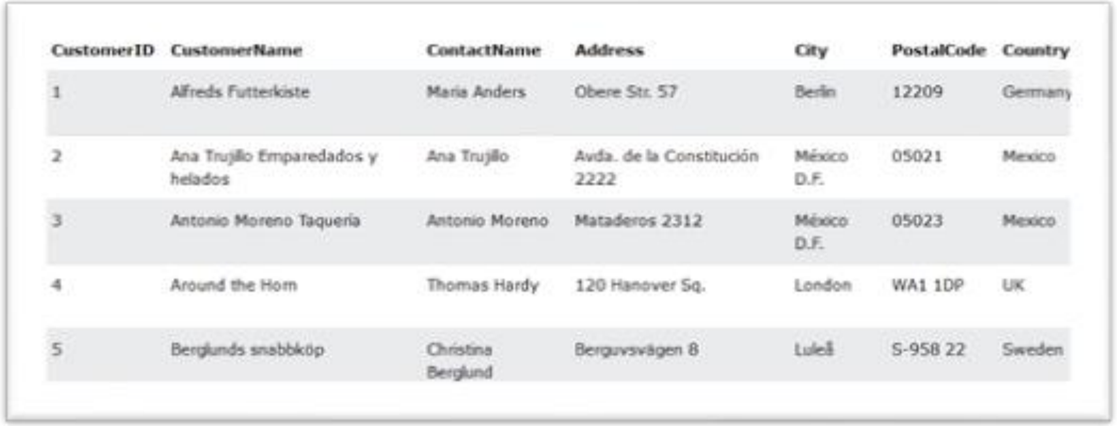


Oracle Database Instance

PL|SQL - QUERY

```sql
SELECT * FROM Customers;
```

RESULT



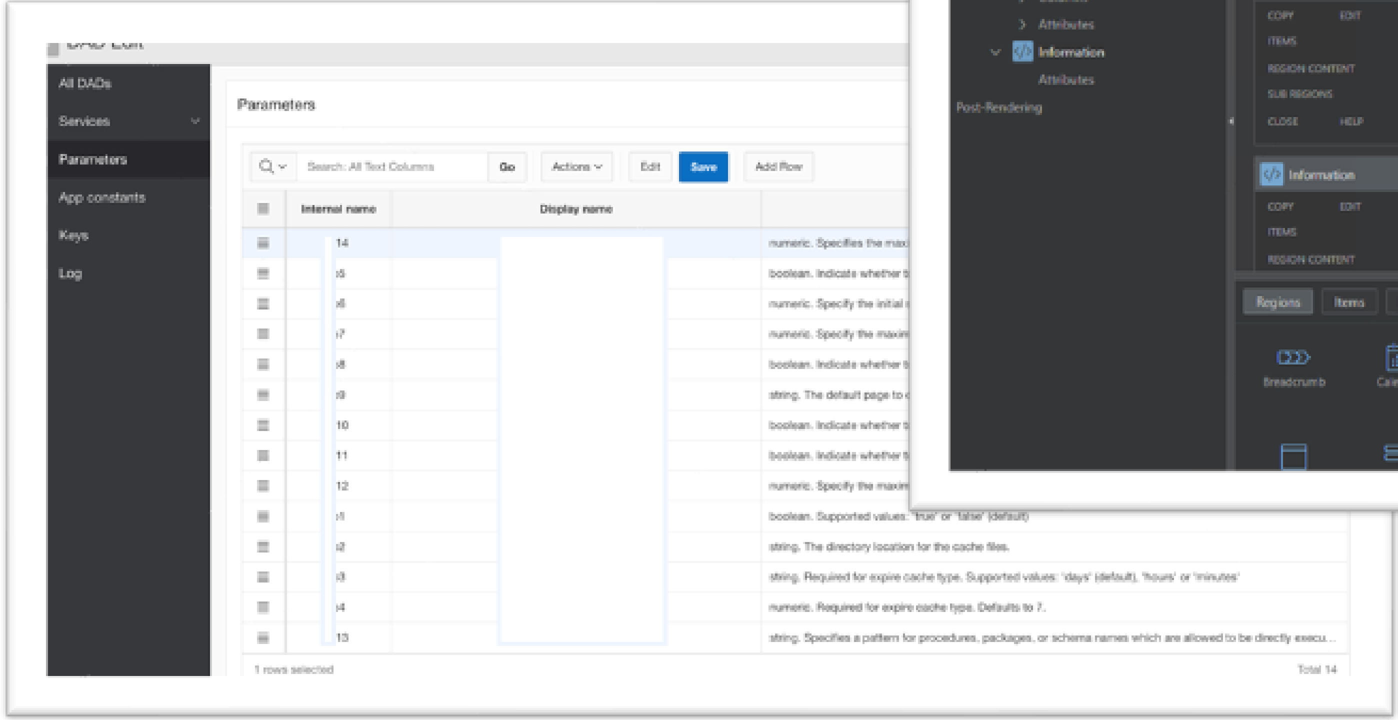
CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

