Oracle Cloud Infrastructure Labs

Oracle OCI Oracle Rest Data Services Lab

V2.0

ORACLE LAB BOOK | JUNE 2018

By
Oracle Sales Consulting Norway
Inge Os

1. Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

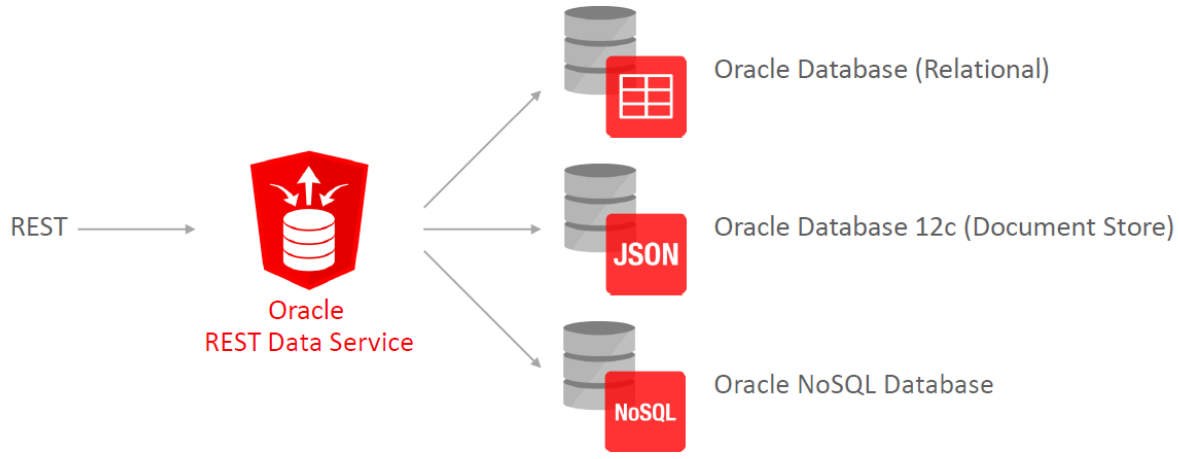
Overview

Lab Overview

Oracle REST Data Services (ORDS) is a powerful tool that enables developers with database skills to build enterprise class, data access APIs to Oracle Databases that today's modern, state-of-the-art application developers want to use, and indeed increasingly demand to use, to build applications.
The data access APIs can fully exploit all the power of Oracle database technology to deliver the highest levels of reliability, security, and performance. Application developers can use these data access APIs to build enterprise class applications employing the same methods and techniques that have become dominant in the startup world for over a decade, at universities where new developers are trained, at the leading web companies like Google, Facebook, and many more, and the most advanced mainstream enterprise IT departments.

The labs are intended to give a brief first touch on Oracle REST Data Services, ORDS. The goal is to show a simple, non HA, setup of the ORDS rest server, and show how data can be exposes. Prerequisites to the lab is a running cloud or VM with Oracle RDBMS and Java 8 installed, with port openings for port 1521 and port 80. In addition it is strongly recommended to use Oracle SQL Developer 18 as it simplified the ORDS administration on the database side. In the LAB examples from a VM hosted in Oracle OCI Cloud will be used.

The lab assumes that a 12c database with a proper schema is available. Throughout the documentations a user ORDSLAB prepopulated with the classic demobld.sql (emp/dept) is used as an example.



Oracle ORDS architecture

Introduction to Oracle REST Services, Oracle ORDS

Oracle ORDS is an excellent way of exposing existing PL/SQL and SQL Data from an Oracle 11g, 12c or 18 database as REST services. The labs are simple labs to give a first touch, but the more advanced features can easy be exposed from the Oracle By Example or learning library. For more in depth information please refer to:

https://www.oracle.com/database/technologies/appdev/rest.html

https://apexapps.oracle.com/pls/apex/f?p=44785:24:3676186256130:::24:P24_CONTENT_ID,P24_PREV_PAGE:13282,24

https://docs.oracle.com/database/ords-18.1/

ORDS requires the following activities:

- Define deployment option of the ORDS java REST service
- Download, unzip ORDS and verify the environment
- Configure the ORDS rest service to communicate with the database instance
- Configure ORDS metadata in the database for by REST enabling a schema
- Configure REST enabled schema objects.

Define deployment option of the ORDS java REST service

The ORDS REST driver can be deployed in a number of ways like:

- Run as a Web application histed on WebLogic, and benefit from WebLogic Clustering
- Run with other J2EE or HTTP/Servlet engine like GlasFish or jetty
- Standalone java SE program. Simple deployment for test/dev

In the lab we will use the standalone model,

Download and install ords.war. Download from the following page:

http://www.oracle.com/technetwork/developer-tools/rest-data-services/downloads/index.html

As of writing of the lab the download version is ords.18.1.1.95.1251.zip

Create a directory for holding the ORDS executable and ORDS configuration. Unzip the ORDS download into this directory.

```
[oracle@ordslab ~]$ mkdir ords
[oracle@ordslab ~]$ cd ords
[oracle@ordslab ords]$ unzip ../ords.18.1.1.95.1251.zip
Archive:  ../ords.18.1.1.95.1251.zip
   creating: logs/
   creating: docs/
   creating: params/
   creating: examples/
   creating: examples/soda/
   creating: examples/soda/getting-started/
   creating: examples/db_auth/
   creating: examples/db_auth/sql/
   creating: examples/plugins/
```

When the download is completed and the zipfile is stored in the HOME directory for the user do the following:

- Verify java version
- Verify Database username/password for dba
- Verify listerner and connection properties

Java version verification

```
[oracle@ordslab ~]$ java -version
java version "1.8.0_131"
Java(TM) SE Runtime Environment (build 1.8.0_131-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.131-b11, mixed mode)
[oracle@ordslab ~]$
```

Database username/password verification

```
[oracle@ordslab ~]$ sqlplus system@eusdb

SQL*Plus: Release 12.2.0.1.0 Production on Wed Jun 27 13:41:35 2018

Copyright (c) 1982, 2016, Oracle.  All rights reserved.

Enter password:
Last Successful login time: Wed Jun 27 2018 13:40:34 +00:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL> exit
```

Listener verification

```
[oracle@ordslab ~]$ lsnrctl status

LSNRCTL for Linux: Version 12.2.0.1.0 - Production on 27-JUN-2018 13:58:42

Copyright (c) 1991, 2016, Oracle.  All rights reserved.

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.2.0.1.0 - Production
Start Date                27-JUN-2018 13:29:31
Uptime                    0 days 0 hr. 29 min. 10 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/home/oracle/app/oracle/product/12.2.0/dbhome_1/network/admin/listener.ora
Listener Log File
/home/oracle/app/oracle/diag/tnslsnr/ordslab/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=127.0.0.1)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=ordslab.xxxxxxxx.oraclevcn.com)(PORT=55
00))(Security=(my_wallet_directory=/home/oracle/app/oracle/admin/eusdb/xdb_wallet)
)(Presentation=HTTP)(Session=RAW))
Services Summary...
Service "eusdb.oraclevcn.com" has 1 instance(s).
  Instance "eusdb", status READY, has 1 handler(s) for this service...
Service "eusdbXDB.oraclevcn.com" has 1 instance(s).
```

```
  Instance "eusdb", status READY, has 1 handler(s) for this service...
The command completed successfully
```

Configure the ORDS rest service to communicate with the database instance

The ords.war act both as a CLI/config tool for ORDS and as well as running the standalone ORDS server.

To start the configuration do the following:

java -jar ords.war install

This will require the following questions to be answered:

Enter the location to store configuration data:**/home/oracle/ords**

Enter the name of the database server [localhost]:**localhost**

Enter the database listen port [1521]:**1521**

Enter 1 to specify the database service name, or 2 to specify the database SID [1]:**2**

Enter the database SID [xe]:e**usdb**

Enter the database password for ORDS_PUBLIC_USER: **xxxx**

Confirm password:**xxxx**

Enter the password for SYS AS SYDBA: **xxxx**

If using Oracle Application Express or migrating from mod_plsql then you must enter 1 [1]:**2**

Enter 1 if you wish to start in standalone mode or 2 to exit [1]:**1**

Enter 1 if using HTTP or 2 if using HTTPS [1]:**1**

This will start the ORDS service immediately after it is configured.

