



Oracle Application Express

Using RESTful Web Services in Oracle Application Express (APEX)

Hands on Lab: Using RESTful Web Services in Oracle Application Express.

Topic	Details
Overview	In this Hands on Lab, you will learn RESTful Web Services using hands-on classroom activities to allow a java program to connect to an APEX database.
Key Concepts	<ul style="list-style-type: none">• APEX RESTful Web Service Lab Setup.• Retrieve, modify and delete data using RESTful Services.• Create an APEX Application Using RESTful service.• Consuming the RESTful Web Service Created in Application Express Using a Java Client.
Difficulty	Advanced
Duration	Approximately 180 minutes
Notes	<p>Access to an APEX account at apex.oracle.com is required. Accounts can be requested at https://apex.oracle.com or install APEX with Oracle REST Data Services and configured it to connect to an Oracle database.</p> <p>It is strongly recommended that you install a browser extension that enables you to view JSON in a web browser. Recommended extension for Mozilla Firefox: JSON View.</p> <p>A RESTful Services Testing Tool is required. Postman, can be downloaded from https://www.getpostman.com/ Students should have prior knowledge of how to create and execute java programs, and have experience of writing and executing SQL statements in APEX.</p>

Introduction

This tutorial covers creating a RESTful Web Service declaratively using Oracle Application Express's SQL Workshop tool to connect to a database table, and then consuming this service by creating an application and adding a Web Service reference to the RESTful Web Service. A client java program is then created and executed that will consume the RESTful Web Service to return data stored in an APEX database table to the java application.

Web Services enable applications to interact with one another over the web in a platform-neutral, language independent environment. In a typical Web Services scenario, a business application sends a request to a service at a given URL by using the HTTP protocol. The service receives the request, processes it, and returns a response.

Once you have defined a RESTful Web Service, you can call it with a unique Uniform Resource Identifier (URI). The Web Service uses the URI to call methods such as GET, POST, PUT, and DELETE.

Section 1

Verify the emp table used in the lab

1. Login into your APEX schema, click SQL Workshop.



App Builder



SQL Workshop



Team Development



Packaged Apps

2. Choose Object Browser.



Object Browser



SQL Commands



SQL Scripts



Utilities



RESTful Services

3. Verify that the EMP table and Data are in the Schema. If the Contents are already created then move to the next section.

Object Browser

Schema: OA

Tables

DEPT

EMP

EMP

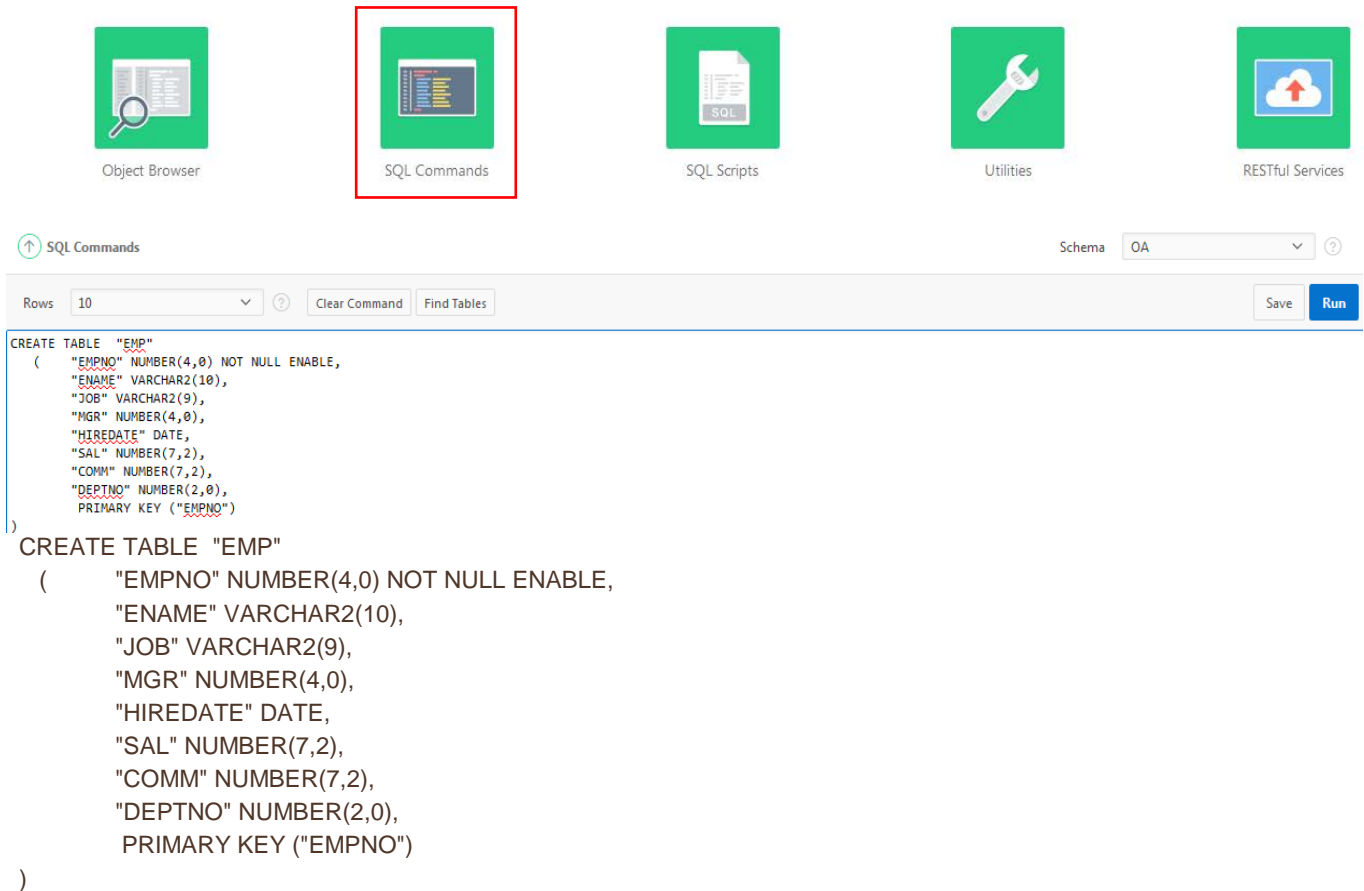
Table Data Indexes Model Constraints Grants Statistics UI Defaults Triggers Dependencies SQL

Query Count Rows Insert Row

Data

EDIT	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
	7839	KING	PRESIDENT		11/17/1981	5234	-	10
	7698	BLAKE	MANAGER	7839	05/01/1981	2850	-	30
	7782	CLARK	MANAGER	7839	06/09/1981	2450	-	10
	7566	JONES	MANAGER	7839	04/02/1981	2975	-	20
	7788	SCOTT	ANALYST	7566	12/09/1982	3000	-	20
	7902	FORD	ANALYST	7566	12/03/1981	3000	-	20
	7369	SMITH	CLERK	7902	12/17/1980	800	-	20
	7499	ALLEN	SALESMAN	7698	02/20/1981	1600	300	30
	7521	WARD	SALESMAN	7698	02/22/1981	1250	500	30
	7654	MARTIN	SALESMAN	7698	09/28/1981	1250	1400	30
	7844	TURNER	SALESMAN	7698	09/08/1981	1500	0	30

4. If the EMP table does not exist, create the EMP table and data manually. Choose SQL Commands.



The screenshot shows the SQL Developer interface with the 'SQL Commands' tab selected. The 'EMP' table creation SQL is entered in the command window:

```
CREATE TABLE "EMP"  
(  
  "EMPNO" NUMBER(4,0) NOT NULL ENABLE,  
  "ENAME" VARCHAR2(10),  
  "JOB" VARCHAR2(9),  
  "MGR" NUMBER(4,0),  
  "HIREDATE" DATE,  
  "SAL" NUMBER(7,2),  
  "COMM" NUMBER(7,2),  
  "DEPTNO" NUMBER(2,0),  
  PRIMARY KEY ("EMPNO")  
)  
  
CREATE TABLE "EMP"  
(  
  "EMPNO" NUMBER(4,0) NOT NULL ENABLE,  
  "ENAME" VARCHAR2(10),  
  "JOB" VARCHAR2(9),  
  "MGR" NUMBER(4,0),  
  "HIREDATE" DATE,  
  "SAL" NUMBER(7,2),  
  "COMM" NUMBER(7,2),  
  "DEPTNO" NUMBER(2,0),  
  PRIMARY KEY ("EMPNO")  
)
```

