Oracle® REST Data Services

Quick Start Guide

Release 26.2
G58046-01
July 2026

Oracle REST Data Services Quick Start Guide, Release 26.2

G58046-01

Copyright © 2011, 2026, Oracle and/or its affiliates.

Primary Authors: Mamata Basapur, Chuck Murray

Contributors: Colm Divilly, Sharon Kennedy, Ganesh Pitchaiah, Kris Rice, Elizabeth Saunders

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Contents

Preface

Audience i
Documentation Accessibility i
Related Documents i
Conventions ii
Third-Party License Information ii

1 Getting Started with Oracle REST Data Services

1.1 Getting Started with RESTful Services 1
1.1.1 Creating a RESTful Service Using Oracle SQLcl 2
1.1.1.1 Creating a New Database User and REST-enabling Schema 3
1.1.1.2 REST-enabling a Schema 4
1.1.1.3 Accessing the Database Actions 4
1.1.1.4 Auto REST-enabling a Table 6
1.1.1.5 Testing the Auto REST-enabled Object 10
1.1.2 Creating a RESTful Service Through the REST Workshop 12
1.1.2.1 Navigate to the REST Workshop 13
1.1.2.2 Create a Module 13
1.1.2.3 Create Template 15
1.1.2.4 Create Handler 16
1.1.2.5 Test the RESTful Service 18
1.1.3 Creating a Privilege Using Database Actions 19
1.1.3.1 Steps to Create a Privilege 19
1.1.4 Register an OAuth Client Application to Access the REST API 23
1.1.4.1 Registering your Application to Access a REST API 24
1.1.5 Creating a RESTful Service Using Oracle SQL Developer 29
1.1.5.1 REST-Enable a Database Table 30
1.1.5.2 Creating a RESTful Service through the Connections Navigator 33
1.1.5.3 Creating a RESTful Service from a SQL Query 42
1.1.5.4 Protect Resources 48
1.1.5.5 Register an OAuth Client Application 50

Copyright © 2011, 2026, Oracle and/or its affiliates.

Index

Copyright © 2011, 2026, Oracle and/or its affiliates.

List of Figures

1-1	Connecting to the Database as an Administrator or an User with DBA Role	3
1-2	ORDSTEST User with required Privileges, Roles, and Tablespace	3
1-3	Executing the ORDS.ENABLE_SCHEMA PL/SQL Procedure	4
1-4	ORDS.ENABLE_SCHEMA PL/SQL Procedure	4
1-5	Accessing Database Actions as the ORDSTEST User	5
1-6	Database Actions Launchpad	5
1-7	Selecting SQL Worksheet	6
1-8	Auto REST-enabling the EMP table	8
1-9	REST Enable Object Screen	9
1-10	Table REST-enabled Confirmation	9
1-11	Icon showing Database Object is Auto REST-enabled	10
1-12	Locating Curl Command to Test the Rest Endpoint	11
1-13	Results Atfer Testing the Emp URL in the Browser	12
1-14	Navigating to the REST Workshop from the Launchpad	13
1-15	Modules Dashboard	14
1-16	Entering Values in the Create Module Screen	14
1-17	Create Template Screen	16
1-18	Create Handler	17
1-19	Resource handler SQL query results	18
1-20	New Tab for Testing the Endpoint	18
1-21	JSON Response from GET Method	19
1-22	Selecting Privileges Menu Option	20
1-23	Associating Resource Module with the Privilege	20
1-24	Privilege Created	21
1-25	Navigating to the demo module to Test the Privilege	22
1-26	Copying the URI to Test the Privilege	22
1-27	Sign-in to Test Path Privilege	23
1-28	Navigating to OAuth Client Menu Option	24
1-29	Create OAuth Client	25
1-30	Checking Client Credentials Selected	26
1-31	Entering Values in Create OAuth Client Slider	27
1-32	OAuth Client Created	27
1-33	Selecting Correct Shell Environment	28
1-34	Access Token with Expiration	28
1-35	jq Response from the Get Request	29
1-36	Enabling the Schema of the EMP Table for REST	31

Copyright © 2011, 2026, Oracle and/or its affiliates.

1-37	REST Enabling the EMP Table	32
1-38	Testing the REST Enabled Table	33
1-39	REST Data Services option under Connections Navigator	34
1-40	Entering Information on the Specify Module Page	35
1-41	Entering Information on the Specify Template Page	36
1-42	Entering Information on Create Resource Handler Page:	37
1-43	Testing URL in a Web Browser	38
1-44	Create Privilege Dialog Box	40
1-45	JSON Document After Signing in	41
1-46	Entering the New Role Name	42
1-47	Entering Information for New RESTful Services Connection	44
1-48	Entering Information on the Specify Module Page	45
1-49	Entering Information on the Specify Template Page	46
1-50	Entering Information on the Specify Handler Page:	47
1-51	Testing the RESTful Service Created from a SQL Query	48
1-52	Edit Privilege Dialog Box	49

Copyright © 2011, 2026, Oracle and/or its affiliates.

Preface

Oracle REST Data Services Quick start Guide is designed to let you get started quickly developing RESTful services using Oracle REST Data Services.

Topics:

- Audience
- Documentation Accessibility
- Related Documents
- Conventions
- Third-Party License Information

- Audience
- Documentation Accessibility
- Related Documents
- Conventions
- Third-Party License Information

Audience

This document is intended for system administrators or application developers who are installing and configuring Oracle REST Data Services. This guide assumes you are familiar with web technologies, especially REST (Representational State Transfer), and have a general understanding of Windows and UNIX platforms.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For more information and resources relating to Oracle REST Data Services, see the following the Oracle Technology Network (OTN) site:

`http://www.oracle.com/technetwork/developer-tools/rest-data-services/`

Copyright © 2011, 2026, Oracle and/or its affiliates.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that is displayed on the screen, or text that you enter.

Third-Party License Information

Oracle REST Data Services contains third-party code. See the *Oracle Database Licensing Information* book for notices Oracle is required to provide.

Note, however, that the Oracle program license that accompanied this product determines your right to use the Oracle program, including the third-party software, and the terms contained in the following notices do not change those rights.

Copyright © 2011, 2026, Oracle and/or its affiliates.

1

Getting Started with Oracle REST Data Services

This tutorial is designed to let you get started quickly developing RESTful services using Oracle REST Data Services.

- Getting Started with RESTful Services

1.1 Getting Started with RESTful Services

Prerequisites

The following are the prerequisites before you start performing the steps in this tutorial:

- Ensure that you have installed Oracle REST Data Services and configured with a currently supported version of Oracle database.
- Ensure that you have installed a currently supported version of client applications. This tutorial uses the following clients to create a RESTful service:
  - Oracle Database Actions
  - Oracle SQLcl

    > **Note**
    >
    > The latest version of SQLcl can be downloaded from one of the following:
    >
    > * SQLcl Downloads
    > * Through Homebrew using the command: `brew install --cask sqlcl`. Additional SQLcl installation information can be found on SQLcl Homebrew.

  - Oracle SQL Developer
  - Oracle strongly recommends you to install a browser extension that enables you to view JSON in the web browser. Popular browser extensions include one of the following:
    * JSON Formatter for Google Chrome
    * JSONView Add-on for Mozilla Firefox

  This tutorial assumes the following:

  - Oracle REST Data Services has been installed and configured on the following server, port, and context path: `localhost:8080/ords/`
  - Oracle REST Data Services is running in a standalone mode
  - Oracle REST Data Services installation was performed using the *Basic Connection* type with the following attributes:
    * Server: `localhost`

Copyright © 2011, 2026, Oracle and/or its affiliates.

  * Port: 1521
  * Service name: ORCLPBD1

The examples in this tutorial assume that Oracle REST Data Services has been installed and configured in a single instance database or Pluggable Database (PDB). The examples and images in this tutorial refer to the PDB as ORCLPDB1.

Client Applications used in this tutorial for Creating the RESTful Services

The examples in this tutorial use the following client applications:

- Oracle Database Actions
- SQLcl

Web browser requirements

For a complete list of currently supported web browsers, refer to Oracle Software Web Browser Support Policy.

- Creating a RESTful Service Using Oracle SQLcl
- Creating a RESTful Service Through the REST Workshop
- Creating a Privilege Using Database Actions
  This section describes how to create a privilege and control access to protected resources.
- Register an OAuth Client Application to Access the REST API
  This topic explains how to register your applications (called third-party applications here) to access a REST API.
- Creating a RESTful Service Using Oracle SQL Developer

1.1.1 Creating a RESTful Service Using Oracle SQLcl

This section describes the steps for creating a RESTful service using Oracle SQLcl.

> **Note**
>
> Oracle recommends that you perform the following steps in this tutorial using the specified names for schemas and database objects. After you have completed the tutorial, you can follow the same steps again using alternate schema and database object names.

Perform the following steps to create a RESTful service:

1. Creating a New Database User and REST-enabling Schema
2. REST-enabling a Schema
3. Accessing the Database Actions
4. Auto REST-enabling a Table
5. Testing the Auto REST-enabled Object

- Creating a New Database User and REST-enabling Schema
- REST-enabling a Schema
- Accessing the Database Actions
- Auto REST-enabling a Table

Copyright © 2011, 2026, Oracle and/or its affiliates.

- Testing the Auto REST-enabled Object

1.1.1.1 Creating a New Database User and REST-enabling Schema

To create a new user, perform the following steps:

1. Using SQLcl, connect to your database as an administrator or using an account with the DBA role.

Figure 1-1 Connecting to the Database as an Administrator or an User with DBA Role

```
~ — -zsh                                   ~ — java ◂ sql sys/oracle@localhost:1521/ORCLPDB1 as sysdba
Last login: Wed May 31 14:07:44 on ttys002
[redacted]@[redacted] ~ % sql sys/oracle@localhost:1521/ORCLPDB1 as sysdba

SQLcl: Release 23.1 Production on Wed May 31 14:25:10 2023

Copyright (c) 1982, 2023, Oracle.  All rights reserved.

Connected to:
Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL>
```

2. Run the following commands to create a new `ORDSTEST` user with the required privileges, roles, and tablespace:

```
CREATE USER ORDSTEST IDENTIFIED BY <password>;
GRANT "CONNECT" TO ORDSTEST;
GRANT "RESOURCE" TO ORDSTEST;
GRANT UNLIMITED TABLESPACE TO ORDSTEST;
```

Figure 1-2 ORDSTEST User with required Privileges, Roles, and Tablespace

```
~ — -zsh                                   ~ — java ◂
Last login: Thu Jun  1 10:54:47 on ttys004
[redacted]@[redacted] ~ % sql sys/oracle@localhost:1521/ORCLPDB1 as sysdba

SQLcl: Release 23.1 Production on Thu Jun 01 10:56:18 2023

Copyright (c) 1982, 2023, Oracle.  All rights reserved.

Connected to:
Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> CREATE USER ORDSTEST IDENTIFIED BY oracle;

User ORDSTEST created.

SQL> GRANT "CONNECT" TO ORDSTEST;

Grant succeeded.

SQL> GRANT "RESOURCE" TO ORDSTEST;

Grant succeeded.

SQL> GRANT UNLIMITED TABLESPACE TO ORDSTEST;

Grant succeeded.

SQL>
```


Copyright © 2011, 2026, Oracle and/or its affiliates.

1.1.1.2 REST-enabling a Schema

To REST-enable a schema, connect to your database as the ORDSTEST user and then run the following PL/SQL procedure:

```
ORDS_ADMIN.ENABLE_SCHEMA
```

Figure 1-3 Executing the ORDS.ENABLE_SCHEMA PL/SQL Procedure

```
[redacted]@[redacted] ~ % sql ordstest/password1234@//localhost:33431/FREEPDB1


SQLcl: Release 23.3 Production on Mon Dec 11 12:49:55 2023

Copyright (c) 1982, 2023, Oracle.  All rights reserved.

Connected to:
Oracle Database 23c Free Release 23.0.0.0.0 - Develop, Learn, and Run for Free
Version 23.3.0.23.09

SQL> Execute ORDS.ENABLE_SCHEMA;

PL/SQL procedure successfully completed.

SQL>
```

Figure 1-4 ORDS.ENABLE_SCHEMA PL/SQL Procedure

> **See Also**
>
> ORDS PL/SQL Package Reference

1.1.1.3 Accessing the Database Actions

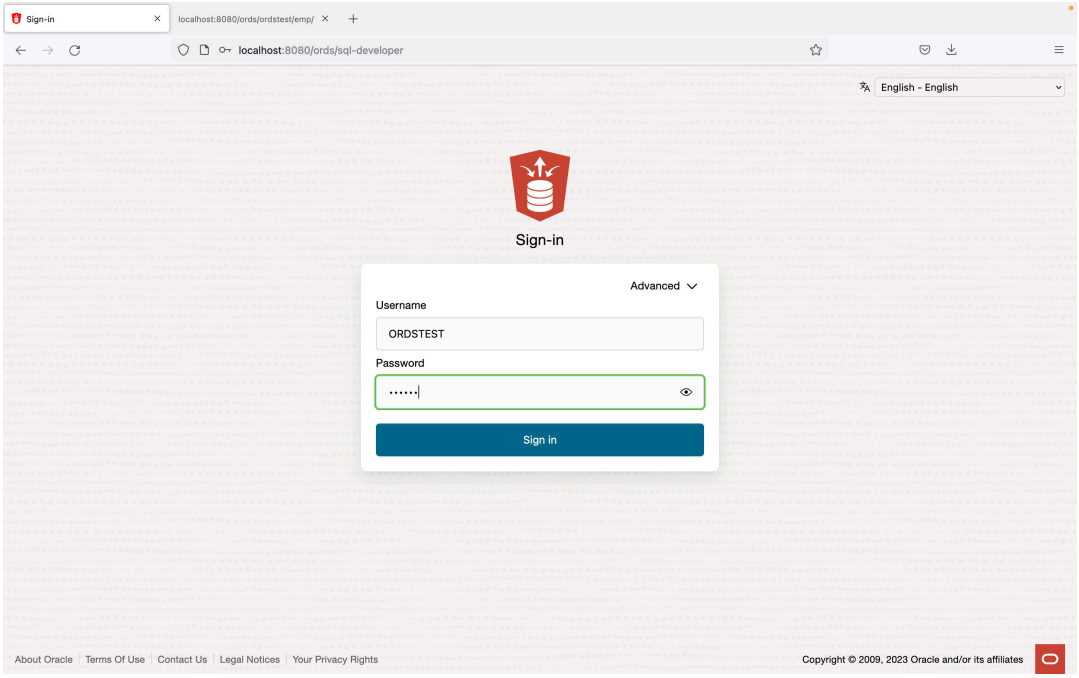
To access the Database Actons, perform the following steps:

Now that `ORDSTEST` user schema is REST-enabled, you can now access Database Actions as the `ORDSTEST` user.

1. Navigate to the URL: `http://[server]:[port]/ords/sql-developer` to display the Sign-in page.

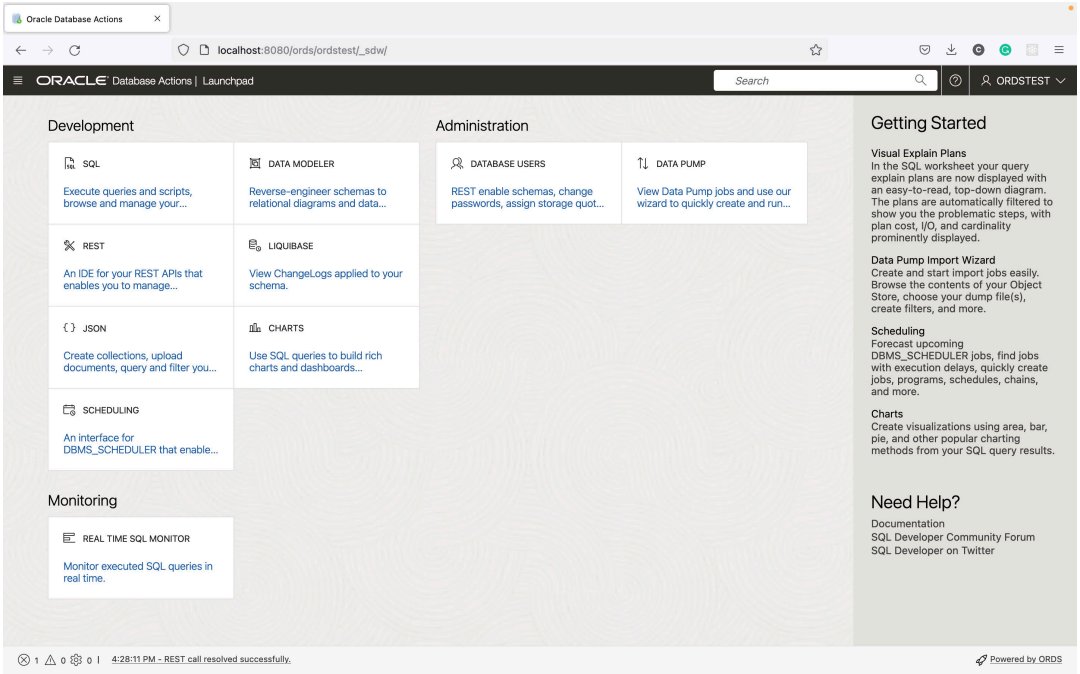
Copyright © 2011, 2026, Oracle and/or its affiliates.

Figure 1-5 Accessing Database Actions as the ORDSTEST User



2. Sign-in as the `ORDSTEST` user with <username>: `ORDSTEST` and <`password`>: `[Password]`.
3. Click on *Sign in* button.
4. Database Actions Launchpad home page appears. It comprises of six main categories: **Development**, **Data Studio**, **Administration**, **Monitoring**, **Downloads**, and **Related Services**. Each category consists of feature-based icons that you can click to navigate to the respective pages available to the `ORDSTEST` user.

Figure 1-6 Database Actions Launchpad



Copyright © 2011, 2026, Oracle and/or its affiliates.

> **See Also**
>
> About Oracle Database Actions

1.1.1.4 Auto REST-enabling a Table

Perform the following steps to connect as a new user and auto REST-enable a table:

> **Note**
>
> The tasks listed in this section are completed with the user logged-in as an ORDSTEST user.

1. Creating a Database Table
2. Inserting the Sample Data
3. REST-enabling the EMP Table

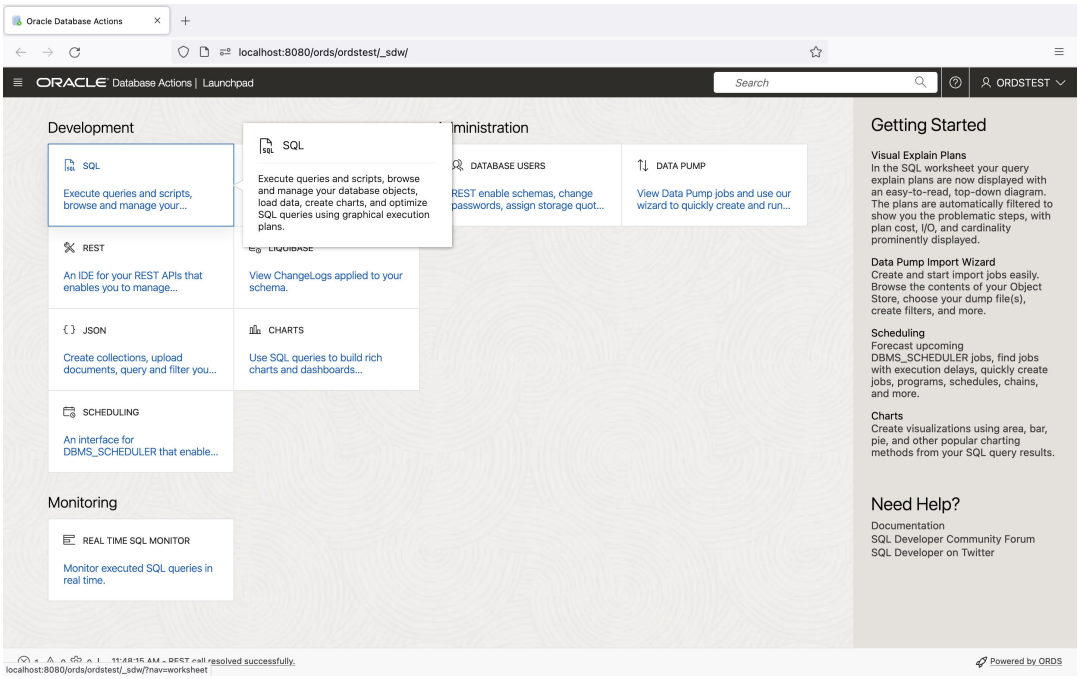
- Creating a Database Table
- Inserting the Sample Data
- REST-enabling the EMP Table

1.1.1.4.1 Creating a Database Table

Perform the following steps to create an EMP table:

1. From the Database Actions launchpad, select **SQL**, under the **Development** category of the dashboard.

Figure 1-7 Selecting SQL Worksheet



Copyright © 2011, 2026, Oracle and/or its affiliates.