To stop the service press ^C. to start just run java –jar ords.war

```
[oracle@ordslab ords]$ java -jar ords.war install

This Oracle REST Data Services instance has not yet been configured.

Please complete the following prompts:

Enter the location to store configuration data:/home/oracle/ords
Enter the name of the database server [localhost]:
Enter the database listen port [1521]:
Enter 1 to specify the database service name, or 2 to specify the database SID
[1]:2
Enter the database SID [xe]:eusdb
Enter the database password for ORDS_PUBLIC_USER:

Confirm password:
```

```
Enter the database password for SYS AS SYDBA: xxxx

Confirm password:xxxx


Retrieving information.
Enter 1 if you want to use PL/SQL Gateway or 2 to skip this step.
If using Oracle Application Express or migrating from mod_plsql then you must
enter 1 [1]:2
Jun 27, 2018 2:10:08 PM
INFO: Updated configurations: defaults, apex_pu
Jun 27, 2018 2:10:08 PM oracle.dbtools.rt.config.setup.SchemaSetup install
INFO: Oracle REST Data Services schema version 18.1.1.95.1251 is installed.
Enter 1 if you wish to start in standalone mode or 2 to exit [1]:
Enter 1 if using HTTP or 2 if using HTTPS [1]:1
2018-06-27 14:10:15.738:INFO::main: Logging initialized @68275ms to
org.eclipse.jetty.util.log.StdErrLog
Jun 27, 2018 2:10:15 PM
INFO: HTTP and HTTP/2 cleartext listening on port: 8080
Jun 27, 2018 2:10:15 PM
INFO: Disabling document root because the specified folder does not exist:
/home/oracle/ords/ords/standalone/doc_root
2018-06-27 14:10:16.341:INFO:oejs.Server:main: jetty-9.4.z-SNAPSHOT, build
timestamp: 2017-11-21T21:27:37Z, git hash:
82b8fb23f757335bb3329d540ce37a2a2615f0a8
2018-06-27 14:10:16.421:INFO:oejs.session:main: DefaultSessionIdManager
workerName=node0

2018-06-27 14:10:16.421:INFO:oejs.session:main: No SessionScavenger set, using
defaults

2018-06-27 14:10:16.422:INFO:oejs.session:main: Scavengi
```

The ORDS REST Sever is now configured and running

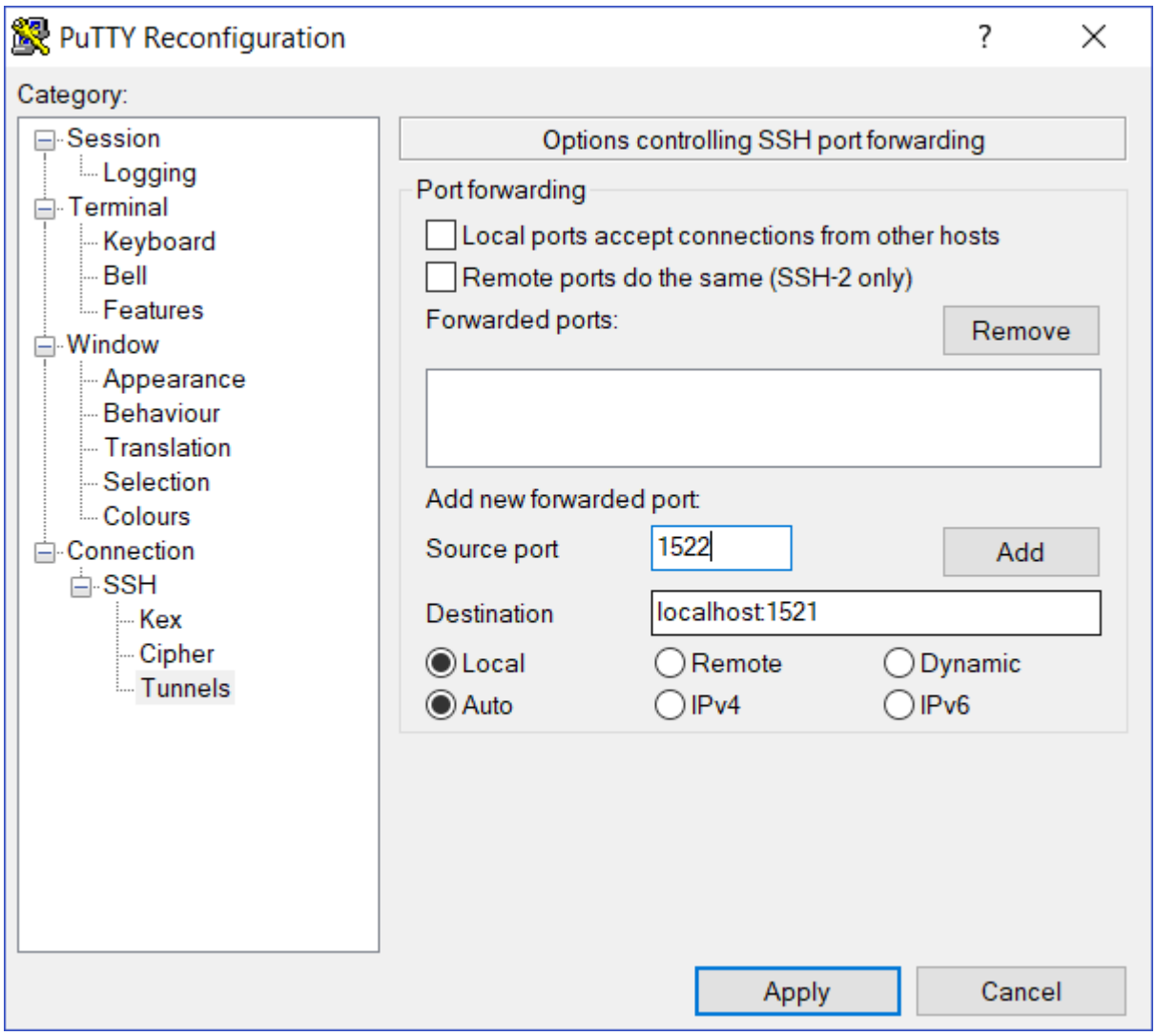
Configure ORDS metadata in the database for REST enabling

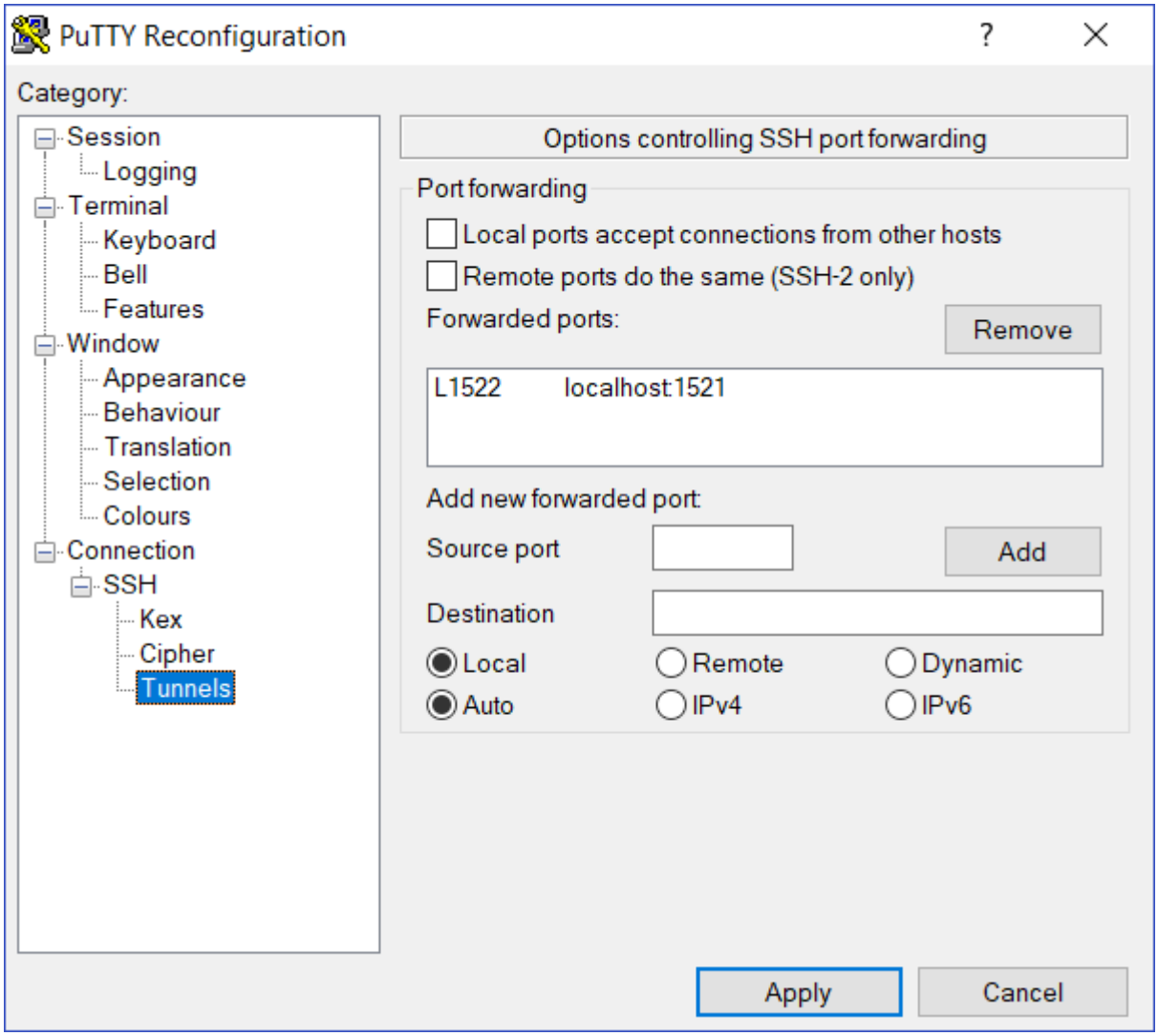
If the configuration is done above with ords.war –install the proper metadata is added to the database.