5. Insert the following data into the EMP table.



The screenshot shows the SQL Scripts | Script Editor window with the following SQL insert statements:

```
1 Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values (7369,'SMITH','CLERK',7902,to_date('17-DEC-80','DD-MON-RR'),800,null,20);  
2 Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values (7499,'ALLEN','SALESMAN',7698,to_date('20-FEB-81','DD-MON-RR'),1600,300,30);  
3 Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values (7521,'WARD','SALESMAN',7698,to_date('22-FEB-81','DD-MON-RR'),1250,500,30);  
4 Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values (7566,'JONES','MANAGER',7839,to_date('02-APR-81','DD-MON-RR'),2975,null,20);  
5 Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values (7654,'MARTIN','SALESMAN',7690,to_date('28-SEP-81','DD-MON-RR'),1250,1400,30);  
6 Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values (7698,'BLAKE','MANAGER',7839,to_date('01-MAY-81','DD-MON-RR'),2850,null,30);  
7 Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values (7702,'CLARK','MANAGER',7839,to_date('09-JUN-81','DD-MON-RR'),2450,null,10);  
8 Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values (7788,'SCOTT','ANALYST',7566,to_date('19-APR-87','DD-MON-RR'),3000,null,20);  
9 Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values (7839,'KING','PRESIDENT',null,to_date('17-NOV-81','DD-MON-RR'),5000,null,10);  
10 Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values (7844,'TURNER','SALESMAN',7698,to_date('08-SEP-81','DD-MON-RR'),1500,0,30);  
11 Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values (7876,'ADAMS','CLERK',7708,to_date('23-MAY-87','DD-MON-RR'),1100,null,20);  
12 Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values (7900,'JAMES','CLERK',7698,to_date('03-DEC-81','DD-MON-RR'),950,null,30);  
13 Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values (7902,'FORD','ANALYST',7566,to_date('03-DEC-81','DD-MON-RR'),3000,null,20);  
14 Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values (7934,'MILLER','CLERK',7782,to_date('23-JAN-82','DD-MON-RR'),1300,null,10);  
15 commit;
```

```
Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values  
(7369,'SMITH','CLERK',7902,to_date('17-DEC-80','DD-MON-RR'),800,null,20);  
Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values  
(7499,'ALLEN','SALESMAN',7698,to_date('20-FEB-81','DD-MON-RR'),1600,300,30);  
Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values  
(7521,'WARD','SALESMAN',7698,to_date('22-FEB-81','DD-MON-RR'),1250,500,30);
```

```

Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values
(7566,'JONES','MANAGER',7839,to_date('02-APR-81','DD-MON-RR'),2975,null,20);
Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values
(7654,'MARTIN','SALESMAN',7698,to_date('28-SEP-81','DD-MON-RR'),1250,1400,30);
Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values
(7698,'BLAKE','MANAGER',7839,to_date('01-MAY-81','DD-MON-RR'),2850,null,30);
Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values
(7782,'CLARK','MANAGER',7839,to_date('09-JUN-81','DD-MON-RR'),2450,null,10);
Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values
(7788,'SCOTT','ANALYST',7566,to_date('19-APR-87','DD-MON-RR'),3000,null,20);
Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values
(7839,'KING','PRESIDENT',null,to_date('17-NOV-81','DD-MON-RR'),5000,null,10);
Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values
(7844,'TURNER','SALESMAN',7698,to_date('08-SEP-81','DD-MON-RR'),1500,0,30);
Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values
(7876,'ADAMS','CLERK',7788,to_date('23-MAY-87','DD-MON-RR'),1100,null,20);
Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values
(7900,'JAMES','CLERK',7698,to_date('03-DEC-81','DD-MON-RR'),950,null,30);
Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values
(7902,'FORD','ANALYST',7566,to_date('03-DEC-81','DD-MON-RR'),3000,null,20);
Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values
(7934,'MILLER','CLERK',7782,to_date('23-JAN-82','DD-MON-RR'),1300,null,10);
commit;

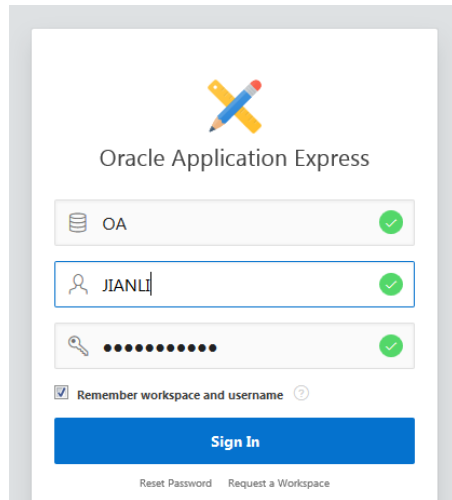
```

Section 2

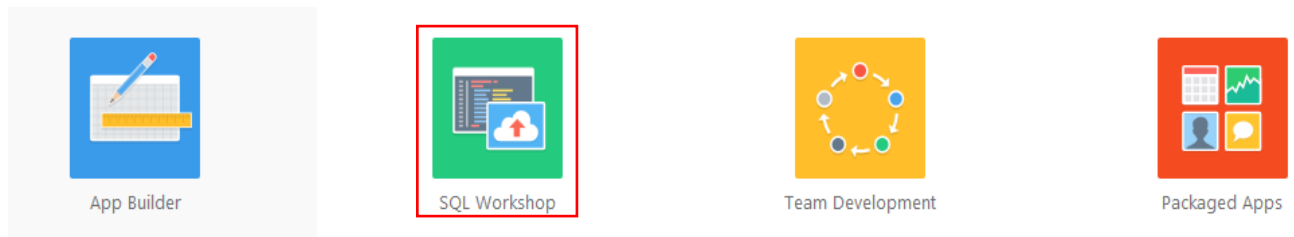
Oracle Academy RESTful Web Service

Lab Setup

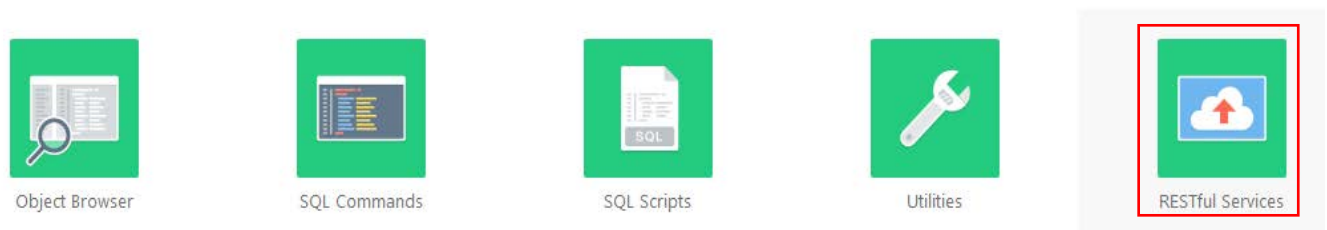
1. Login into Oracle Application Express (APEX).



2. On the Workspace home page, click SQL Workshop.





3. Click RESTful Services. RESTful Services:



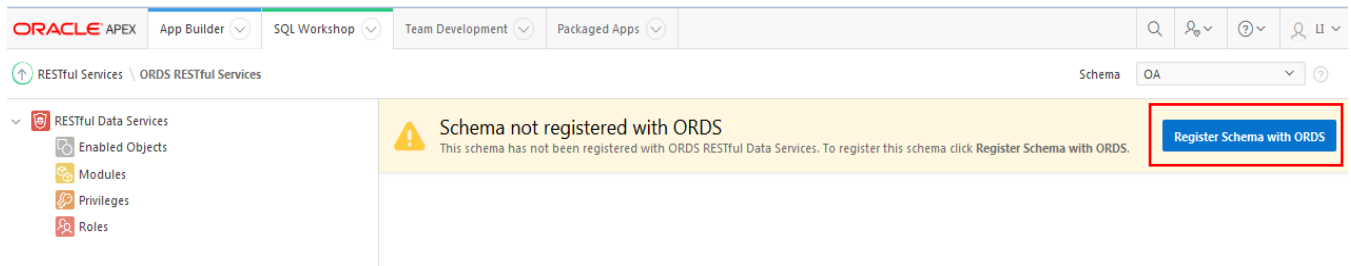
4. Click ORDS Based RESTful Services.

RESTful Service Options

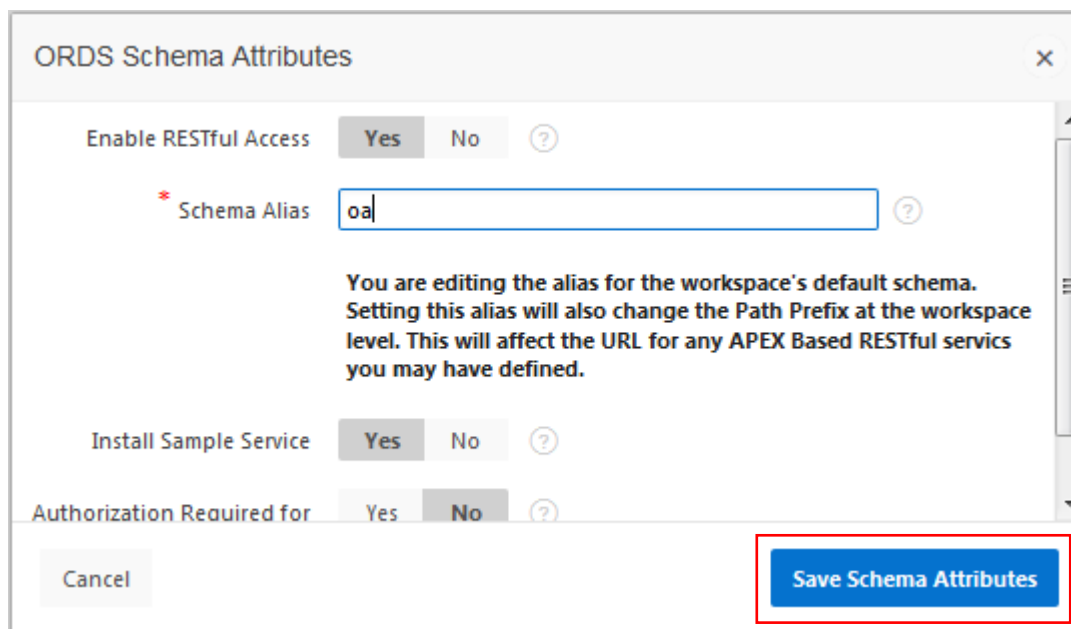
**APEX** APEX Based RESTful Services
View and migrate APEX based RESTful Services to the ORDS REST repository.

**ORDS Based RESTful Services**
Create and manage RESTful services using the ORDS based repository.

5. If your schema is not registered with ORDS RESTful Data Services, click Register Schema with ORDS.



6. ORDS Schema Attributes dialog (Note: your schema name will be different). Click Save Schema Attributes.



7. ORDS RESTful Services dashboard:

ORACLE APEX

App Builder

SQL Workshop

Team Development

Packaged Apps

RESTful Services \ ORDS RESTful Services

Schema OA

RESTful Data Services

Enabled Objects

Modules

Privileges

Roles

Schema enabled for use with ORDS RESTful Services and sample RESTful Service successfully installed.