CERN openlab

CONTEXT | APEX

FEATURES

- Low Code Plattform
- Drag&Drop building block system
- Templates for easy table modification

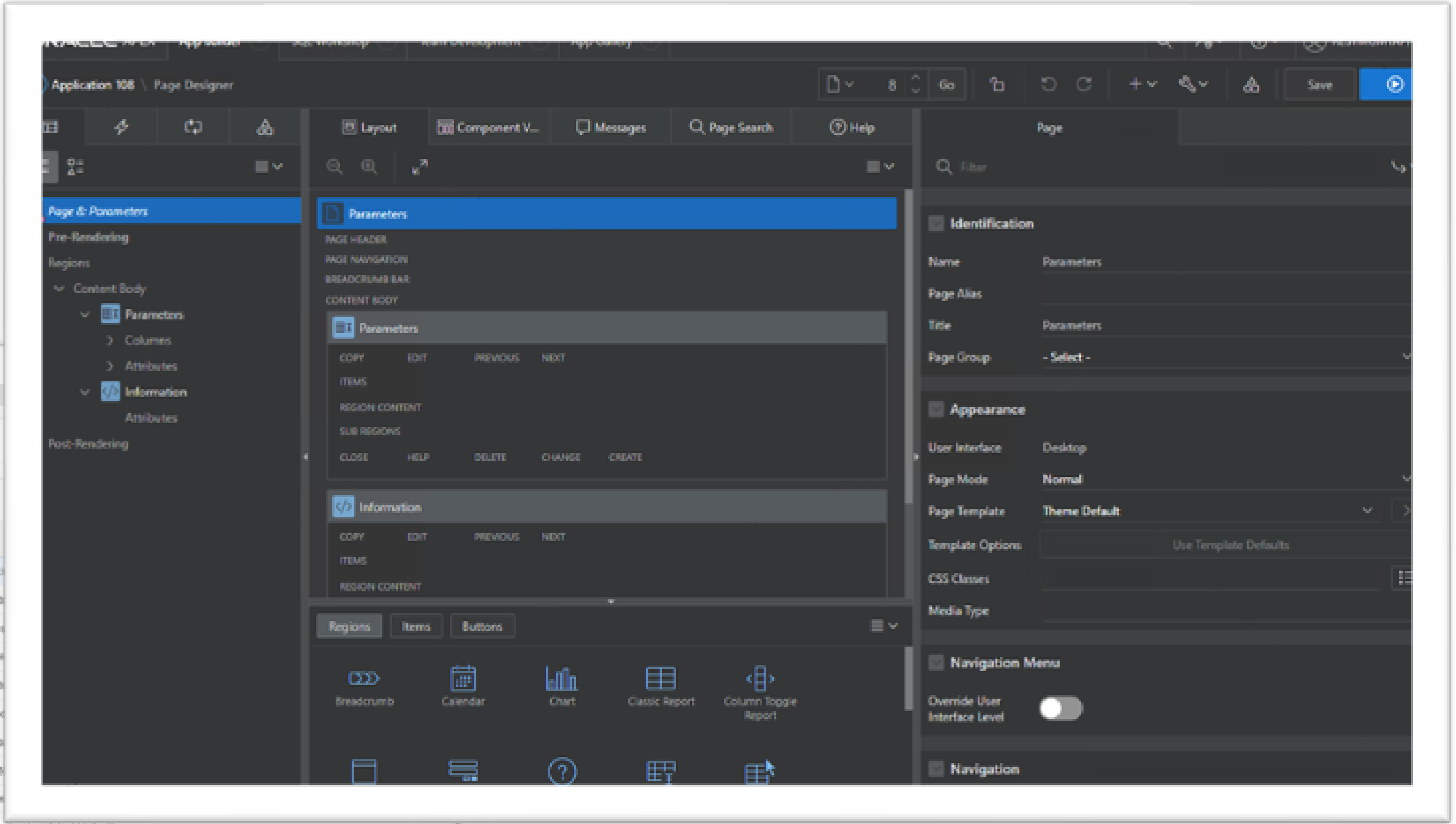


All DADs
Services
Parameters
App constants
Keys
Log

Parameters

	Internal name	Display name	
	14		numeric. Specifies the max
	5		boolean. Indicate whether t
	6		numeric. Specify the initial
	7		numeric. Specify the maxim
	8		boolean. Indicate whether t
	9		string. The default page to
	10		boolean. Indicate whether t
	11		boolean. Indicate whether t
	12		numeric. Specify the maxim
	1		boolean. Supported values: 'true' or 'false' (default)
	2		string. The directory location for the cache files.
	3		string. Required for expire cache type. Supported values: 'days' (default), 'hours' or 'minutes'
	4		numeric. Required for expire cache type. Defaults to 7.
	13		string. Specifies a pattern for procedures, packages, or schema names which are allowed to be directly execu...

1 rows selected Total 14



Application 108 \ Page Designer

Page & Parameters
Pre-Rendering
Regions
Content Body
Parameters
Columns
Attributes
Information
Attributes
Post-Rendering

Identification
Name: Parameters
Page Alias:
Title: Parameters
Page Group: - Select -

Appearance
User Interface: Desktop
Page Mode: Normal
Page Template: Theme Default
Template Options: Use Template Defaults
CSS Classes:
Media Type:

Navigation Menu
Override User Interface Level

Navigation

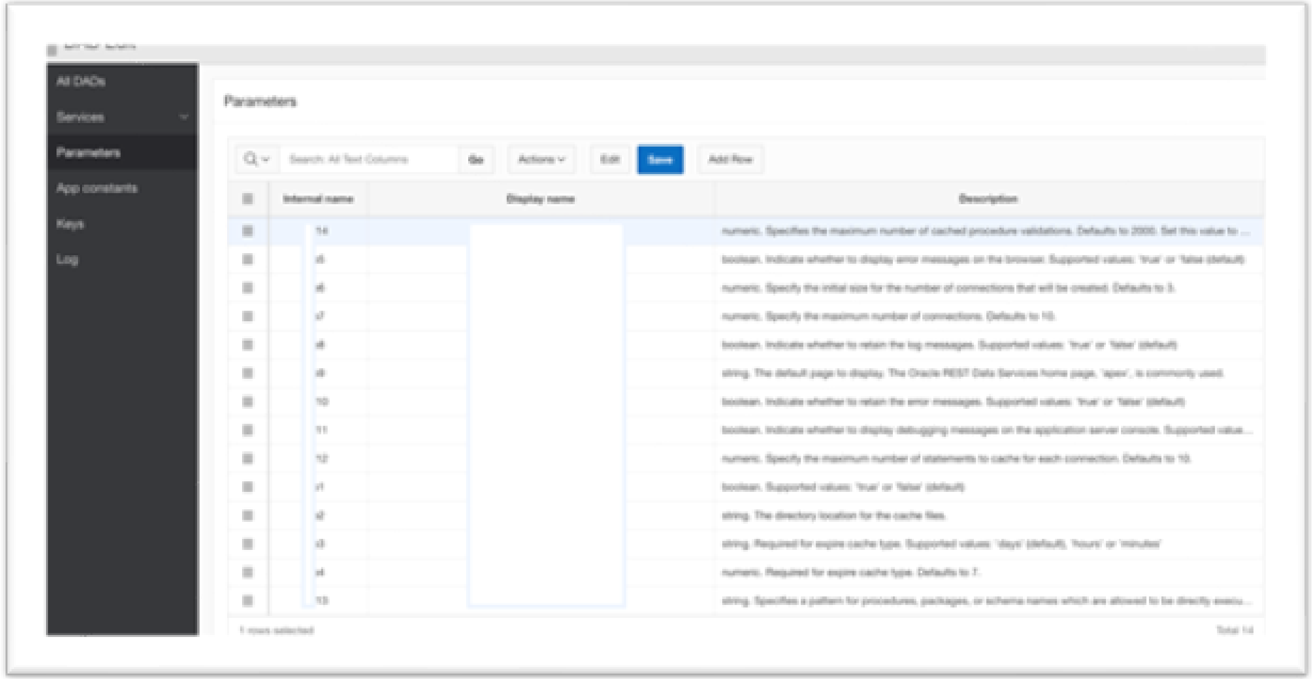
CERN openlab

THE PROBLEM

How to automate an APEX application

CERN IT-DAR-Department uses an APEX application to manage entries in several database schemas.

- Easy to use
- No PL/SQL writing



Internal name	Display name	Description
14		numeric. Specifies the maximum number of cached procedure validations. Defaults to 2000. Set this value to ...
i5		boolean. Indicate whether to display error messages on the browser. Supported values: 'true' or 'false' (default)
i6		numeric. Specify the initial size for the number of connections that will be created. Defaults to 3.
i7		numeric. Specify the maximum number of connections. Defaults to 10.
i8		boolean. Indicate whether to retain the log messages. Supported values: 'true' or 'false' (default)
i9		string. The default page to display. The Oracle REST Data Services home page, 'apex', is commonly used.
10		boolean. Indicate whether to retain the error messages. Supported values: 'true' or 'false' (default)
11		boolean. Indicate whether to display debugging messages on the application server console. Supported value...
12		numeric. Specify the maximum number of statements to cache for each connection. Defaults to 10.
i1		boolean. Supported values: 'true' or 'false' (default)
i2		string. The directory location for the cache files.
i3		string. Required for expire cache type. Supported values: 'days' (default), 'hours' or 'minutes'
i4		numeric. Required for expire cache type. Defaults to 7.
13		string. Specifies a pattern for procedures, packages, or schema names which are allowed to be directly execu...

1 rows selected

Total 14

Scripts and other apps uses direct database access [JBDC] to access tables.

