



**UNLEASHING THE POTENTIAL  
OF ORACLE APEX:**

# **MASTERING REST API FOR MODERN APPS**

**5th May 2023, 5 PM, IST**

**Rutvik Prajapati**  
Programmer Analyst  
Loblaw Companies Ltd.



# About Me

---

- 9+ years of experience as Oracle Database Developer
- Started my career as Oracle FORMS and Reports Developer
- Working on Oracle APEX since 2014
- Programmer Analyst, Loblaw Companies Limited, Winnipeg, Canada



Rutvik Prajapati

 @rutveek

 [bit.ly/rutvik\\_linkedin](https://bit.ly/rutvik_linkedin)

 [bit.ly/rutvik\\_blog](https://bit.ly/rutvik_blog)



# About me



# About me





# Canada Oracle APEX Special Interest Group

 @canapexsig

 <https://www.linkedin.com/company/canapexsig>

 canapexsig@gmail.com





**MOST THINGS WORTH HAVING  
LIVE OUTSIDE OUR COMFORT ZONE  
ON THE OTHER SIDE OF FEAR**





## Mentor and Speaker Hub

Our goal is to *connect* speakers with mentors  
to assist in *preparing* technical sessions and  
*improving* presentation skills

Interested? Read more and get in touch

<https://mashprogram.wordpress.com>

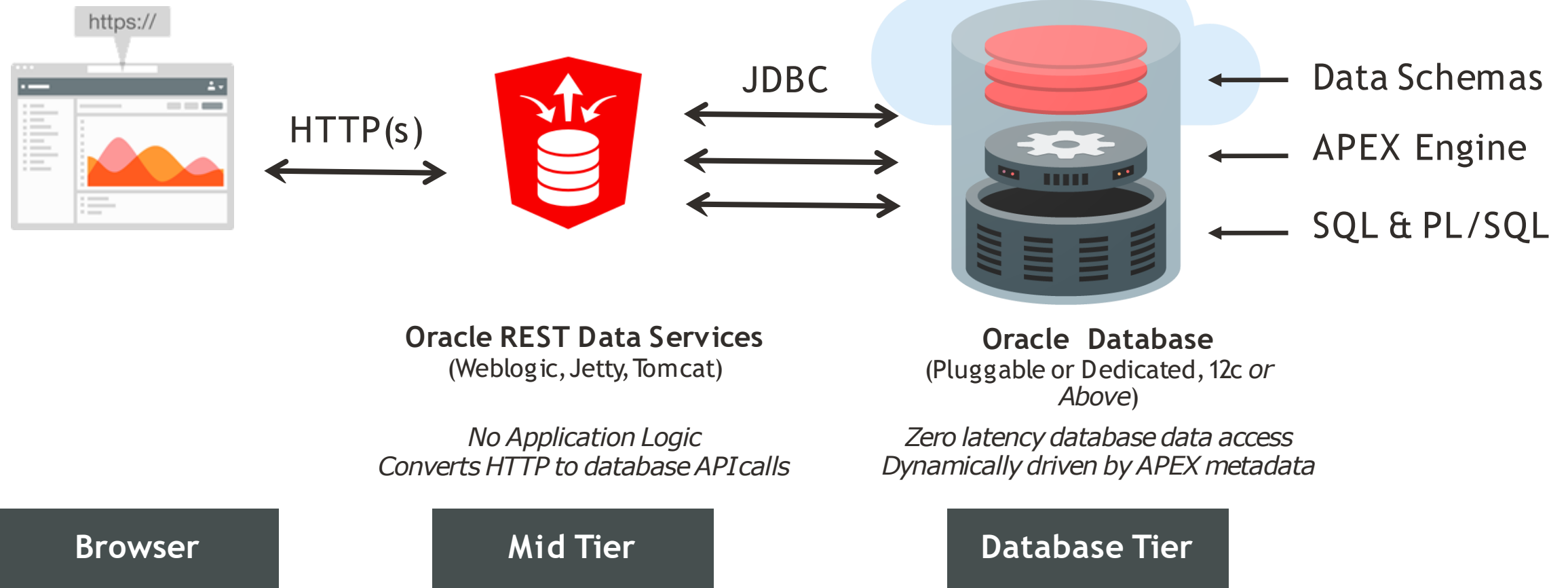
# Agenda

---

- Oracle APEX Architecture
- What is ORDS?
- What is REST API?
- Auto Rest Feature of ORDS
- Demo on Auto REST on a Table and Package Procedure
- Manually Create REST API on a Table
- Demo on Manually Creating REST API
- How to Secure REST API?
- Demo: Secure REST API