ORDS Version 18.1.1.95.1251

De-Register Schema from ORDS

Reset Sample Service

Import

Export

Configure

Schema Access	Metadata Access	Schema Aliased
<div> <div>✓</div> <div>Access Status</div> <div>ENABLED</div> </div>	<div> <div>⚠</div> <div>Authorization Required</div> <div>DISABLED</div> </div>	<div> <div>⚠</div> <div>Schema Alias</div> <div>oa</div> </div>

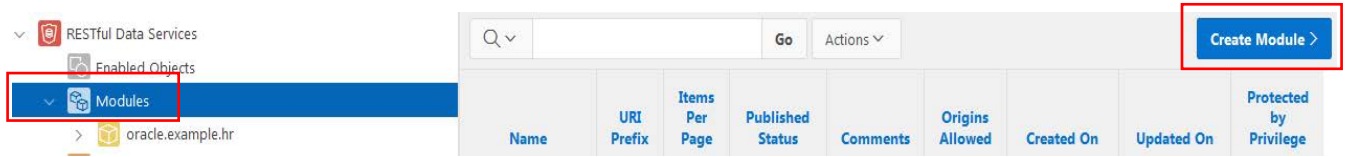
Modules	Privileges	Roles	Enabled Objects
<div> <div>1</div> <div>Total Modules</div> </div>	<div> <div>2</div> <div>Total Privileges</div> </div>	<div> <div>143</div> <div>Total Roles</div> </div>	<div> <div>0</div> <div>Total Enabled Objects</div> </div>

Module Status	Module Security	Object Aliases
<div> <div> <div></div> <div>Published</div> </div> <div> <div></div> <div>Unpublished</div> </div> </div>	<div> <div> <div></div> <div>Secured</div> </div> <div> <div></div> <div>Unsecured</div> </div> <div> <div></div> <div>Partially Secured</div> </div> </div>	<div>No RESTful Enabled Objects Found</div> <div> <div> <div></div> <div>Aliased</div> </div> <div> <div></div> <div>Unaliased</div> </div> </div>

Section 3

Creating the first Restful Web Service

1. The first Restful Web service will retrieve all records in the emp table. Highlight Modules. Click Create Module button.



2. Provide the Module Name `oa_test`, and the Base Path `/oa_test/`. Change Pagination size to 25. Click "Create Module"

The 'ORDS Module Definition' form is shown. The fields are filled as follows: Module Name is 'oa_test', Base Path is '/oa_test/', Is Published is 'Yes', and Pagination Size is '25'. The 'Create Module' button is highlighted in the top right corner.

3. Click "Create Template".

The 'ORDS Module Definition' form is shown again, but with an additional 'Full URL' field filled with 'https://apex.oracle.com/pls/apex/oa/oa_test/'. The 'Create Template >' button is highlighted in the bottom right corner.

4. Create the Resource Template empall/.

ORDS Template Definition

Cancel>Create Template

RESTful Service Module

oa_test

Module Base Path

/oa_test/

* URI Template

empall/

?

* Priority

0

?

* HTTP Entity Tag Type

Secure Hash

?

Comments

5. Create the Resource Handler. Click “Create Handler”

RESTful Service Module

oa_test

Module Base Path

/oa_test/

* URI Template

empall/

?

Full URL

https://apex.oracle.com/pls/apex/oa/oa_test/empall/

* Priority

0

?

* HTTP Entity Tag Type

Secure Hash

?

Comments

Resource Handlers

Create Handler >

Q

Go

Actions

6. Create the Resource Handler. Enter the following query in the “Source” area: `SELECT * FROM emp` Click “Create Handler”

The screenshot shows the "ORDS Handler Definition" form. At the top right, the "Create Handler" button is highlighted with a red box. The form fields are as follows:

- RESTful Service Module: **oa_test**
- Module Base Path: **/oa_test/**
- URI Template: **empall/**
- Method: **GET** (dropdown menu, highlighted with a red box)
- Source Type: **Query** (dropdown menu, highlighted with a red box)
- Format: **JSON** (dropdown menu)
- Pagination Size: **6** (input field)
- Comments: (empty text area)
- Source: (code editor area, highlighted with a red box) containing the query: `1 select * from emp`

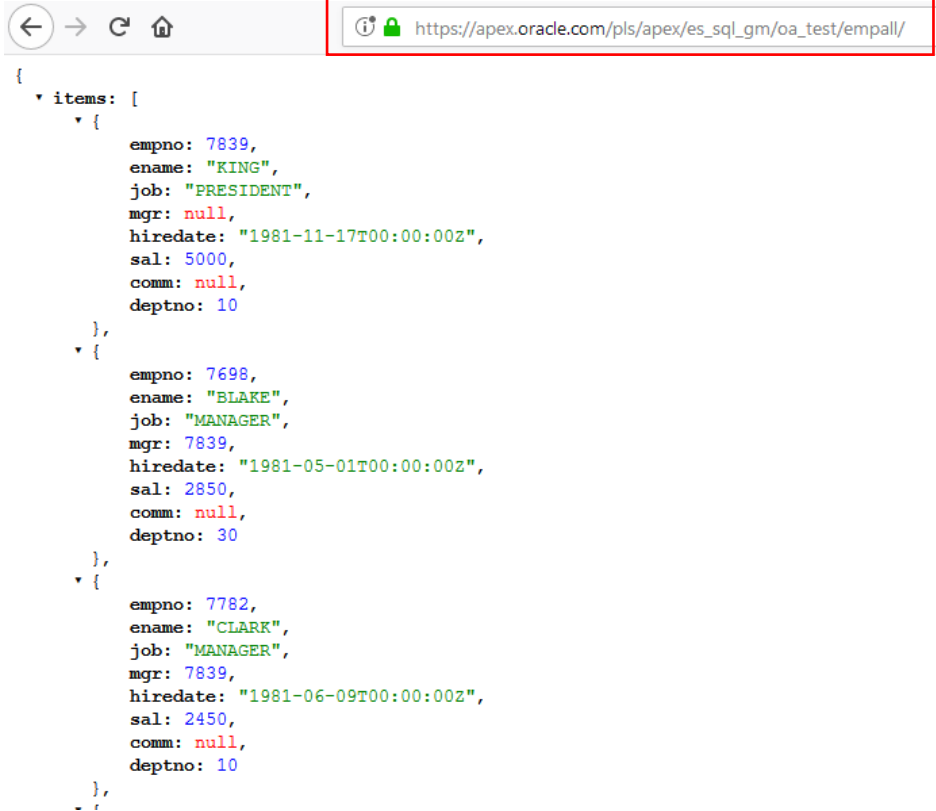
7. Copy the module URL. Ensure the GET Resource Handler is selected, and click the icon to copy the URL.

The screenshot shows the "ORDS Handler Definition" form for an existing handler. At the top right, the "Apply Changes" button is visible. The form fields are as follows:

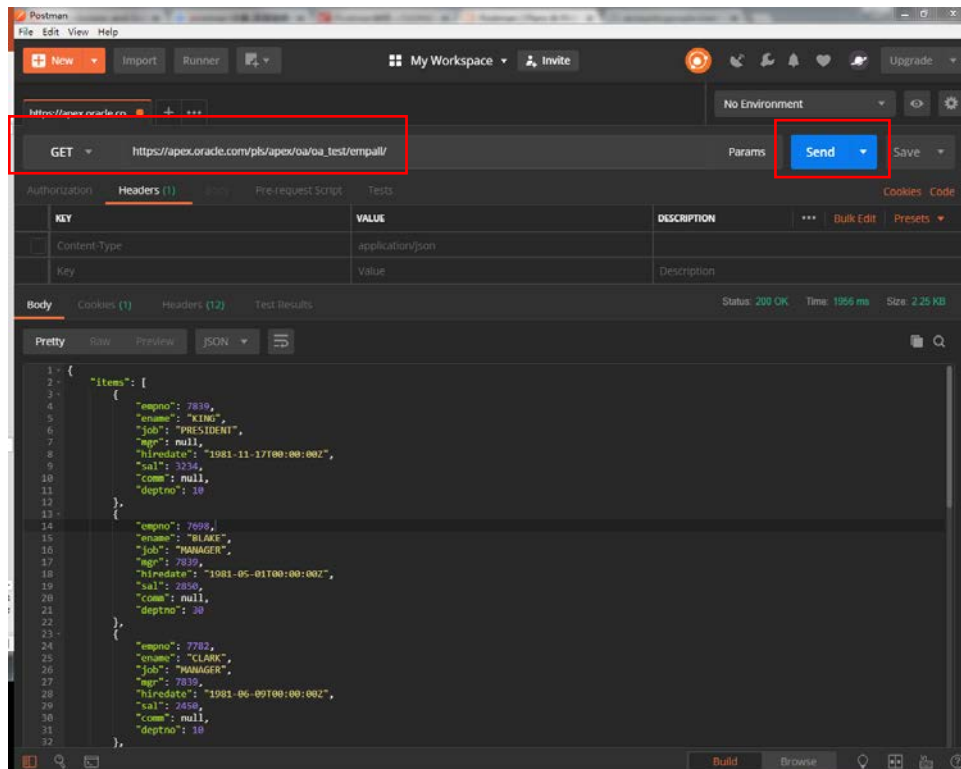
- RESTful Service Module: **oa_test**
- Module Base Path: **/oa_test/**
- URI Template: **empall/**
- Full URL: **https://apex.oracle.com/pls/apex/es_sql_gm/oa_test/empall/** (copy icon highlighted with a red box)
- Method: **GET** (dropdown menu)
- Source Type: **Query** (dropdown menu)
- Format: **JSON** (dropdown menu)
- Pagination Size: **25** (input field)
- Comments: (empty text area)
- Source: (code editor area) containing the query: `1 SELECT * FROM emp`

- Test the Rest Web Service from the web browser.