2. After the SQL Worksheet is loaded, enter the following SQL query to create the EMP table:

```
CREATE TABLE EMP (
        EMPNO NUMBER(4,0),
        ENAME VARCHAR2(10 BYTE),
        JOB VARCHAR2(9 BYTE),
        MGR NUMBER(4,0),
        HIREDATE DATE,
        SAL NUMBER(7,2),
        COMM NUMBER(7,2),
        DEPTNO NUMBER(2,0),
        CONSTRAINT PK_EMP PRIMARY KEY (EMPNO)
        );
```

1.1.1.4.2 Inserting the Sample Data

1. After the `EMP` table is successfully created, insert the following sample data:

```
     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;
```

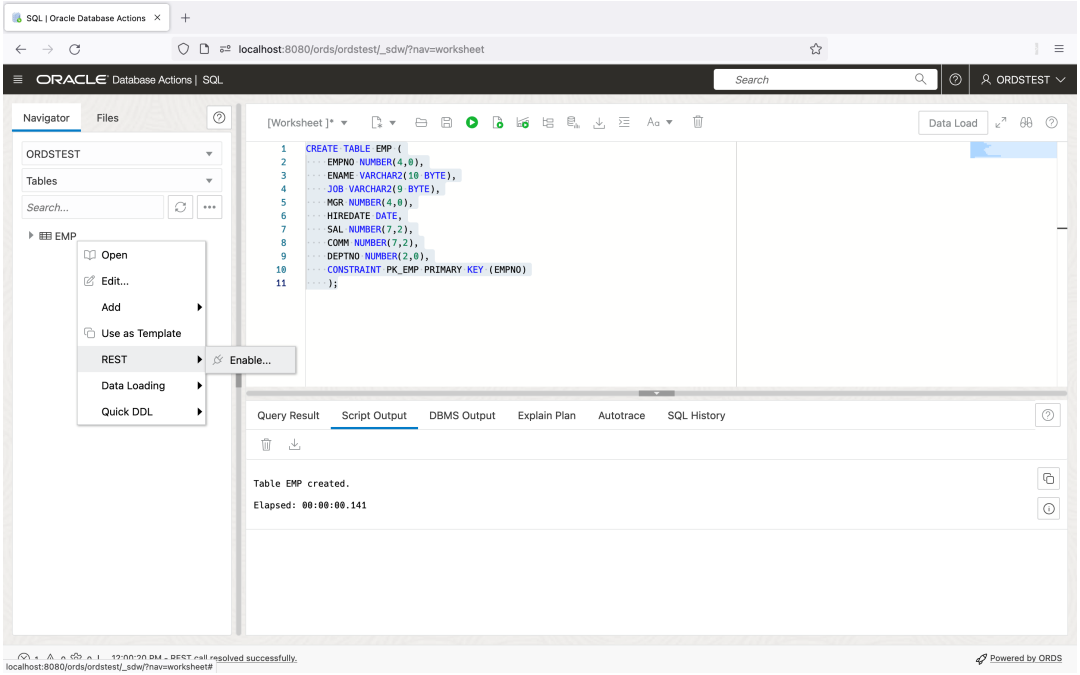

Copyright © 2011, 2026, Oracle and/or its affiliates.

1.1.1.4.3 REST-enabling the EMP Table

Perform the following steps to auto REST-enable the table:

1. From the navigator panel, right mouse-click on the table name, navigate to **REST** and then click **Enable**.

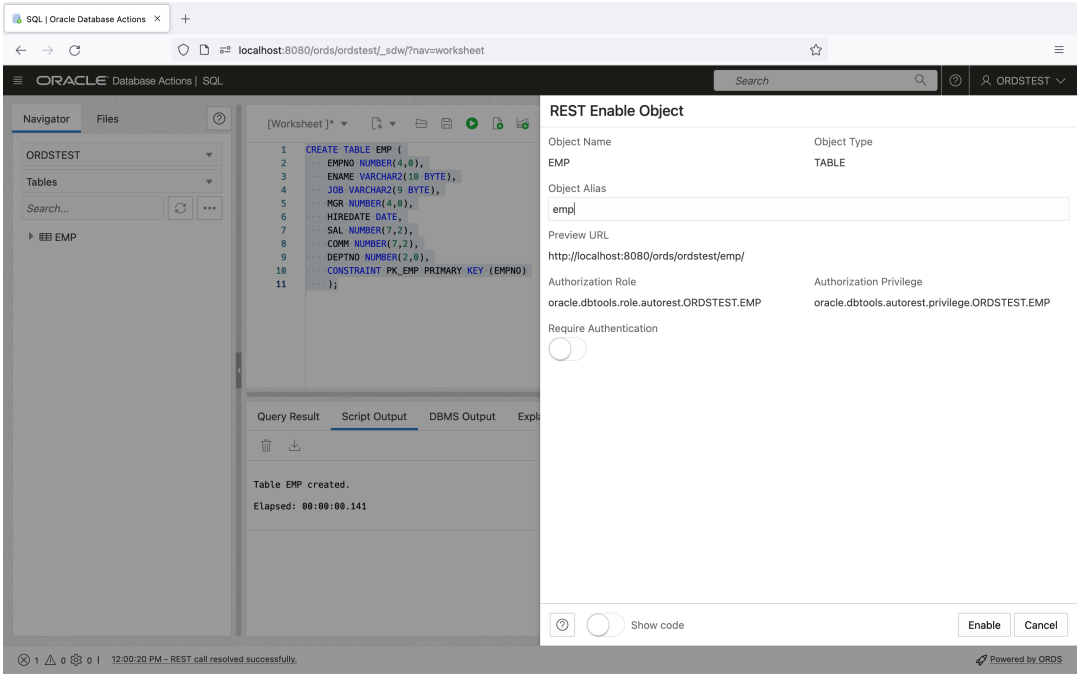
Figure 1-8 Auto REST-enabling the EMP table



2. A **REST Enable Object** screen appears. After reviewing the parameters that got automatically generated, click on **Enable** located at the bottom of the screen.

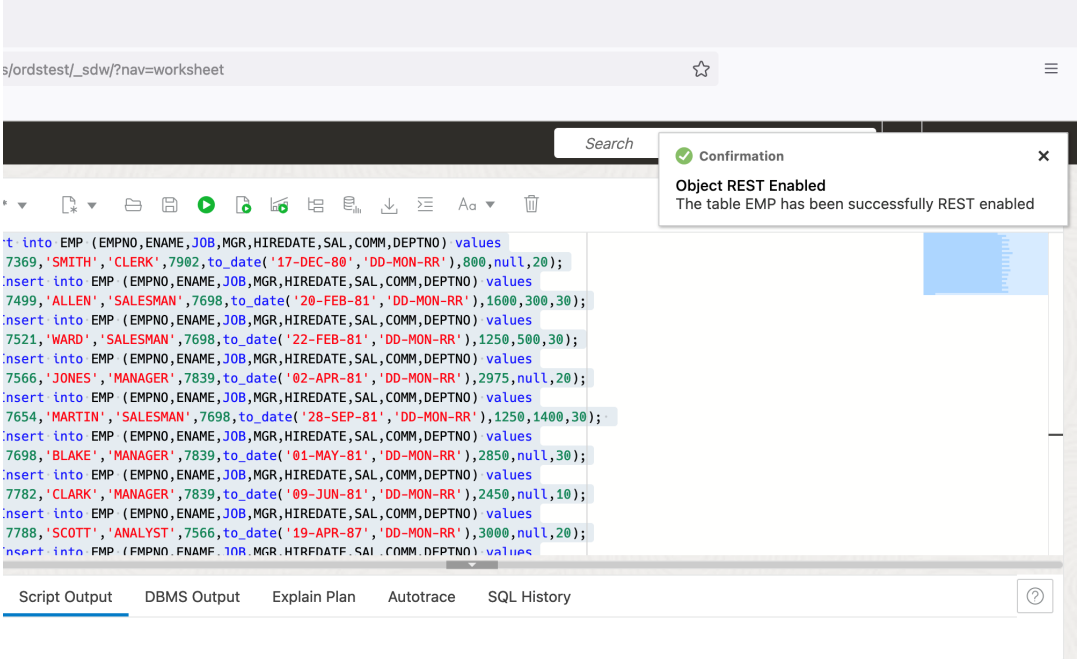
Copyright © 2011, 2026, Oracle and/or its affiliates.

3. **Figure 1-9 REST Enable Object Screen**



4. A message slider appears confirming that the EMP table has been REST-enabled.

Figure 1-10 Table REST-enabled Confirmation

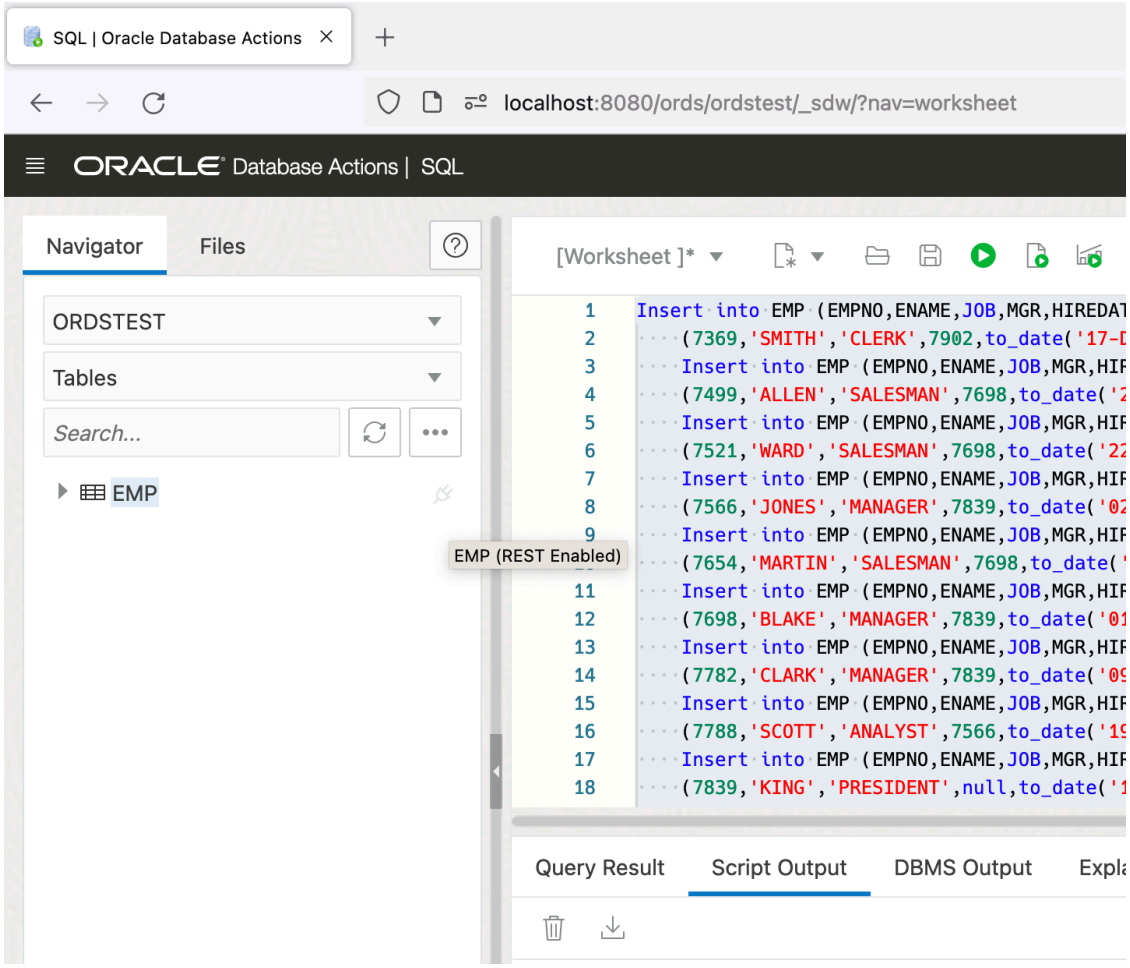


Copyright © 2011, 2026, Oracle and/or its affiliates.

1.1.1.5 Testing the Auto REST-enabled Object

> **Note**
>
> An icon next to the database objects indicates that the database objects have been auto-REST enabled.

Figure 1-11 Icon showing Database Object is Auto REST-enabled



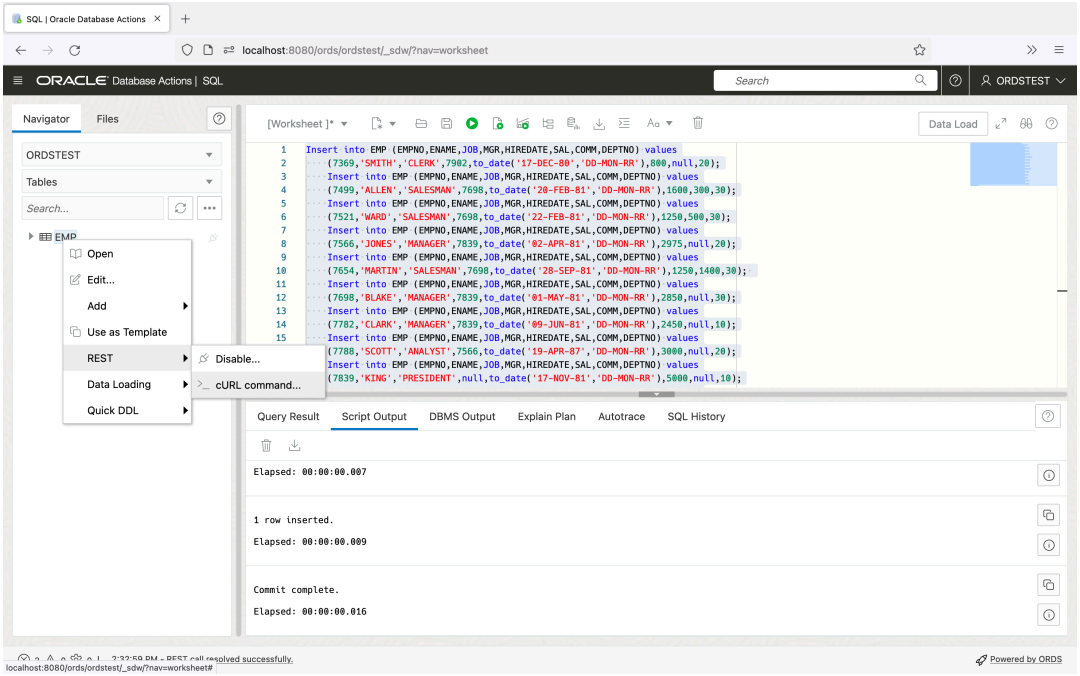
> **Note**
>
> If you do not see the icon, then click **Refresh** in the Navigator Panel to display the icon.

To review and retrieve the REST endpoints for the EMP table, perform the following steps:

1. Right-click on the name of the object, click **REST** and then select **cURL command**:

Copyright © 2011, 2026, Oracle and/or its affiliates.

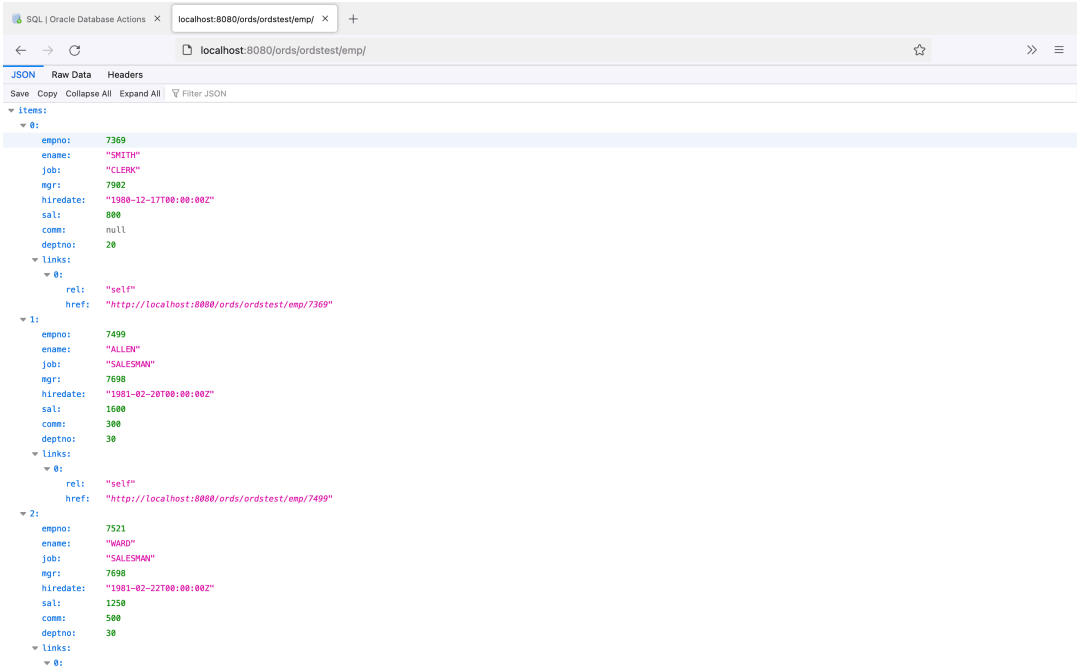
Figure 1-12 Locating Curl Command to Test the Rest Endpoint



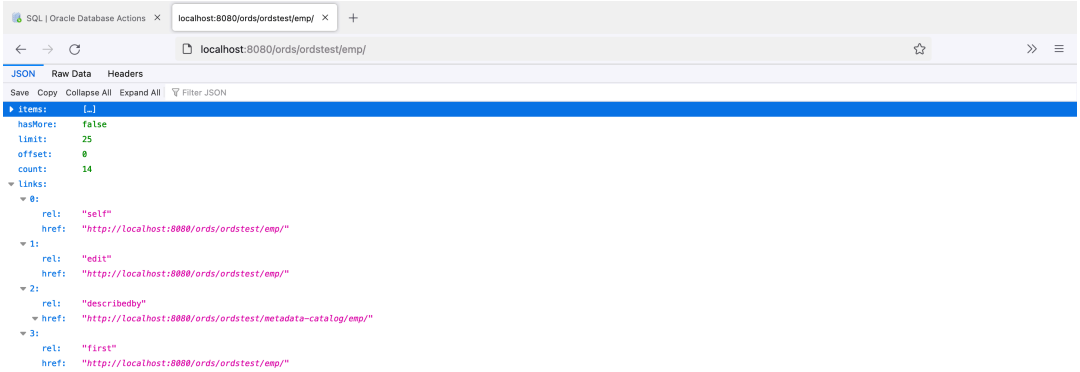
2. A slider **cURL for the table EMP** appears with the following HTTP methods available to an auto-REST enabled resource:
   - `GET ALL`
   - `GET Single`
   - `POST`
   - `BATCH LOAD`
   - `PUT`
   - `DELETE`
3. Copy the URL portion of the `GET ALL` cURL command.
4. Open a new browser and paste the URL in the address bar, and press **Enter** on your keyboard.

Copyright © 2011, 2026, Oracle and/or its affiliates.

Figure 1-13 Results Atfer Testing the Emp URL in the Browser



5. A list of the first 25 items in the EMP table are displayed.
6. If you collapse the preceding items list, it reveals the other helpful links that are automatically included with all the auto-REST enabled resources.



1.1.2 Creating a RESTful Service Through the REST Workshop

This section explains how to create a RESTful service using REST Workshop of Database Actions. The REST Workshop enables you to create and edit the RESTful service definitions.

Copyright © 2011, 2026, Oracle and/or its affiliates.

Perform the following steps to create and test a RESTful service in the REST Workshop:

1. Navigate to the REST Workshop
2. Create a Module
3. Create Template
4. Create Handler
5. Test the RESTful Service

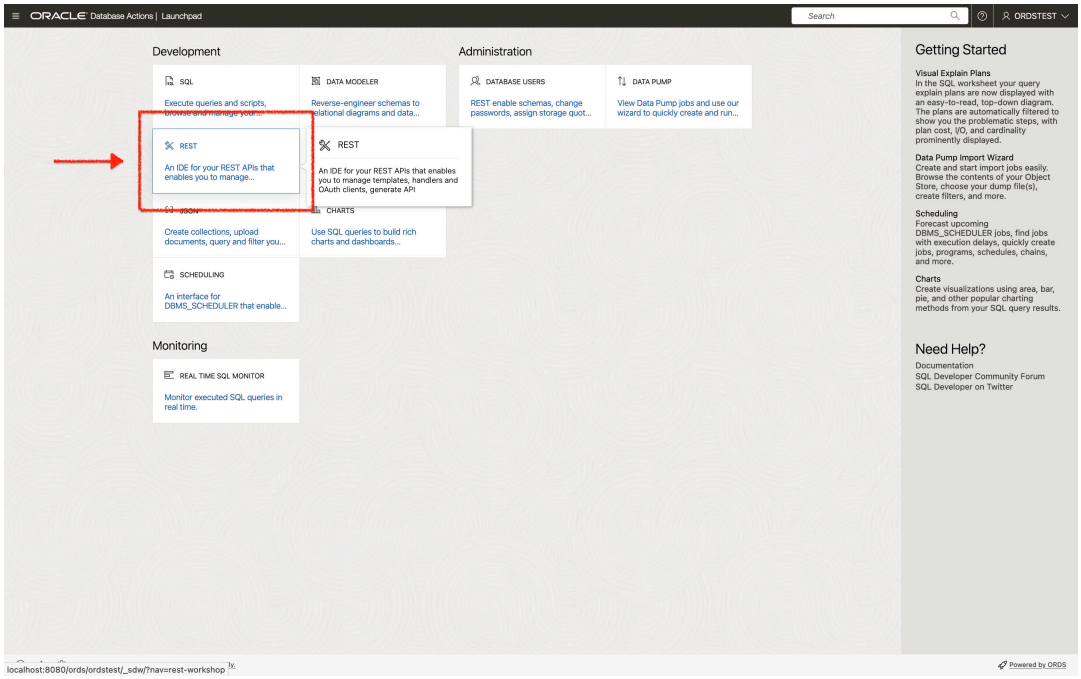
- Navigate to the REST Workshop
- Create a Module
- Create Template
- Create Handler
- Test the RESTful Service

1.1.2.1 Navigate to the REST Workshop

Perform the following steps to navigate to the REST Workshop:

1. Log in as the ORDSTEST user and navigate to the Database Actions Launchpad.
2. Select **REST** under the **Development** section.