REST enable schema data in the database.

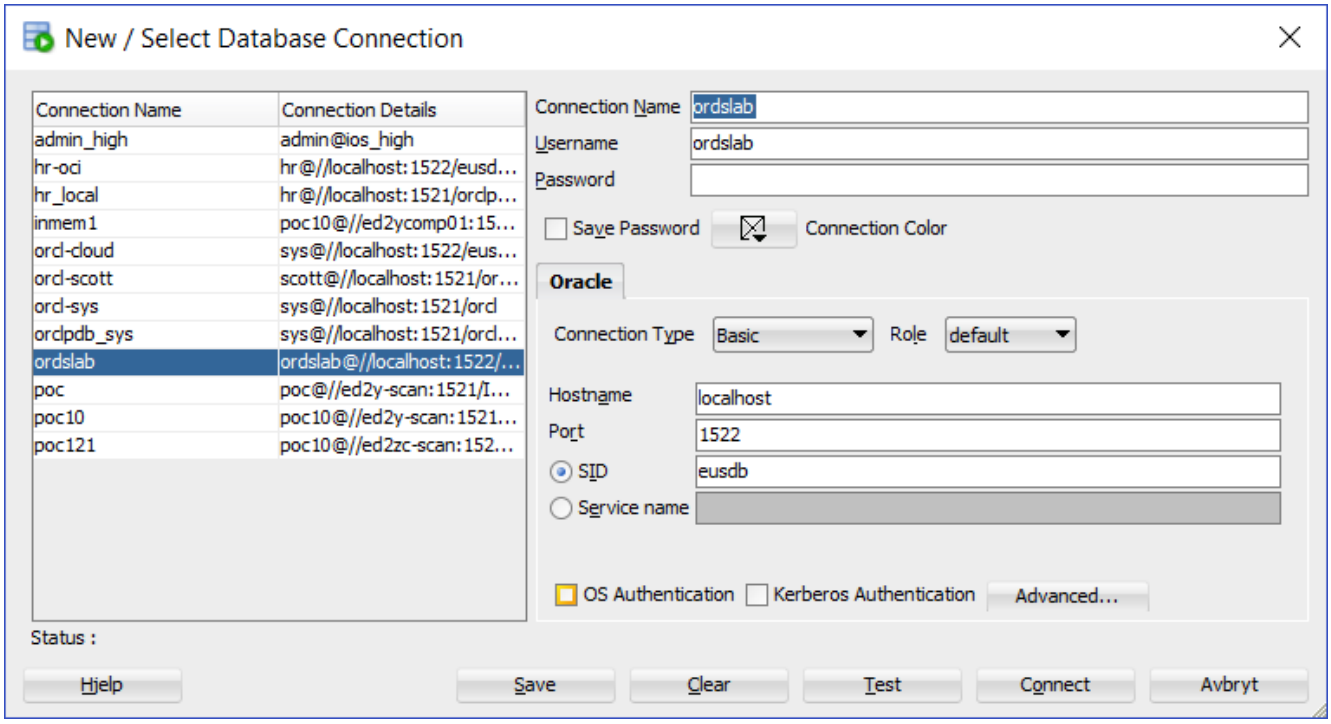
The easiest way to perform this step is to set up a localhost ssh tunnel from the local sqldeveloper installation.

Configure the localhost tunnel as follows: (tunneling via local port 1522 to avoid conflict with local installed Oracle DB)





Then create the sqldeveloper 18c sql connection to the database as the user that will expose services with ORDS, in the lab case the ORDSLAB user:



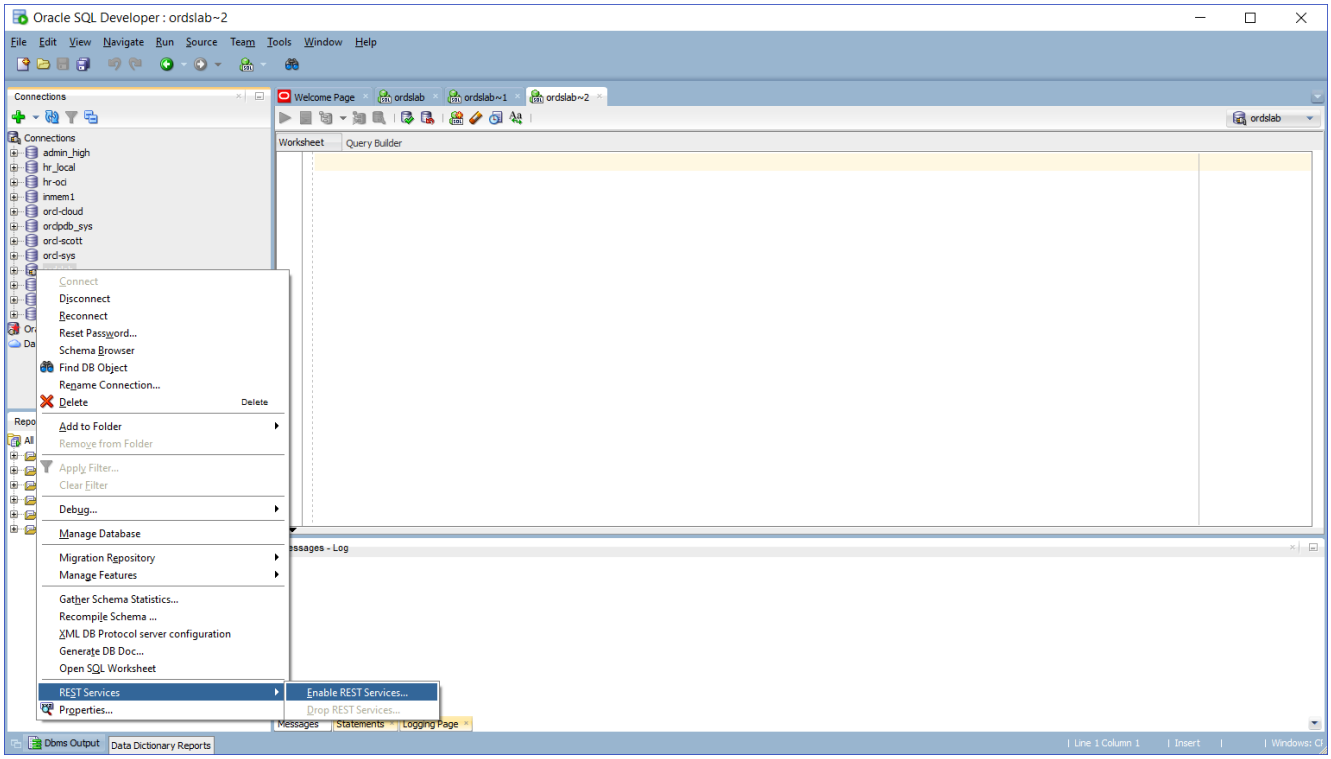
(Note localhost and port reflect the SSH tunnel above)

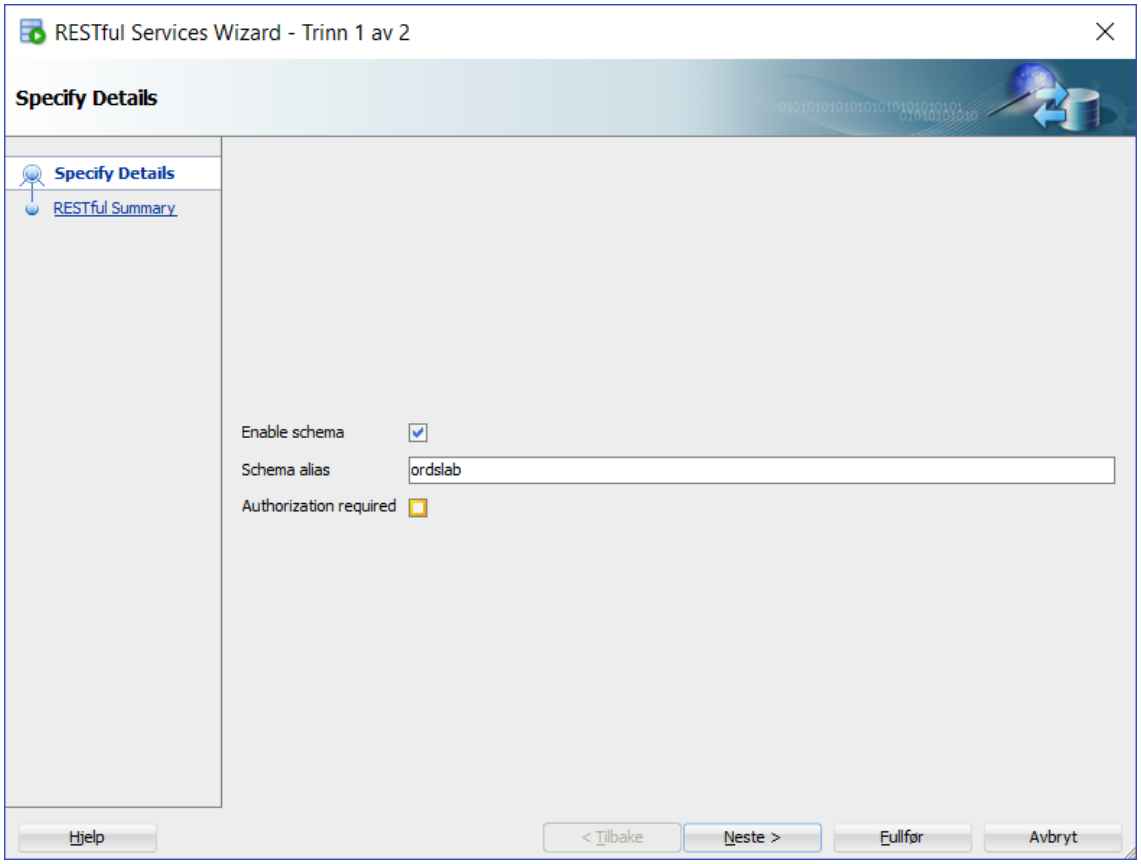
The Actual schema needs to be enabled for ORDS, and add the schema metadata to ORDS metadata. This is either done by running the following SQL interactively, signed on as the schema owner, in this case ORDSLAB:

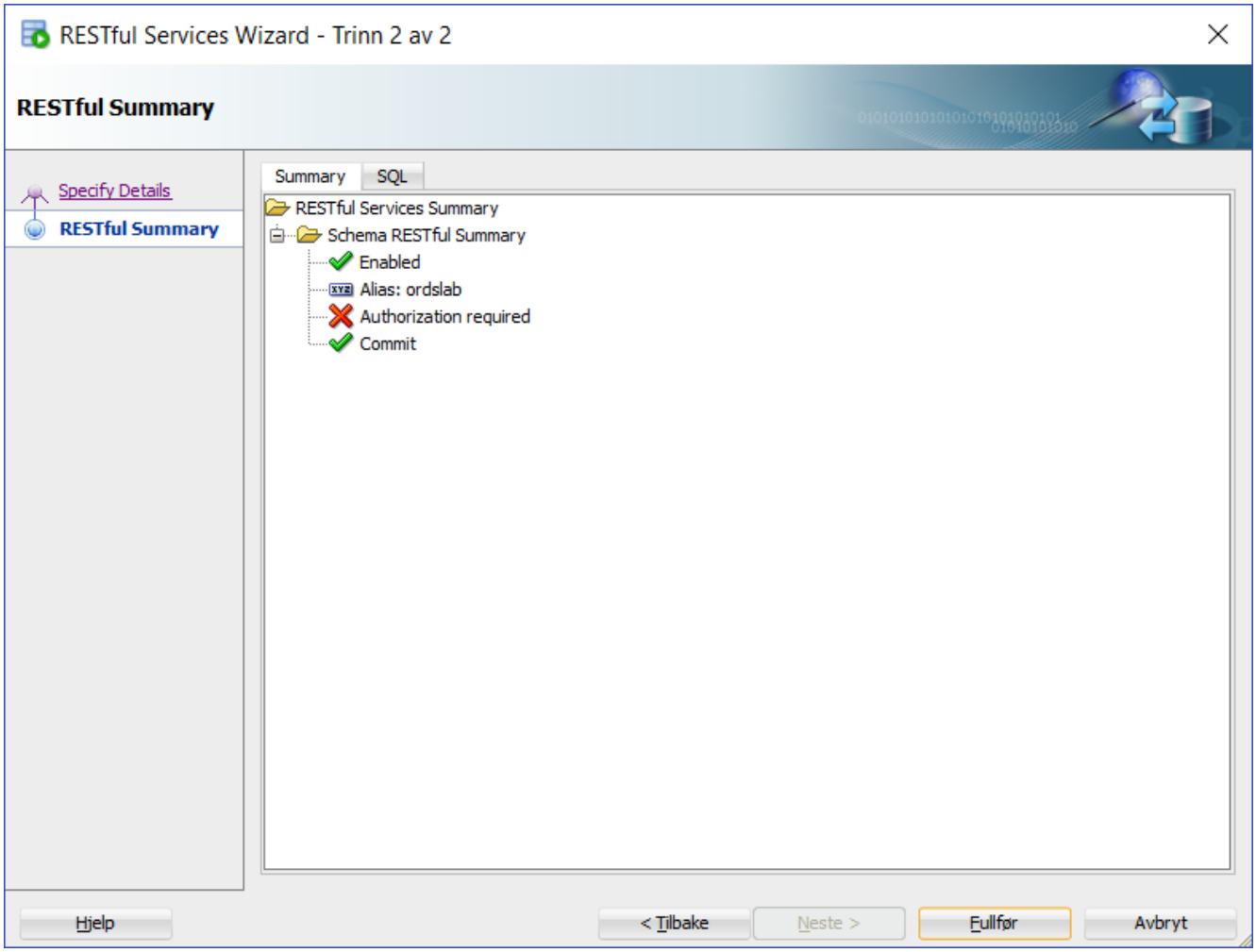
```
SQL> exec ords.enable_schema;
anonymous block completed
SQL> commit;
```

```
[oracle@ordslab ~]$ sqlplus ordslab@eusdb

SQL*Plus: Release 12.2.0.1.0 Production on Wed Jun 27 13:41:35 2018

Copyright (c) 1982, 2016, Oracle.  All rights reserved.

Enter password:
Last Successful login time: Wed Jun 27 2018 13:40:34 +00:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
SQL> exec ords.enable schema;
anonymous block completed
SQL> commit


SQL> exit
```

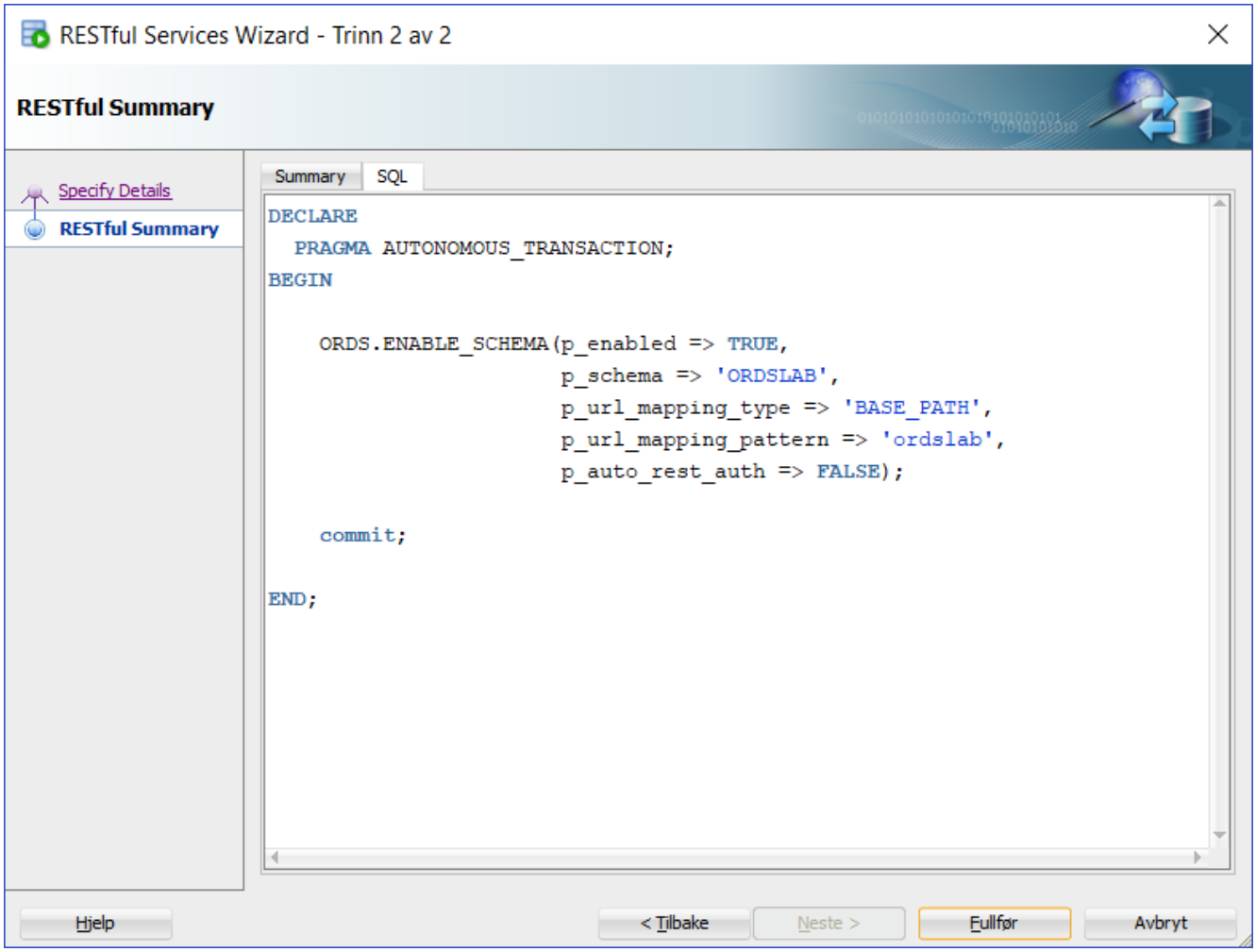
Or with SQLDeveloper, by right click on schema name:




RESTful Services Wizard - Trinn 1 av 2
Specify Details
Specify Details
RESTful Summary
Enable schema
Schema alias
ordslab
Authorization required
Hjelp
< Tilbake
Neste >
Fullfør
Avbryt



Just for fun, view the full SQL syntax:


RESTful Services Wizard - Trinn 2 av 2
RESTful Summary
Specify Details
RESTful Summary
Summary
SQL
DECLARE
PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
ORDS.ENABLE_SCHEMA(p_enabled => TRUE,
p_schema => 'ORDSLAB',
p_url_mapping_type => 'BASE_PATH',
p_url_mapping_pattern => 'ordslab',
p_auto_rest_auth => FALSE);
commit;
END;
Hjelp
< Tilbake
Neste >
Fullfør
Avbryt

Consume Database tables, Select statements via ORDS

After ORDS enabling the schema, objects to be exposed via ORDS need to be configured. There ae in general two ways of doing this:

- Automatic enable a schema object, basically open all features for ORDS for a table or view
- Create a template to finegrain the behavior and exposure of REST services.