- Complex to integrate

THE PROBLEM

How to automate an APEX application

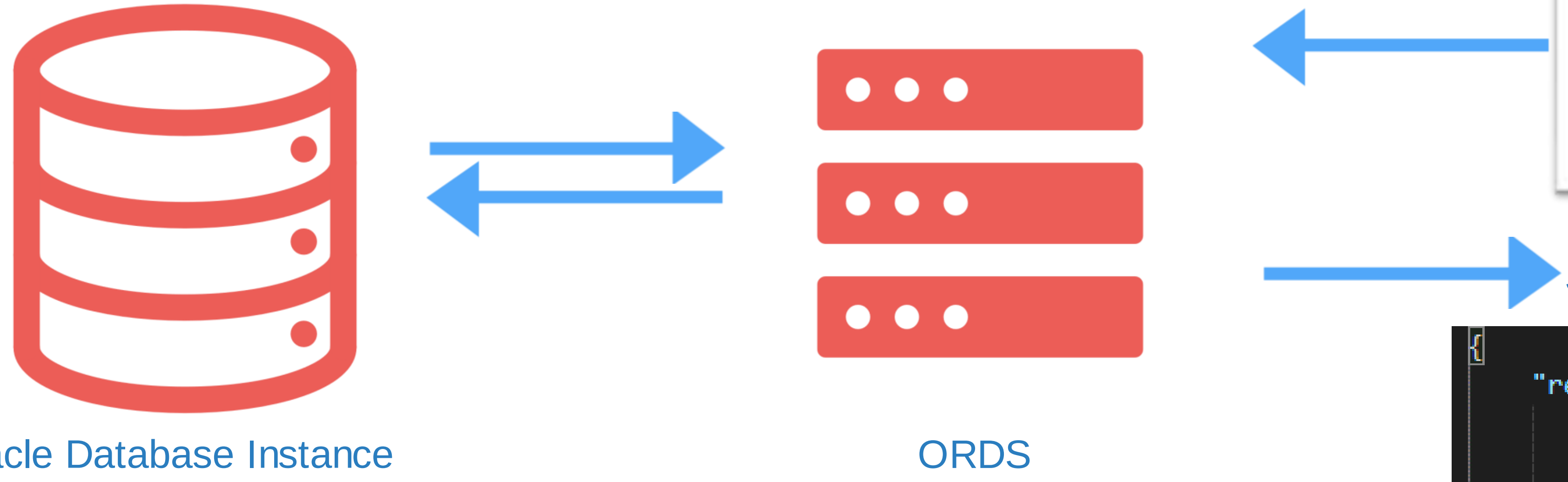
Every third-party application or user needs the following things, to get data in and out of the database schema:

- User-account
- Direct database server access
- Table structure knowledge
- PL/SQL knowledge

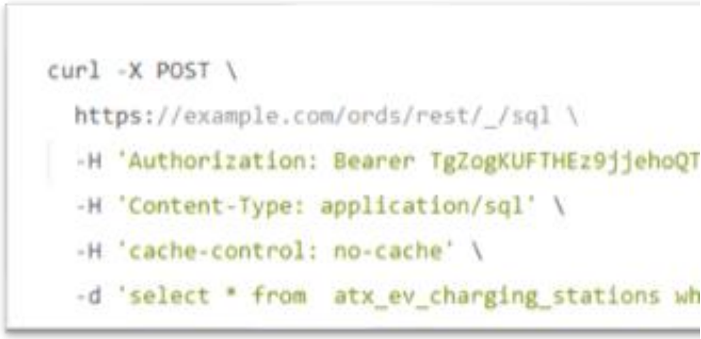
CONTEXT | ORDS

Oracle **REST** Data **Service**

- Provides an RESTFUL API „Proxy“ between Database and User
- **Avoids the direct database access**
- **REST ENABLED SQL** translates given SQL queries inside of a HTTP Request and executes it on the database

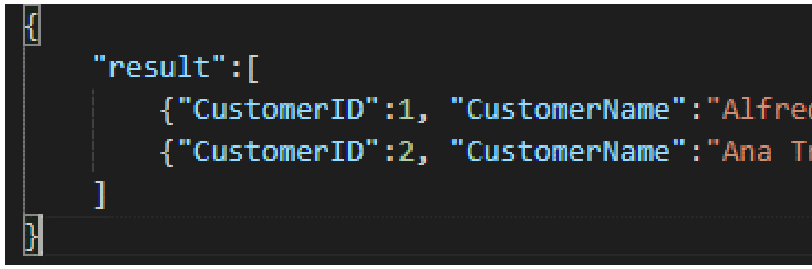


REST REQUEST [QUERY]



```
curl -X POST \
  https://example.com/ords/rest/_/sql \
  -H 'Authorization: Bearer TgZogKUFTHEz9jjehoQT
  -H 'Content-Type: application/sql' \
  -H 'cache-control: no-cache' \
  -d 'select * from  atx_ev_charging_stations wh
```

JSON RESPONSE [RESULT]



```
{
    "result":[
        {"CustomerID":1, "CustomerName":"Alfre
        {"CustomerID":2, "CustomerName":"Ana T
    ]
}
```

THE SOLUTION

Use ORDS

Implement a custom ORDS module, which maps the same functionality as the APEX application and offers this to a RESTFUL API endpoint