Figure 1-14 Navigating to the REST Workshop from the Launchpad



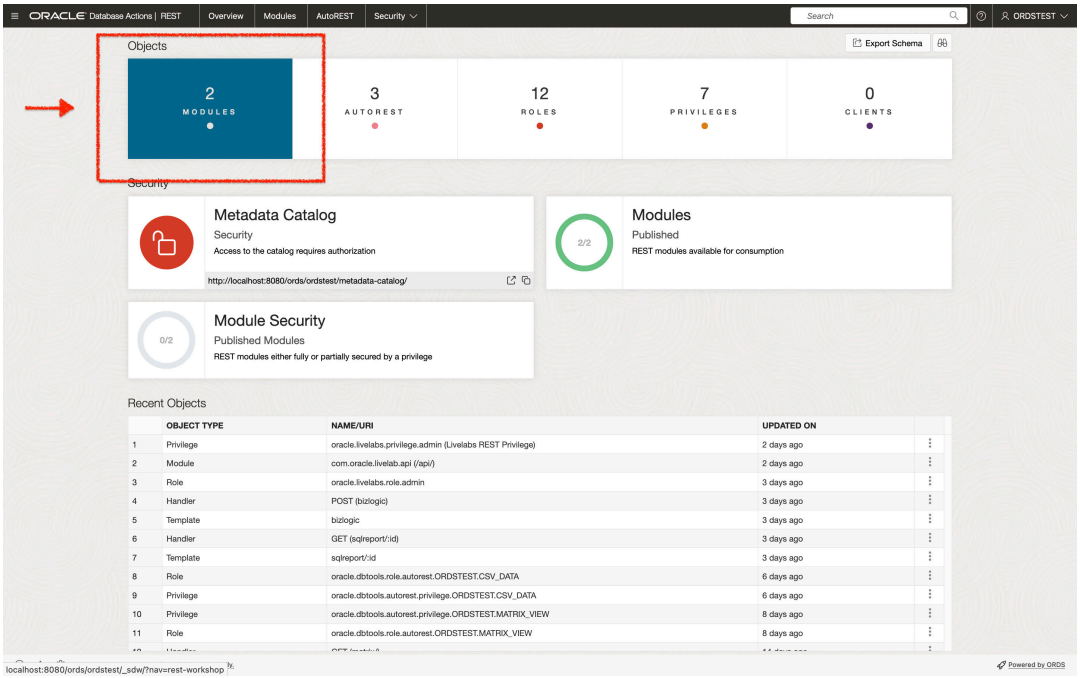
1.1.2.2 Create a Module

To create a module, perform the following steps:

1. After the REST workshop screen loads, click on **Modules** widget.

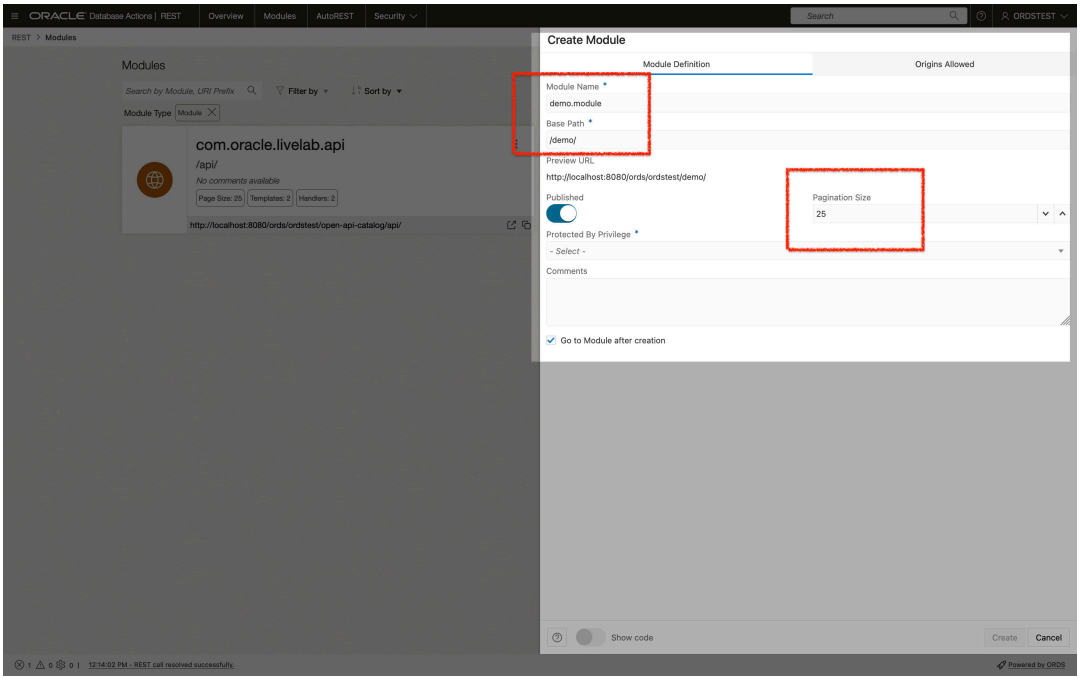
Copyright © 2011, 2026, Oracle and/or its affiliates.

Figure 1-15 Modules Dashboard



2. Modules dashboard appears. Click on **Create Module** button located at the upper right-hand corner of the dashboard.
3. A **Create Module** slider appears.

Figure 1-16 Entering Values in the Create Module Screen



Enter he following values in the respective fields:

Copyright © 2011, 2026, Oracle and/or its affiliates.

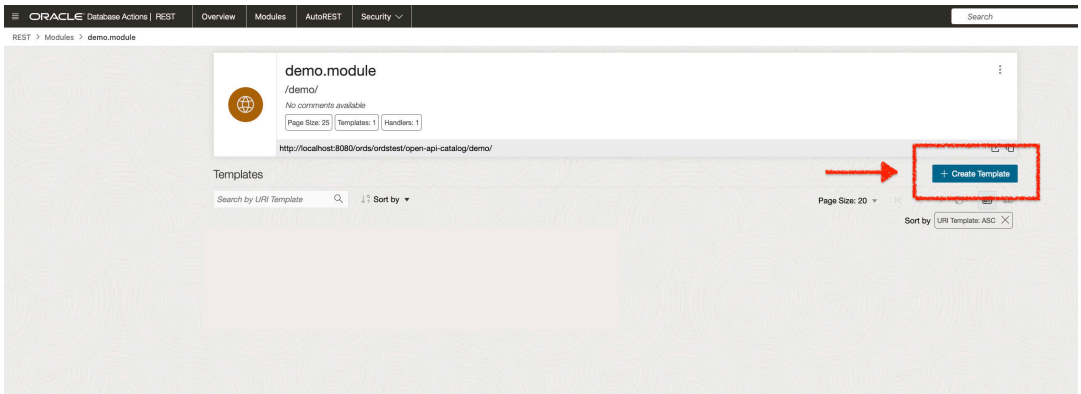
- **Module Name**: Any desired name for the connection. For example, `demo.module`.
- **URI Prefix**: `/demo/`
- **Pagination Size**: 25
- In the **Protected by Privilege** field, select `Not Protected`

4. Click on **Create**, the module settings are saved, and a confirmation message is displayed to confirm that the module is created.

1.1.2.3 Create Template

Perform the following steps to create a template:

1. After creating a Module, you will be automatically taken to the **Create Template** screen. Click on **Create Template** button.

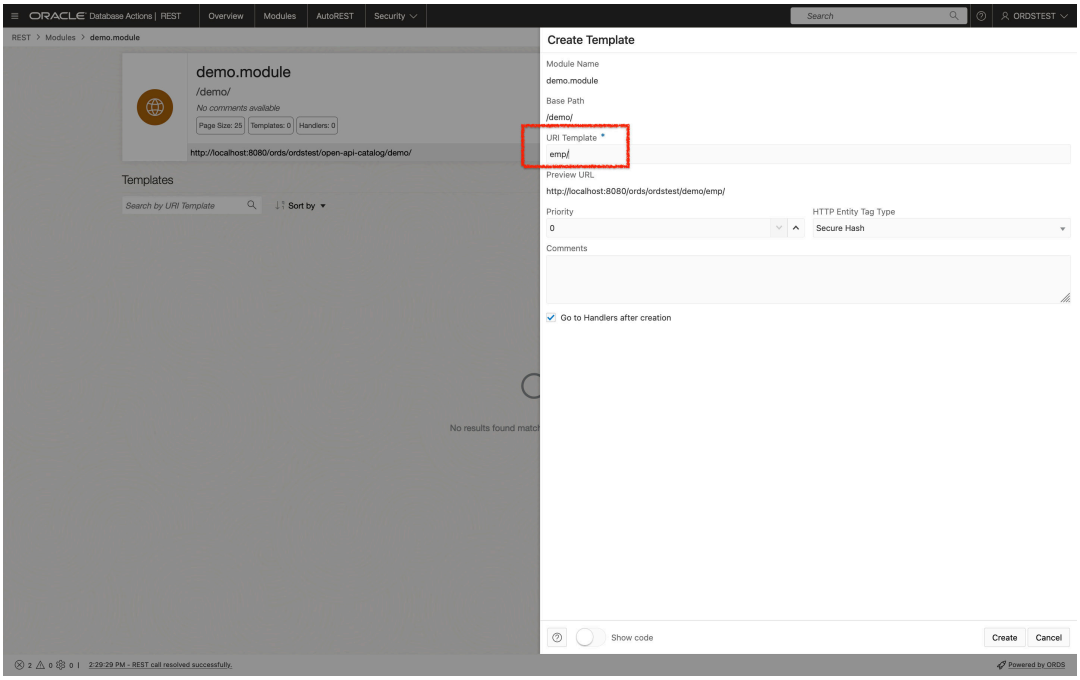


2. A **Create Template** screen appears. In the **URI Template** field, enter `emp/`.

> **Note**
>
> For this demonstration, retain the detault settings.

Copyright © 2011, 2026, Oracle and/or its affiliates.

Figure 1-17 Create Template Screen



3. You are automatically taken to the **Create Handler** page.

1.1.2.4 Create Handler

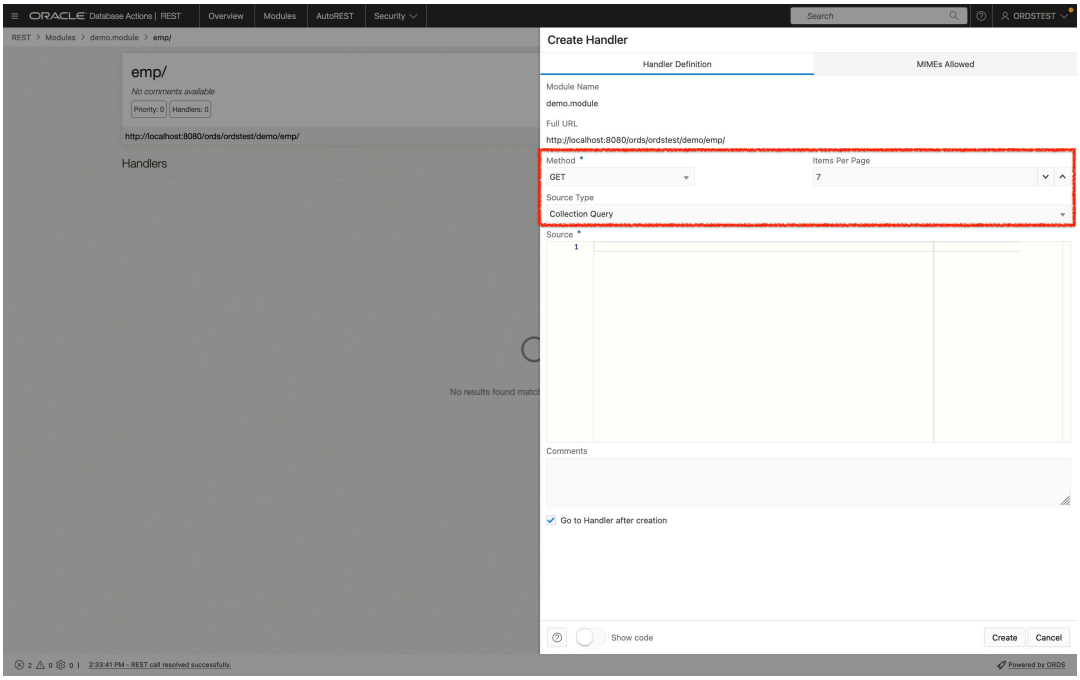
Perform the following steps to create the Get handler and review the SQL query results:

> **Note**
>
> For this tutorial, a `GET` method is created.

1. Click on **Create Handler** button to display the `Create Handler` screen. Verify the following settings:
   - **Method**: `GET`
   - **Items Per Page**: `7`
   - **Source Type**: `Collection Query`

Copyright © 2011, 2026, Oracle and/or its affiliates.

Figure 1-18 Create Handler



2. In the **Source** field, enter the following SQL query:

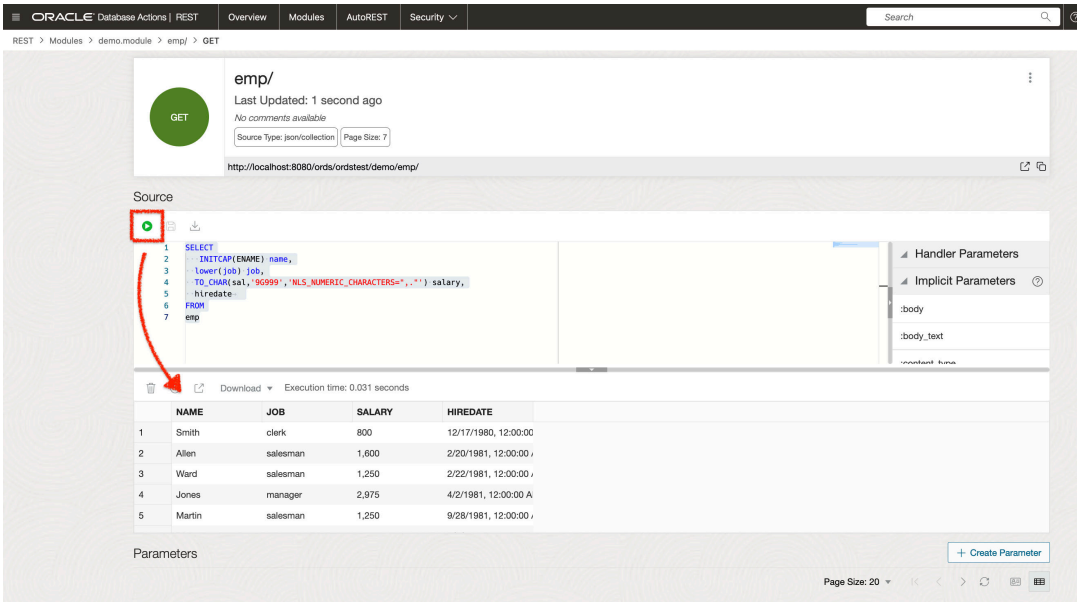
```
SELECT
  INITCAP(ENAME) name,
  lower(job) job,
  TO_CHAR(sal,'9G999','NLS_NUMERIC_CHARACTERS=",."') salary,
  hiredate
FROM
emp
```

Click on the **Create** button to automatically open the **Resource Handler** page with a confirmation message indicating that the handler is created.

3. You can then test the SQL query. To do so, click on the **Play** icon. The results of the query appears in the output window.

Copyright © 2011, 2026, Oracle and/or its affiliates.

Figure 1-19 Resource handler SQL query results

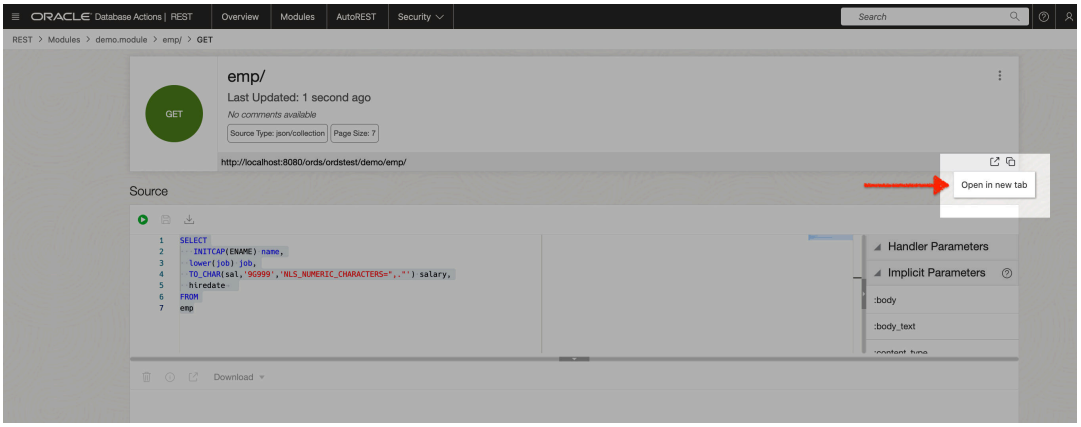


1.1.2.5 Test the RESTful Service

Perform the following steps to test the RESTful service:

1. Click on the **Open in a new tab** icon.

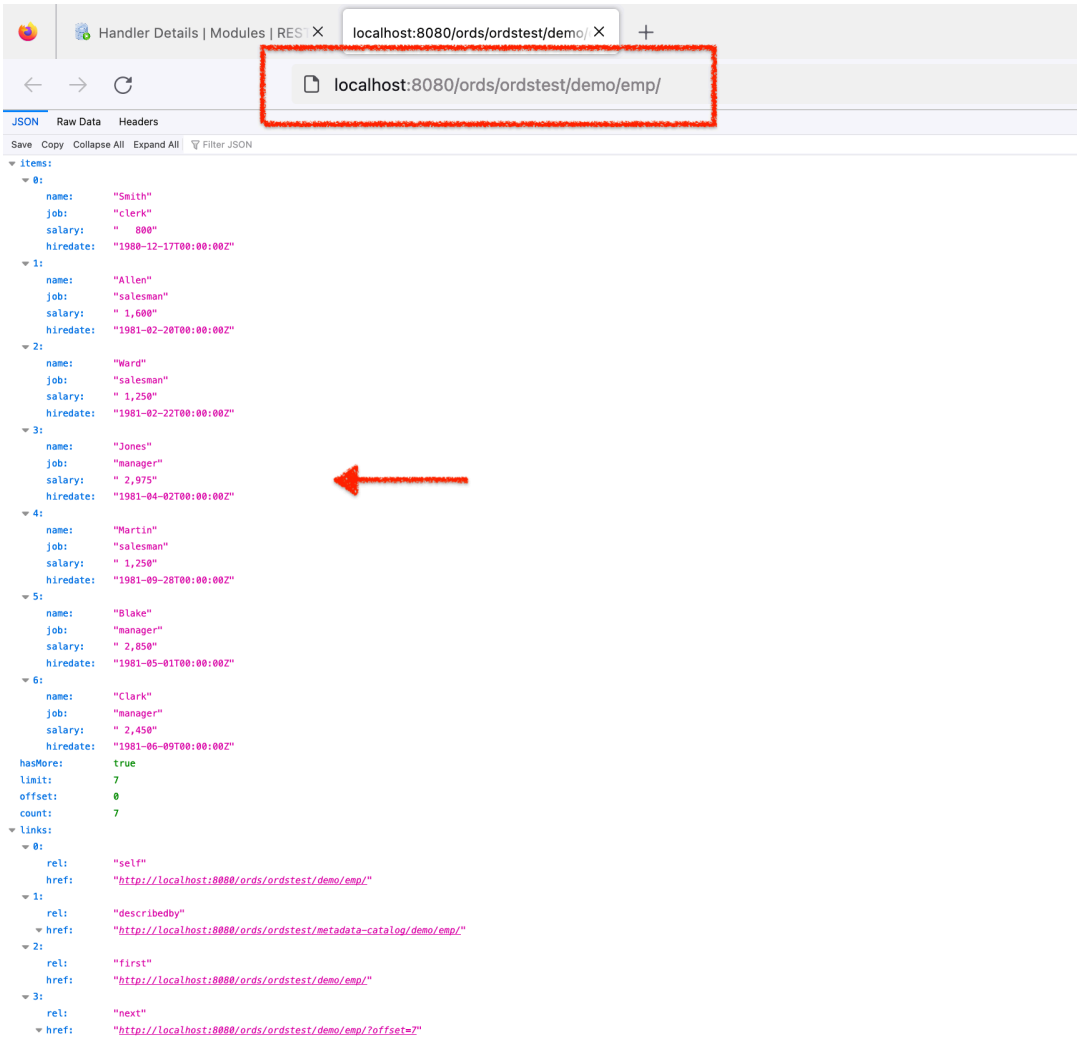
Figure 1-20 New Tab for Testing the Endpoint



2. A new browser tab appears. Enter the Get URI in the browser tab.
3. A JSON response is returned in the browser window.

Copyright © 2011, 2026, Oracle and/or its affiliates.

Figure 1-21 JSON Response from GET Method



1.1.3 Creating a Privilege Using Database Actions

This section describes how to create a privilege and control access to protected resources.

Privileges are defined to control access to protected resources. Privileges restrict access to those users who have access to at least one from a set of the specified roles. A privilege is then associated with one or more resource modules. Before accessing those resource modules, the user must be authenticated and then authorized to ensure that the user has one of the required roles.

- Steps to Create a Privilege
  This section describes the steps to control access to the protected resources.

1.1.3.1 Steps to Create a Privilege

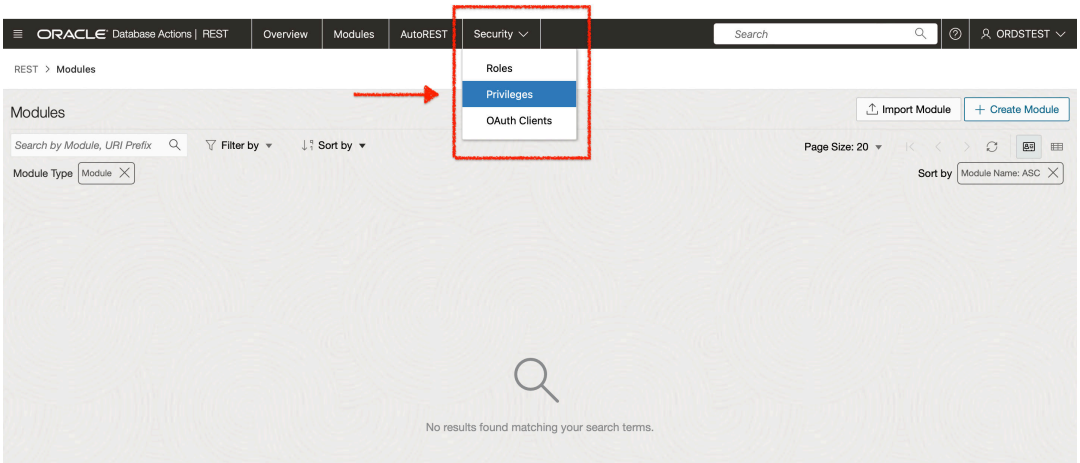
This section describes the steps to control access to the protected resources.

Perform the following steps to create a privilege:

Copyright © 2011, 2026, Oracle and/or its affiliates.