In the lab we will do:

- Expose EMP and DEPT tables
- Create a view with a join between EMP and DEPT and expose the view
- Run some CURL examples and HTTPRequester (Firefox) examples

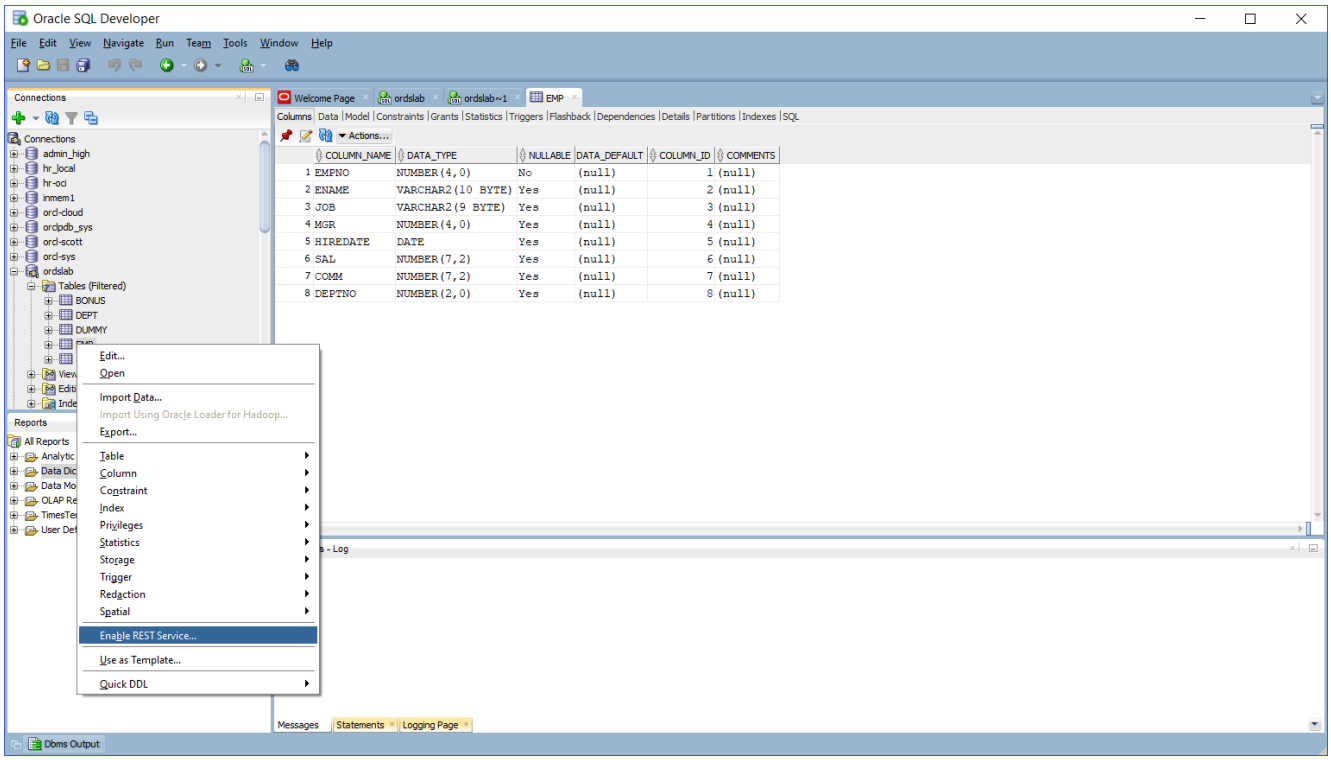
First lest create the view and verify it:

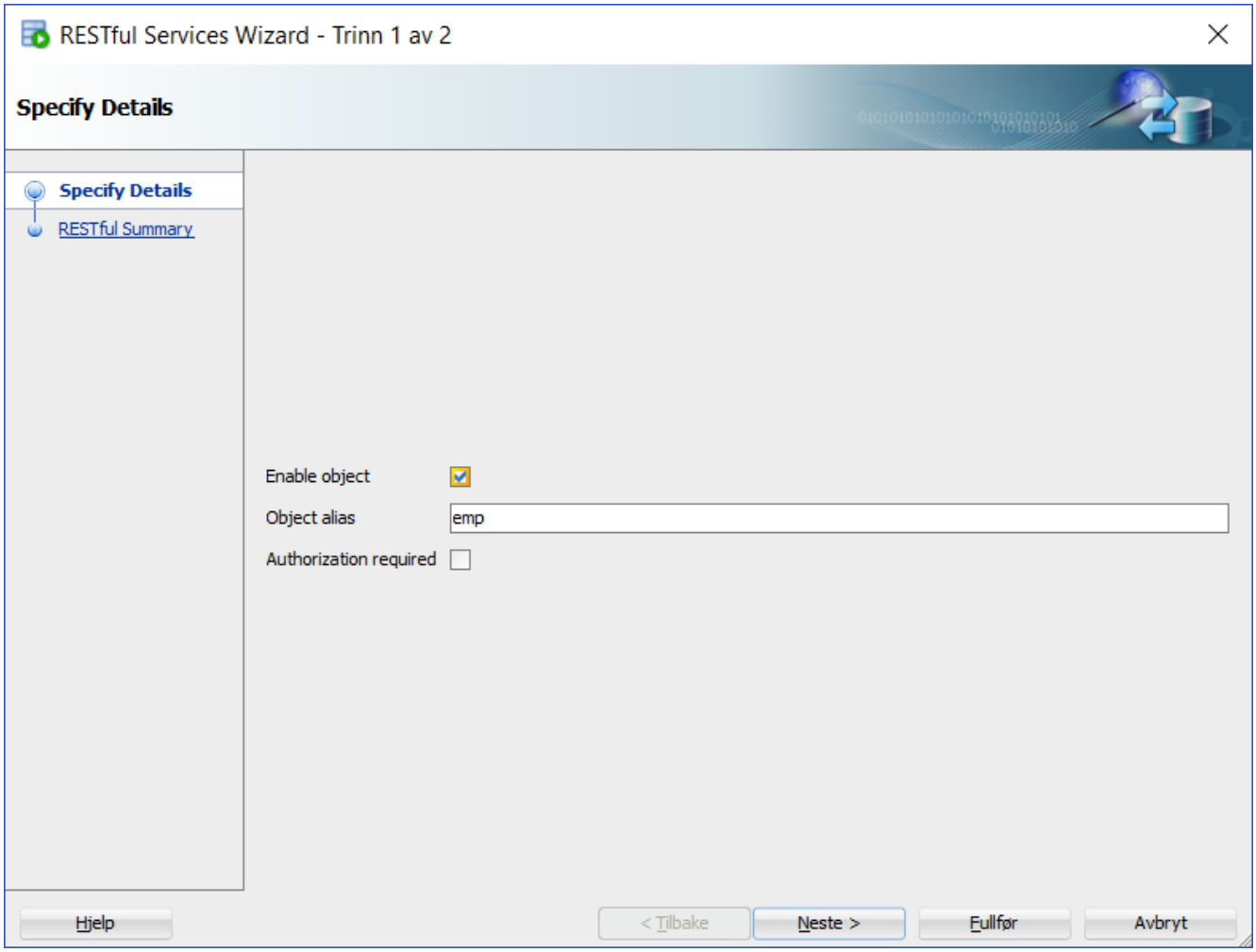
```
SQL> create view v_employees as (
  2  select ename,job,dname from emp e,dept d
  3   where d.deptno=e.deptno);

View created.
SQL> select * from v_employees;

ENAME      JOB       DNAME
---------- --------- --------------
SMITH      CLERK     RESEARCH
ALLEN      SALESMAN  SALES
WARD       SALESMAN  SALES
JONES      MANAGER   RESEARCH
MARTIN     SALESMAN  SALES
BLAKE      MANAGER   SALES
CLARK      MANAGER   ACCOUNTING
SCOTT      ANALYST   RESEARCH
KING       PRESIDENT ACCOUNTING
TURNER     SALESMAN  SALES
ADAMS      CLERK     RESEARCH

ENAME      JOB       DNAME
---------- --------- --------------
JAMES      CLERK     SALES
FORD       ANALYST   RESEARCH
MILLER     CLERK     ACCOUNTING

14 rows selected.
```