- GET and MODIFY table entries
- Implement several filter and sorting options
- Add SSO, logging, permissions

THE SOLUTION

basic implementation and structure of the solution

API DEFINITION

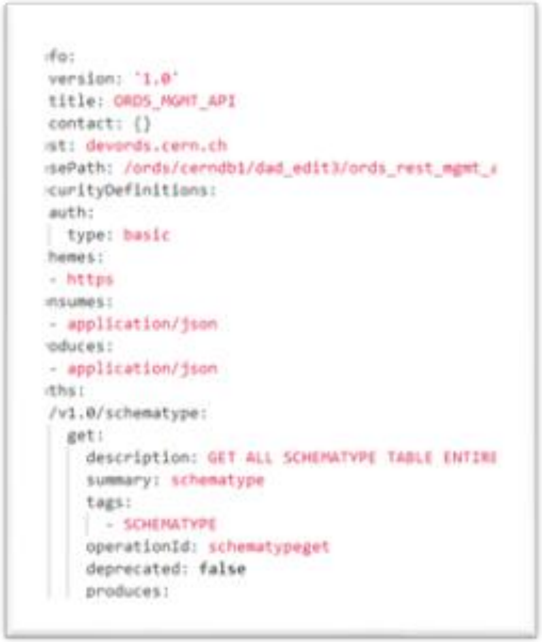
- Documentation for the User
- Basic structure of an API call
- Swagger for interactive testing

ORDS HANDLER

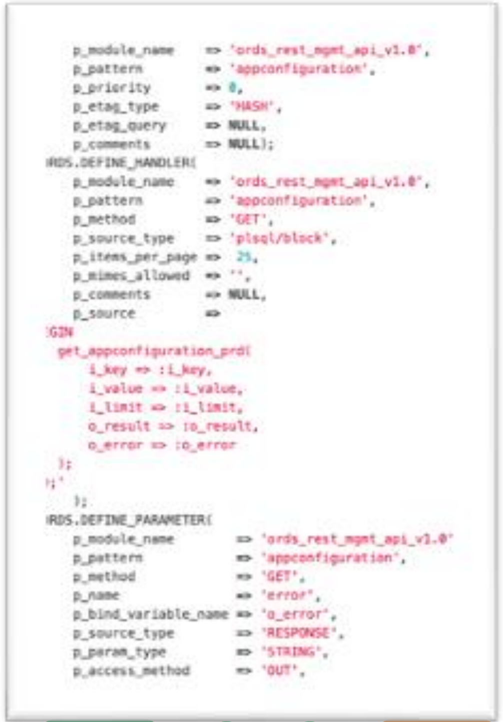
- Handles the incoming API call
- Executes the specified PL/SQL query
- return sql response as JSON