# Oracle APEX Architecture



# What is ORDS?

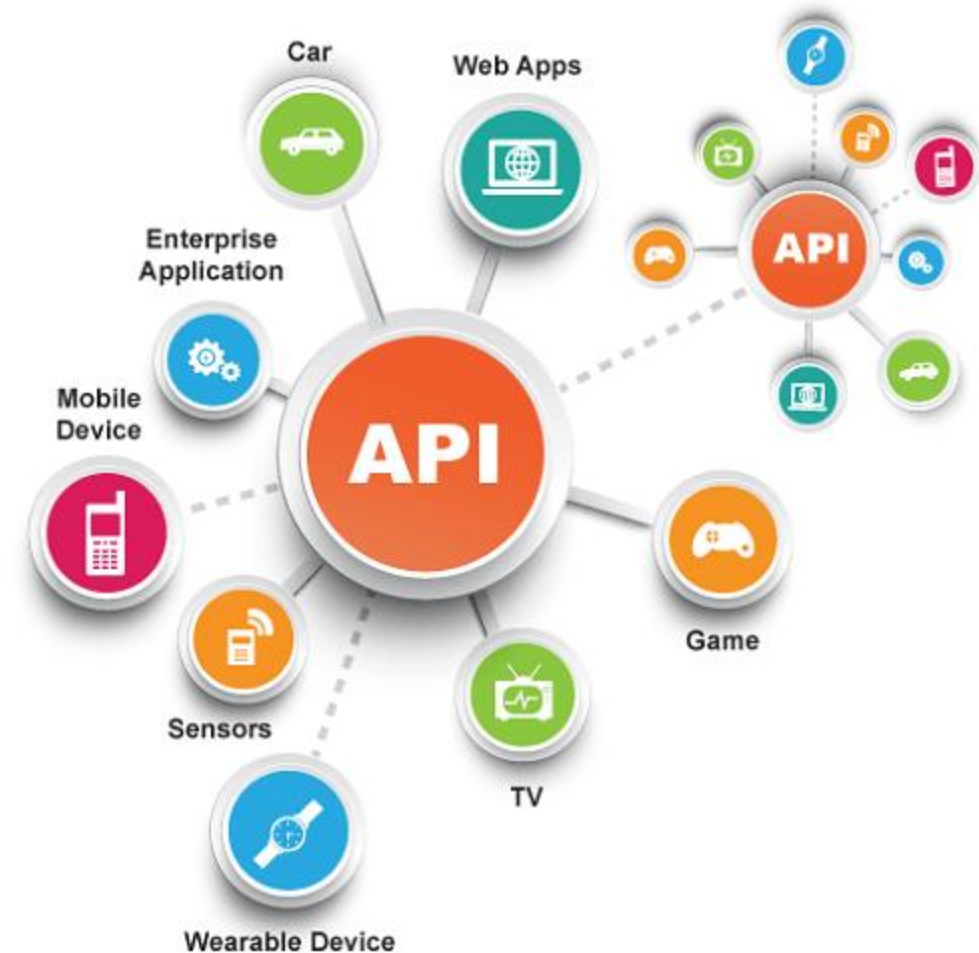


Oracle REST Data Services

- A mid-tier Java application
- Java EE-based alternative for Oracle HTTP serve which runs in Java application server like WebLogic or Tomcat.
- Maps standard http(s) RESTful requests to database transactions.
- Allows you to expose and capture data in your Oracle Database using the REST Protocol
- Access to Relational data over HTTP(s) without installing JDBC/ODBC drivers
- Automatically returns SQL queries in JSON.
- Automatically pages response for GET method services.
- Supports OAuth2.0 authentication

# What is API?

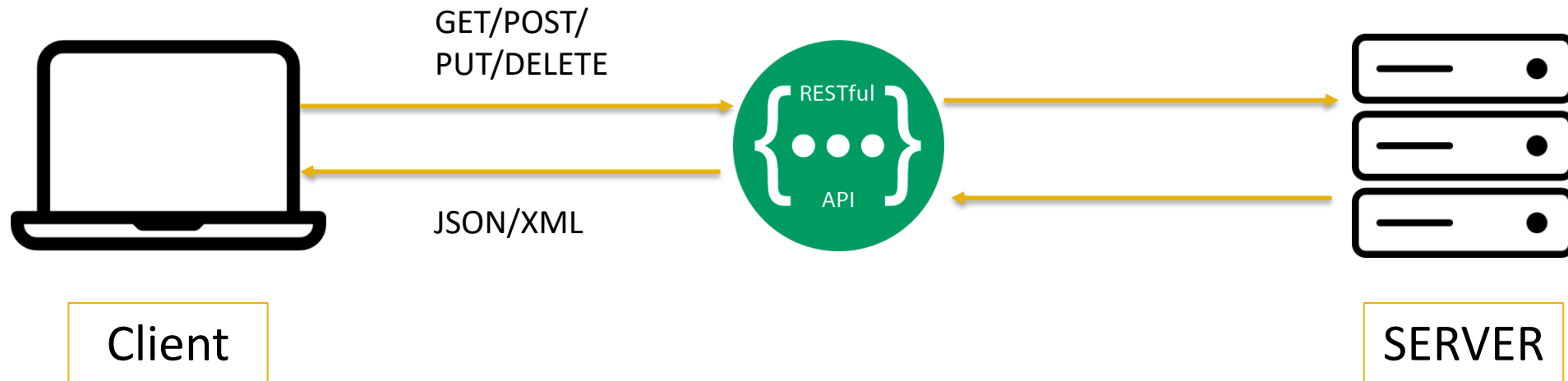
- API stands for Application Programming Interface
- Application can interact with each other using API.
- An API is an interface that computer programs use to interact with an application.
- APIs are directly used by programs, not humans.
- API let computers talk to each other.



# What is REST API?

## Representational State Transfer

- An architecture that provides interoperability between two systems.
- Use HTTP(s) protocol for communication



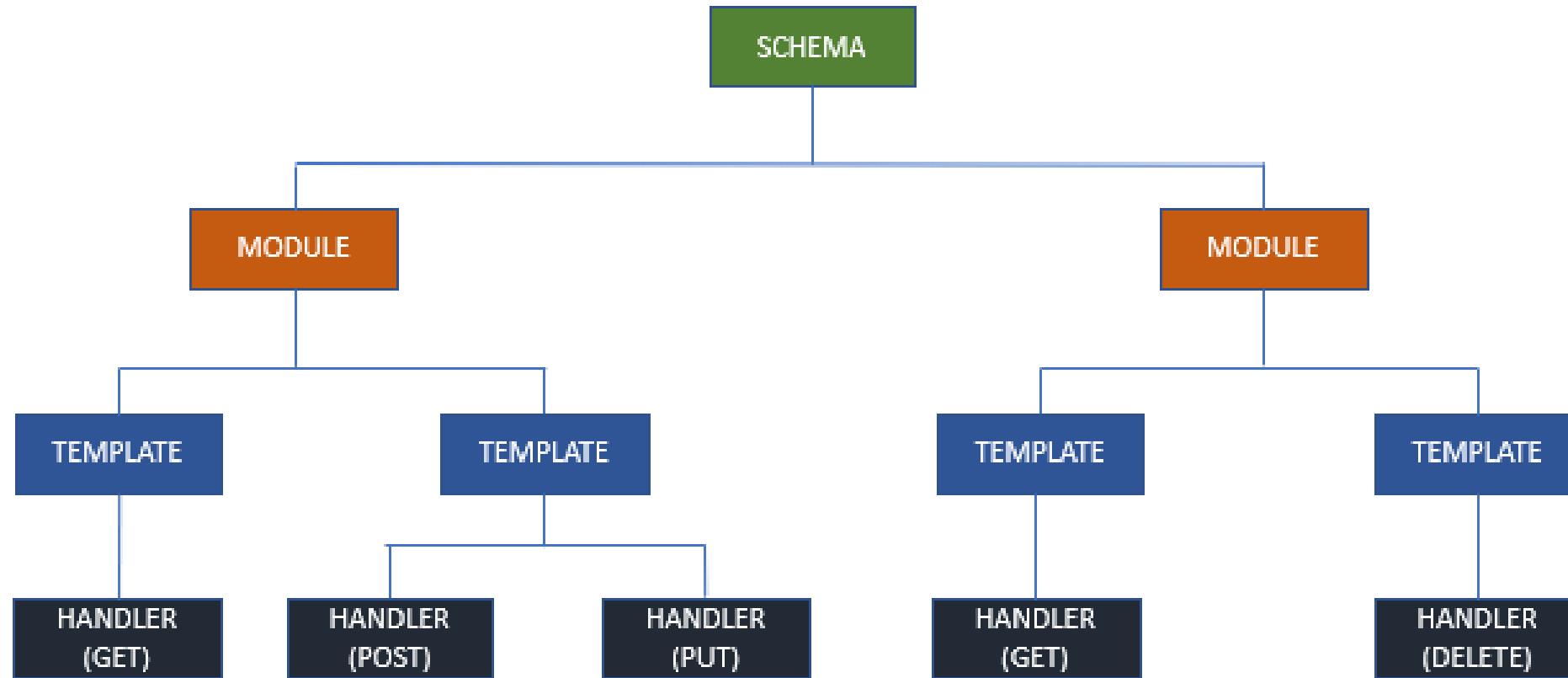


# HTTPS Methods

- REST communicates over HTTP, therefore uses standard HTTP methods

Operation	HTTP Method	SQL Equivalent
CREATE	POST	INSERT
READ	GET	SELECT
UPDATE	PUT	UPDATE
DELETE	DELETE	DELETE

