Oracle® REST Data Services

Installation, Configuration, and Development Guide

Release 20.3
F35491-01
October 2020

Oracle REST Data Services Installation, Configuration, and Development Guide, Release 20.3

F35491-01

Copyright © 2011, 2020, Oracle and/or its affiliates.

Primary Authors: Mamata Basapur, Chuck Murray

Contributors: Colm Divilly, Sharon Kennedy, Ganesh Pitchaiah, Kris Rice, Elizabeth Saunders, Jason Straub, Vladislav Uvarov

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 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" or "commercial computer software documentation" 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 and Java 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.

Contents

Preface

Audience xviii
Documentation Accessibility xviii
Related Documents xix
Conventions xix

Changes in This Release for Oracle REST Data Services Installation, Configuration, and Development Guide

Changes in Oracle REST Data Services Release 20.3 xx

1 Introduction to Oracle REST Data Services

About Oracle REST Data Services 1-1
Understanding the Installation Process 1-1
- Supported Java EE Application Servers 1-2
- Supported Oracle Application Express (APEX) Versions 1-2
- System Requirements 1-2
- About Installing Oracle REST Data Services 1-3

Configuring and Installing Oracle REST Data Services 1-3
- About Using the Command-Line Interface 1-4
- About the Database Users Used by Oracle REST Data Services 1-4
- Privileges Required for Oracle REST Data Services 1-5
- Downloading, Configuring and Installing Oracle REST Data Services 1-7
  - ORDS Installer Privileges Script 1-8
  - Advanced Installation Using Command-Line Prompts 1-8
  - ORDS Parameter File 1-14
  - Simple Installation Using a Parameter File 1-27
  - Silent Installation Using a Parameter File 1-28
  - Changing Default Configuration from the Command Line 1-29
  - Validating the Oracle REST Data Services Installation 1-30
  - If You Want to Reinstall or Uninstall (Remove) Oracle REST Data Services 1-31
- Using SQL Developer Oracle REST Data Services Administration (Optional) 1-32

About SQL Developer Oracle REST Data Services Administration 1-32
Configuring an Administrator User 1-32
Using OAuth2 in Non-HTTPS Environments 1-33
Running in Standalone Mode 1-33
Starting in Standalone Mode 1-34
Converting a Private Key to DER (Linux and Unix) 1-35
Stopping the Server in Standalone Mode 1-35
Configuring a Doc Root for Non-Application Express Static Resources 1-35
Deploying to Oracle WebLogic Server 1-36
About Oracle WebLogic Server 1-37
Downloading, Installing, and Configuring Oracle REST Data Services 1-37
Configuring Oracle Application Express Images 1-37
Launching the Administration Server Console 1-38
Installing the Oracle WebLogic Server Deployment 1-38
Configuring WebLogic to Handle HTTP Basic Challenges Correctly 1-40
Verifying the State and Health of ords and i 1-41
Deploying to Apache Tomcat 1-41
About Apache Tomcat 1-41
Downloading, Installing, and Configuring Oracle REST Data Services 1-42
Configuring Oracle Application Express Images 1-42
Installing the Apache Tomcat Deployment 1-42
Upgrading Oracle REST Data Services 1-43
Using a Bequeath Connection to Install, Upgrade, Validate, or Uninstall Oracle REST Data Services 1-44
Authorizing Oracle REST Data Services to Access Oracle Data Guard Protected Users 1-45

2 Configuring Oracle REST Data Services (Advanced)

Configuring Multiple Databases 2-1
About the Request URL 2-1
Configuring Additional Databases 2-2
Routing Based on the Request Path Prefix 2-3
Example of Routing Based on the Request Path Prefix 2-3
Routing Based on the Request URL Prefix 2-4
Example of Routing Based on the Request URL Prefix 2-4
Support for Oracle RAC Fast Connection Failover 2-4
Configuring Security, Caching, Pre- and Post Processing, Environment, and Excel Settings 2-5
Configuring REST-Enabled SQL Service Settings 2-6
Configuring the Maximum Number of Rows Returned from a Query 2-6
Configuring the Custom Error Pages 2-7

Developing RESTful Services for Use with Oracle REST Data Services 2-7
Managing ORDS Administrator Privilege 2-7
Provisioning ORDS_ADMINISTRATOR_ROLE to a User 2-8
Unprovisioning ORDS_ADMINISTRATOR_ROLE from a User 2-8
Managing ORDS Runtime Privilege 2-8
Provisioning ORDS_RUNTIME_ROLE to a User 2-9
Unprovisioning ORDS_RUNTIME_ROLE from a User 2-9

3 Installing and Configuring Customer Managed ORDS on Autonomous Database

About Customer Managed Oracle REST Data Services on Autonomous Database 3-1
Downloading Wallet and Verifying Connection to Autonomous Database 3-2
Creating Customer Managed Oracle REST Data Services User Role 3-3
Downloading and Configuring Oracle REST Data Services 3-4
Preparing and Starting ORDS 3-5