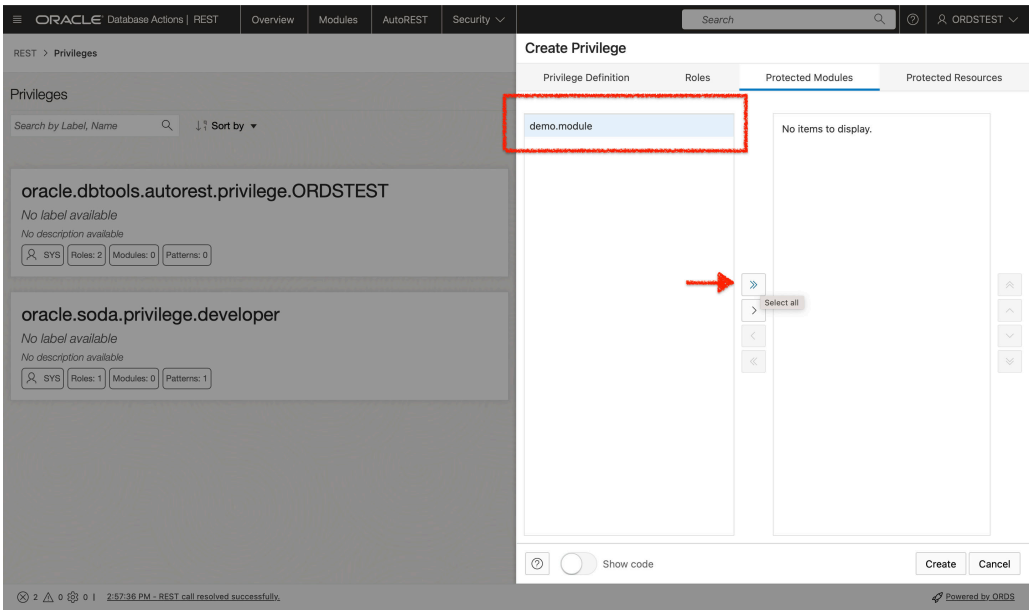
1. Navigate to the REST Workshop
2. Under the **Security** menu item, select **Privileges**.

Figure 1-22 Selecting Privileges Menu Option



3. Click on **Create Privilege** to display the **Create Privilege** screen.
4. Enter the following values in the respective fields:
   - **Label**: `Demo module privilege`
   - **Name**: `demo.module.privilege`
   - **Description**: `A Privilege created for demonstrating privileges for the demo.module Resource Module.`
   - Navigate to the **Protected Modules** table. Move the `demo.module` Resource Module from the left column to the right column. This ensures that the `demo.module` Resource Module is associated with this Privilege.

Figure 1-23 Associating Resource Module with the Privilege

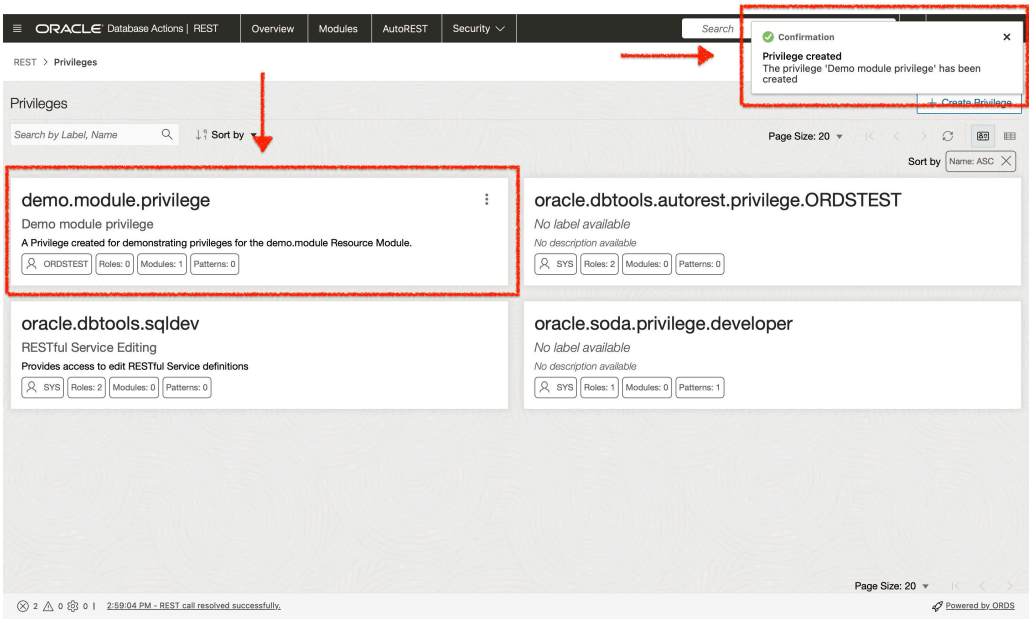


Copyright © 2011, 2026, Oracle and/or its affiliates.

- Click on the **Create** button. A confirmation message appears indicating successful creation of a new privilege.

> **Note**
>
> The newly created privilege can now be viewed in the **Privileges** dashboard.

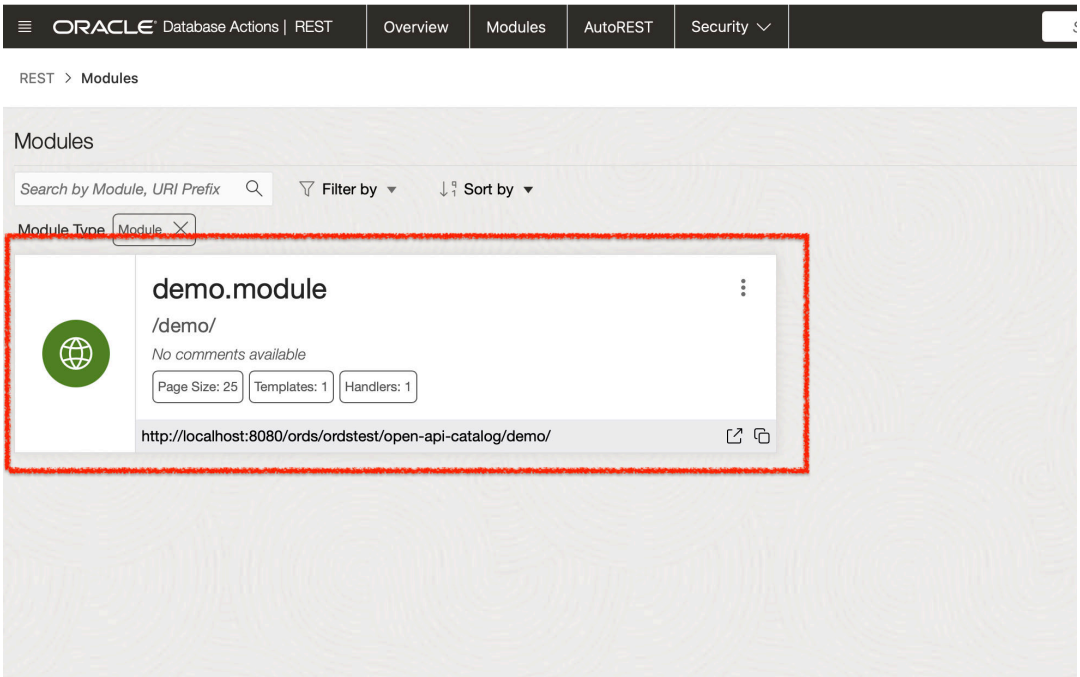
Figure 1-24 Privilege Created



5. To test the Privilege created, navigate to the path of the `demo.module`.

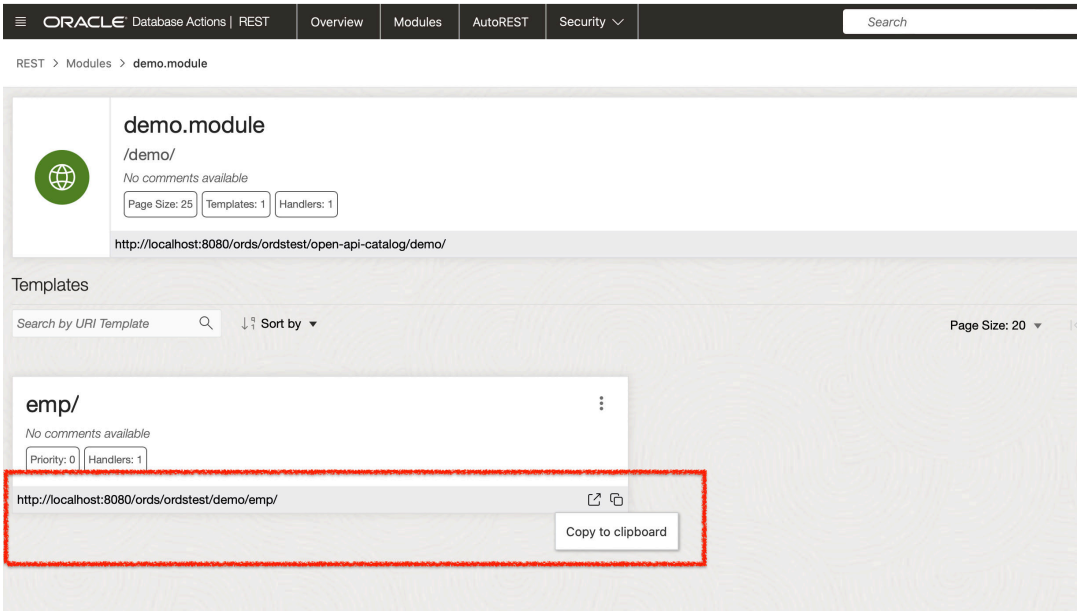
Copyright © 2011, 2026, Oracle and/or its affiliates.

Figure 1-25 Navigating to the demo module to Test the Privilege



6. Copy the `emp/` URI.

Figure 1-26 Copying the URI to Test the Privilege



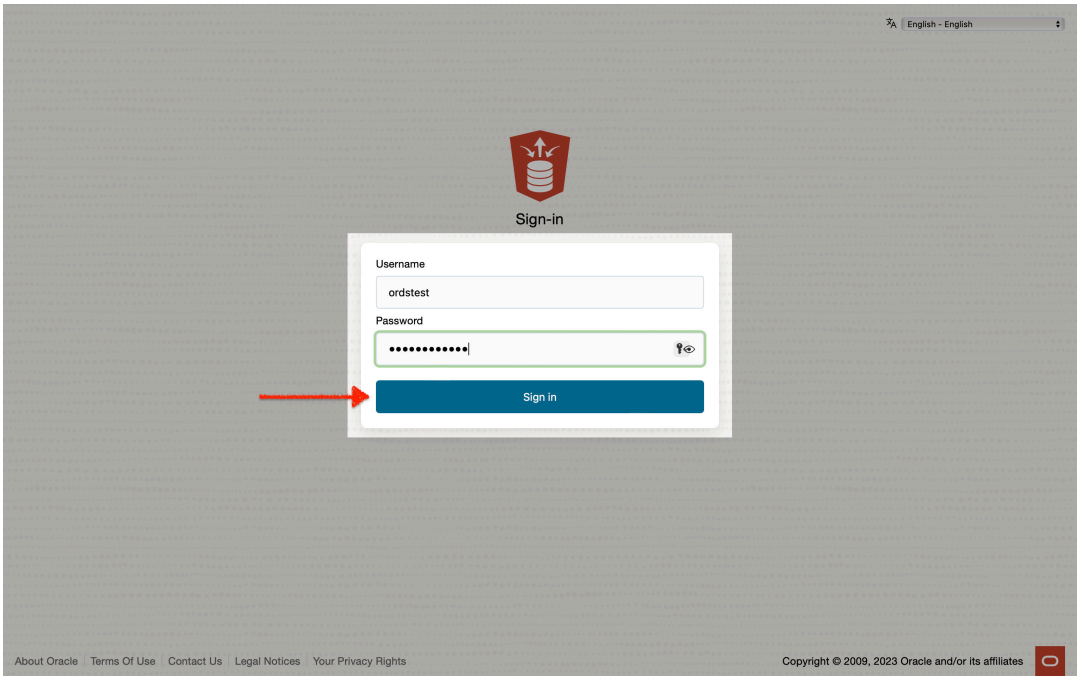
7. Sign out of Database Actions. Open a new browser window. Paste the URI in the address bar and press **Enter**
8. A 401 unauthorized error message is displayed to indicate that the resource is protected. Notice that a sign-in prompt appears. Since this Privilege has not been associated with a

Copyright © 2011, 2026, Oracle and/or its affiliates.

specific role, any user who has been granted the Connect role can sign-in to view the response from this request.

9. Sign-in with your database credentials to view the resource.

Figure 1-27 Sign-in to Test Path Privilege



10. After you sign-in, the contents of the JSON document can be viewed.

1.1.4 Register an OAuth Client Application to Access the REST API

This topic explains how to register your applications (called third-party applications here) to access a REST API.

OAuth 2.0 is a standard internet protocol that provides a means for HTTP servers providing REST APIs to give limited access to third party applications on behalf of an authenticated end user.

Before a third party application can access a REST API:

- It must be registered and
- The authenticated end user must approve access

Prior to registering the application, it must be assigned a user identity so that the user is allowed to register the applications. Users possessing the SQL Developer role are permitted to register OAuth 2.0 clients.

> **Note**
>
> In a real application, you must provision specific users who can register OAuth clients. Such users must be granted the OAuth Client Developer role.

Copyright © 2011, 2026, Oracle and/or its affiliates.

> **Note**
>
> The following example is not intended to serve as a full-featured demonstration for creation and integration for a third party application. The example provided in this section, only outlines the core concepts of the OAuth 2.0 protocol.

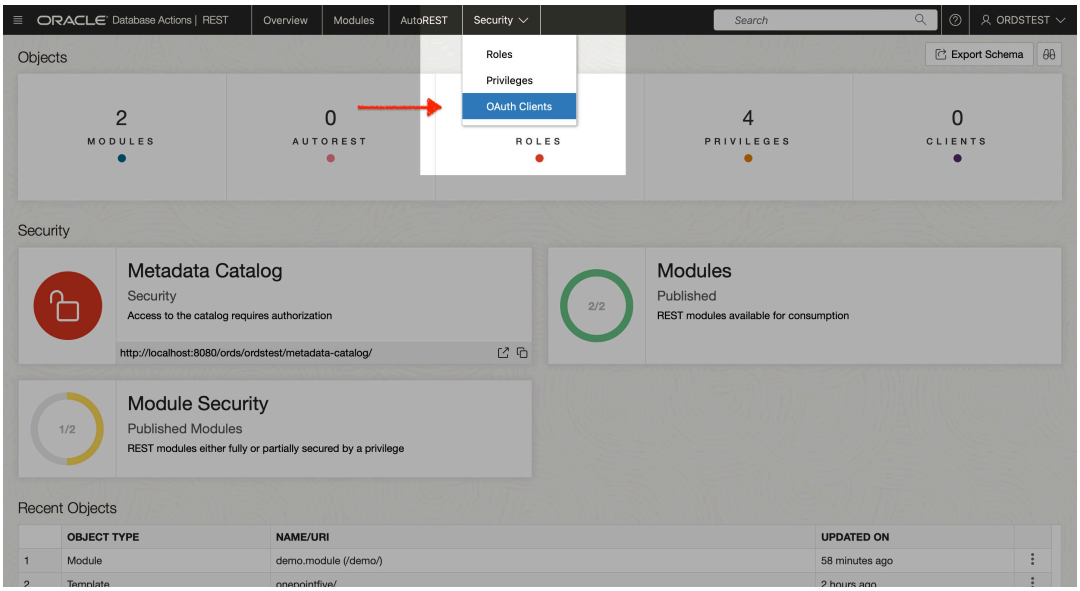
- Registering your Application to Access a REST API

1.1.4.1 Registering your Application to Access a REST API

Perform the following steps to register your application to access a REST API:

1. From the REST Workshop dashboard, select **OAuth Clients** option from the **Security** menu.

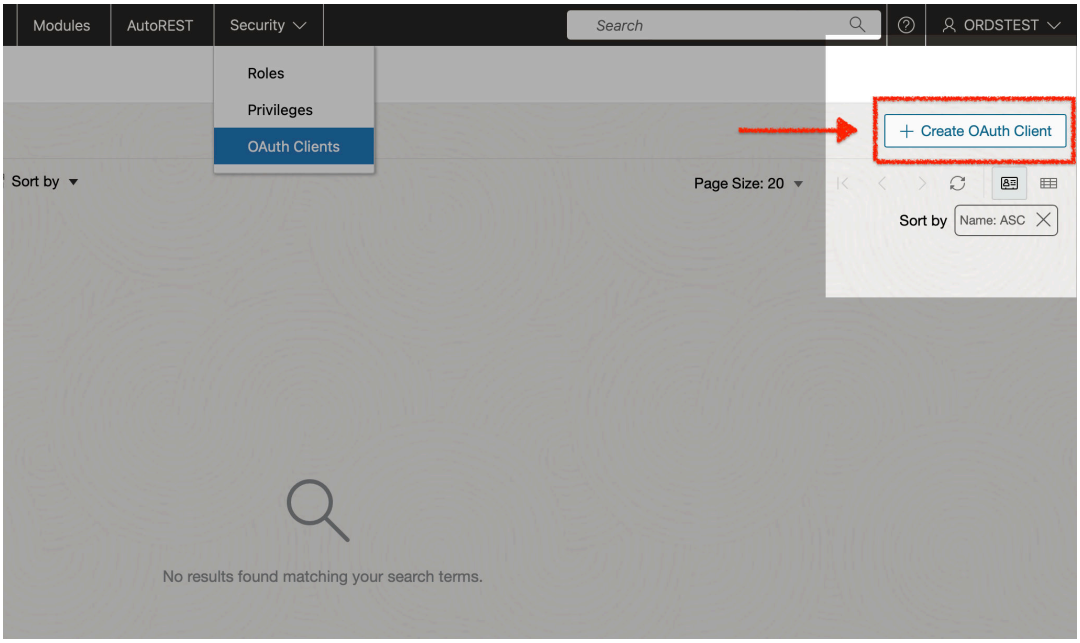
Figure 1-28 Navigating to OAuth Client Menu Option



2. The OAuth Clients dashboard appears. From the OAuth Clients dashboard, click on **Create OAuth Client**.

Copyright © 2011, 2026, Oracle and/or its affiliates.

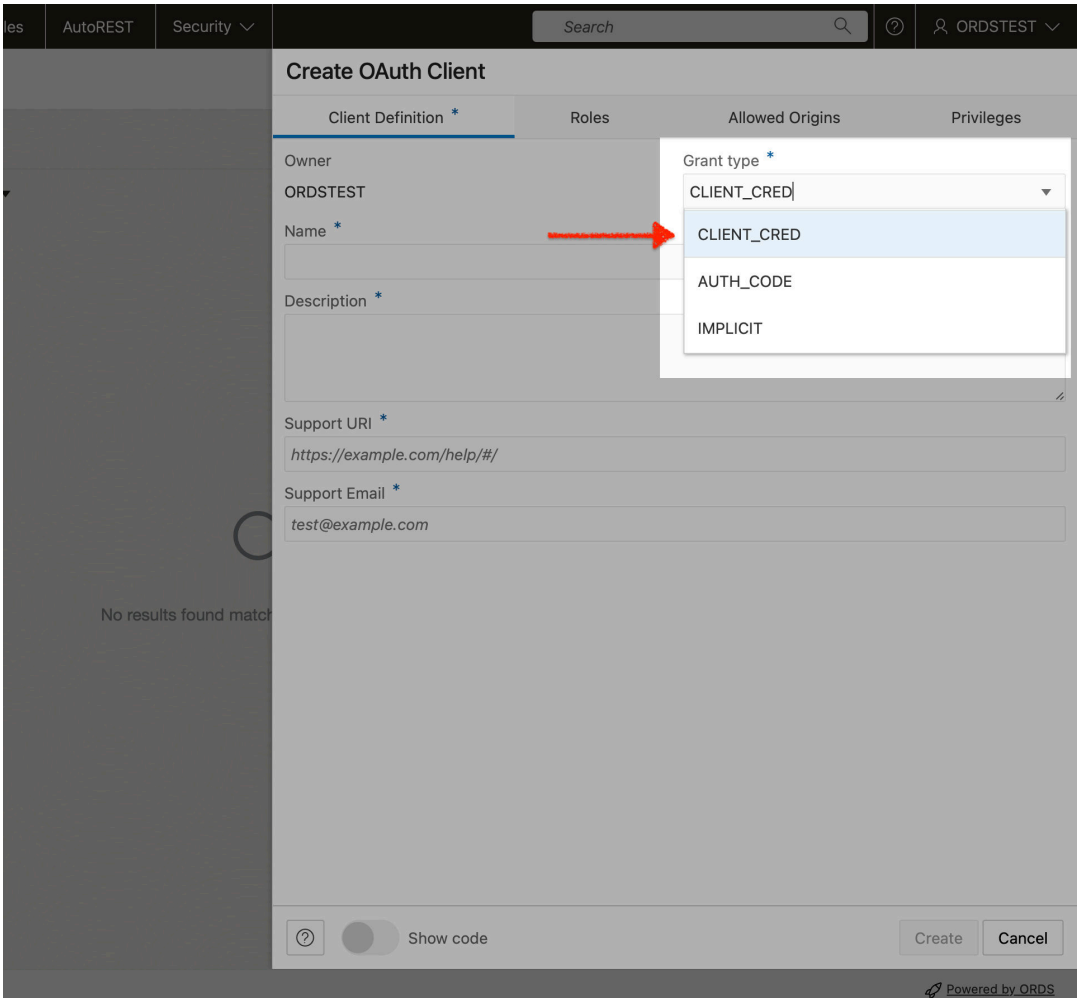
Figure 1-29 Create OAuth Client



3. The **Create OAuth Client** slider appears.

Copyright © 2011, 2026, Oracle and/or its affiliates.