# Structure of a ORDS Webservice



# Understanding ORDS URI Terminology

https://server.com/ords/myschema/hr/employees/:id

server.com	Server URL	
ords	ORDS Alias	Alias for the ORDS engine (Defined at the web server level)
myschema	Schema Alias	Defines the path used to reach the ORDS Enables Schema
hr	Module Base Path	Defines the BASE_PATH for a group of related services
employees	URI Template	Defines the path that is used to access a specific resource
:id	Bind Variable	Defines the bind variable used in the handler

# Creating REST API in ORDS

---

ORDS provides two different ways to expose data/functionality via REST:

- **AutoREST Enablement**
- **Manual REST Service Creation**



# AutoREST Enablement

---

- Concept of making database resource available via ORDS with Zero Code
- Can expose Tables, Views, Packages, Procedures, and Functions
  - Tables (GET, POST, PUT, DELETE)
  - Views (GET)
  - Procedures, Functions, Packages (POST)
- You sacrifice flexibility and customizability for ease of creation
- No customization of included columns or data format
- No way to introduce extra validation or logic

# How to enable schema for ORDS?

The screenshot shows the APEX SQL Workshop interface. The top navigation bar includes the APEX logo, App Builder, SQL Workshop, Team Development, and Gallery. A search bar and user profile (admin ws1) are also present. The left sidebar shows the RESTful Services section expanded, with sub-items: RESTful Data Services, Enabled Objects, Modules, Privileges, and Roles. The main content area displays a warning message: "Schema not registered with ORDS". The message states: "This schema has not been registered with ORDS RESTful Data Services. To register this schema click **Register Schema with ORDS**." A green button labeled "Register Schema with ORDS" is visible on the right side of the warning message. The schema name "WKSP\_WS1" is shown in the top right corner of the main content area.

# How to enable schema for ORDS?

The screenshot shows the APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The user is logged in as 'admin ws1'. The 'RESTful Services' section is active, showing a warning: 'Schema not registered with ORDS'. This warning states: 'This schema has not been registered with ORDS RESTful Data Services. To register this schema click Register Schema with ORDS.' A 'Register Schema with ORDS' button is visible. A dialog box titled 'ORDS Schema Attributes' is open, allowing configuration of the schema. The dialog includes the following options:

- Enable RESTful Access:** A toggle switch that is currently turned on (green).
- \* Schema Alias:** A text input field containing 'apidemo'. A note below this field states: 'You are editing the alias for the workspace's default schema. Setting this alias will also change the Path Prefix at the workspace level.'
- Install Sample Service:** A toggle switch that is currently turned off (grey).
- Authorization Required for Metadata Access:** A toggle switch that is currently turned off (grey).

At the bottom of the dialog, there are 'Cancel' and 'Save Schema Attributes' buttons.

# How to enable schema for ORDS?

- Using ENABLE\_SCHEMA procedure of ORDS API, we can enable or disable the schema.
- We only need to perform this action once.

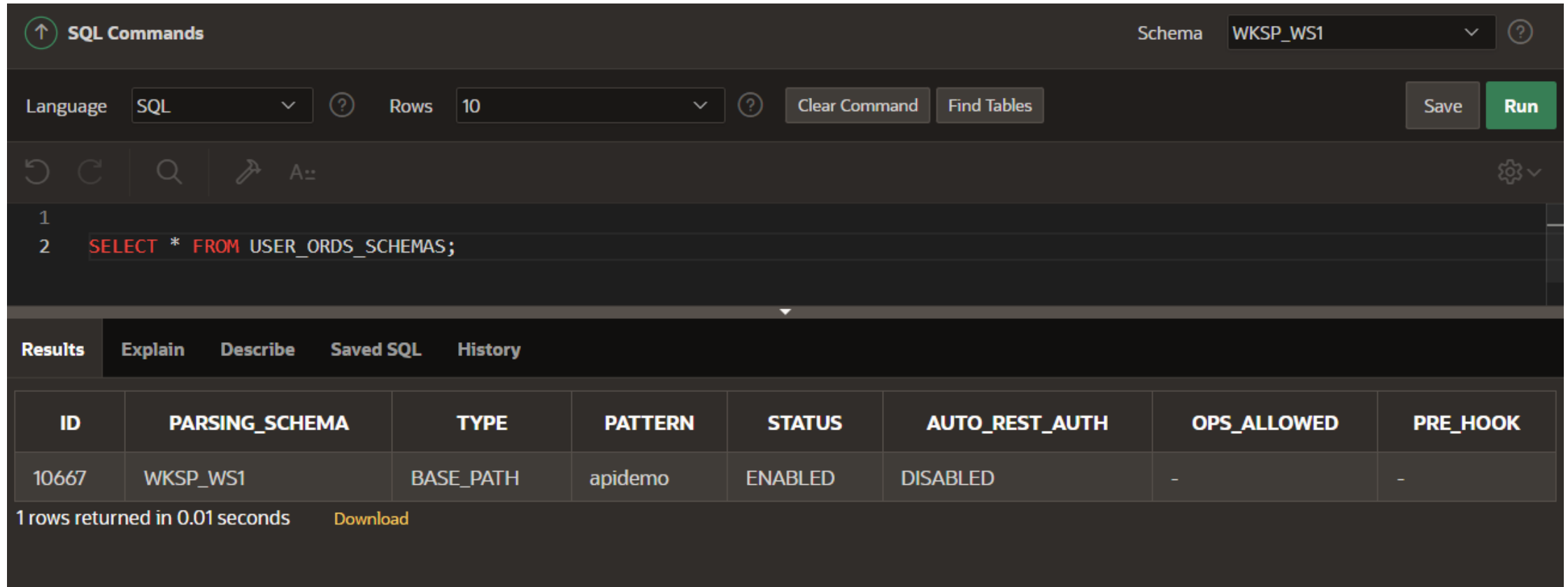
```
ORDS.ENABLE_SCHEMA (  
    p_enabled          => TRUE,  
    p_schema           => 'WKSP_RUTVIKWS',  
    p_url_mapping_type  => 'BASE_PATH',  
    p_url_mapping_pattern => 'apidemo', --- Schema Alias which will appear in the URL  
    p_auto_rest_auth    => FALSE  
);
```

- **Note:** Only database users with the DBA role can enable or disable a schema other than their own.