PL/SQL FUNCTIONS

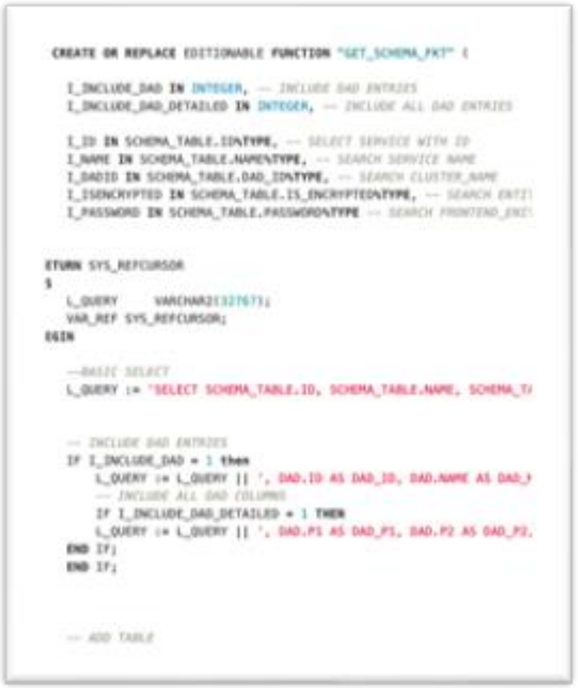
- Dynamically build the SQL query statement
- Returns resultset
- user-friendly error messages



```
fo:
 version: '1.0'
 title: ORDS_MGMT_API
 contact: {}
st: devords.cern.ch
sePath: /ords/cerndb1/dad_edit3/ords_rest_mgmt_
curityDefinitions:
 auth:
   type: basic
hemes:
 - https
nsumes:
 - application/json
oduces:
 - application/json
ths:
 /v1.0/schematype:
   get:
     description: GET ALL SCHEMATYPE TABLE ENTIRE
     summary: schematype
     tags:
       - SCHEMATYPE
     operationId: schematypeget
     deprecated: false
     produces:
```



```
    p_module_name     => 'ords_rest_mgmt_api_v1.0',
    p_pattern         => 'appconfiguration',
    p_priority        => 0,
    p_etag_type       => 'HASH',
    p_etag_query      => NULL,
    p_comments        => NULL);
RDS.DEFINE_HANDLER(
    p_module_name     => 'ords_rest_mgmt_api_v1.0',
    p_pattern         => 'appconfiguration',
    p_method          => 'GET',
    p_source_type     => 'plsql/block',
    p_items_per_page  =>  25,
    p_mimes_allowed   => '',
    p_comments        => NULL,
    p_source          =>
GIN
  get_appconfiguration_prd(
      i_key => :i_key,
      i_value => :i_value,
      i_limit => :i_limit,
      o_result => :o_result,
      o_error => :o_error
  );
);'
    );
RDS.DEFINE_PARAMETER(