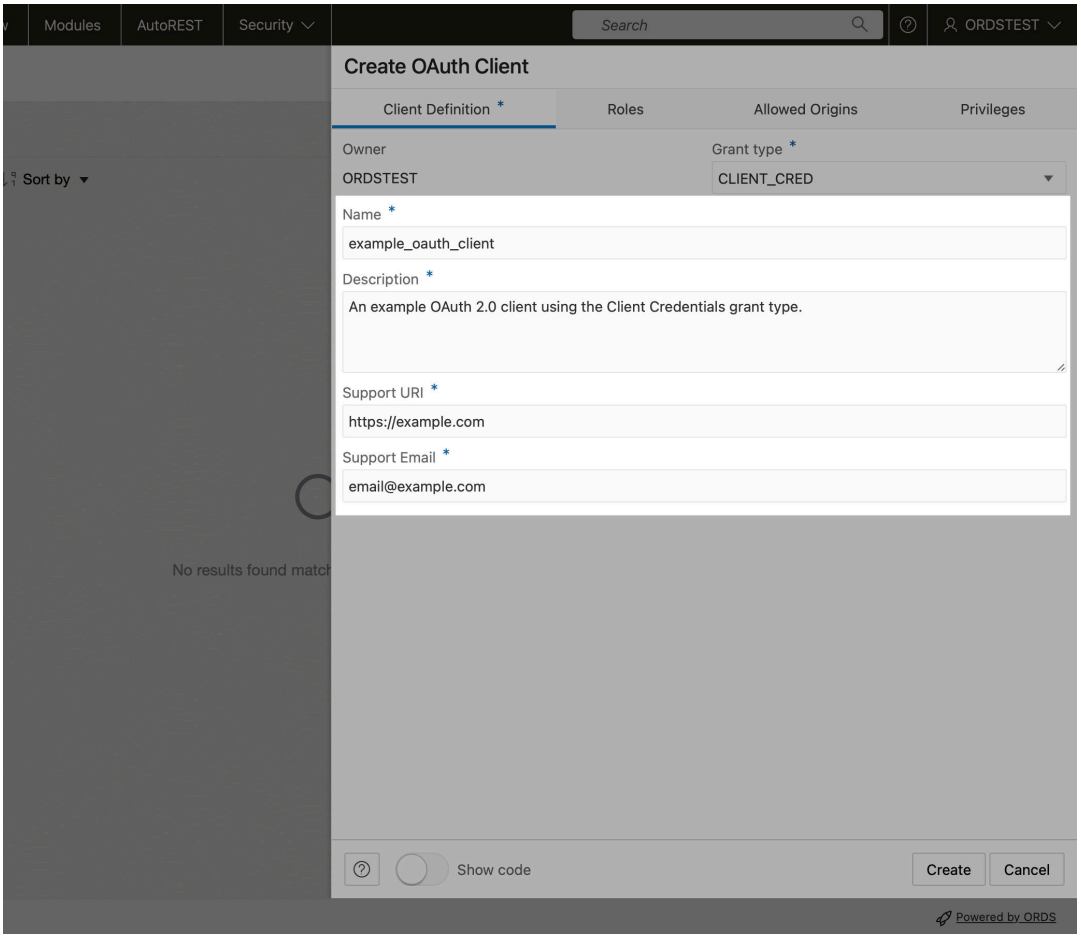
Figure 1-30 Checking Client Credentials Selected



4. Enter the following values in the **Create OAuth Client** slider:
   - **Name**: `example_oauth_client`
   - **Description**: An example OAuth 2.0 client using the Client Credentials grant type.
   - **Support URI**: `https://example.com`
   - **Support Email:** `email@example.com`

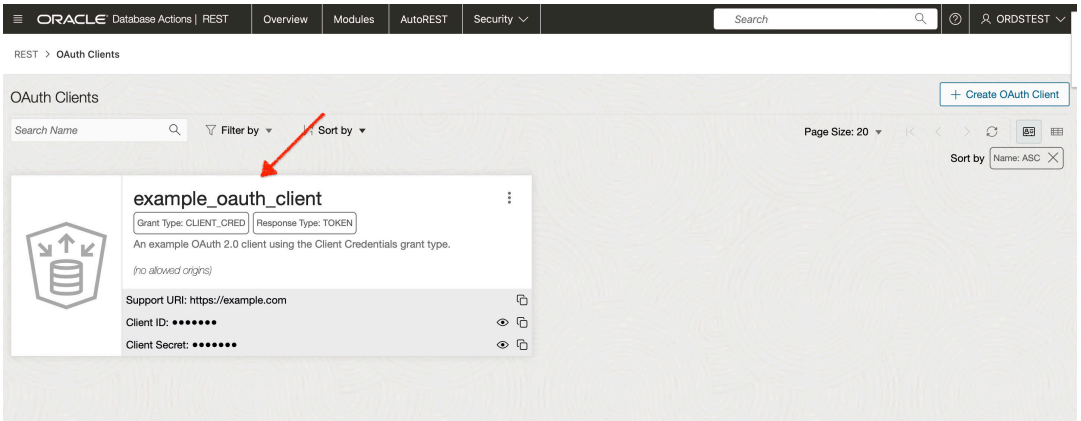
Copyright © 2011, 2026, Oracle and/or its affiliates.

Figure 1-31 Entering Values in Create OAuth Client Slider



5. Click on **Create**.
6. After returning to the **OAuth Clients** dashboard, you can find the newly-created OAuth client.

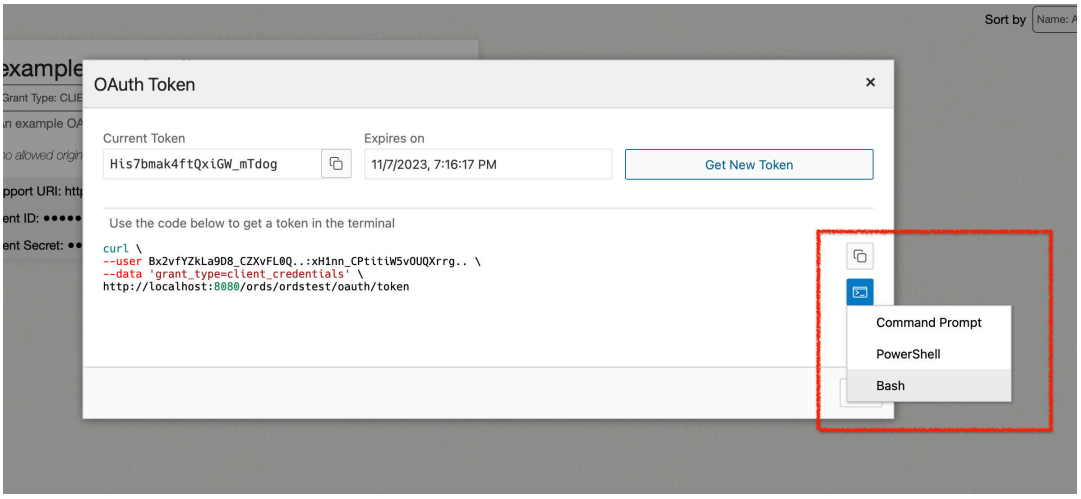
Figure 1-32 OAuth Client Created



7. After the **OAuth Token** appears, select the correct shell environment.

Copyright © 2011, 2026, Oracle and/or its affiliates.

Figure 1-33 Selecting Correct Shell Environment



8. Copy brearer token curl command to your clipboard.
9. Using the Client ID and Client Secret provided to you, issue the following curl command to obtain an Access Token:

```
curl \
--user Bx2vfYZkLa9D8_CZXvFL0Q..:xH1nn_CPtitiW5vOUQXrrg.. \
--data 'grant_type=client_credentials' \
http://localhost:8080/ords/ordstest/oauth/token
```

10. You will receive an Access Token, with expiration time.

Figure 1-34 Access Token with Expiration

```
Last login: Tue Nov  7 12:18:34 on ttys003
choina@choina-mac ~ % curl \
--user Bx2vfYZkLa9D8_CZXvFL0Q..:xH1nn_CPtitiW5vOUQXrrg.. \
--data 'grant_type=client_credentials' \
http://localhost:8080/ords/ordstest/oauth/token
{"access_token":"PWMGyNYtzhg9J1r85_oJGQ","token_type":"bearer","expires_in":3600}%
choina@choina-mac ~ %
```

11. You can now access the `emp/` endpoint. Create your curl command ensuring that you have included the Access Token as a header in your curl command:

```
curl -H "Authorization: Bearer PWMGyNYtzhg9J1r85_oJGQ" http://
localhost:8080/ords/ordstest/demo/emp/ | jq
```


Copyright © 2011, 2026, Oracle and/or its affiliates.

Figure 1-35 jq Response from the Get Request

```
...assword1234@localhost:33431/FREEPDB1 as sysdba ...   ~ — -zsh   ~ — java • ords serve   ~ — -zsh
    "type": "tag:oracle.com,2020:error/Unauthorized",
    "instance": "tag:oracle.com,2020:ecid/KJPTrQ-tIKEPuMU31CoMyQ"
choina@choina-mac ~ % curl -H "Authorization: Bearer PWMGyNYtzhg9J1r85_oJGQ" http://localhost:8080/ords/ordstest/demo/emp/
{"items":[{"name":"Smith","job":"clerk","salary":"   800","hiredate":"1980-12-17T00:00:00Z"},{"name":"Allen","job":"salesman","sal
ary":" 1,600","hiredate":"1981-02-20T00:00:00Z"},{"name":"Ward","job":"salesman","salary":" 1,250","hiredate":"1981-02-22T00:00:00
Z"},{"name":"Jones","job":"manager","salary":" 2,975","hiredate":"1981-04-02T00:00:00Z"},{"name":"Martin","job":"salesman","salary
":" 1,250","hiredate":"1981-09-28T00:00:00Z"},{"name":"Blake","job":"manager","salary":" 2,850","hiredate":"1981-05-01T00:00:00Z"}
,{"name":"Clark","job":"manager","salary":" 2,450","hiredate":"1981-06-09T00:00:00Z"}],"hasMore":true,"limit":7,"offset":0,"count"
:7,"links":[{"rel":"self","href":"http://localhost:8080/ords/ordstest/demo/emp/"},{"rel":"describedby","href":"http://localhost:80
80/ords/ordstest/metadata-catalog/demo/emp/"},{"rel":"first","href":"http://localhost:8080/ords/ordstest/demo/emp/"},{"rel":"next"
,"href":"http://localhost:8080/ords/ordstest/demo/emp/?offset=7"}]}%
choina@choina-mac ~ % curl -H "Authorization: Bearer PWMGyNYtzhg9J1r85_oJGQ" http://localhost:8080/ords/ordstest/demo/emp/ | jq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   977    0   977    0     0  16040      0 --:--:-- --:--:-- --:--:-- 18788
{
  "items": [
    {
      "name": "Smith",
      "job": "clerk",
      "salary": "   800",
      "hiredate": "1980-12-17T00:00:00Z"
    },
    {
      "name": "Allen",
      "job": "salesman",
      "salary": " 1,600",
      "hiredate": "1981-02-20T00:00:00Z"
    },
    {
      "name": "Ward",
      "job": "salesman",
      "salary": " 1,250",
      "hiredate": "1981-02-22T00:00:00Z"
    },
    {
      "name": "Jones",
```

> **Note**
>
> You can *optionally* pipe in the `jq` command so that the JSON response payload is structured in a readable format.

1.1.5 Creating a RESTful Service Using Oracle SQL Developer

This section describes the steps involved in developing the RESTful services using Oracle SQL developer desktop application (aka client).

Topics:

- REST-Enable a Database Table
- Creating a RESTful Service through the Connections Navigator
- Creating a RESTful Service from a SQL Query
- Protect Resources
- Register an OAuth Client Application

- REST-Enable a Database Table
- Creating a RESTful Service through the Connections Navigator
- Creating a RESTful Service from a SQL Query
- Protect Resources
- Register an OAuth Client Application

Copyright © 2011, 2026, Oracle and/or its affiliates.

1.1.5.1 REST-Enable a Database Table

To enable a table for REST access, follow these steps.

> **Note**
>
> It is recommended that you follow the steps as closely as possible, including using the specified names for schemas and database objects. After you have successfully completed the tutorial using this approach, feel free to try it again using other values if you wish.

1. Create a user `ordstest` with the following privileges or roles:

```
CREATE USER ordstest IDENTIFIED BY <password>;
GRANT "CONNECT" TO ordstest;
GRANT "RESOURCE" TO ordstest;
GRANT UNLIMITED TABLESPACE TO ordstest
```

2. Connect to the `ordstest` schema. In SQL Developer create a connection to the `ordstest` schema, connect to it, and open a SQL worksheet.

3. Create a database table. For example, enter the following in the SQL Worksheet to create an example table named EMP:

```
CREATE TABLE EMP (
  EMPNO NUMBER(4,0),
  ENAME VARCHAR2(10 BYTE),
  JOB VARCHAR2(9 BYTE),
  MGR NUMBER(4,0),
  HIREDATE DATE,
  SAL NUMBER(7,2),
  COMM NUMBER(7,2),
  DEPTNO NUMBER(2,0),
  CONSTRAINT PK_EMP PRIMARY KEY (EMPNO)
  );
```

4. Insert some sample data into the table. For example:

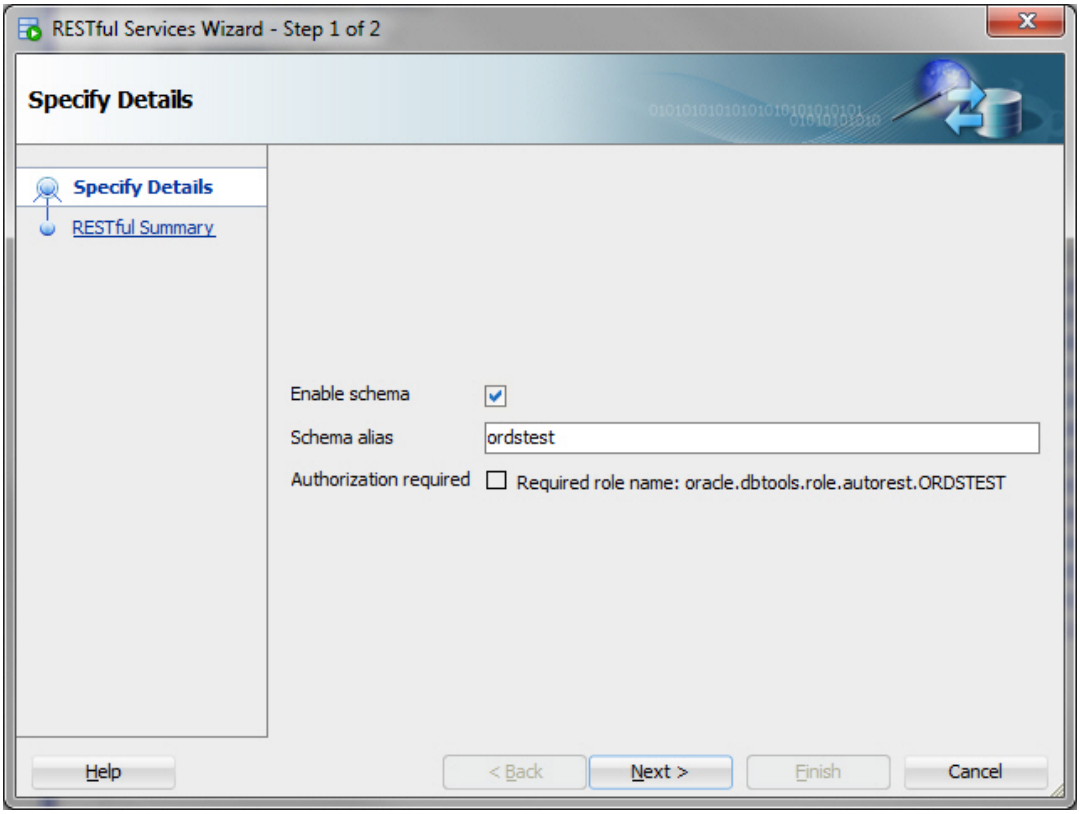
```
Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values
(7369,'SMITH','CLERK',7902,to_date('17-DEC-80','DD-MON-RR'),800,null,20);
Insert into EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO) values
(7499,'ALLEN','SALESMAN',7698,to_date('20-FEB-81','DD-MON-RR'),1600,300,30);
Insert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


Copyright © 2011, 2026, Oracle and/or its affiliates.