# How to check if Schema is ORDS Enabled or not?

- By querying USER\_ORDS\_SCHEMAS and ORDS\_METADATA.ORDS\_SCHEMAS we can check the status of schema.



The screenshot shows an SQL IDE interface. At the top, there's a 'SQL Commands' tab. Below it, a 'Schema' dropdown is set to 'WKSP\_WS1'. The 'Language' is set to 'SQL' and 'Rows' is set to '10'. There are buttons for 'Clear Command', 'Find Tables', 'Save', and a green 'Run' button. Below the command area, there's a toolbar with icons for undo, redo, search, and a settings gear. The SQL command entered is: `SELECT * FROM USER_ORDS_SCHEMAS;`. Below the command area, there's a tabbed interface with 'Results' selected. The results are displayed in a table with 8 columns: ID, PARSING\_SCHEMA, TYPE, PATTERN, STATUS, AUTO\_REST\_AUTH, OPS\_ALLOWED, and PRE\_HOOK. One row is returned with ID 10667, PARSING\_SCHEMA WKSP\_WS1, TYPE BASE\_PATH, PATTERN apidemo, STATUS ENABLED, AUTO\_REST\_AUTH DISABLED, OPS\_ALLOWED -, and PRE\_HOOK -. Below the table, it says '1 rows returned in 0.01 seconds' and there's a 'Download' link.

ID	PARSING_SCHEMA	TYPE	PATTERN	STATUS	AUTO_REST_AUTH	OPS_ALLOWED	PRE_HOOK
10667	WKSP_WS1	BASE_PATH	apidemo	ENABLED	DISABLED	-	-

1 rows returned in 0.01 seconds [Download](#)

# Demo

# AUTOREST Using APEX

The screenshot shows the APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. A search bar and user profile 'admin ws1' are on the right. The 'Object Browser' on the left lists various tables, with 'OEHR\_EMPLOYEES' selected. The main panel displays the 'REST' configuration for 'OEHR\_EMPLOYEES' in the 'WKSP\_WS1' schema. The configuration includes a 'REST Enable Object' toggle set to 'Yes', an 'Object Alias' of 'emp', an 'Authorization Required' toggle set to 'No', and an 'Authorization Role' of 'oracle.dbtools.role.autorest.WKSP\_WS1.OEHR\_EMPLOYEES'. An 'Apply' button is at the bottom right.

**Object Browser**

Tables

DEPT

EMP

OEHR\_COUNTRIES

OEHR\_CUSTOMERS

OEHR\_DEPARTMENTS

**OEHR\_EMPLOYEES**

OEHR\_INVENTORIES

OEHR\_JOBS

OEHR\_JOB\_HISTORY

OEHR\_LOCATIONS

OEHR\_ORDERS

OEHR\_ORDER\_ITEMS

OEHR\_PRODUCT\_DESCRIPTIONS

OEHR\_PRODUCT\_INFORMATION

**OEHR\_EMPLOYEES**

Schema: WKSP\_WS1

Table Data Indexes Model Constraints Statistics UI Defaults Triggers Dependencies SQL **REST** Sample Que

ORDS Version 23.1.2.r1151944

REST Enable Object **Yes** No ?

Object Alias emp ?

Authorization Required **No** Yes ?

Authorization Role oracle.dbtools.role.autorest.WKSP\_WS1.OEHR\_EMPLOYEES ?

**Apply**

# AUTOREST Using APEX

The screenshot shows the Oracle APEX Object Browser interface. On the left, the 'Object Browser' pane lists various database objects, with 'OEHR\_EMPLOYEES' selected. The main pane displays the configuration for the REST endpoint of this table. A green banner at the top indicates 'TABLE OEHR\_EMPLOYEES successfully processed'. Below this, a tabbed interface shows various properties, with the 'REST' tab selected. The 'REST' tab displays the 'ORDS Version 23.1.2.r1151944' and configuration options for the REST endpoint. The 'REST Enable Object' is set to 'Yes', the 'Object Alias' is 'emp', and 'Authorization Required' is set to 'No'. The 'RESTful URI' is highlighted with a yellow box and shows the URL 'https://g797c891abe3879-db1.adb.us-phoenix-1.oraclecloudapps.com/ords/apidemo/emp/'. An 'Apply' button is located at the bottom right of the configuration pane.

Object Browser

Schema: WKSP\_WS1

Tables

DEPT

EMP

OEHR\_COUNTRIES

OEHR\_CUSTOMERS

OEHR\_DEPARTMENTS

**OEHR\_EMPLOYEES**

OEHR\_INVENTORIES

OEHR\_JOBS

OEHR\_JOB\_HISTORY

OEHR\_LOCATIONS

OEHR\_ORDERS

OEHR\_ORDER\_ITEMS

OEHR\_PRODUCT\_DESCRIPTIONS

OEHR\_PRODUCT\_INFORMATION

OEHR\_PROMOTIONS

OEHR\_REGIONS

**OEHR\_EMPLOYEES**

Table Data Indexes Model Constraints Statistics UI Defaults Triggers Dependencies SQL **REST** Sample Query

ORDS Version 23.1.2.r1151944

REST Enable Object **Yes** No ?

Object Alias emp ?

Authorization Required **No** Yes ?

RESTful URI **https://g797c891abe3879-db1.adb.us-phoenix-1.oraclecloudapps.com/ords/apidemo/emp/**

Apply



# AUTOREST Using ORDS API

```
ORDS.ENABLE_OBJECT (  
    p_enabled          => TRUE, -- Is AutoREST Access enabled  
    p_schema           => 'WKSP_WS1', -- Schema owning the object  
    p_object            => 'OEHR_EMPLOYEES' -- Object Name  
    p_object_type       => 'TABLE', -- Object Type  
    p_object_alias      => 'emp', --- Object Alias  
    p_auto_rest_auth    => FALSE  
);
```