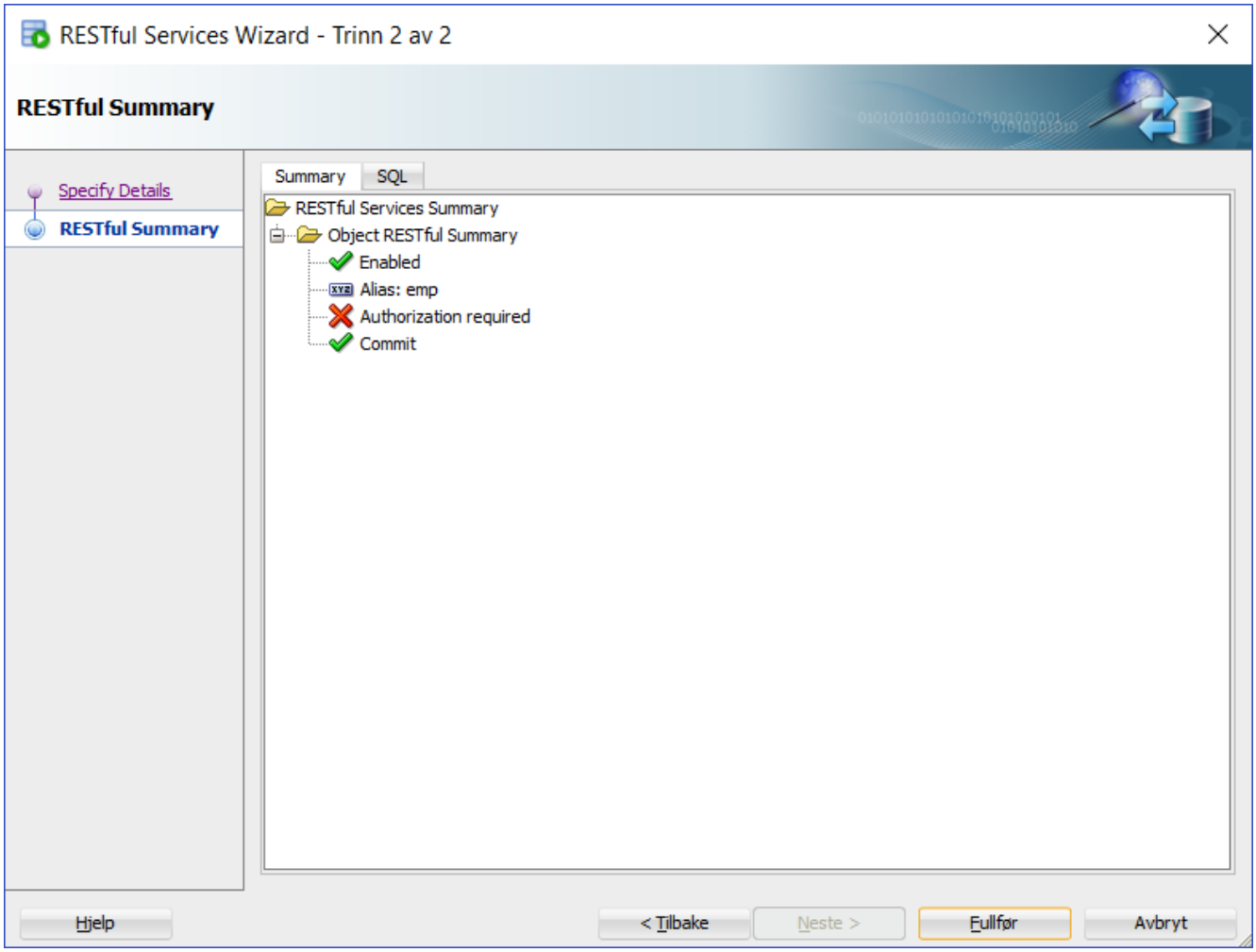
The object EMP, DEPT and V_EMPLOYEES can be enabled for ORDS exposure either with PL/SQL or with SQLDeveloper, which generates the proper PL/SQL and executes it.

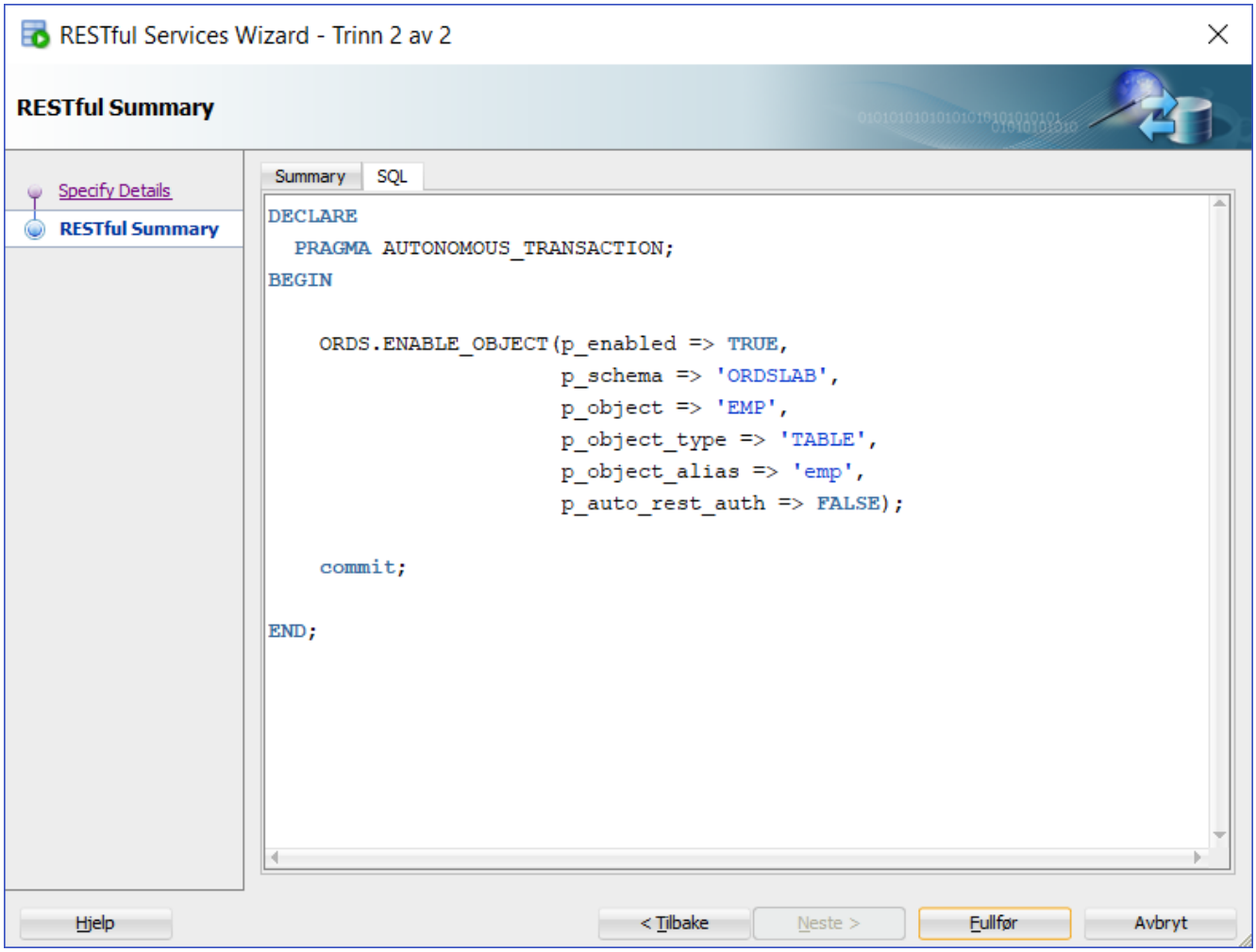
ORDS enable EMP

Right click on the table name, and select Enable REST




RESTful Services Wizard - Trinn 1 av 2
Specify Details
Specify Details
RESTful Summary
Enable object
Object alias
emp
Authorization required
Hjelp
< Tilbake
Neste >
Fullfør
Avbryt


RESTful Services Wizard - Trinn 2 av 2
RESTful Summary
Specify Details
RESTful Summary
Summary
SQL
RESTful Services Summary
Object RESTful Summary
Enabled
Alias: emp
Authorization required
Commit
Hjelp
< Tilbake
Neste >
Fullfør
Avbryt



ORDS enable DEPT

Repeat the steps above or execute the SQL directly as ordslab:

```
DECLARE
  PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
    ORDS.ENABLE_OBJECT(p_enabled => TRUE,
                       p_schema => 'ORDSLAB',
                       p_object => 'DEPT',
                       p_object_type => 'TABLE',
                       p_object_alias => 'dept',
                       p_auto_rest_auth => TRUE);
    commit;
END;
/
```