Paste the URL into the address bar of your browser and press the enter key to see the results.



- Select GET, paste the URL, and click Send to test the Rest Web Service from Postman.



Section 4


Retrieving Data Using a Parameter


1. Perform the following steps to create a RESTful Service which retrieves the employee information based on a parameter id using the HTTP Method GET.
2. In the oa_test module, create a new Template emp/:empid following the steps shown in the previous section.



RESTful Service Module **oa_test**

Module Base Path **/oa_test/**

URI Template **emp/:empid**


Full URL **https://apex.oracle.com/pls/apex/oa/oa_test/emp/:empid** 






* Method **GET** 

* Source Type **Collection Query**  

Format **JSON**

3. Add the Source code for the GET Handler.

Source 

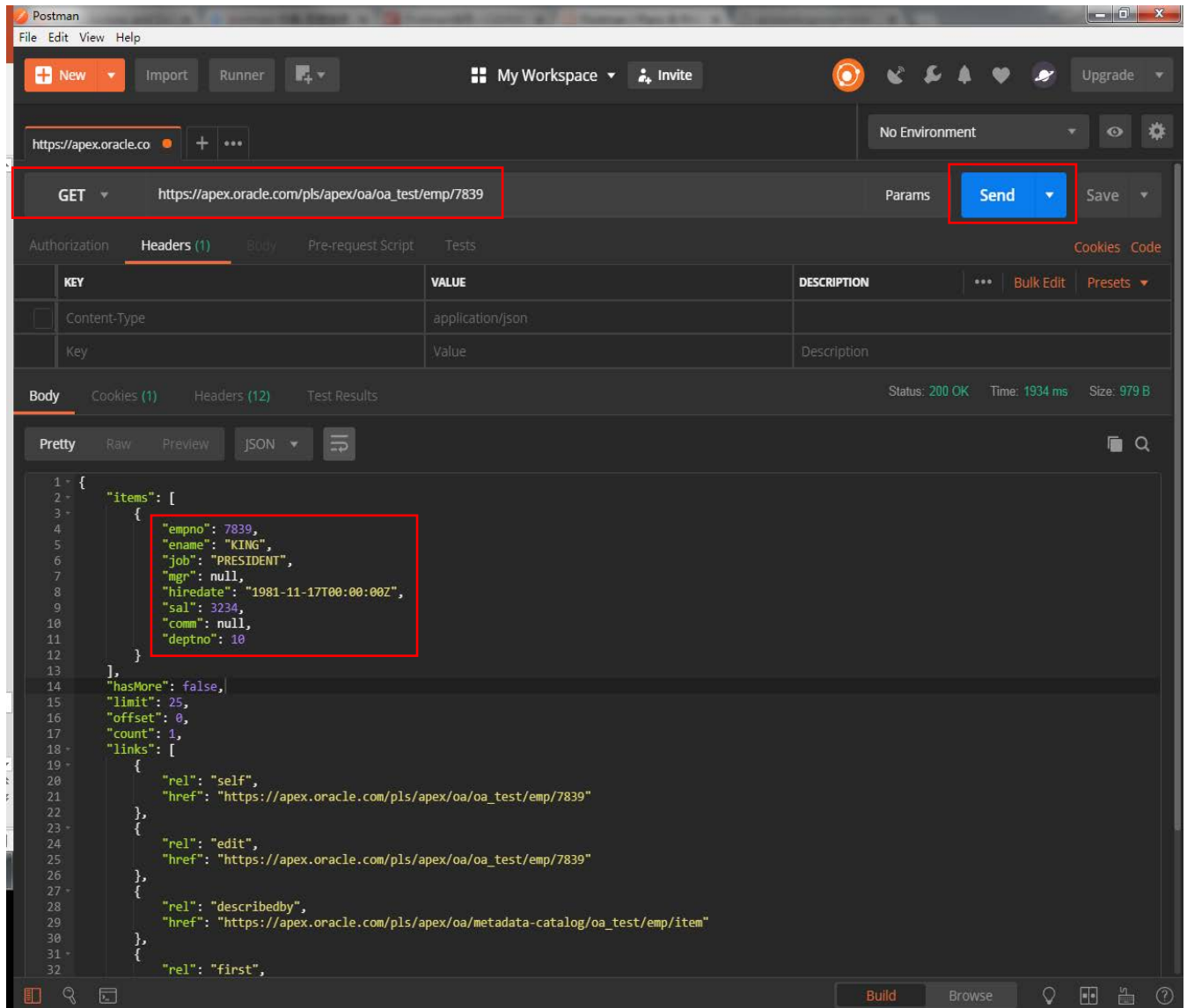
```
1 select * from emp where empno=:empid
```

4. Test the new Web service from the Browser, adding a valid employee number to the end of the URL for your schema. 7839 is the empno for "King".



```
{
  items: [
    {
      empno: 7839,
      ename: "KING",
      job: "PRESIDENT",
      mgr: null,
      hiredate: "1981-11-17T00:00:00Z",
      sal: 3234,
      comm: null,
      deptno: 10
    }
  ],
  hasMore: false,
  limit: 25,
  offset: 0,
  count: 1,
  links: [
    {
      rel: "self",
      href: https://apex.oracle.com/pls/apex/oa/oa_test/emp/7839
    },
    {
      rel: "edit",
      href: https://apex.oracle.com/pls/apex/oa/oa_test/emp/7839
    },
    {
      rel: "describedby",
      href: https://apex.oracle.com/pls/apex/oa/metadata-catalog/oa_test/emp/item
    },
    {
      rel: "first",
      href: https://apex.oracle.com/pls/apex/oa/oa_test/emp/7839
    }
  ]
}
```

5. Test the new Web service from postman.



Section 5

Inserting Data Using RESTful Service

1. In this section, you will create a RESTful Service to insert data using the HTTP Method POST. You will use the postman tool to test your RESTful Service and to verify the results.
2. In the empall Template, create a new POST Handler.

RESTful Service Module **oa_test**

Module Base Path **/oa_test/**

* URI Template ?

Full URL **https://apex.oracle.com/pls/apex/oa/oa_test/empall/**

* Priority ?

* HTTP Entity Tag Type ?

Comments

Resource Handlers Create Handler >

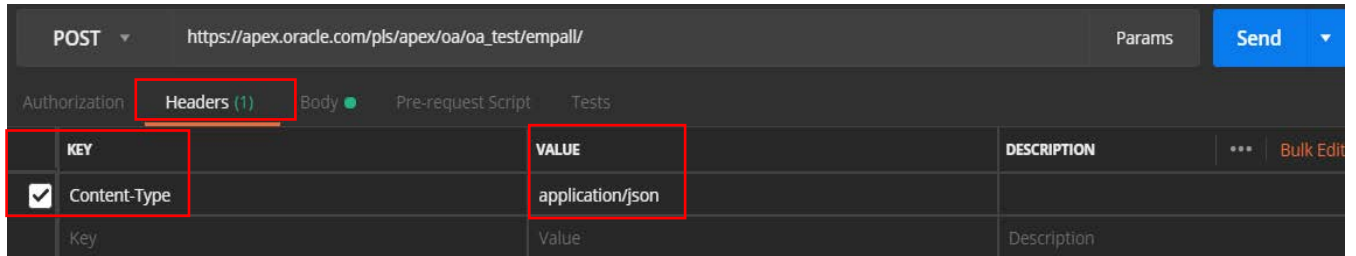
HTTP Method	Items Per Page	Mimes Allowed	Comments	Created On	Updated On
GET	25	-	-	13-AUG-2018 14:23:37	17-AUG-2018 09:07:22

3. Provide the anonymous PL/SQL source code to the POST handler.

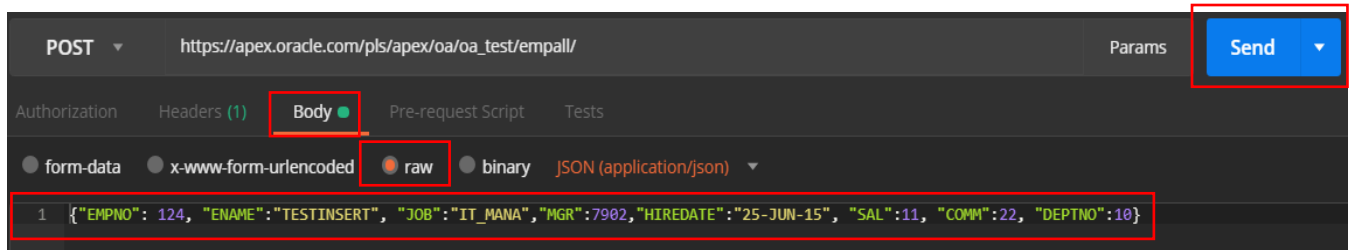
Source ?

```
1 BEGIN
2   INSERT INTO emp (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)
3     VALUES (:EMPNO, :ENAME, :JOB, :MGR, TO_DATE(:HIREDATE, 'DD-MON-RR'), :SAL, :COMM, :DEPTNO);
4   http.prn('employee no ' || :EMPNO || ' added successfully');
5 end;
```

4. In the postman tool to test the POST request, we have to set the Content-type to application/json. Click the Headers tab under the URL bar. Under the heading KEY, enter Content-Type, and for VALUE enter application/json.