# Demo

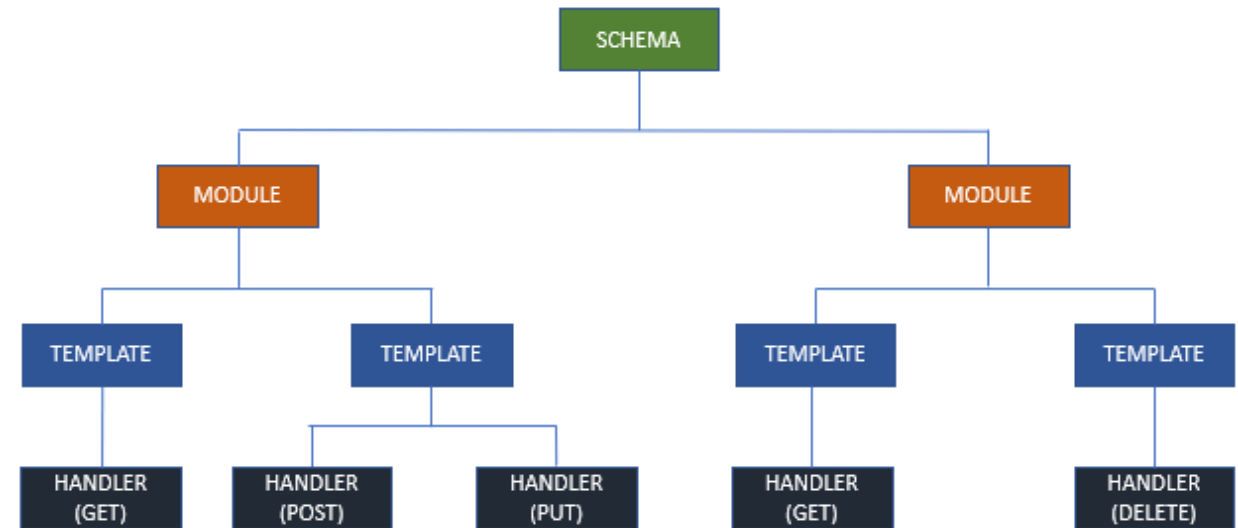
# Manual REST Service Creation

---

- Requires you to specify the SQL or PL/SQL to support required actions
- More effort but more flexibility
- Can customize columns, join across multiple tables, etc.
- Validate incoming data using PL/SQL
- Include complex logic to decide what action to take

# Manual REST Service Creation

- Create Module
- Create Template
- Create Handler
  - For each action we need to create a new handler
    - GET
    - POST
    - PUT
    - DELETE



# Manual REST Service Creation

HR.EMPLOYEES		
P *	EMPLOYEE_ID	NUMBER (6)
	FIRST_NAME	VARCHAR2 (20 BYTE)
	* LAST_NAME	VARCHAR2 (25 BYTE)
	* EMAIL	VARCHAR2 (25 BYTE)
	PHONE_NUMBER	VARCHAR2 (20 BYTE)
	* HIRE_DATE	DATE
F *	JOB_ID	VARCHAR2 (10 BYTE)
	SALARY	NUMBER (8,2)
	COMMISSION_PCT	NUMBER (2,2)
F	MANAGER_ID	NUMBER (6)
F	DEPARTMENT_ID	NUMBER (4)
◆ EMP_EMAIL_UK		
▶ EMP_EMP_ID_PK		
◆ EMP_DEPARTMENT_IX		
◆ EMP_JOB_IX		
◆ EMP_MANAGER_IX		
◆ EMP_NAME_IX		

Module : hr/v1/

Template : employees/:id?

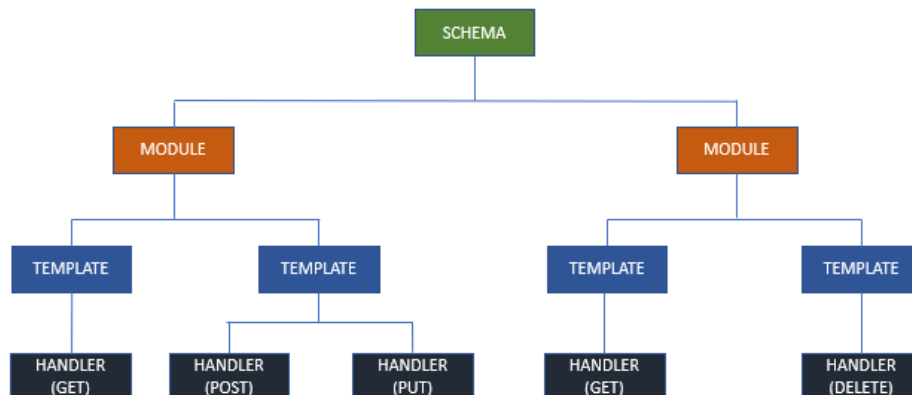
Handler:

GET : Get all emps or single emps

POST : Create single emp record

PUT : Update single emp record

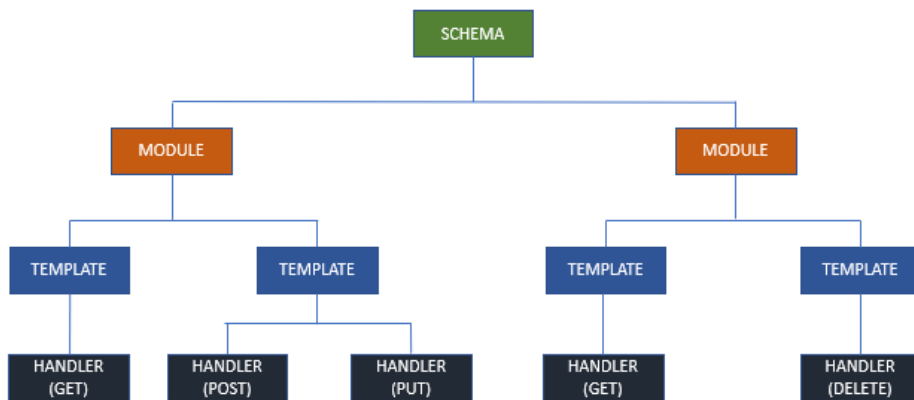
DELETE : Delete single emp record



# Manual REST Service Creation

Added 3 columns to Employees table

MIME_TYPE	VARCHAR2(256)	Yes
FILE_NAME	VARCHAR2(256)	Yes
CONTENT	BLOB	Yes



Module : hr/v2/

Template : employees/

Handler:

GET : Get all emps

POST : Create single emp record

Template : employees/:id

Handler:

GET : Get single emp record

PUT : Update single emp record

DELETE : Delete single emp record

Template : employees/:id/photo/

Handler:

GET : Get photo of emp

PUT : Update emp photo

# Manual REST Service Creation

---

## ORDS Parameters

- Implicit
  - :body, :body\_text, :status\_code, :content\_type, and more...
- Explicit
  - User defined

Find out more about implicit parameters:

<https://docs.oracle.com/en/database/oracle/oracle-rest-data-services/22.1/orddg/implicit-parameters.html#GUID-B0BB1694-715C-4948-84A5-307EA3868063>

# Manual REST Service Creation

Parameter name that you can see in request of response.

Bind variable name that you can reference in SQL or PL/SQL code.

Indicates the source type of parameters.  
IN: URI, HTTP Header  
OUT: Response, HTTP Header

		Name	Bind Variable	Access Method	Source Type	Data Type
<input checked="" type="checkbox"/>	≡	status_msg	status_msg	OUT	RESPONSE	STRING
<input type="checkbox"/>	≡	X-ORDS-FORWARD-LOCATION	res_location	OUT	HTTP HEADER	STRING
<input type="checkbox"/>	≡	X-ORDS-STATUS-CODE	status	OUT	HTTP HEADER	INTEGER

Access Method: IN or OUT

IN : to receive the value from the client during the request


OUT : to send the value to the client during response



# Manual REST Service Creation


:id is a mandatory bind variable

\* URI Template  ?