```
(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;
```

5. Enable the schema of the EMP table for REST. In SQL Developer, right-click the `ordstest` connection, and select **REST Services > Enable RESTful Services** to display the following wizard page:

Figure 1-36 Enabling the Schema of the EMP Table for REST



Enable schema: Enable this option.

Schema alias: Accept `ordstest` for the schema alias.

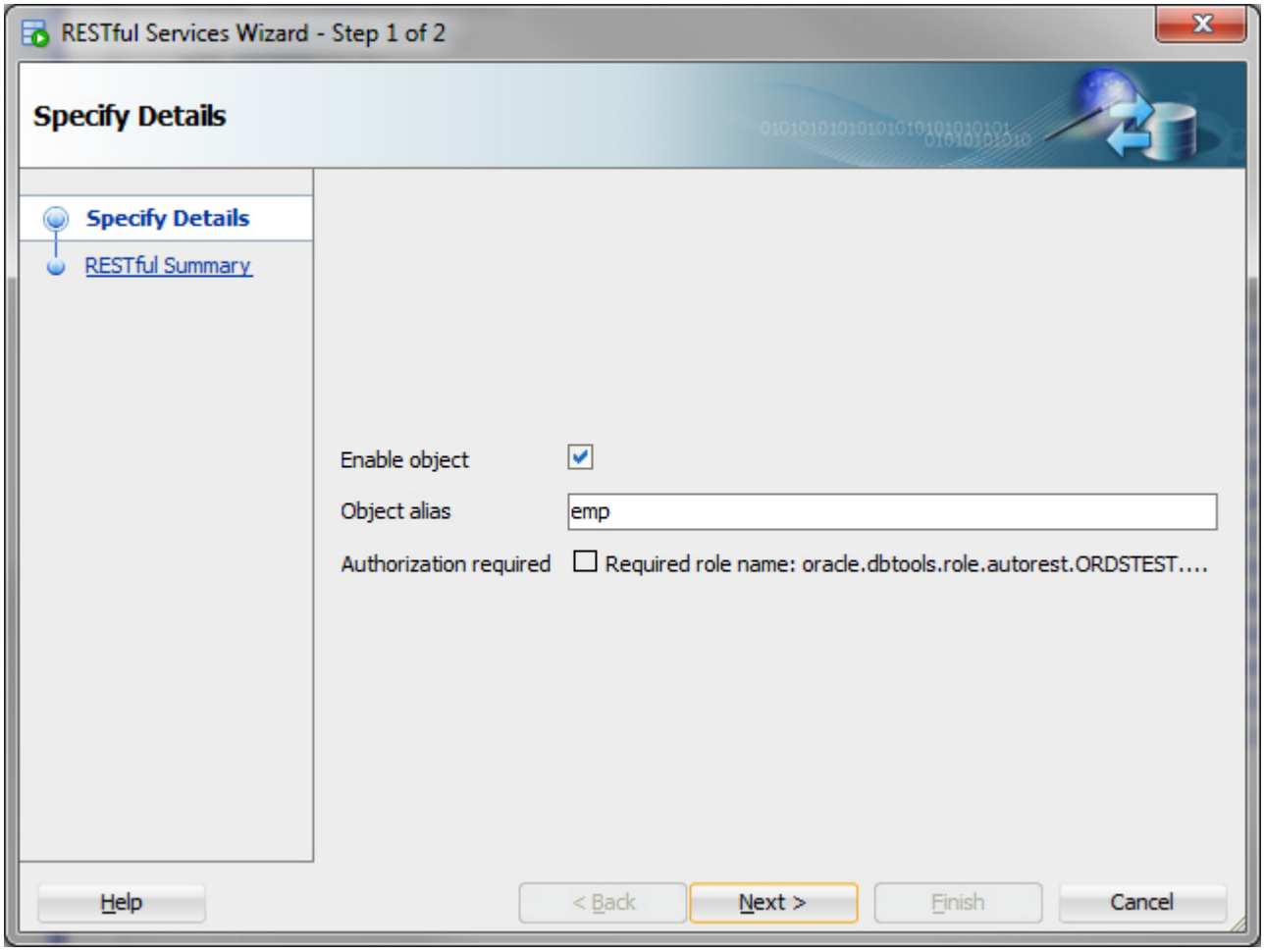
Authorization required: For simplicity, this tutorial does not require authorization, so disable this option.

Click **Next**.

6. On the RESTful Summary page of the wizard, click **Finish**.
7. Enable the EMP table. In SQL Developer, right-click EMP table in the Connections navigator, and select **REST Services > Enable RESTful Services** to display the following wizard page:

Copyright © 2011, 2026, Oracle and/or its affiliates.

Figure 1-37 REST Enabling the EMP Table



Enable object: Enable this option (that is, enable REST access for the EMP table).

Object alias: Accept `emp` for the object alias.

Authorization required: For simplicity, this tutorial does not require authorisation, so disable this option.

8. On the RESTful Summary page of the wizard, click **Finish**.

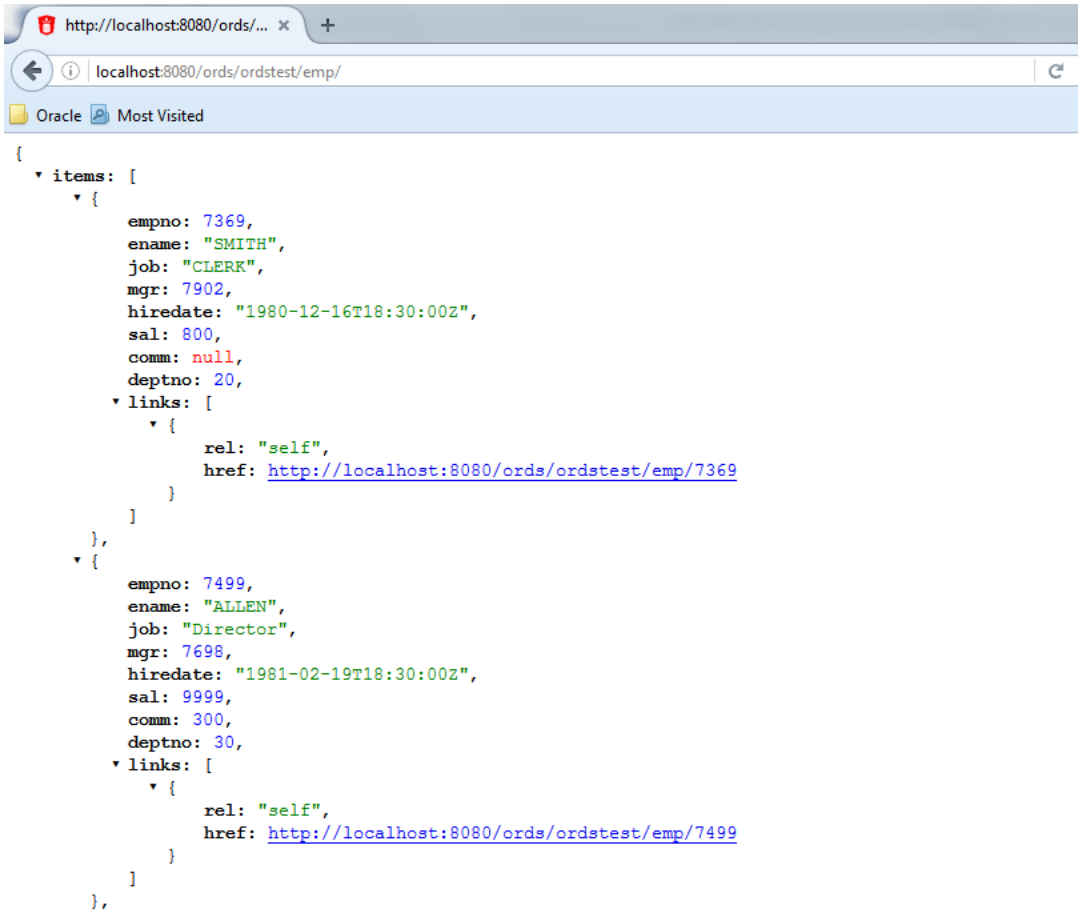
   The EMP table is now exposed as a REST HTTP endpoint .

   > **Note**
   >
   > DELETE, PUT, POST, and metadata-catalog endpoints are also auto-generated.

9. Test the REST endpoint. In a web browser, enter the URL `http://localhost:8080/ords/ordstest/emp/` as shown in the following figure:
   - The ORDSTEST schema has been exposed at the `/ordstest/` path.
   - The EMP table has been exposed at the `/emp/` path.

Copyright © 2011, 2026, Oracle and/or its affiliates.

Figure 1-38 Testing the REST Enabled Table



Related Topics

- Automatic Enabling of Schema Objects for REST Access (AutoREST)

1.1.5.2 Creating a RESTful Service through the Connections Navigator

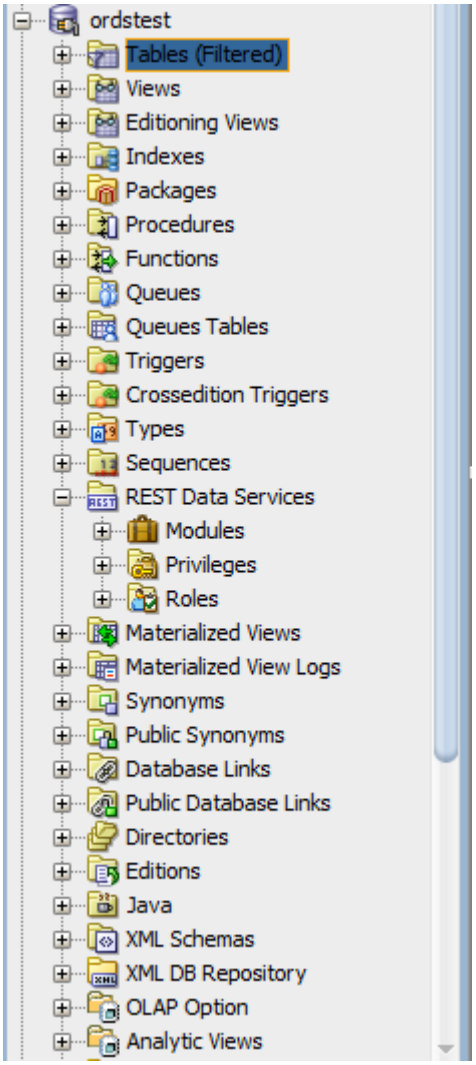
This section explains how to create a RESTful service by using REST Data Services node in the Connections navigator. Oracle REST Data Services provides an option through the Connections navigator that enables you to create and edit RESTful service definitions.

To create and test a RESTful service by using REST Data Services node in the Connections navigator, follow these steps:

1. Under **ordstest** schema, select **REST Data Services.**

Copyright © 2011, 2026, Oracle and/or its affiliates.

Figure 1-39 REST Data Services option under Connections Navigator

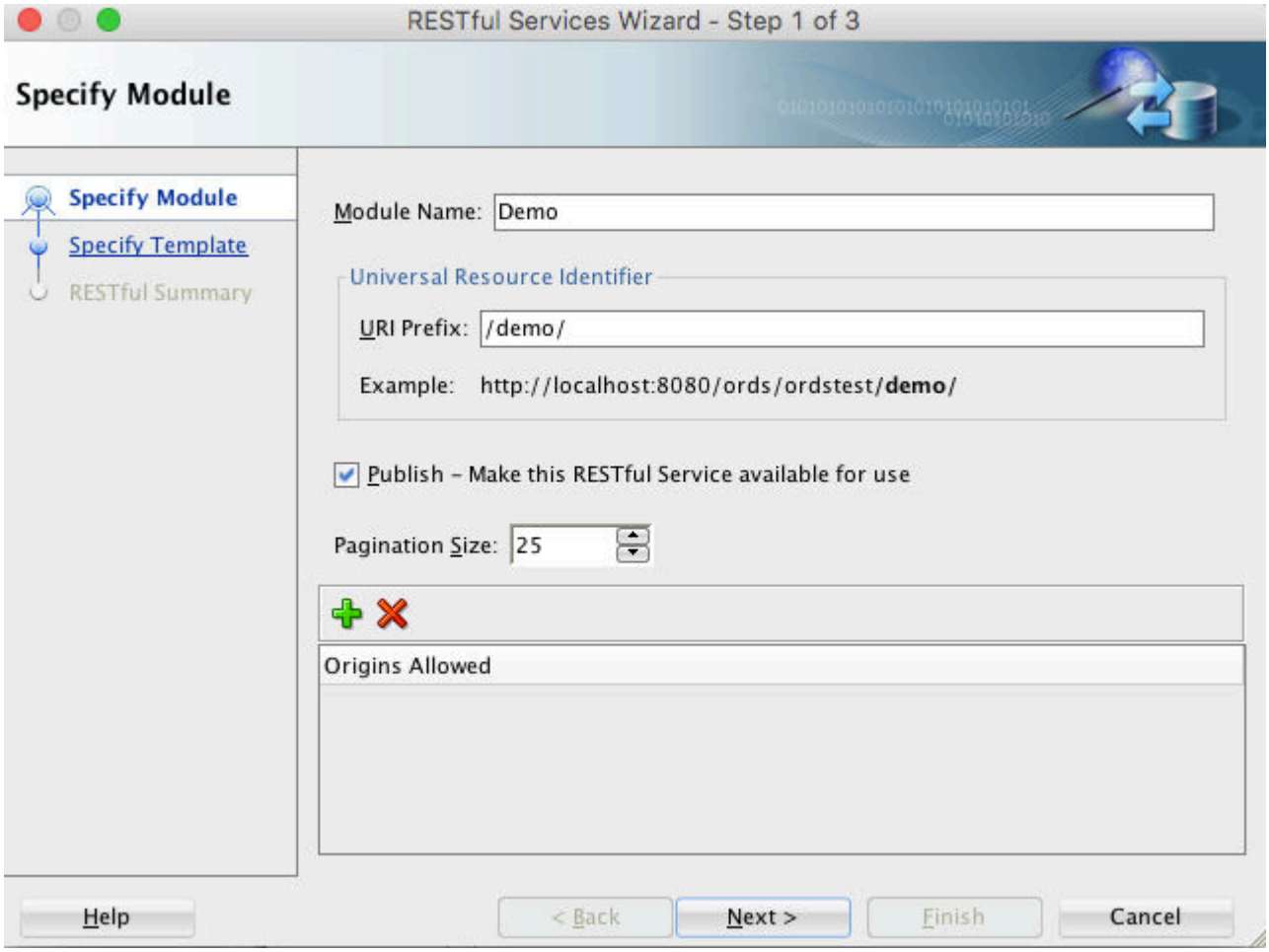


The following steps create and test the RESTful service.

2. Under **REST Data Services** node, right-click the `Modules` node, click **New Module**, and enter information on the Specify Module page:

Copyright © 2011, 2026, Oracle and/or its affiliates.

Figure 1-40 Entering Information on the Specify Module Page



Module Name: Any desired name for the connection. For example, `demo`

URI Prefix: `/demo/`

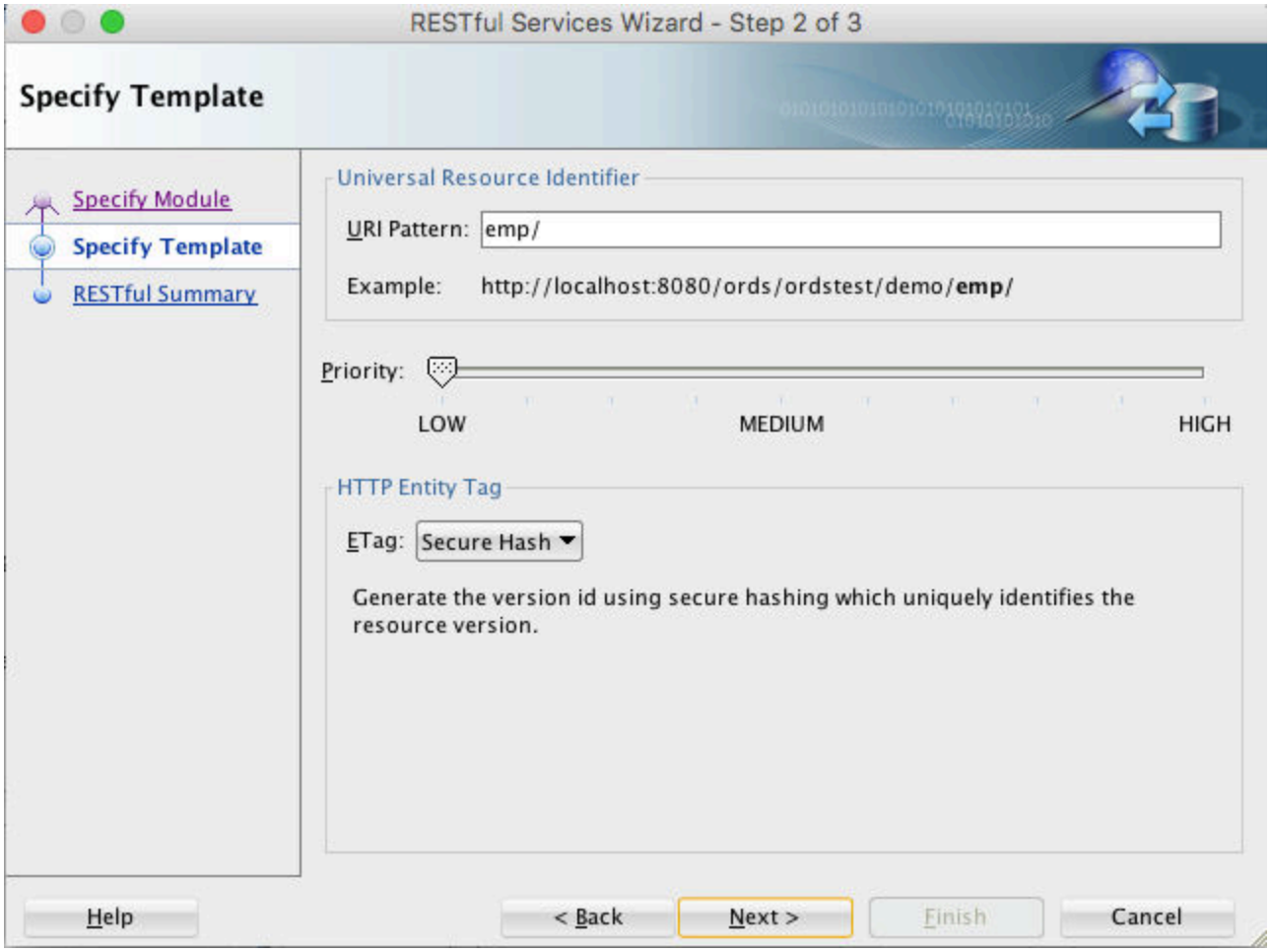
Publish - Make this RESTful Service available for use: Enable (check).

Pagination Size: `25`

3. Click **Next**, and enter information on the Specify Template page:

Copyright © 2011, 2026, Oracle and/or its affiliates.

Figure 1-41 Entering Information on the Specify Template Page



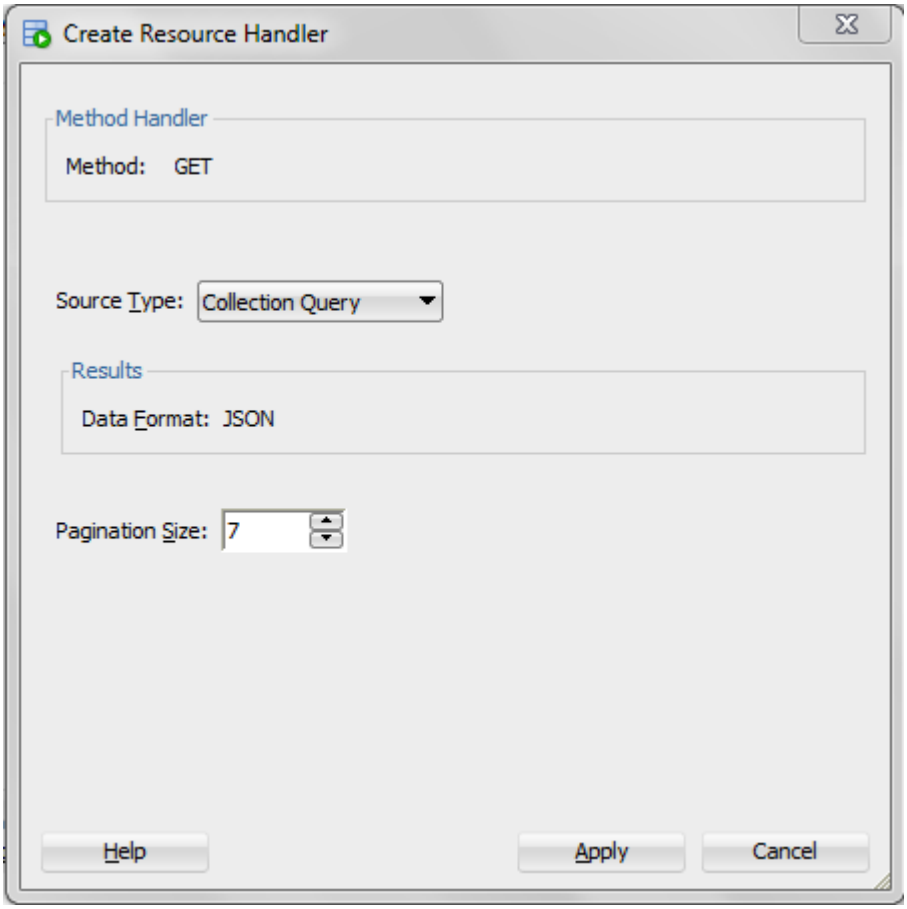
URI Pattern: `emp/`

Accept the defaults for the remaining options.

4. Click Next to go to the RESTful Summary page of the wizard, then click **Finish**. Expand the Modules node to display the resource module that you created.
5. Expand the module, Demo and right click on the `emp/`node, select **Add** handler and then select **GET** method.
6. Enter the information on the Create Resource Handler page.

Copyright © 2011, 2026, Oracle and/or its affiliates.

Figure 1-42 Entering Information on Create Resource Handler Page:



Source Type: `Collection Query`

Pagination Size: `7`

Click **Apply**.

Next step is to define the query for the GET resource handler.

7. In the SQL Worksheet, enter the following query:

```
SELECT
   INITCAP(ENAME) name,
  lower(job) job,
  TO_CHAR(sal,'9G999','NLS_NUMERIC_CHARACTERS=",."') salary,
  hiredate
FROM
emp
```

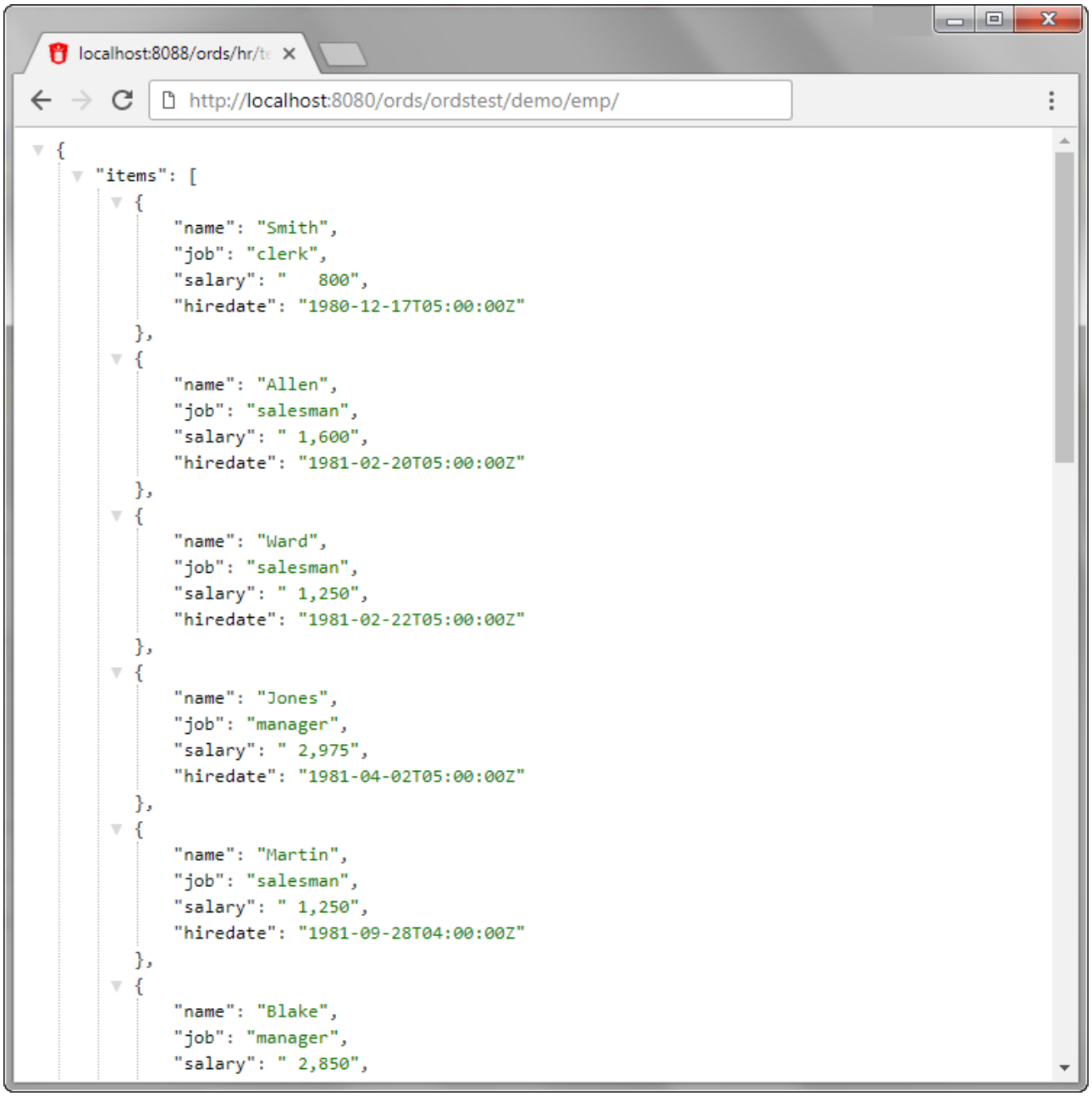
8. Click **Save REST Handler** icon. A confirmation message appears in the **Messages - Log** pane to confirm that the handler is saved to the database.

Copyright © 2011, 2026, Oracle and/or its affiliates.

> **Note**
>
> If you do not see the Messages - Log pane, go to the **View** menu and then select **Log**.

9. Test the RESTful service. In a web browser enter the URL http://localhost:8080/ords/ordstest/demo/emp/ as shown in the following figure:
   - The ORDSTEST schema has been exposed at the /ordstest/ path.
   - The query has been exposed at the /demo/emp/ path.

Figure 1-43 Testing URL in a Web Browser