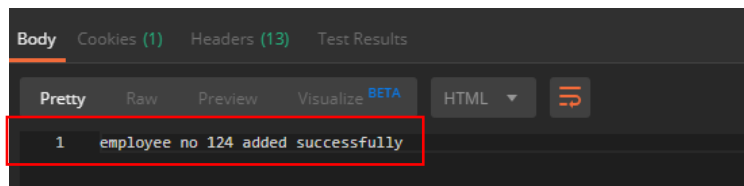


5. In the postman tool to test the POST request, click the Body tab, check raw, and enter the json code shown below. Click Send



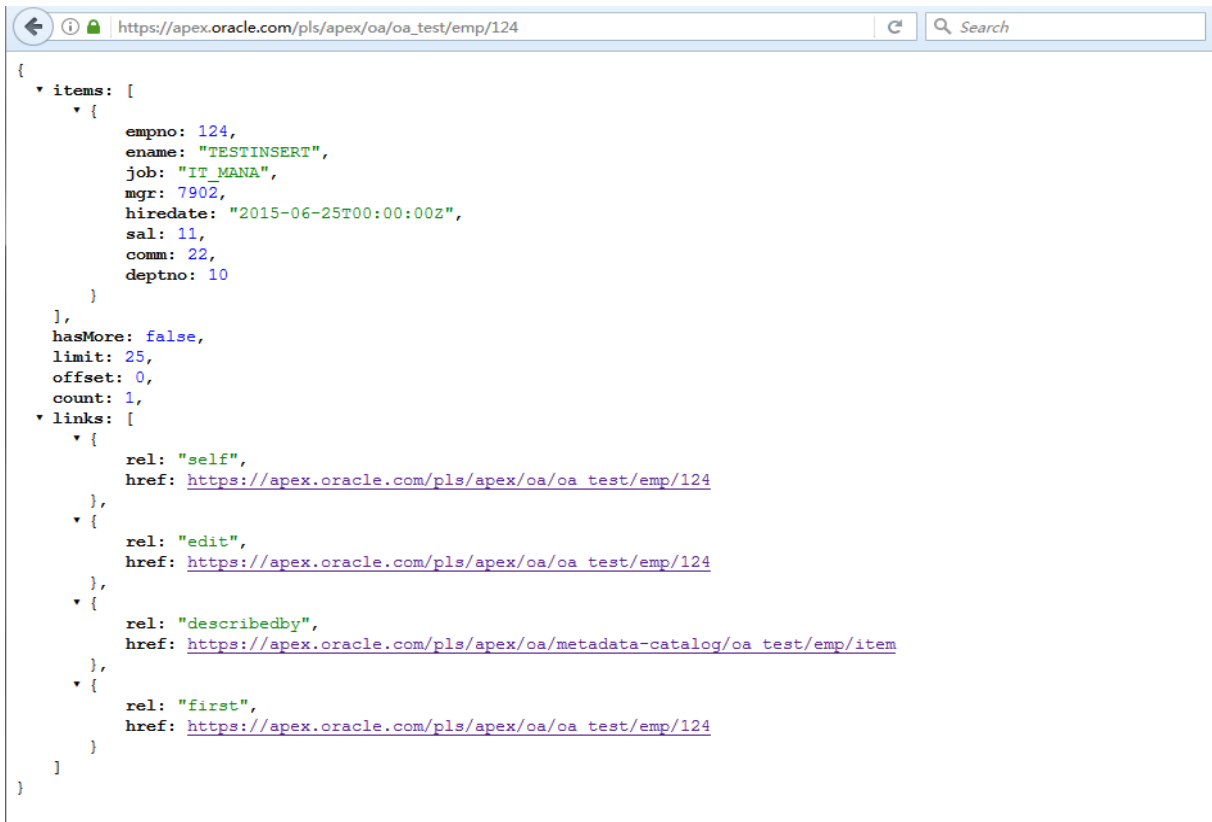
```
{"EMPNO":124, "ENAME":"testinsert", "JOB":"IT_MANA", "MGR":7902, "HIREDATE":"25-JUN-15", "SAL":11, "COMM":22, "DEPTNO":10}
```

If the insert is successful, you will see the message below.



Note that the emp table has a different structure from the employees table included in the Oracle Academy script, and the column sizes are not the same. The "ename" column in the emp table has a maximum size of 10, so if your insert is unsuccessful, ensure that you are not exceeding the maximum number of characters for this column.

6. Check the new inserted record information by using the GET handler created in the previous section, adding 124 to the end of the URL.



The screenshot shows a web browser window with the address bar displaying `https://apex.oracle.com/pls/apex/oa/oa_test/emp/124`. The browser's developer tools are open, showing a REST client response. The response is a JSON object with the following structure:

```
{
  "items": [
    {
      "empno": 124,
      "ename": "TESTINSERT",
      "job": "IT MANA",
      "mgr": 7902,
      "hiredate": "2015-06-25T00:00:00Z",
      "sal": 11,
      "comm": 22,
      "deptno": 10
    }
  ],
  "hasMore": false,
  "limit": 25,
  "offset": 0,
  "count": 1,
  "links": [
    {
      "rel": "self",
      "href": "https://apex.oracle.com/pls/apex/oa/oa_test/emp/124"
    },
    {
      "rel": "edit",
      "href": "https://apex.oracle.com/pls/apex/oa/oa_test/emp/124"
    },
    {
      "rel": "describedby",
      "href": "https://apex.oracle.com/pls/apex/oa/metadata-catalog/oa_test/emp/item"
    },
    {
      "rel": "first",
      "href": "https://apex.oracle.com/pls/apex/oa/oa_test/emp/124"
    }
  ]
}
```

Section 6

Updating Data Using RESTful Service

1. In this section, you will create a RESTful Service to update data using the HTTP Method PUT. You will use the postman tool to test your RESTful Service, and to verify the results.
2. In the emp:/empid Template, create a new PUT Handler.

Resource Handlers						Create Handler >
<input type="text"/> <input type="button" value="Go"/> <input type="button" value="Actions"/>						
HTTP Method	Items Per Page	Mimes Allowed	Comments	Created On	Updated On	
DELETE	-	-	-	17-AUG-2018 13:10:47	17-AUG-2018 13:12:24	
GET	-	-	-	17-AUG-2018 13:06:52	17-AUG-2018 13:41:32	

3. Specify the PUT Method and Source Type as PL/SQL. Provide the Source Code for the PL/SQL.

* Method **PUT** ?

Source Type **PL/SQL**

Mime Types Allowed

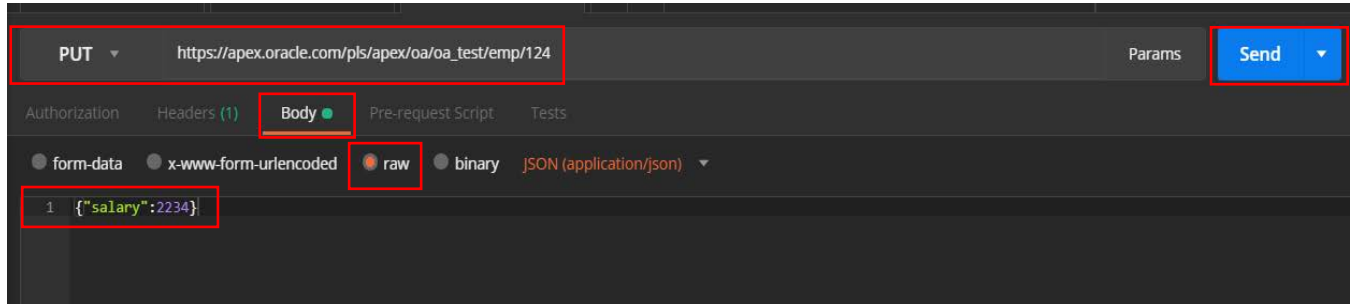
Comments

Source

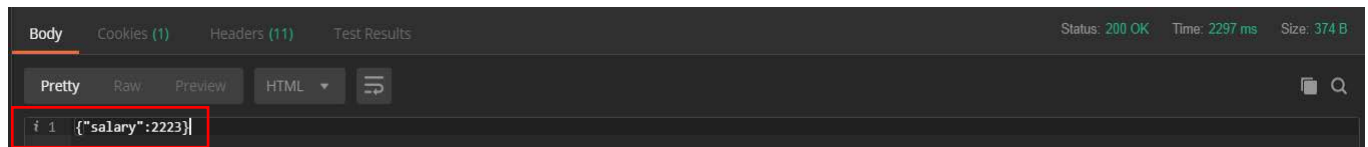
↶ ↷ 🔍 ↔ A

```
1 DECLARE
2   l_oldsalary NUMBER;
3   l_salarychange NUMBER;
4 BEGIN
5   SELECT NVL(e.sal, 0) INTO l_oldsalary
6   FROM emp e
7   WHERE e.empno = :empid;
8   UPDATE emp e
9   SET e.sal = NVL(:salary, e.sal)
10  WHERE e.empno = :empid;
11  l_salarychange:=NVL(:salary, l_oldsalary)-l_oldsalary;
12  http.prn('{'||'"salary":'||l_salarychange||' '}');
13 END;
```

4. Test in Postman, select PUT, enter the correct URL adding /124, click the Body tab, check raw, and enter the json code shown below. Click Send.



If the update was successful, you will see the message below.



Section 7

Deleting Data Using RESTful Service

1. In this section, you will create a RESTful Service to delete data using the HTTP Method DELETE. You will use the postman tool to test your RESTful Service and to verify the results.
2. In the emp:/empid Template, create a new DELETE Handler.