    p_module_name          => 'ords_rest_mgmt_api_v1.0'
    p_pattern              => 'appconfiguration',
    p_method               => 'GET',
    p_name                 => 'error',
    p_bind_variable_name   => 'o_error',
    p_source_type          => 'RESPONSE',
    p_param_type           => 'STRING',
    p_access_method        => 'OUT',
```



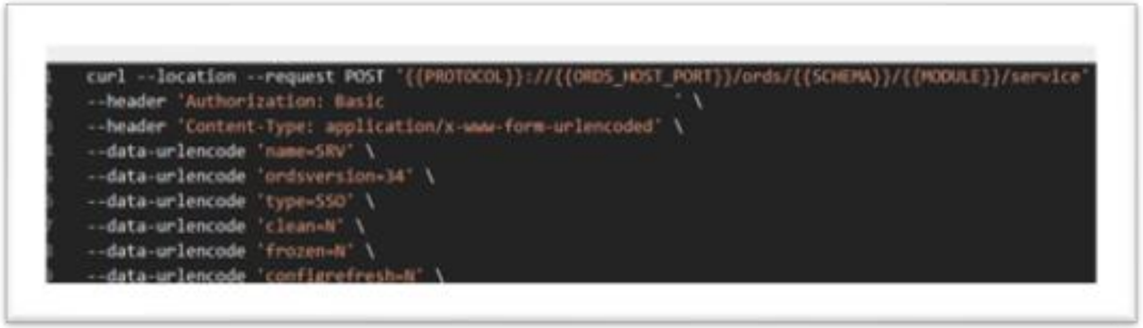
```
CREATE OR REPLACE EDITIONABLE FUNCTION "GET_SCHEMA_FKT" (

   I_INCLUDE_DAD IN INTEGER, -- INCLUDE DAD ENTRIES
   I_INCLUDE_DAD_DETAILED IN INTEGER, -- INCLUDE ALL DAD ENTRIES

   I_ID IN SCHEMA_TABLE.ID%TYPE, -- SELECT SERVICE WITH ID
   I_NAME IN SCHEMA_TABLE.NAME%TYPE, -- SEARCH SERVICE NAME
   I_DADID IN SCHEMA_TABLE.DAD_ID%TYPE, -- SEARCH CLUSTER_NAME
   I_ISENCRYPTED IN SCHEMA_TABLE.IS_ENCRYPTED%TYPE, -- SEARCH ENTI
   I_PASSWORD IN SCHEMA_TABLE.PASSWORD%TYPE -- SEARCH FRONTEND_ENT

ETURN SYS_REFCURSOR
S
   L_QUERY     VARCHAR2(32767);
   VAR_REF SYS_REFCURSOR;
EGIN

   --BASIC SELECT
   L_QUERY := 'SELECT SCHEMA_TABLE.ID, SCHEMA_TABLE.NAME, SCHEMA_T


   -- INCLUDE DAD ENTRIES
   IF I_INCLUDE_DAD = 1 then
       L_QUERY := L_QUERY || ', DAD.ID AS DAD_ID, DAD.NAME AS DAD_
       -- INCLUDE ALL DAD COLUMNS
       IF I_INCLUDE_DAD_DETAILED = 1 THEN
       L_QUERY := L_QUERY || ', DAD.P1 AS DAD_P1, DAD.P2 AS DAD_P2,
   END IF;
   END IF;



   -- ADD TABLE
```