```
{
  "items": [
    {
        "name": "Smith",
        "job": "clerk",
        "salary": "   800",
        "hiredate": "1980-12-17T05:00:00Z"
    },
    {
        "name": "Allen",
        "job": "salesman",
        "salary": " 1,600",
        "hiredate": "1981-02-20T05:00:00Z"
    },
    {
        "name": "Ward",
        "job": "salesman",
        "salary": " 1,250",
        "hiredate": "1981-02-22T05:00:00Z"
    },
    {
        "name": "Jones",
        "job": "manager",
        "salary": " 2,975",
        "hiredate": "1981-04-02T05:00:00Z"
    },
    {
        "name": "Martin",
        "job": "salesman",
        "salary": " 1,250",
        "hiredate": "1981-09-28T04:00:00Z"
    },
    {
        "name": "Blake",
        "job": "manager",
        "salary": " 2,850",
```


Copyright © 2011, 2026, Oracle and/or its affiliates.

- Creating a Privilege under REST Data Services
- Creating a Role
  This section explains how to create and delete a role.

Related Topics

- Creating a RESTful Service from a SQL Query

1.1.5.2.1 Creating a Privilege under REST Data Services

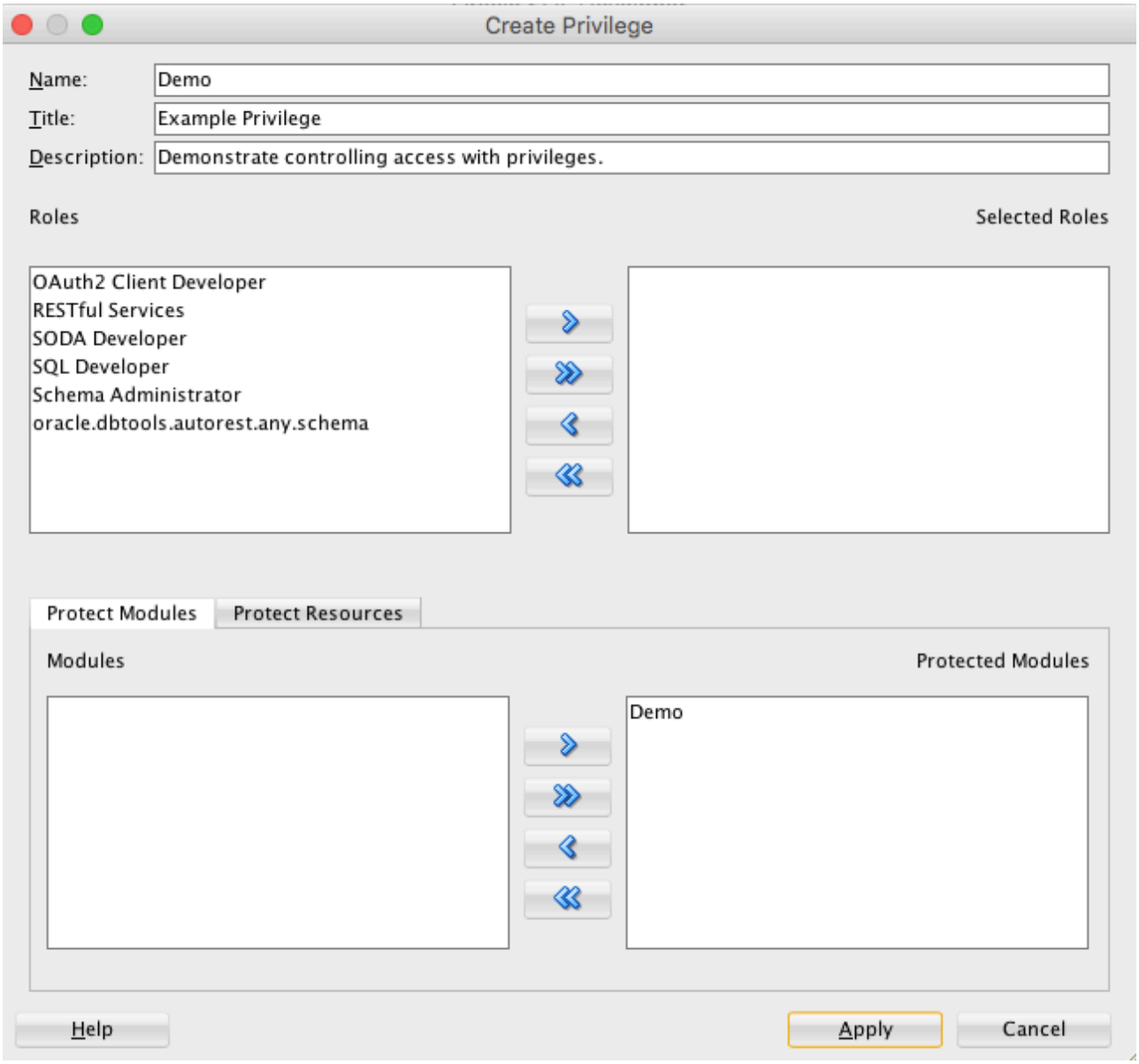
Controlling access to protected resources is done by defining privileges. **Privileges** restrict access to only users having at least one of a set of specified roles. A privilege is then associated with one or more resource modules: before those resource modules can be accessed, the user must be authenticated and then authorized to ensure that the user has one of the required roles.

To protect resources, follow these steps.

1. Create a privilege. In SQL Developer, right-click the `Privileges` node under REST Data Services and select **New Privileges** to display the Create Privilege dialog box:

Copyright © 2011, 2026, Oracle and/or its affiliates.

Figure 1-44 Create Privilege Dialog Box



Name: `Demo`

Title: `Example Privilege`

Description: `Demonstrate controlling access with privileges`

Protected Modules: Ensure that the list includes the `Demo` module. Use the arrow button to move it if necessary.

Click **Apply**.

You have now created a privilege that protects the demo module. However, you have not restricted the privilege to any particular role; this will just require that the user be authenticated before accessing the demo module (the next step).

2. Test the RESTful service. In a web browser enter the following URL:

   http://localhost:port/ords/ordstest/demo/emp/

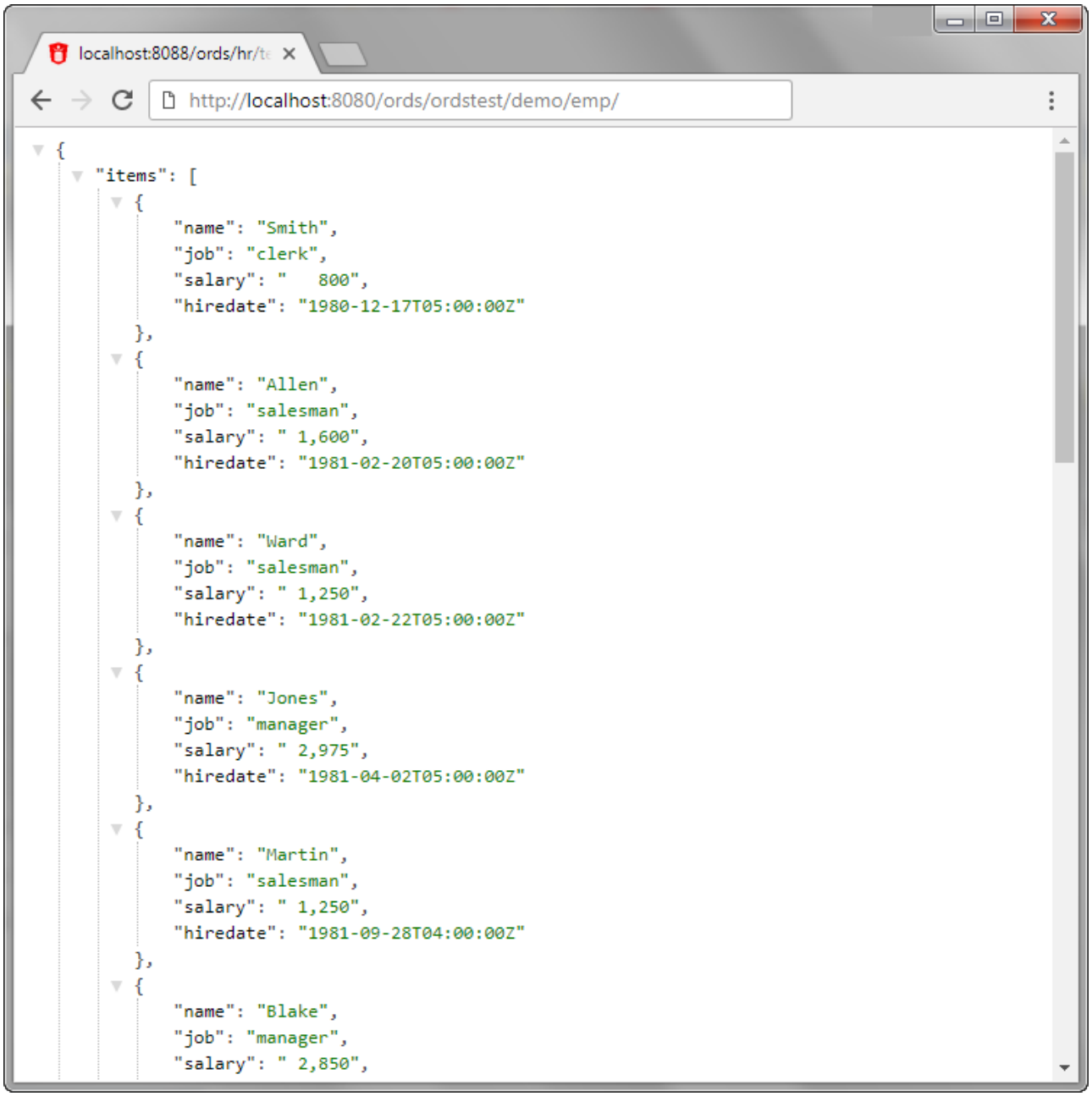
Copyright © 2011, 2026, Oracle and/or its affiliates.

3. Click the link to sign in, and enter the `test_developer` credentials.

> **Note**
>
> To create a test_developer user refer to **Create a RESTful Service from a SQL Query** section.

A JSON document similar to the following is displayed:

Figure 1-45 JSON Document After Signing in



Copyright © 2011, 2026, Oracle and/or its affiliates.

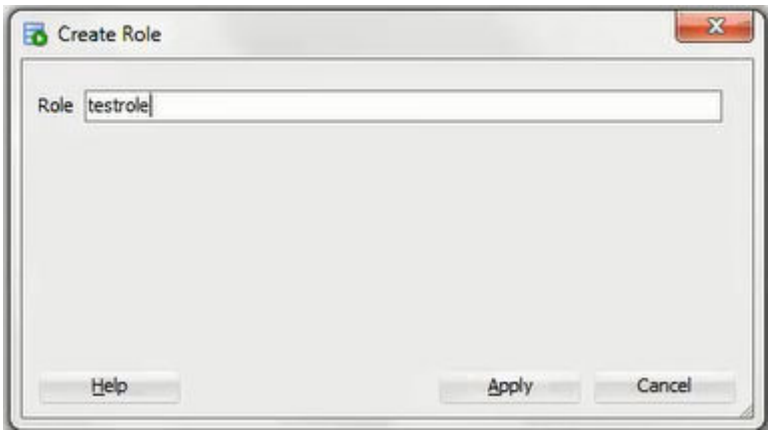
1.1.5.2.2 Creating a Role

This section explains how to create and delete a role.

To create a role, follow these steps:

1. Under **REST Data Services**, right click **Roles** and then click **New Role**.
2. In the **Create Role** dialog box, enter the name of the role you want to create.

Figure 1-46 Entering the New Role Name



3. Click **Apply**, the new role is now created.

   To rename or delete a role, right click on the role name and choose one of the following options:

   - **Rename**: to change the role name.
   - **Delete**: to remove the role.

1.1.5.3 Creating a RESTful Service from a SQL Query

Oracle REST Data Services provides a REST API (called the Resource Modules API) that enables Oracle SQL Developer to create and edit RESTful service definitions. This option is available when you do not have direct access to the database. Access to the Resource Modules API is protected, a user with the correct role must be provisioned, and the created user's credentials must be used when accessing the API from SQL Developer.

To create a RESTful service from a SQL query, follow these steps.

1. In the folder where Oracle REST Data Services was installed, enter the following command at a command prompt:

Copyright © 2011, 2026, Oracle and/or its affiliates.

> **Note**
>
> - If the `ORDS_CONFIG` environment variable has not been set, then run the following command:
>
>  ```
>  ords --config [/directory/path/to/ords_config] config user add
>  [Username] [Role]
>  ```
>
> – For multiple roles, specify `["Role_01, Role_02, Role_03, and so on"]`
>
> - If the `ORDS_CONFIG` environment variable has already been set (that is, you would not need to include the `--config [/directory/path/to/ords_config]` option), then run the following command:
>
>  ```
>  ords config user add [Username] [Role]
>  ```
>
> – For multiple roles, specify `["Role_01, Role_02, Role_03, and so on"]`.
>
> The text within square brackets [] is intended to be replaced with user-defined input or directory path.