Resource Handlers						Create Handler >
<input type="text" value="Q"/> <input type="button" value="Go"/> <input type="button" value="Actions"/>						
HTTP Method	Items Per Page	Mimes Allowed	Comments	Created On	Updated On	
PUT	-	-	-	15-AUG-2018 14:40:32	17-AUG-2018 04:23:20	


3. Specify the DELETE Method, and Source Type as PL/SQL. Provide the Source Code for the PL/SQL.

RESTful Service Module **oa_test**

Module Base Path **/oa_test/**

URI Template **emp:/empid**

Full URL **https://apex.oracle.com/pls/apex/oa/oa_test/emp:/empid** 

* Method **DELETE** 

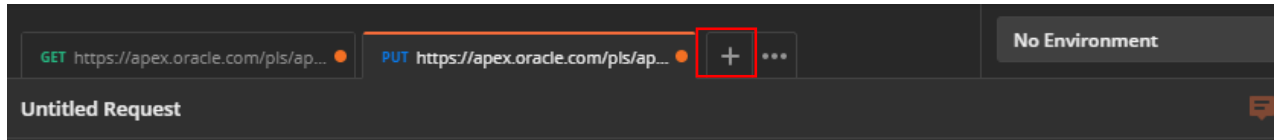
Source Type **PL/SQL**

Comments

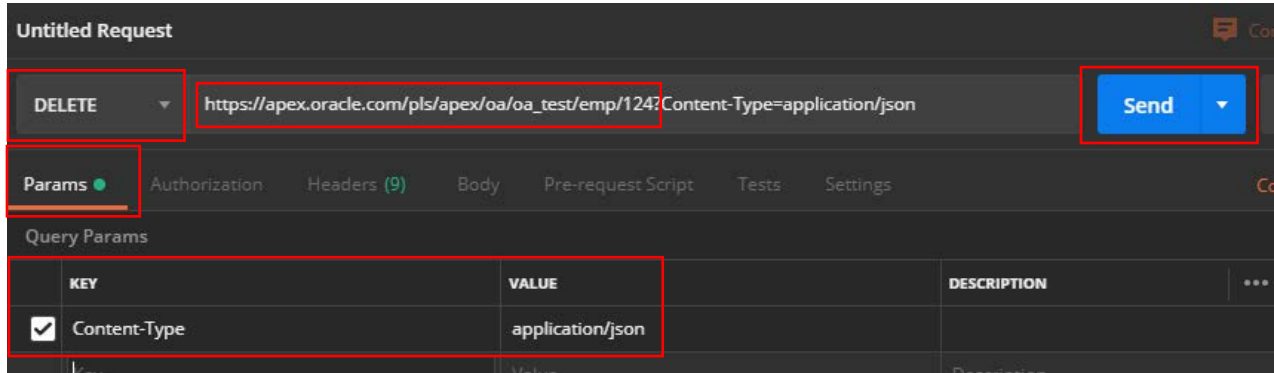
Source

```
1 BEGIN
2   DELETE FROM emp WHERE empno = :empid;
3   http.prn('employee no: ' || :empid || ' deleted successfully');
4 END;
```

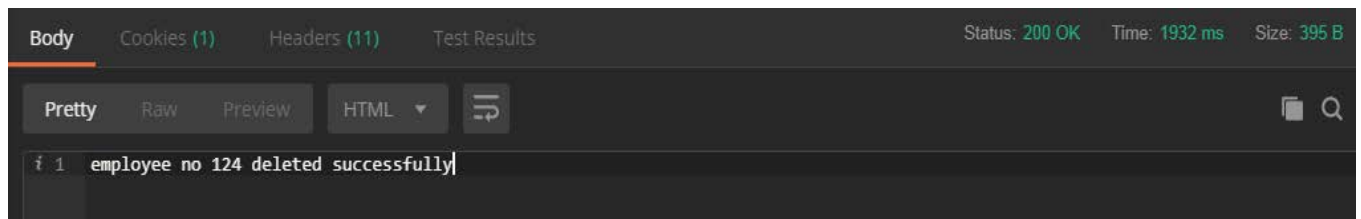
- To test in Postman, click the plus icon to create a new tab.



- Select DELETE, enter the correct URL adding /124. Click the Params tab. For KEY enter Content-Type, and for VALUE enter application/json (note that this information will automatically be added to the end of the URL). Click Send.



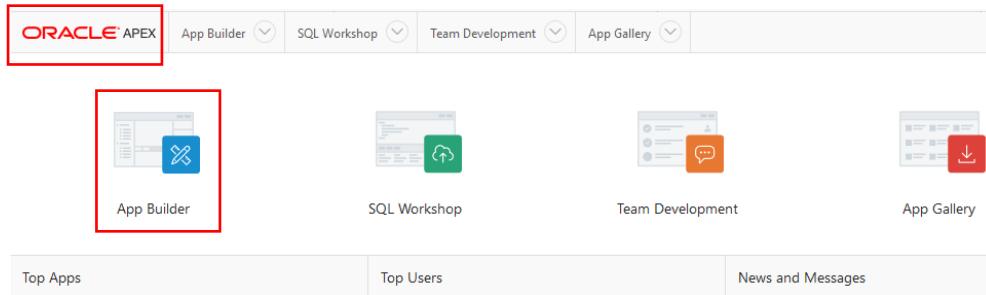
If the delete was successful, you will see the message below.



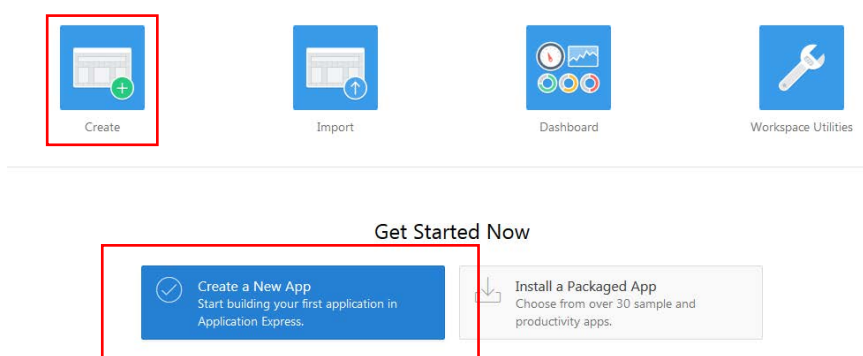
Section 8

Create an APEX Application Using RESTful service

1. In this section, you will create a simple application using the Application Builder component of APEX to display the employees in the emps table. Click the Oracle logo to go to the APEX home page, then click App Builder.



2. Create a simple application with home page. Click Create, Create a New App



3. Create the EMP DISPLAY application. Add the name: EMP DISPLAY, accept all other default values and click Create application.

Create an Application

Name
EMP DISPLAY

Appearance
Vita, Side Menu

Pages

+ Add Page

Home

Blank

Edit

Features

Check All

☒ About Page
Add about this application page

☒ Access Control
Enable role-based user authorization

☒ Activity Reporting
Include user activity and error reports

☒ Configuration Options
Enable or disable application features

☒ Feedback
Allow users to provide feedback

☒ Theme Style Selection
Update default application look and feel

Settings

Application ID
35041

Schema
QA

Authentication
Application Express Accounts

Language
English (en)

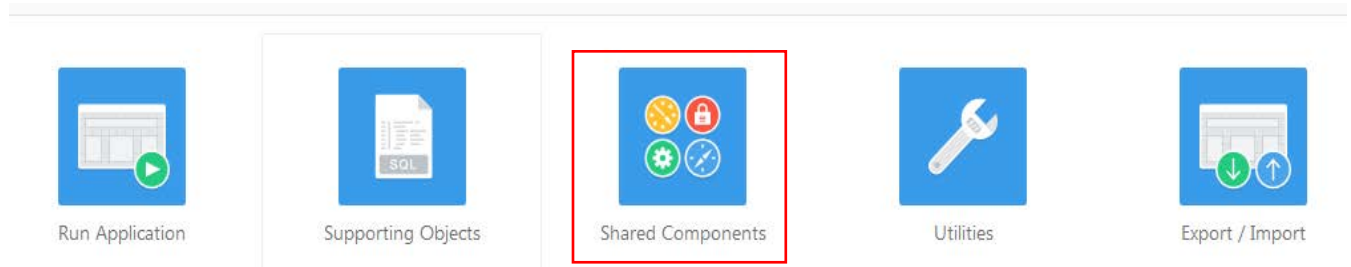
Advanced Settings

User Interface Defaults

Cancel

Create Application

4. Choose Shared Components.



5. In the Data Sources section, choose Web Source Modules.

Application Logic	Security	Other Components
Application Definition Attributes Application Items Application Processes Application Computations Application Settings Build Options	Security Attributes Authentication Schemes Authorization Schemes Application Access Control Session State Protection Web Credentials >	List of Values Plug-ins Component Settings Shortcuts Email Templates
Navigation	User Interface	Files
Lists Navigation Menu Breadcrumbs Navigation Bar List	User Interface Attributes Themes Templates	Static Application Files Static Workspace Files
Data Sources	Reports	Globalization
Data Load Definitions REST Enabled SQL > Web Source Modules Legacy Web Service References	Report Queries Report Layouts	Globalization Attributes Text Messages Translate Application

- Click the create button, select from scratch and click Next

Web Source Modules Utilization History

Go Actions

Create >

Create Web Source Module - Method

Method

When you create a new web source module, you have two options. You can create a new web source module from scratch or you reuse implementations that already exist in your application or in other applications within your workspace.

Create Web Source Module: ☒ From scratch ☐ As copy of an existing web source module

Cancel Next >

- Add the name and URL for your empall module created earlier, click next

Create Web Source

General

Web Source Type Simple HTTP

* Name APEX REST EMPALL Service

* URL Endpoint https://apex.oracle.com/pls/apex/oa/oa_test/empall/

< Next >

8. The next step will automatically split up the endpoint URL into a *server-specific* and a *service-specific* part. Click Next

Create Web Source - Remote Server

✓

●

●

●

Remote Server

Remote Server - Create New - ?

* Base URL https://apex.oracle.com/pls/apex/ ?

* Service URL Path /oa/oa_test/empall/ ?

< Next >

9. For authentication select No, click Discover

Authentication

✓

✓

●

●

Authentication

* Authentication Required Yes No ?