ORDS Enable V_EMPLOYEES

Repeat the steps above or execute the SQL directly as ordslab:

```
DECLARE
  PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
    ORDS.ENABLE_OBJECT(p_enabled => TRUE,
                       p_schema => 'ORDSLAB',
                       p_object => 'V_EMPLOYEES',
                       p_object_type => 'VIEW',
                       p_object_alias => 'employees',
                       p_auto_rest_auth => TRUE);
    commit;
END;
/
```

Consume Oracle Data as REST services with CURL and HTTP Requestor

The simple curl command to list all metadata, showing all REST enabled objects

curl -i http://localhost:8080/ords/ordslab/metadata-catalog/

```
[oracle@ordslab ~]$ curl -i http://localhost:8080/ords/ordslab/metadata-catalog/
HTTP/1.1 200 OK
Date: Thu, 28 Jun 2018 08:23:45 GMT
Content-Type: application/json
X-Frame-Options: SAMEORIGIN
Transfer-Encoding: chunked

{"items":[{"name":"DEPT","links":[{"rel":"describes","href":"http://localhost:8080
/ords/ordslab/dept/"},{"rel":"canonical","href":"http://localhost:8080/ords/ordsla
b/metadata-
catalog/dept/","mediaType":"application/json"},{"rel":"alternate","href":"http://l
ocalhost:8080/ords/ordslab/open-api-
catalog/dept/","mediaType":"application/openapi+json"}]},{"name":"EMP","links":[{"
rel":"describes","href":"http://localhost:8080/ords/ordslab/emp/"},{"rel":"canonic
al","href":"http://localhost:8080/ords/ordslab/metadata-
catalog/emp/","mediaType":"application/json"},{"rel":"alternate","href":"http://lo
calhost:8080/ords/ordslab/open-api-
catalog/emp/","mediaType":"application/openapi+json"}]},{"name":"V_EMPLOYEES","lin
ks":[{"rel":"describes","href":"http://localhost:8080/ords/ordslab/employees/"},{"
rel":"canonical","href":"http://localhost:8080/ords/ordslab/metadata-
catalog/employees/","mediaType":"application/json"},{"rel":"alternate","href":"htt
p://localhost:8080/ords/ordslab/open-api-
catalog/employees/","mediaType":"application/openapi+json"}]}],"hasMore":false,"li
mit":25,"offset":0,"count":3,"links":[{"rel":"self","href":"http://localhost:8080/
ords/ordslab/metadata-
catalog/"},{"rel":"first","href":"http://localhost:8080/ords/ordslab/metadata-
catalog/"}]}[oracle@ordslab ~]
```

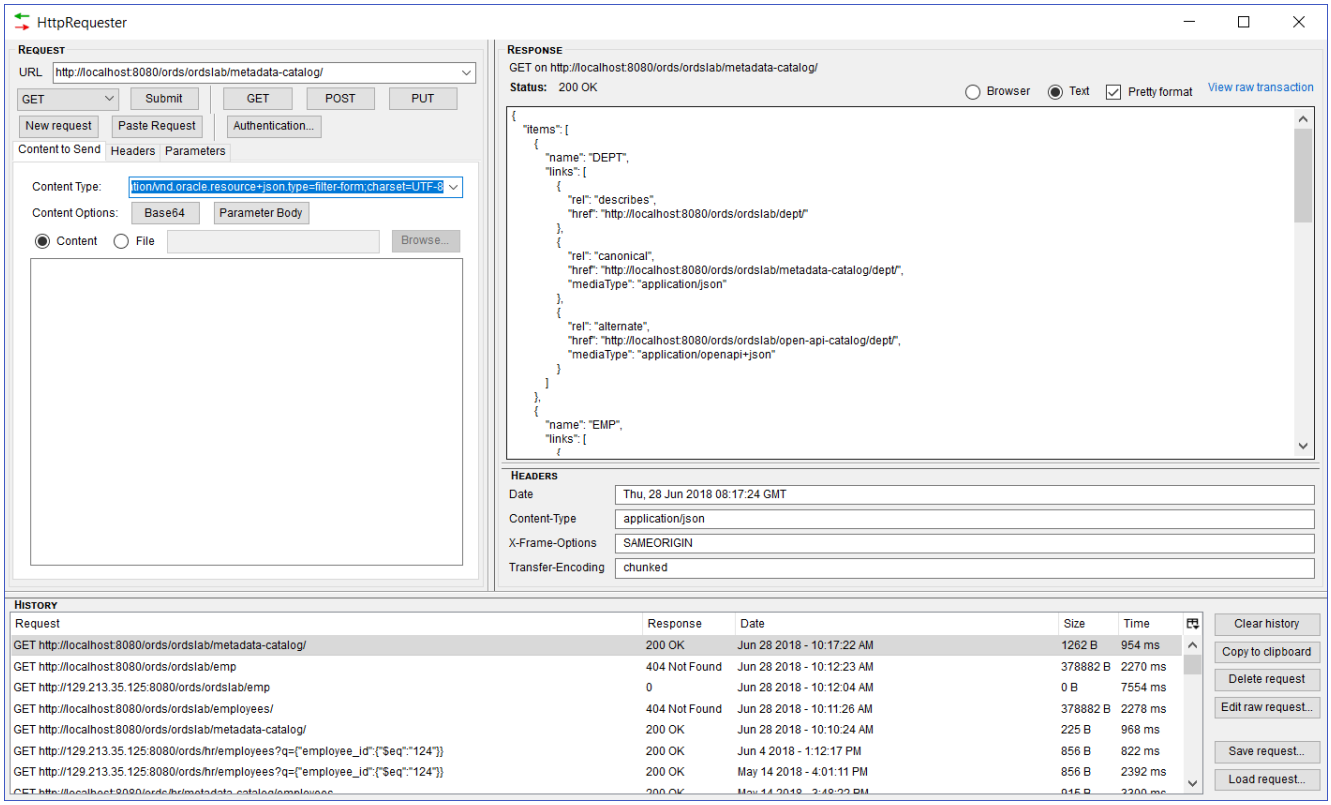
Simple curl command to dump the EMP table

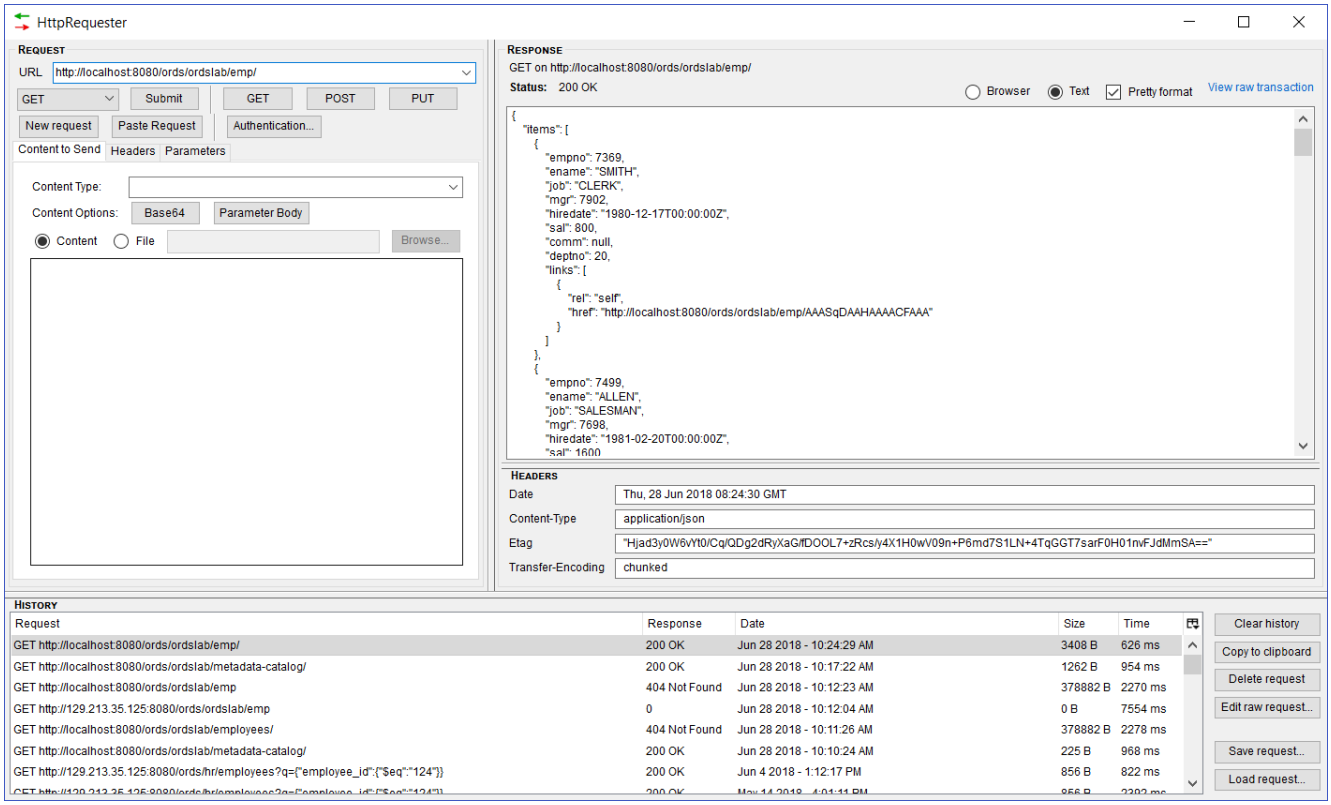
curl -i http://localhost:8080/ords/ordslab/emp/

```
[oracle@ordslab ~]$ curl -i http://localhost:8080/ords/ordslab/emp/
HTTP/1.1 200 OK
Date: Thu, 28 Jun 2018 08:20:01 GMT
Content-Type: application/json
ETag:
"Hjad3y0W6vYt0/Cq/QDg2dRyXaG/fDOOL7+zRcs/y4X1H0wV09n+P6md7S1LN+4TqGGT7sarF0H01nvFJ
dMmSA=="
Transfer-Encoding: chunked
```