```
ords config user add test_developer "SQL Developer"
```

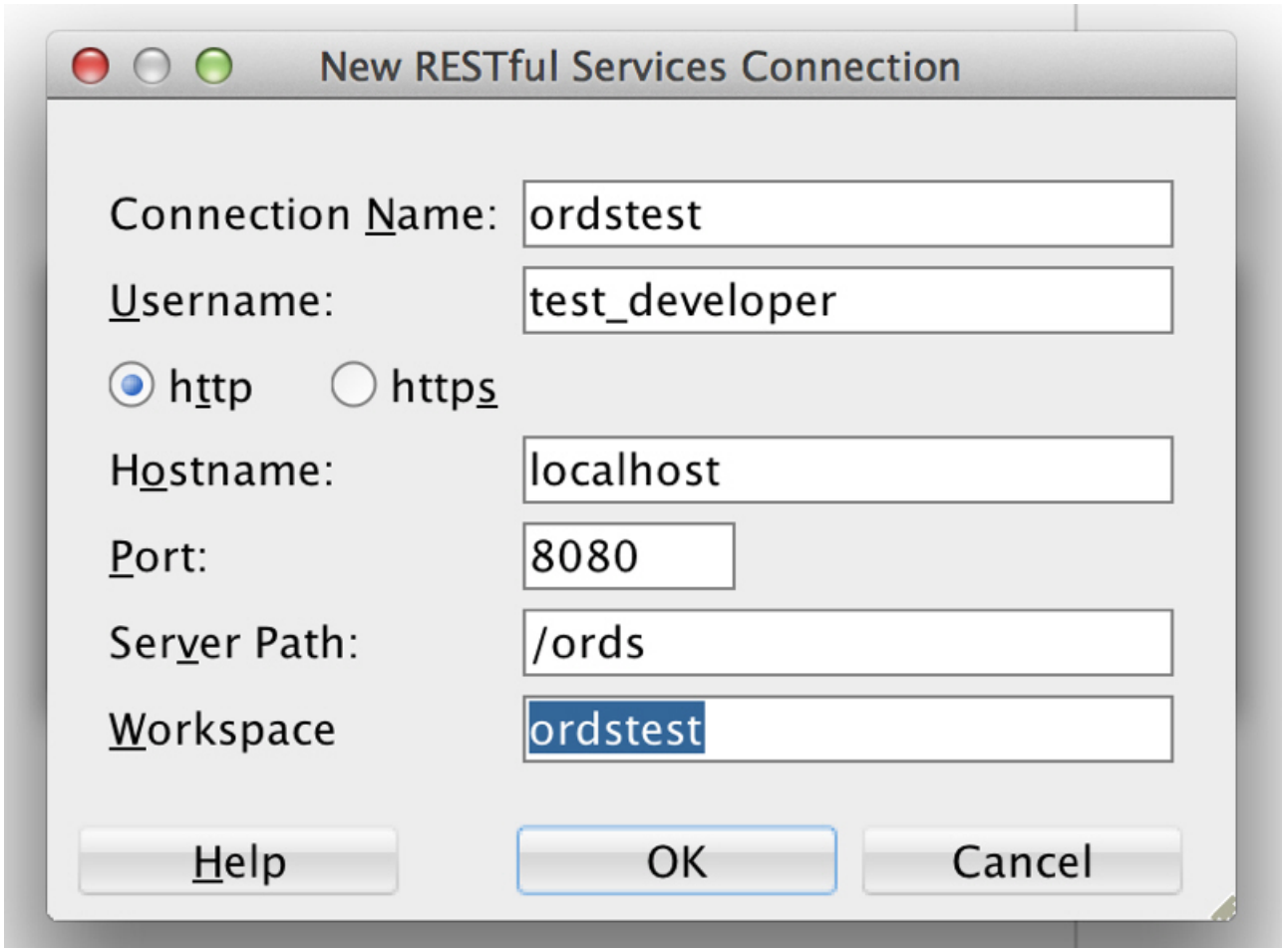
- You will be prompted to enter a password.
- This command creates a user named `test_developer` and grants the user the role named `SQL Developer`. Only users with the `SQL Developer` role are permitted to access the resource module's API.
- The user details are stored in a file named `credentials` in the ORDS configuration folder. However, it is not recommended to store user credentials in the credentials file in production deployments; instead, users should be provisioned in the host application server.

The remaining steps create and test the RESTful service.

2. Create RESTful connection. In SQL Developer, select **View > REST Data Services > Development**.
3. In the REST Development pane, right-click **REST Data Services > Connect**.
4. In the RESTful Services Connection dialog box, click the **+** (plus sign) icon to add a connection to the list available for selection.
5. In the New RESTful Services Connection dialog box, enter the necessary information:

Copyright © 2011, 2026, Oracle and/or its affiliates.

Figure 1-47 Entering Information for New RESTful Services Connection



Connection Name: Any desired name for the connection. Example: `ordstest`

Username: `test_developer`

http or **https**: Select `http` for simplicity in this tutorial.

Hostname: `localhost`

Port: `8080`

Server Path: `/ords`

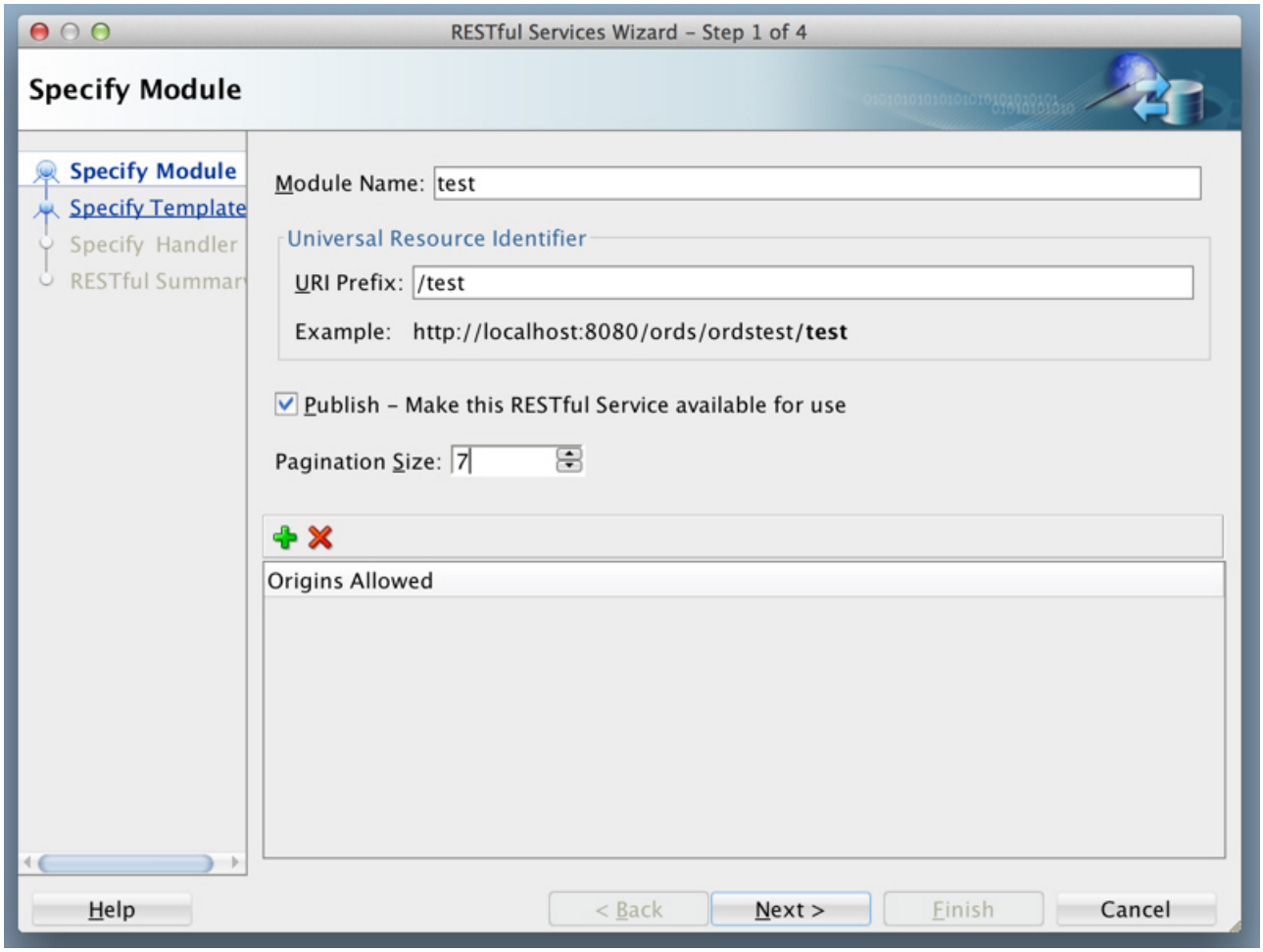
Workspace: `ordstest`

Click **OK**, then enter the password for the `test_developer` user at the prompt.

6. Create the module. Right-click the `Modules` node in the REST Development view, click **New Module**, and enter information on the Specify Module page:

Copyright © 2011, 2026, Oracle and/or its affiliates.

Figure 1-48 Entering Information on the Specify Module Page



Module Name: Any desired name for the connection. Example: `test`

URI Prefix: `/test`

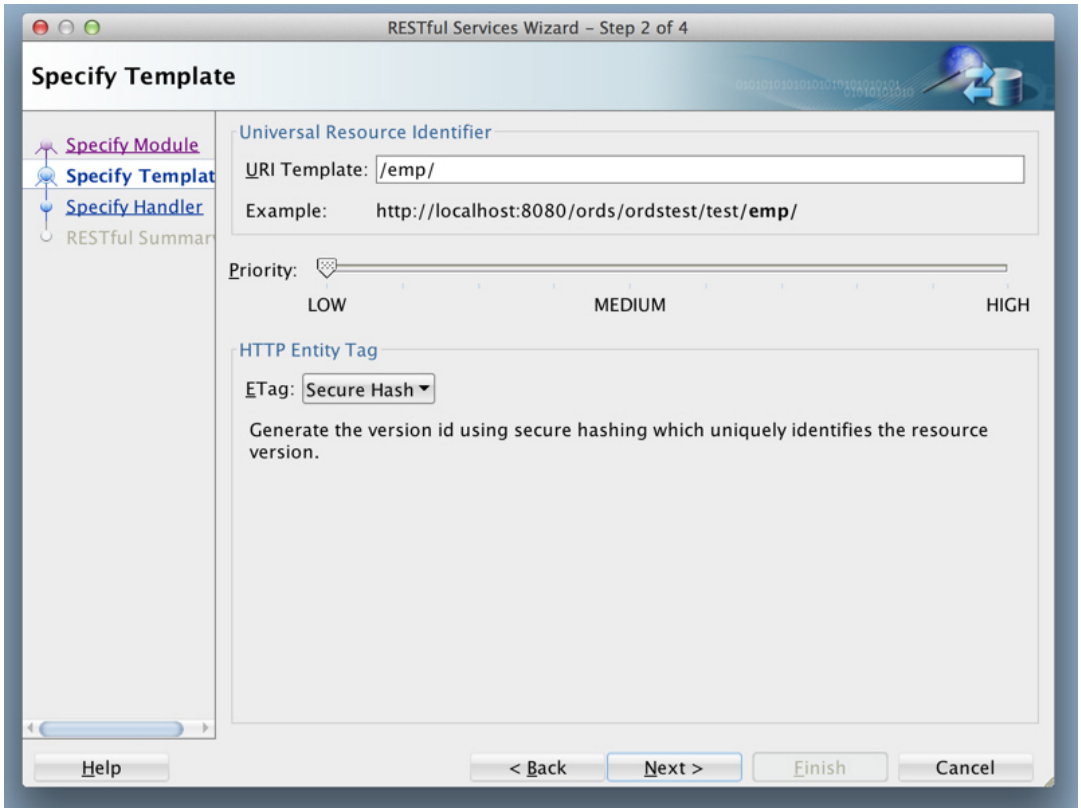
Publish - Make this RESTful Service available for use: Enable (check).

Pagination Size: `7`

7. Click **Next**, and enter information on the Specify Template page:

Copyright © 2011, 2026, Oracle and/or its affiliates.

Figure 1-49 Entering Information on the Specify Template Page



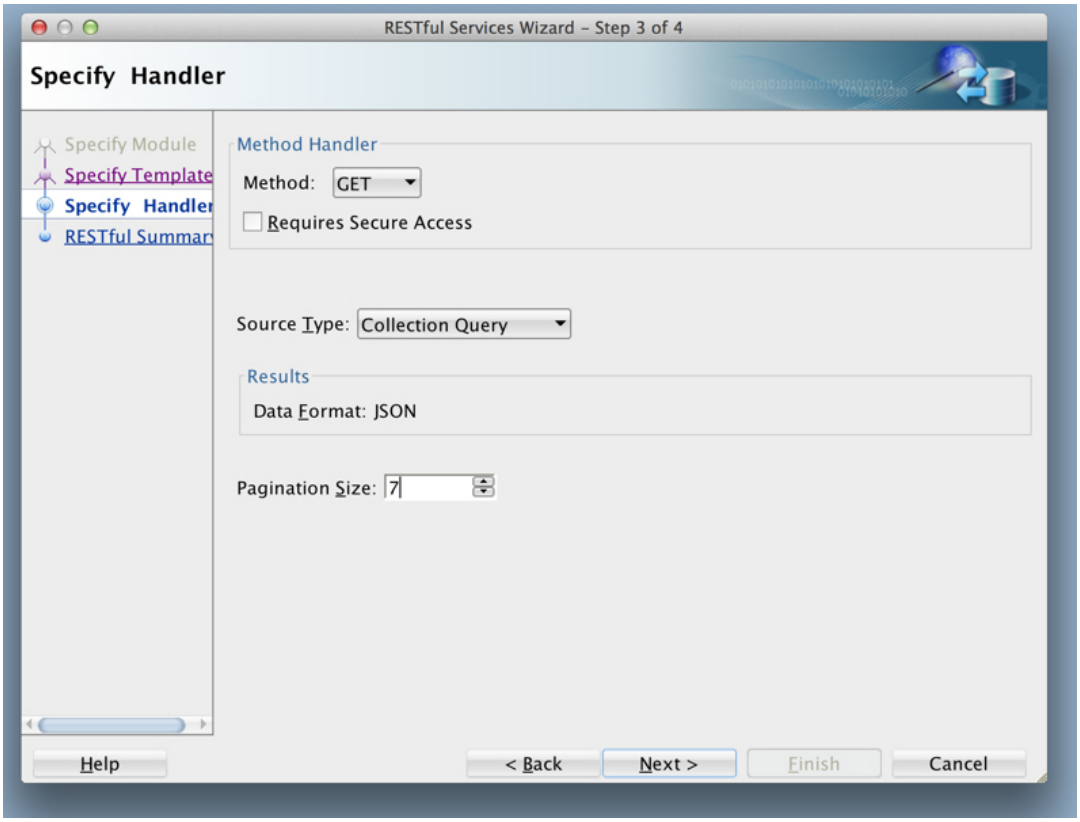
URI Template: `/emp/`

Accept the defaults for the remaining options.

8. Click **Next**, and enter information on the Specify Handler page:

Copyright © 2011, 2026, Oracle and/or its affiliates.

Figure 1-50 Entering Information on the Specify Handler Page:



Method: `GET`

Requires Secure Access: Disable (uncheck) for this tutorial.

Source Type: `Collection Query`

Pagination Size: `7`

9. Click Next to go to the RESTful Summary page of the wizard, then click **Finish**.

   The resource module is now created, the next step is to define the query for the GET resource handler.

10. Define the query for the GET resource handler.

    a. Expand the `test` node under the `Modules` node in the REST Development view.

    b. Expand the `/emp/` node, right-click the `GET` node, and select **Open**.

    c. In the SQL Worksheet that opens for `GET /emp/`, enter the following SQL query:

    ```
    select * from emp
    ```

    d. Right-click on the test node under the 'Modules' node in the 'REST Development' view

    e. Click 'Upload...'. A confirmation dialog will appear confirming the module has been uploaded.

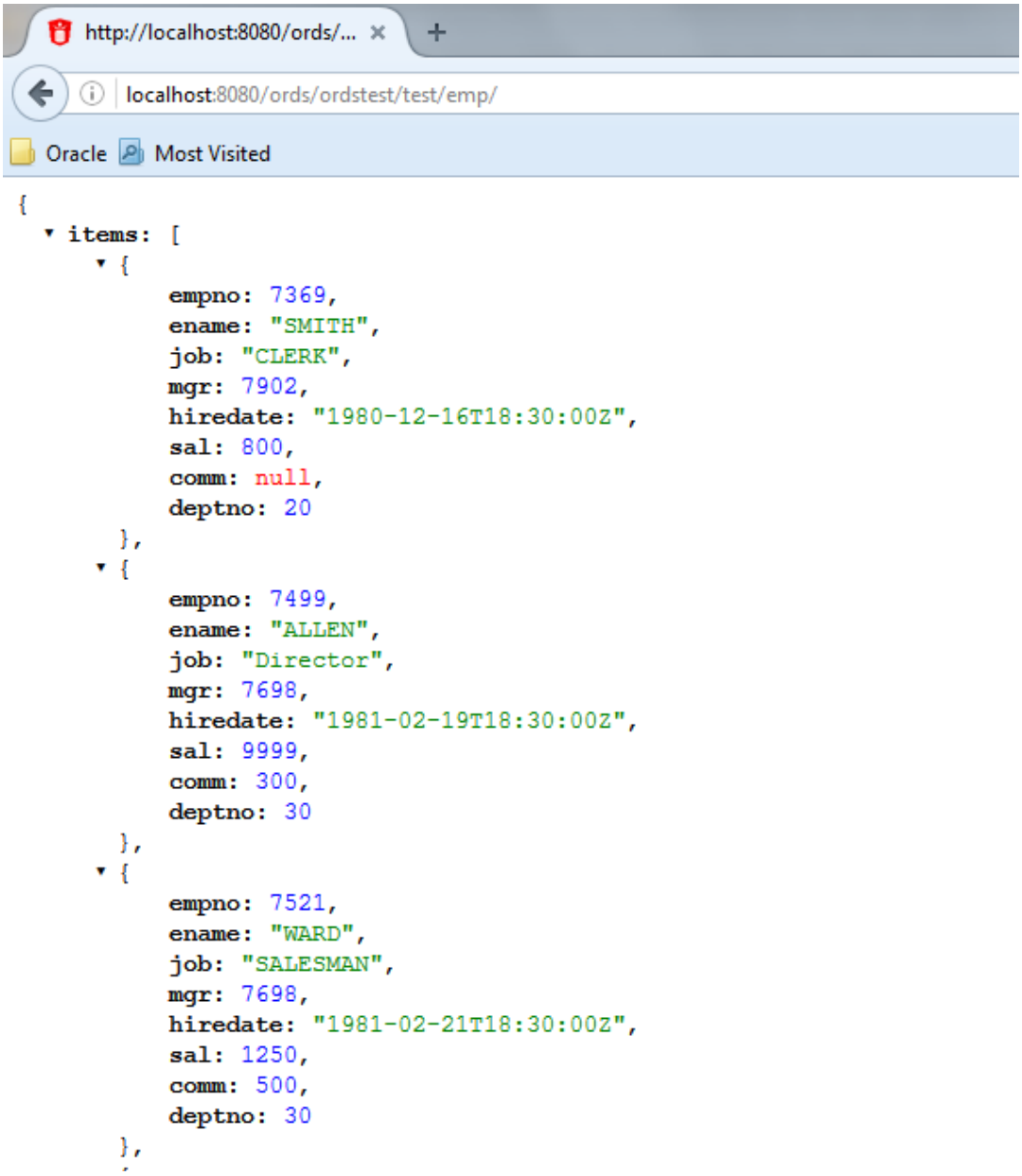
11. Test the RESTful service. In a web browser enter the URL `http://localhost:8080/ords/ordstest/test/emp/` as shown in the following figure:

    - The ORDSTEST schema has been exposed at the `/ordstest/` path.

Copyright © 2011, 2026, Oracle and/or its affiliates.

- The query has been exposed at the `/test/emp/` path.

Figure 1-51 Testing the RESTful Service Created from a SQL Query



Related Topics

- Creating a RESTful Service through the Connections Navigator

1.1.5.4 Protect Resources

Up to this point the tutorial has deliberately disabled security on the RESTful endpoints you created, because it is easier to test them without security. In this topic you protect the `/test/emp/` service, requiring users to authenticate before accessing the service.

Controlling access to protected resources is done by defining privileges. **Privileges** restrict access to only users having at least one of a set of specified roles. A privilege is then associated with one or more resource modules: before those resource modules can be

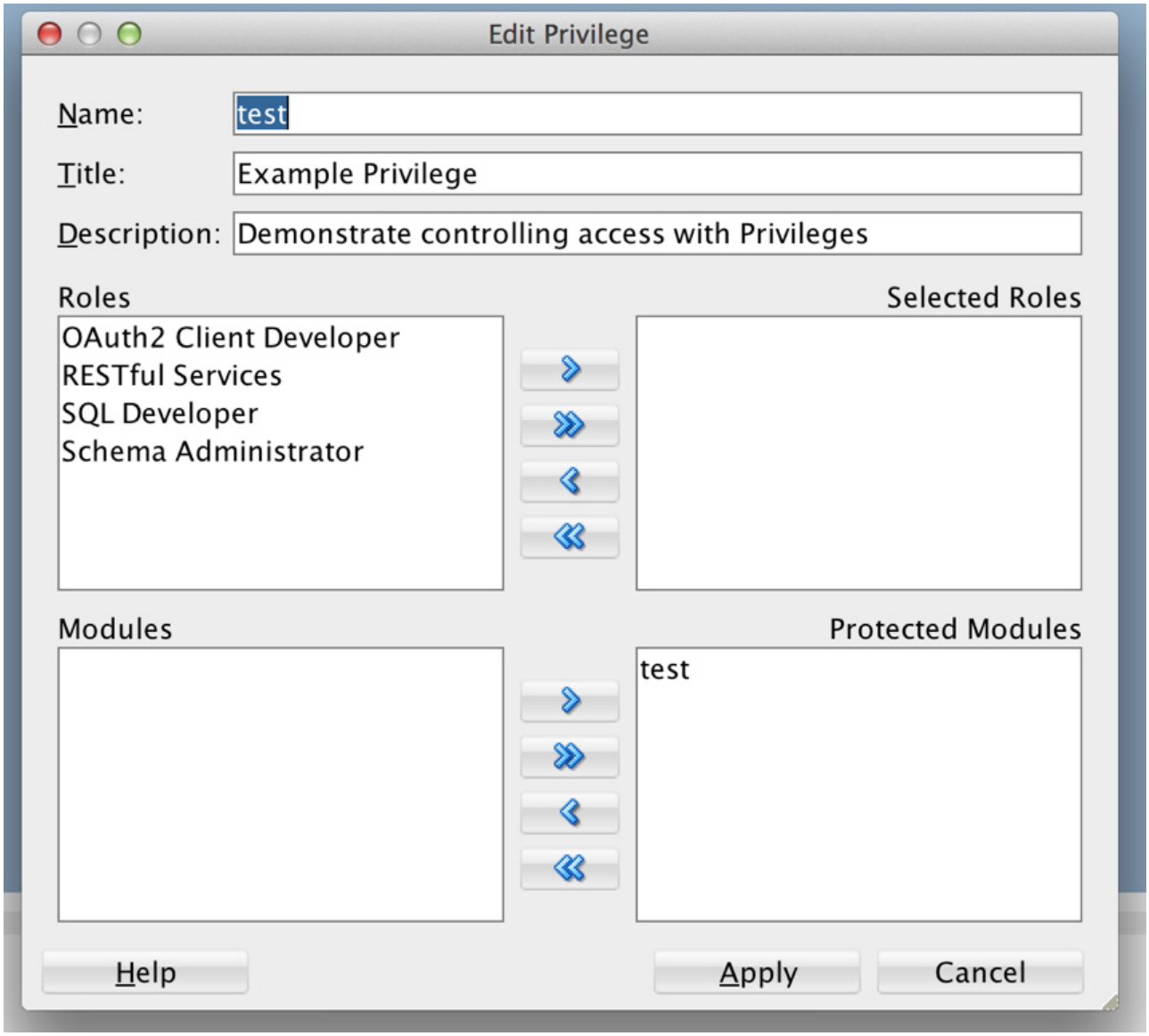
Copyright © 2011, 2026, Oracle and/or its affiliates.

accessed, the user must be authenticated and then authorized to ensure that the user has one of the required roles.

To protect resources, follow these steps.

1. Create a privilege. In SQL Developer, right-click the `Privileges` node in the REST Development view and select **New Privileges** to display the Edit Privilege dialog box:

Figure 1-52 Edit Privilege Dialog Box



Name: `test`

Title: `Example Privilege`

Description: `Demonstrate controlling access with privileges`

Protected Modules: Ensure that the list includes the `test` module. Use the arrow button to move it if necessary.

Click **Apply**.

Copyright © 2011, 2026, Oracle and/or its affiliates.

2. Right click the `test` privilege and click **Upload**.

   A dialog box confirms that the privilege has been uploaded.

   You have now created a privilege that protects the test module. However, you have not restricted the privilege to any particular role; this will just require that the user be authenticated before accessing the test module (the next step).

3. Test the RESTful service. In a web browser enter the following URL:

   `http://localhost:8080/ords/ordstest/test/emp/`

4. Click the link to sign in, and enter the `test_developer` credentials.

   The contents of the JSON document are displayed.

1.1.5.5 Register an OAuth Client Application

This topic explains how to register your applications (called "third-party" applications here) to access a REST API.

OAuth 2.0 is a standard Internet protocol that provides a means for HTTP servers providing REST APIs to give limited access to third party applications *on behalf of* an end user.

- The author of the third-party application must register the application to gain client credentials.
- Using the client credentials the third party application starts a web flow that prompts the end-user to approve access.

So, before a third party application can access a REST API, it must be registered and the user must approve access. And before the application can be registered, it must be assigned a user identity that enables the user to register applications. Users possessing the `SQL Developer` role (such as the `test_developer` user created in Creating a RESTful Service from a SQL Query) are permitted to register OAuth clients.

> **Tip**
>
> In a real application, you may want to provision specific users that can register OAuth clients; these users should be granted the `OAuth Client Developer` role.

This topic outlines how to complete these actions. It is not a full-featured demonstration of how to create and integrate a third party application; it just outlines the concepts involved.

1. Register the client application.

   a. In a web browser enter the following URL:

      `http://localhost:8080/ords/ordstest/oauth/clients/`

   b. At the prompt, click the link to sign in and enter the credentials for the `test_developer` user.

   c. Click **New Client** and enter the following information:

      Name: `Test Client`

      Description: `An example OAuth Client`

      Redirect URI: `http://example.org/redirect`

      Support e-mail: `info@example.org`

Copyright © 2011, 2026, Oracle and/or its affiliates.

Support URI: `http://example.org/support`

Required Privileges: `Example Privilege`

d. Click **Create**.

The client registration is created, and the Authorization URI for the client is displayed. You have created a client that will use the Implicit Grant authorization flow (explained at https://tools.ietf.org/html/rfc6749#section-4.2).

Note the Client Identifier assigned to the client and the Authorization URI value. These values are used to start the authorization flow (next major step).

2. Approve the client application.

In a real third-party client application, the client will initiate the approval flow by directing a web browser to the Authorization URI. The end user will be prompted to sign in and approve access to the client application. The browser will be redirected back to the client's registered Redirect URI with a URI fragment containing the `access_token` for the approval. To simulate this process:

a. In a web browser, enter the Authorization URI that you noted in the previous step. The URL should look like the following (though you should not copy and paste in this example value):

```
http://localhost:8080/ords/ordstest/oauth/auth?
response_type=token&client_id=5B77A34A266EFB0056BE3497ED7099.&state=d5b7944-
d27d-8e2c-4d5c-fb80e1114490&_auth_=force
```

The `client_id` value must be the value of the client identifier assigned to the application. Be sure you are using the correct `client_id` value. Do not use the value in the preceding example; replace it with the client identifier assigned to your application.

The `state` value should be a unique, unguessable value that the client remembers, and can use later to confirm that the redirect received from Oracle REST Data Services is in response to this authorisation request. This value is used to prevent Cross Site Request Forgery attacks; it is very important, cannot be omitted, and must not be guessable or discoverable by an attacker.

b. At the prompt, click the link to sign in and enter the credentials for the `test_developer` user.

c. Review the access being requested, and click **Approve**.

The browser is redirected to a URL similar to the following:

```
http://example.org/redirect#token_type=bearer&access_token=-
i_Ows8j7JYu0p07jOFMEA..&expires_in=3600
```

When registering the OAuth client, you specified `http://example.org/redirect` as the Redirect URI. On completion of the approval request, the browser is redirected to this registered redirect URI. Appended to the URI is the information about the access token that was generated for the approval.

In a real application, the third party application would respond to the redirect to the redirect URI by caching the access token, redirecting to another page to show the user that they are now authorized to access the REST API, and including the access token in every subsequent request to the REST API. However, in this tutorial you just make note of the access token value and manually create a HTTP request with the access token included, as explained in the next major step.

The value of the access token (which in the preceding example is `-i_Ows8j7JYu0p07jOFMEA..`) will change on every approval.

Copyright © 2011, 2026, Oracle and/or its affiliates.

Note that the access token expires. In the preceding example it expires after 3600 seconds (`&expires_in=3600`), that is, one hour.

3. Issue an authorized request.

   After an access token has been acquired, the client application must remember the access token and include it with every request to the protected resource. The access token must be included in the HTTP Authorization request header (explained at https://datatracker.ietf.org/doc/html/rfc2616#section-14.8) as in the following example:

```
Host: localhost:8080
GET /ords/ordstest/test/emp/
Authorization: Bearer -i_Ows8j7JYu0p07jOFMEA..
```

   To emulate creating a valid HTTP request, use the cURL command line tool (if necessary, install cURL). In a real application this request would be performed by the client making an HTTP request, such as an `XMLHttpRequest`. For example:

```
curl -i -H'Authorization: Bearer -i_Ows8j7JYu0p07jOFMEA..' http://localhost:8080/
ords/ordstest/test/emp/
```

   However, in this example replace `-i_Ows8j7JYu0p07jOFMEA..` with the access token value that you previously noted.

   Output similar to the following JSON document should be displayed:

```
HTTP/1.1 200 OK
Content-Type: application/json
Transfer-Encoding: chunked

{
 "items":[

{"empno":7369,"ename":"SMITH","job":"CLERK","mgr":7902,"hiredate":"1980-12-17T00:00:0
0Z","sal":800,"comm":null,"deptno":20},

{"empno":7499,"ename":"ALLEN","job":"SALESMAN","mgr":7698,"hiredate":"1981-02-20T00:0
0:00Z","sal":1600,"comm":300,"deptno":30},

{"empno":7521,"ename":"WARD","job":"SALESMAN","mgr":7698,"hiredate":"1981-02-22T00:00
:00Z","sal":1250,"comm":500,"deptno":30},

{"empno":7566,"ename":"JONES","job":"MANAGER","mgr":7839,"hiredate":"1981-04-01T23:00
:00Z","sal":2975,"comm":null,"deptno":20},

{"empno":7654,"ename":"MARTIN","job":"SALESMAN","mgr":7698,"hiredate":"1981-09-27T23:
00:00Z","sal":1250,"comm":1400,"deptno":30},

{"empno":7698,"ename":"BLAKE","job":"MANAGER","mgr":7839,"hiredate":"1981-04-30T23:00
:00Z","sal":2850,"comm":null,"deptno":30},

{"empno":7782,"ename":"CLARK","job":"MANAGER","mgr":7839,"hiredate":"1981-06-08T23:00
:00Z","sal":2450,"comm":null,"deptno":10}
 ],
 "hasMore":true,
 "limit":7,
 "offset":0,
 "count":7,
 "links":[
  {"rel":"self","href":"http://localhost:8080/ords/ordstest/test/emp/"},
  {"rel":"describedby","href":"http://localhost:8080/metadata-catalog/test/emp/"},
  {"rel":"first","href":"http://localhost:8080/ords/ordstest/test/emp/"},
  {"rel":"next","href":"http://localhost:8080/ords/ordstest/test/emp/?offset=7"}
```


Copyright © 2011, 2026, Oracle and/or its affiliates.

```
 ]
}
```

However, if the `Authorization` header is omitted, then the status `401 Unauthorized` is returned instead.

See Also

curl_haxx

Copyright © 2011, 2026, Oracle and/or its affiliates.

Index

Copyright © 2011, 2026, Oracle and/or its affiliates.