< Create Module Manually Advanced > Discover >

10. Employees from the emps table will be discovered. Click Create Web Source.

Web Source Discovery

x

✓

✓

✓

 Preview

Data

Data Profile

Empno ↑	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7369	SMITH	CLERK	7902	17-DEC-80 12.00.00.0000000000 AM +00:00	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81 12.00.00.0000000000 AM +00:00	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81 12.00.00.0000000000 AM +00:00	1250	500	30

<

More Detail

Create Web Source

11. Return to the application home page by clicking the Application number under the Oracle APEX logo (Your application number will be different), and click the Create Page button in order to add a new page.

ORACLE[®] APEX App Builder SQL Workshop Team Development

Application 91911 Shared Components Web Source Modules

✓ Web Source Module created.

Q

Go

Actions

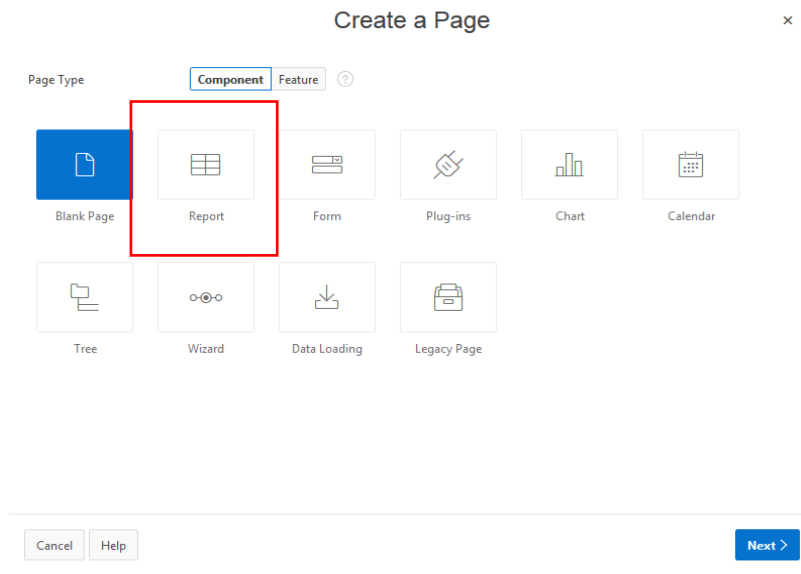
Create Page >

0 - Global Page - Desktop

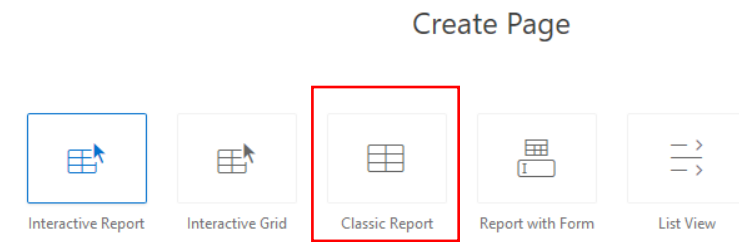
1 - Home

9999 - Login Page

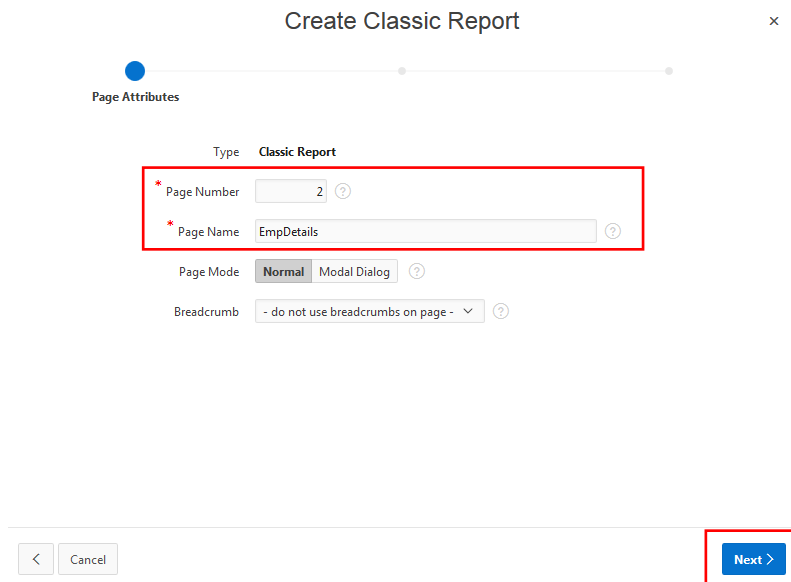
12. Click Report.



13. Choose Classic Report.



14. Enter Page Number 2, and Page Name EmpDetails. Click Next



14. Navigation Menu. Select Create a new Navigation menu entry, add EmpDetails as the name, and select Home as Parent Navigation Menu Entry. Click Next

Create Classic Report

Navigation Menu

Navigation Preference

☐ Do not associate this page with a navigation menu entry

☒ Create a new navigation menu entry

☐ Identify an existing navigation menu entry for this page

* New Navigation Menu Entry

EmpDetails

Parent Navigation Menu Entry

No parent selected -

Home

< Cancel

Next >

15. For Data Source, choose Web Source, select the APEX REST EMPALL service. Click Create.

Create Classic Report

Report Source

Data Source

Local Database

REST Enabled SQL Service

Web Source

Web Source Module

APEX REST EMPALL Service

* Select Columns

JOB (Varchar2)

MGR (Number)

SAL (Number)

COMM (Number)

EMPNO (Number)

ENAME (Varchar2)

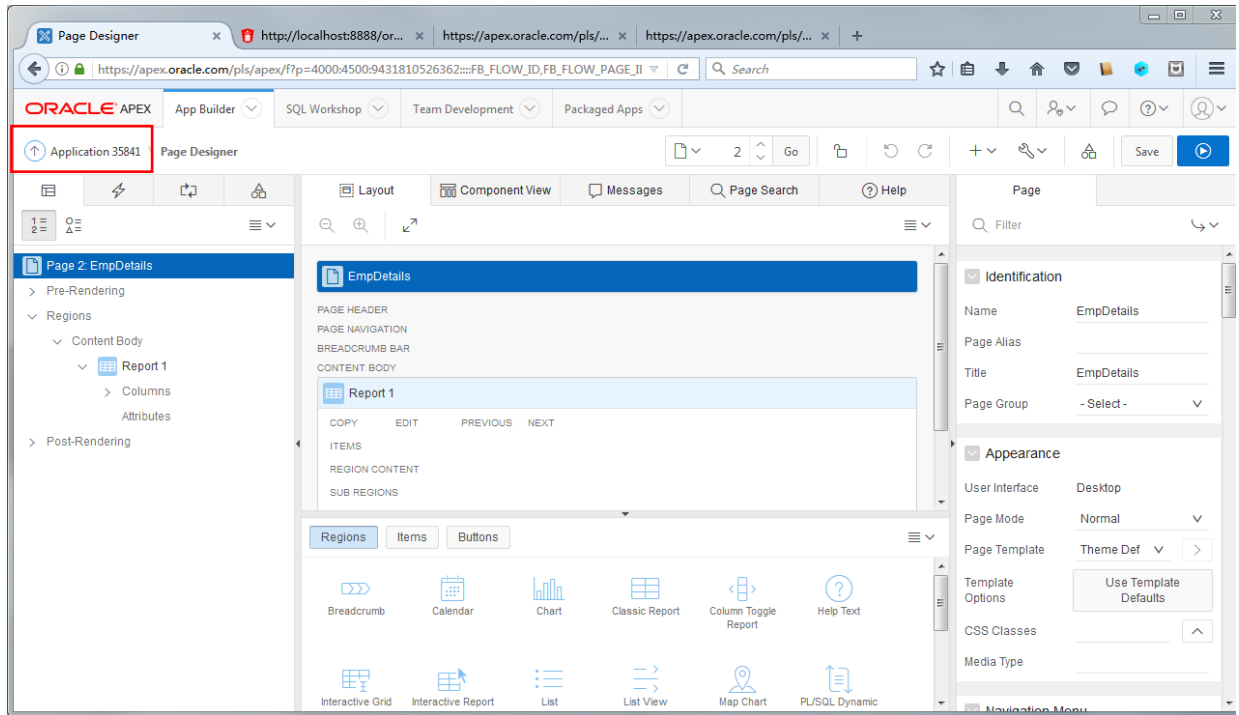
DEPTNO (Number)

HIREDATE (Timestamp_Tz)