Full URL <https://g797c891abe3879-db1.adb.us-phoenix-1.oraclecloudapps.com/ords/apidemo/hr/v2/employees/:id> 

:id is an optional bind variable

\* URI Template  ?

Full URL <https://g797c891abe3879-db1.adb.us-phoenix-1.oraclecloudapps.com/ords/apidemo/hr/v1/employees/:id?> 

# Manual REST Service Creation

Column that starts with \$ : ORDS will generate a link for that JSON attribute.

```
Source
1 select empno "$uri",
2         rn,
3         empno,
4         ename,
5         job,
6         hiredate,
7         mgr,
8         sal,
9         comm,
10        deptno
11 from (
12     select emp.*
13           , row_number() over (order by empno) rn
14     from emp
15 ) tmp
```

```
Body Cookies Headers (10) Test Results
Pretty Raw Preview Visualize JSON
11 "comm": null,
12 "deptno": 20,
13 "links": [
14     {
15         "rel": "uri",
16         "href": "https://apex.oracle.com/pls/apex/apidemo/hr/employees/7369"
17     }
18 ],
19 },
20 {
21     "In": 2,
22     "empno": 7499,
23     "ename": "ALLEN",
```

# Manual REST Service Creation

Parameter that starts with \$ : ORDS will generate a link for that JSON attribute.

```
1  BEGIN
2  HR_API_V1.CREATE_EMP(
3      P_EMPLOYEE_ID => :employee_id,
4      P_FIRST_NAME  => :first_name,
5      P_LAST_NAME   => :last_name,
6      P_EMAIL       => :email,
7      P_PHONE_NUMBER => :phone,
8      P_HIRE_DATE   => :hire_date,
9      P_JOB_ID      => :job_id,
10     P_SALARY       => :salary,
11     P_COMMISSION_PCT => :comm,
12     P_MANAGER_ID   => :manager_id,
13     P_DEPARTMENT_ID => :department_id,
14     P_RES_LOC      => :res_location,
15     P_STATUS_CD    => :status,
16     P_STATUS_MSG   => :status_msg
17 );
18
19 END;
```

Parameters

Search: All Text Columns

<input type="checkbox"/>	<input type="checkbox"/>	Name	Bind Variable	Access Method	Source Type	Data Type
<input checked="" type="checkbox"/>	<input type="checkbox"/>	\$self	res_location	OUT	RESPONSE	STRING
<input type="checkbox"/>	<input type="checkbox"/>	status_msg	status_msg	OUT	RESPONSE	STRING
<input type="checkbox"/>	<input type="checkbox"/>	X-ORDS-STATUS-CODE	status	OUT	HTTP HEADER	INTEGER

# Securing REST Service

---

Mainly two types of authentications

- Basic Auth
  - Protect a service using username and password
  - Database Users
  - APEX Workspace User
- OAUTH2
  - More Secure
  - Involves Provider (Server) and Consumer (Client) of the service
  - Owner of the service creates a “Client”, assigns it privileges and provides details to the consumer

# Securing REST Service using OAuth2

## Server Side

- Define the Module
- Create a Role
- Create a Privilege
- Create Client Credentials
- Link Module, Role, Privilege and Client

## Client Side

- Authenticate as Client
- Validate/Retrieve Token
- Use token to access REST Service

# Securing REST Service using OAuth2

## Create ORDS Role

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes the APEX logo, 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery' menus, along with a search bar and user profile icons. The breadcrumb trail indicates the current location: 'RESTful Services > Roles > Role Definition'. The 'Schema' dropdown is set to 'WKSP\_WS1'. On the left sidebar, 'RESTful Data Services' is expanded, showing 'Enabled Objects', 'Modules', 'Privileges', and 'Roles'. The main panel displays the 'ORDS Role Definition' dialog with the following fields:

- Owner:** WKSP\_WS1
- Role Name:** inaoug.demo.role

At the top right of the dialog are 'Cancel' and 'Create Role' buttons.

# Securing REST Service using OAuth2

## Create Privilege, Assign Role and Protect Modules

ORDS Privileges Definition

Cancel

Create Privilege

Owner

WKSP\_WS1

Name

inaoug.demo.privilege

?

Title

INAUG Demo Privilege

?

Description

?

Comments

Roles

oracle.dbtools.role.autorest.WKSP\_WS1

oracle.dbtools.role.autorest.WKSP\_WS1.PKG\_EMP

OAuth2 Client Developer

SQL Developer

RESTful Services

Schema Administrator

oracle.dbtools.autorest.any.schema

SODA Developer

?

inaoug.demo.role

<

>>

>

<

<<

Protected Modules

Modules

hr/v2/

101

hr/v1/

hr/v3/

<

>>

>

<

<<

\*Rutvik Prajapati, @rutveek

# Securing REST Service using OAuth2

After this step, our webservice is protected and we cannot directly access it

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** `https://g797c891abe3879-db1.adb.us-phoenix-1.oraclecloudapps.com/ords/apidemo/hr/v3/employees/`
- Authorization:** No Auth (selected)
- Status:** 401 Unauthorized
- Time:** 1599 ms
- Size:** 16.13 KB
- Response Body:**

**ORACLE<sup>®</sup>**  
REST Data Services

**401** Unauthorized

2023-05-04T16:04:30.431252Z | 4b0c0b882e2fa75e4648917033ae88ad |

Access to this resource is protected. Please [sign in](#) to access this resource.



# Securing REST Service using OAuth2

Create client using PL/SQL API

```
BEGIN
  OAUTH.create_client(
    p_name      => 'rutvik.prajapati',
    p_grant_type => 'client_credentials',
    p_owner     => 'WKSP_WS1',
    p_description => 'Internal client to access API',
    p_support_email => 'rutveek@gmail.com',
    p_privilege_names => 'inaoug.demo.privilege'
  );

  COMMIT;
END;
```