THE SOLUTION

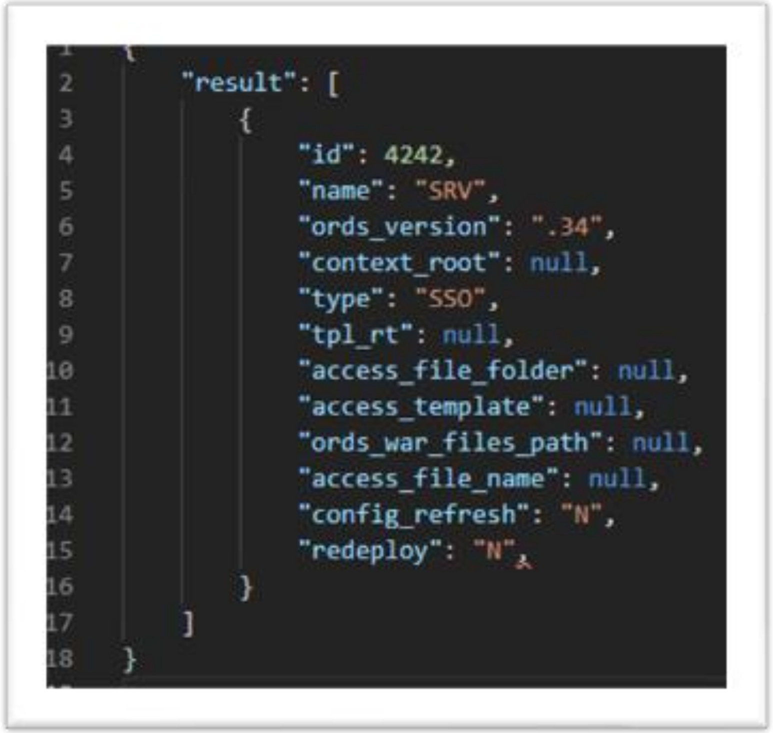
basic implementation and structure of the solution

HTTP REQUEST

RESPONSE



```
curl --location --request POST '{{PROTOCOL}}://{{ORDS_HOST_PORT}}/ords/{{SCHEMA}}/{{MODULE}}/service'
--header 'Authorization: Basic                                   ' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'name=SRV' \
--data-urlencode 'ordsversion=34' \
--data-urlencode 'type=SSO' \
--data-urlencode 'clean=N' \
--data-urlencode 'frozen=N' \
--data-urlencode 'configrefresh=N' \
```



```
{
    "result": [
        {
            "id": 4242,
            "name": "SRV",
            "ords_version": ".34",
            "context_root": null,
            "type": "SSO",
            "tpl_rt": null,
            "access_file_folder": null,
            "access_template": null,
            "ords_war_files_path": null,
            "access_file_name": null,
            "config_refresh": "N",
            "redeploy": "N",
        }
    ]
}
```

SUMMING UP

- Implement a fully working and tested RESTFUL API based on the APEX application using a custom ORDS module

- Two python-based applications have already been successfully ported to use the newly created API

- After testing the API on the development database was also deployed on the production database.

QUESTIONS?