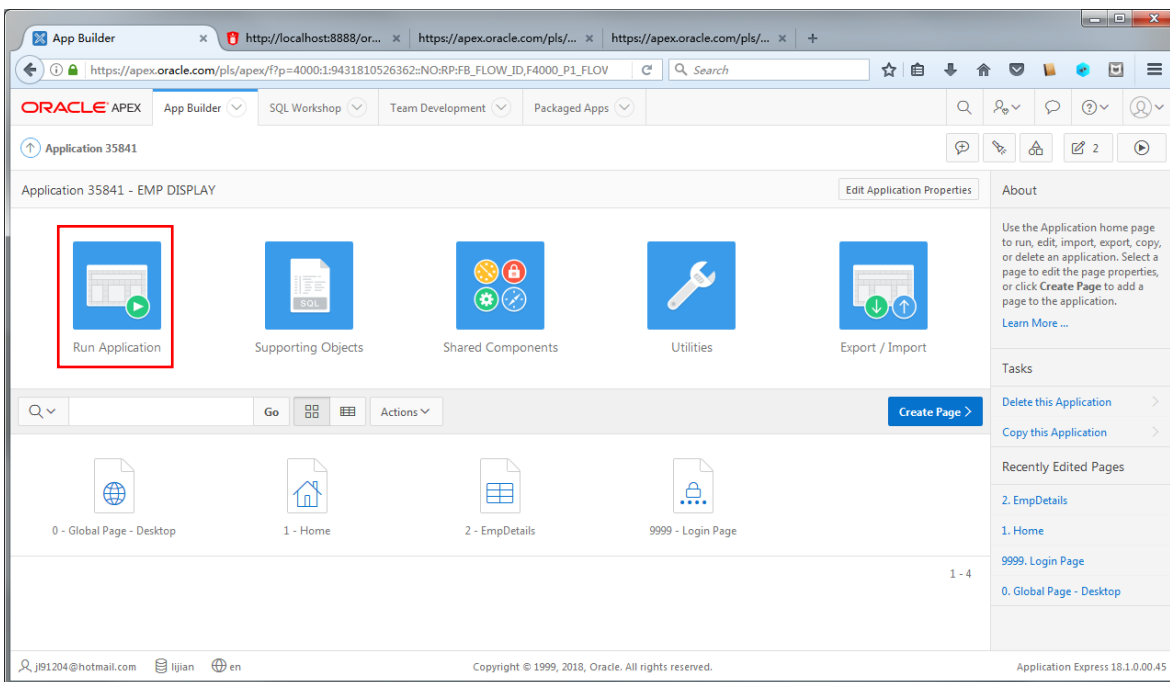
< Cancel

Create

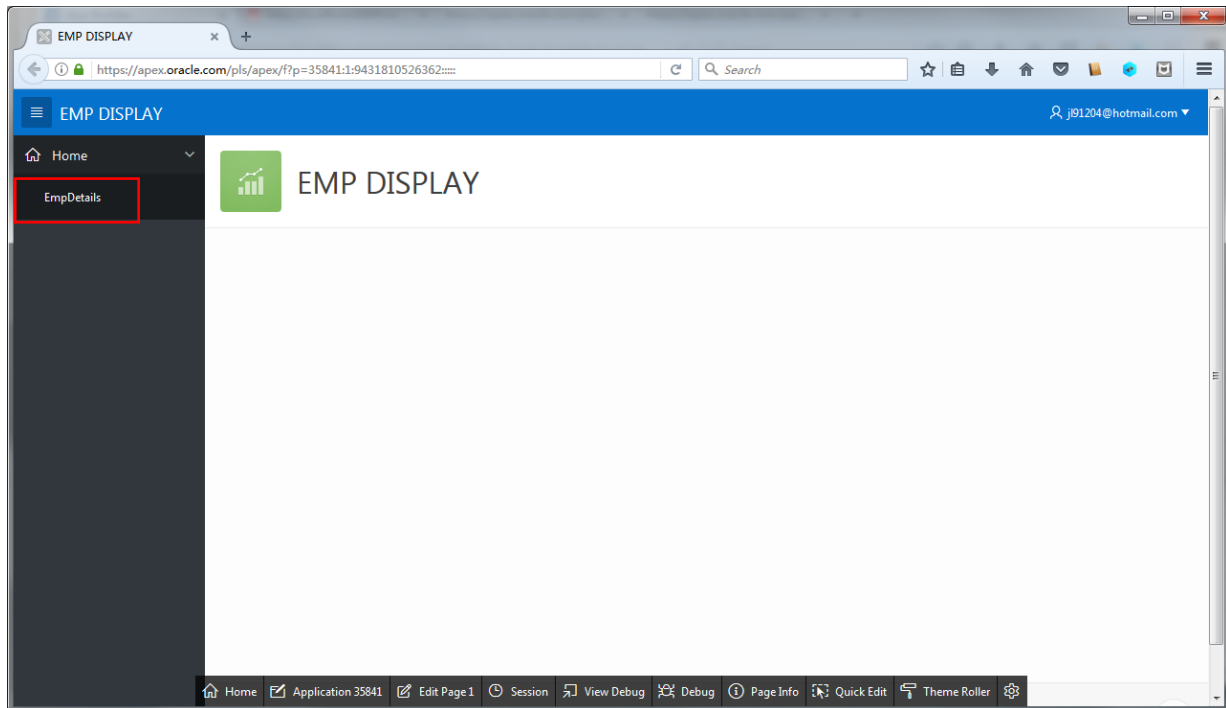
16. The page design appears. Click the application number to return to the application Home page.



17. Click the Run Application icon.



18. Click the EmpDetails menu item link.



19. The employees in the database table emp are displayed in the application using the REST Web Service.

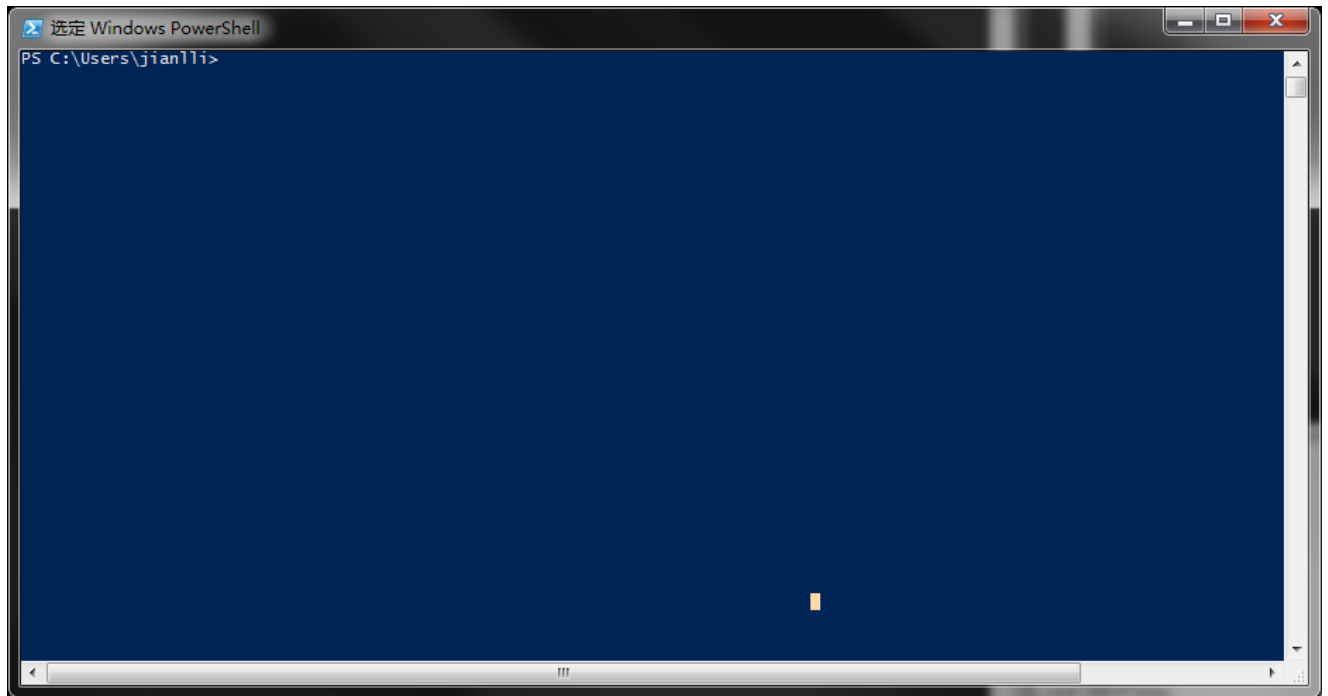
The screenshot shows the EMP DISPLAY application with the 'EmpDetails' page active. The main content area displays a table titled 'Report 1' containing employee data. The table has the following columns: Empno, Ename, Job, Mgr, Hiredate, Sal, Comm, and Deptno. The data is as follows:

Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7369	SMITH	CLERK	7902	17-DEC-1980	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-1981	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-1981	1250	500	30
7566	JONES	MANAGER	7839	02-APR-1981	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-1981	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-1981	2850		30
7782	CLARK	MANAGER	7839	09-JUN-1981	2450		10
7788	SCOTT	ANALYST	7566	09-DEC-1982	3000		20
7839	KING	PRESIDENT		17-NOV-1981	3234		10
7844	TURNER	SALESMAN	7698	08-SEP-1981	1500	0	30
7876	ADAMS	CLERK	7788	12-JAN-1983	1100		20
7900	JAMES	CLERK	7698	03-DEC-1981	950		30
7902	FORD	ANALYST	7566	03-DEC-1981	3000		20

Section 9

Consuming the RESTful Web Service Created in Application Express Using a Java Client

1. Open the Command Prompt.



2. Create a Java Source file RESTemp.java to retrieve the empno 7782 details. Be sure to change the URL to that of your own schema. Alternatively, you could create, compile and run the java file in an IDE such as Eclipse or Netbeans.

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

public class RestMain {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        connectRest();
    }

    public static void connectRest() {
        try {
            //***** Enter the URL of your own schema *****
            String uri = "https://apex.oracle.com/pls/apex/oa/oa_test/emp/7782";
            URL url = new URL(uri);
            HttpURLConnection connection
                = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("GET");
            connection.setRequestProperty("Accept", "application/json");
            readEmp(connection);
        } catch (Exception e) {
            System.out.println("Could not establish the connection");
        }
    }

    public static void readEmp(HttpURLConnection connection) {
        try {
```

```

InputStream json = connection.getInputStream();
InputStreamReader isr = new InputStreamReader(json);
BufferedReader br = new BufferedReader(isr);
String line;
StringBuilder sb = new StringBuilder();
while ((line = br.readLine()) != null) {
    sb.append(line);
}
System.out.println(sb);
} catch (Exception e) {
    System.out.println("Can not read the json object");
}
}
}

```

3. Compile the RESTEmp.java file.

```

PS C:\Users\jianli> java RESTEmp
{"items":[{"empno":7782,"ename":"CLARK","job":"MANAGER","mgr":7839,"hiredate":"1981-06-09T00:00:00Z","sal":2450,"comm":null}]}

```

4. Execute the RESTEmp class and the data for empno 7782 will be displayed.

Congratulations! You have completed **Using RESTful Web Service in Application Express** Lab successfully.