{"items":[{"empno":7369,"ename":"SMITH","job":"CLERK","mgr":7902,"hiredate":"1980-12-17T00:00:00Z","sal":800,"comm":null,"deptno":20,"links":[{"rel":"self","href":"http://localhost:8080/ords/ordslab/emp/AAASqDAAHAAAACFAAA"}]},{"empno":7499,"ename":"ALLEN","job":"SALESMAN","mgr":7698,"hiredate":"1981-02-20T00:00:

The same two, formatted with HTTP Requestor:

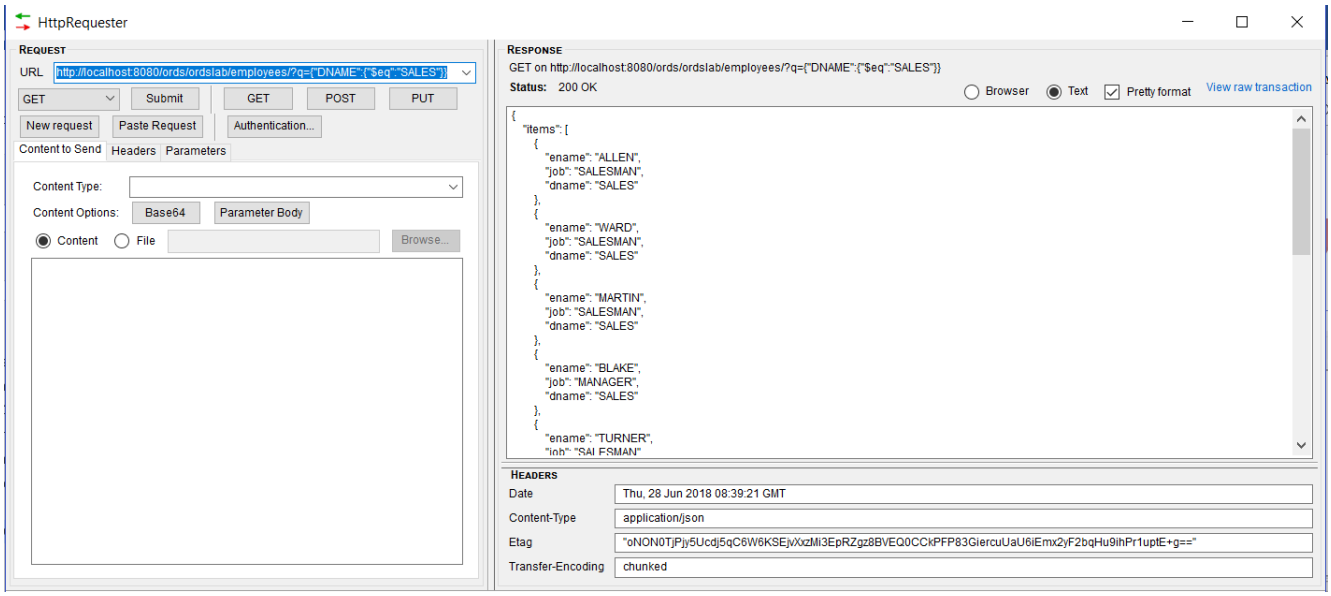




Other HTTPrequester examples

Select from employees view with simple filter

http://localhost:8080/ords/ordslab/employees/?q={"DNAME":{"$eq":"SALES"}}



Select from Employees view with complex filter with POST method

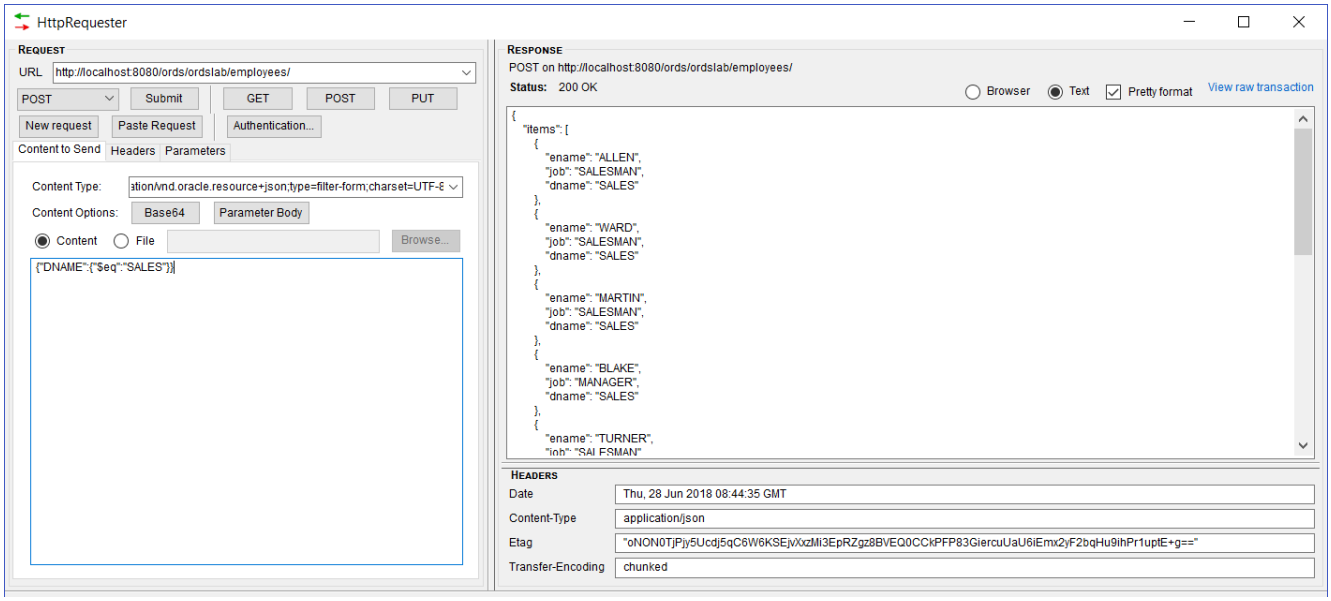
Post is normally used for changing the data, but may also be used for complex query. N this case we order all sales people based on ename

Content_type: application/vnd.oracle.resource+json;type=filter-form;charset=UTF-8

Payload: {"$orderby":{"ename":"ASC"},"DNAME":{"$eq":"SALES"}}

Method: POST

HTTPRequester



CURl command:

```
curl --header "Content-Type: application/vnd.oracle.resource+json;type=filter-
form;charset=UTF-8" \
> --request POST \
> --data '{"$orderby":{"ename":"ASC"},"DNAME":{"$eq":"SALES"}}' \
> http://localhost:8080/ords/ordslab/employees/
```

Formatted JSON Output:

```
{
    "items": [
        {
            "ename": "ALLEN",
            "job": "SALESMAN",
            "dname": "SALES"
```

```
        },
        {
            "ename": "BLAKE",
            "job": "MANAGER",
            "dname": "SALES"
        },
        {
            "ename": "JAMES",
            "job": "CLERK",
            "dname": "SALES"
        },
        {
            "ename": "MARTIN",
            "job": "SALESMAN",
            "dname": "SALES"
        },
        {
            "ename": "TURNER",
            "job": "SALESMAN",
            "dname": "SALES"
        },
        {
            "ename": "WARD",
            "job": "SALESMAN",
            "dname": "SALES"
        }
    ],
    "hasMore": false,
    "limit": 25,
    "offset": 0,
    "count": 6,
    "links": [
        {
            "rel": "self",
            "href":
"http://localhost:8080/ords/ordslab/employees/?q=%7B%22%24orderby%22:%7B%22ename%2
2:%22ASC%22%7D%2C%22DNAME%22:%7B%22%24eq%22:%22SALES%22%7D%7D"
        },
        {
            "rel": "edit",
            "href":
"http://localhost:8080/ords/ordslab/employees/?q=%7B%22%24orderby%22:%7B%22ename%2
2:%22ASC%22%7D%2C%22DNAME%22:%7B%22%24eq%22:%22SALES%22%7D%7D"
        },
        {
            "rel": "describedby",
            "href": "http://localhost:8080/ords/ordslab/metadata-
catalog/employees/"
        },
        {
            "rel": "first",
            "href":
"http://localhost:8080/ords/ordslab/employees/?q=%7B%22%24orderby%22:%7B%22ename%2
2:%22ASC%22%7D%2C%22DNAME%22:%7B%22%24eq%22:%22SALES%22%7D%7D"
        }
    ]
```

```
}
```