Assign Role to client

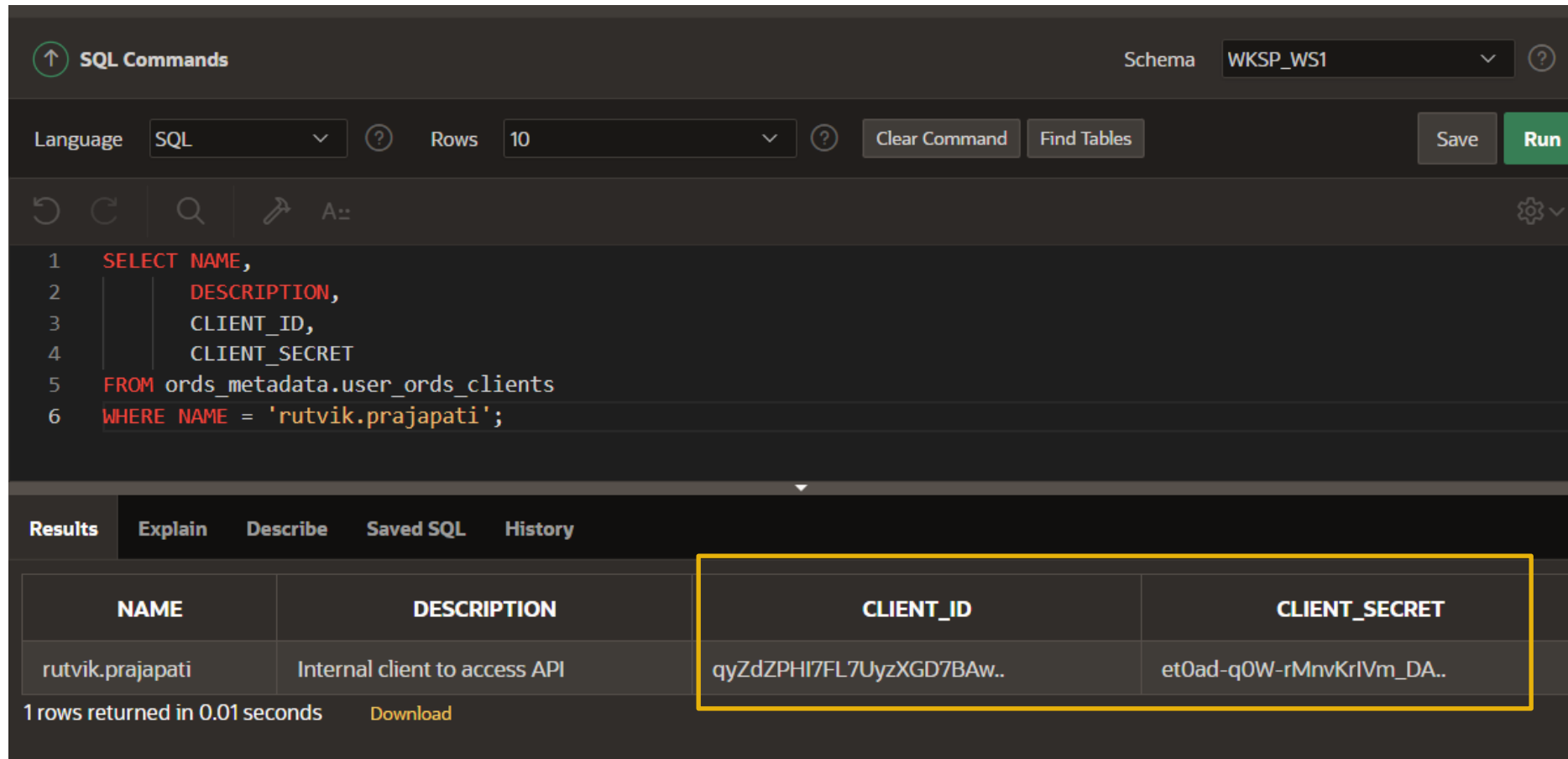
```
BEGIN

  OAUTH.grant_client_role(
    p_client_name => 'rutvik.prajapati',
    p_role_name  => 'inaoug.demo.role'
  );

  COMMIT;
END;
```

# Securing REST Service using OAuth2

- We created client and assign a role to that client. Next, we need to provide client id and client secret to the consumer(client) of this API.
- For that query ords\_metadata.user\_ords\_clients view



The screenshot shows a SQL IDE interface. At the top, the 'SQL Commands' tab is active, and the 'Schema' is set to 'WKSP\_WS1'. The 'Language' is 'SQL' and 'Rows' are set to '10'. The query being executed is:

```
1 SELECT NAME,  
2     DESCRIPTION,  
3     CLIENT_ID,  
4     CLIENT_SECRET  
5 FROM ords_metadata.user_ords_clients  
6 WHERE NAME = 'rutvik.prajapati';
```

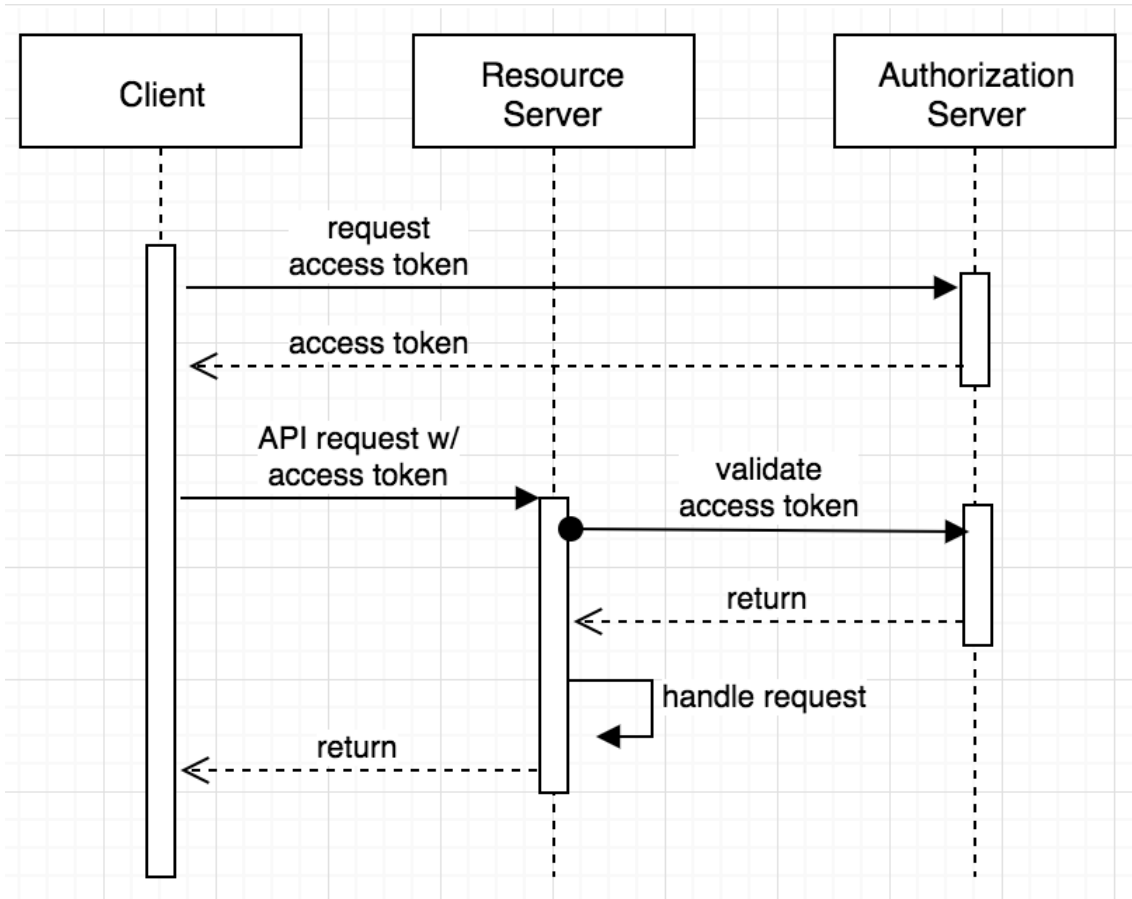
Below the query, the 'Results' tab is active, displaying a table with the following data:

NAME	DESCRIPTION	CLIENT_ID	CLIENT_SECRET
rutvik.prajapati	Internal client to access API	qyZdZPHI7FL7UyzXGD7BAw..	et0ad-q0W-rMnvKrIVm_DA..

At the bottom, it indicates '1 rows returned in 0.01 seconds' and provides a 'Download' link. A yellow box highlights the 'CLIENT\_ID' and 'CLIENT\_SECRET' columns in the results table.

# Securing REST Service using OAuth2

- OAuth2 Client Credential Flow



Access Token URL

`http(s)://<server>/<ords_alias>/<schema_alias>/oauth/token`

# Securing REST Service using OAuth2

The screenshot shows a REST client interface with a dark theme. At the top, there is a method dropdown set to 'GET' and a URL field containing 'https://g797c891abe3879-db1.adb.us-phoenix-1.oraclecloudapps.com/ords/apidemo/hr/v3/employees/'. To the right of the URL is a blue 'Send' button with a dropdown arrow. Below the URL bar are tabs for 'Params', 'Authorization', 'Headers (6)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Authorization' tab is selected and highlighted with an orange underline. To the right of these tabs is a 'Cookies' link. The main area of the 'Authorization' tab contains a form for configuring OAuth2. It includes fields for 'Token Name' (set to 'INAUG Demo Token'), 'Grant Type' (set to 'Client Credentials'), 'Access Token URL' (truncated), 'Client ID' (with a warning icon), 'Client Secret' (with a warning icon), 'Scope' (with a placeholder 'e.g. read:org'), and 'Client Authentication' (set to 'Send as Basic Auth header').

GET	https://g797c891abe3879-db1.adb.us-phoenix-1.oraclecloudapps.com/ords/apidemo/hr/v3/employees/	Send
Params	Authorization	Headers (6)
Body	Pre-request Script	Tests
Settings		
Cookies		
Token Name	INAUG Demo Token	
Grant Type	Client Credentials	
Access Token URL ⓘ	https://g797c891abe3879-db1.adb.us-phoe...	
Client ID ⓘ	qyZdZPHI7FL7UyzXGD7BAw.. ⚠	
Client Secret ⓘ	et0ad-q0W-rMnvKrIVm_DA.. ⚠	
Scope ⓘ	e.g. read:org	
Client Authentication	Send as Basic Auth header	

# Securing REST Service using OAuth2

MANAGE ACCESS TOKENS

All TokensDelete

INAOUG-Demo-Token

INAOUG Demo Token

Token Details

Token NameINAOUG Demo Token

Access Tokenz7R-NwYHtwvKAsBGBly90A

Token Typebearer

expires\_in3600

access\_token\_urlhttps://g797c891abe3879-db1.adb.us-phoenix-1.oraclecloudapps.com/ords/apidemo/oauth/token

client\_idqyZdZPHI7FL7UyzXGD7BAw..

client\_secretet0ad-q0W-rMnvKrlVm\_DA..

timestamp1683217261744

Use Token

# Securing REST Service using OAuth2

The screenshot shows a REST client interface with a dark theme. At the top, a GET request is configured to the URL `https://g797c891abe3879-db1.adb.us-phoenix-1.oraclecloudapps.com/ords/apidemo/hr/v3/employees/`. Below the URL bar, tabs for Params, Authorization, Headers (6), Body, Pre-request Script, Tests, and Settings are visible. The Authorization tab is active, displaying OAuth 2.0 configuration options. On the left, under 'Type', 'OAuth 2.0' is selected. Below it, a note states: 'The authorization data will be automatically generated when you send the request. Learn more about [authorization](#)'. Under 'Add authorization data to', a dropdown menu is open with 'Request Headers' selected. The main area shows 'Current Token' information: 'This token is only available to you. Sync the token to let collaborators on this request use it.' Below this, the 'Token' dropdown is set to 'INAOUG Demo Token', and the token value 'z7R-NwYHtwvKAsBGBly90A' is displayed. The 'Header Prefix' is set to 'Bearer'. At the bottom, there are two toggle switches: 'Auto-refresh token' (disabled) and 'Share token' (disabled). The 'Configure New Token' section at the bottom shows the 'Token Name' as 'INAOUG Demo Token'. A blue 'Send' button is located in the top right corner.

GET `https://g797c891abe3879-db1.adb.us-phoenix-1.oraclecloudapps.com/ords/apidemo/hr/v3/employees/` Send

Params **Authorization** Headers (6) Body Pre-request Script Tests Settings Cookies

Type OAuth 2.0

The authorization data will be automatically generated when you send the request. Learn more about [authorization](#)

Add authorization data to Request Headers

Request URL

Request Headers

**Current Token**

This token is only available to you. Sync the token to let collaborators on this request use it.

Token INAOUG Demo Token

z7R-NwYHtwvKAsBGBly90A

Header Prefix Bearer

Auto-refresh token

Share token

**Configure New Token**

Token Name INAOUG Demo Token

# Securing REST Service using OAuth2

The screenshot displays the Oracle REST Data Services (ORDS) interface for a service named "ORDS HR/V3 / Secure Emps". The main configuration area is set to "Authorization" with "OAuth 2.0" selected as the type. The "Current Token" section shows a "Token" dropdown set to "INAUG Demo Token" and a token value of "z7R-NwYHtwvKAsBGBly90A". The "Body" tab is active, showing a JSON response in "Pretty" format. The response is a list of employee records, with the first record being Steven King.

**Request Configuration:**

- Method: GET
- URL: `https://g797c891abe3879-db1.adb.us-phoenix-1.oraclecloudapps.com/ords/apidemo/hr/v3/employees/`
- Authorization Type: OAuth 2.0
- Token: INAUG Demo Token
- Token Value: z7R-NwYHtwvKAsBGBly90A

**Response (JSON):**

```
1 {
2   "items": [
3     {
4       "employee_id": 100,
5       "first_name": "Steven",
6       "last_name": "King",
7       "email": "SKING",
8       "phone_number": "515.123.4567",
9       "hire_date": "2010-03-14T00:00:00Z",
10      "job_id": "AD_PRES",
11      "salary": 24000,
```

# Demo



# Learn More

---

<https://www.thatjeffsmith.com/archive/2020/06/ords-101-working-with-pl-sql/>

<https://www.thatjeffsmith.com/archive/2019/03/calling-a-pl-sql-function-via-ords/>

<https://www.thatjeffsmith.com/oracle-database-rest-apis/>



**THANK YOU**

# Q & A