4 Using the Multitenant Architecture with Oracle REST Data Services

Setting Up ORDS in a CDB Environment 4-1
Installation Enabling Multiple Releases 4-2
Command Line Installation 4-2
Advanced Installation 4-2
Silent Installation 4-3
Upgrading Oracle REST Data Services in a CDB Environment 4-4
Migrating Oracle REST Data Services in the CDB to Enable Multiple Releases 4-4
Making All PDBs Addressable by Oracle REST Data Services 4-4
Uninstalling Oracle REST Data Services in a CDB Environment 4-4
Setting Up ORDS in an Application Container 4-4
Prerequisites for Creating ORDS in an Application Container 4-5
Creating an Application Root Container 4-6
Installing ORDS in the Application Root Container 4-6
Creating an Application Seed 4-7
Creating an Application PDB from the Application Seed 4-8
ORDS Configuration Files Setup 4-9
Specifying the ORDS Configuration Folder 4-9
Creating the ORDS Configuration Files for the Application Root Container 4-10
Making all Application PDBs in an Application Root Container Addressable by ORDS 4-11
Running ORDS 4-11
Validating ORDS in the Application Root Container 4-12

Upgrading ORDS in the Application Container 4-13
Uninstalling ORDS from the Application Container 4-14
Verifying ORDS in the Application Container 4-15
Making All PDBs Addressable by Oracle REST Data Services (Pluggable Mapping) 4-15

5 Developing Oracle REST Data Services Applications

Introduction to Relevant Software 5-2
About Oracle Application Express 5-2
About RESTful Web Services 5-2
Getting Started with RESTful Services 5-2
RESTful Services Terminology 5-3
About Request Path Syntax Requirements 5-4
"Getting Started" Documents Included in Installation 5-4
About cURL and Testing RESTful Services 5-5
Automatic Enabling of Schema Objects for REST Access (AutoREST) 5-5
Examples: Accessing Objects Using RESTful Services 5-6
Filtering in Queries 5-17
Auto PL/SQL 5-25
Manually Creating RESTful Services Using SQL and PL/SQL 5-31
About Oracle REST Data Services Mechanisms for Passing Parameters 5-32
Using SQL/JSON Database Functions 5-42
About Working with Dates Using Oracle REST Data Services 5-52
About Datetime Handling with Oracle REST Data Services 5-53
About Setting the Time Zone 5-54
Exploring the Sample RESTful Services in Application Express (Tutorial) 5-55
Configuring Secure Access to RESTful Services 5-59
Authentication 5-60
First Party Cookie-Based Authentication 5-60
Third Party OAuth 2.0-Based Authentication 5-60
About Privileges for Accessing Resources 5-61
About Users and Roles for Accessing Resources 5-61
About the File-Based User Repository 5-62
Tutorial: Protecting and Accessing Resources 5-62
OAuth Flows and When to Use Each 5-63
Assumptions for This Tutorial 5-63
Steps for This Tutorial 5-64
About Oracle REST Data Services User Roles 5-74
About Oracle Application Express Users and Oracle REST Data Services Roles 5-75
Granting Application Express Users Oracle REST Data Services Roles 5-75
Automatically Granting Application Express Users Oracle REST Data Services Roles 5-75

Controlling RESTful Service Access with Roles 5-76
About Defining RESTful Service Roles 5-76
Associating Roles with RESTful Privileges 5-77
Authenticating Against WebLogic Server User Repositories 5-77
Authenticating Against WebLogic Server 5-77
Creating a WebLogic Server User 5-78
Verifying the WebLogic Server User 5-79
Integrating with Existing Group/Role Models 5-79
About role-mapping.xml 5-79
Parameterizing Mapping Rules 5-80
Dereferencing Parameters 5-80
Indirect Mappings 5-81
Integrating Oracle REST Data Services and WebLogic Server 5-81
Configuring ORDS to Integrate with WebLogic Server 5-82
Using the Oracle REST Data Services PL/SQL API 5-82
Creating a RESTful Service Using the PL/SQL API 5-83
Testing the RESTful Service 5-84
Oracle REST Data Services Database Authentication 5-85
Installing Sample Database Scripts 5-85
Enabling the Database Authentication 5-86
Configuring the Request Validation Function 5-87
Testing the Database Authenticated User 5-88
Uninstalling the Sample Database Schema 5-88
Overview of Pre-hook Functions 5-88
Configuring the Pre-hook Function 5-89
Using a Pre-hook Function 5-89
Processing of a Request 5-89
Identity Assertion of a User 5-89
Aborting Processing of a Request 5-90
Ensuring Pre-hook is Executable 5-90
Exceptions Handling by Pre-hook Function 5-90
Pre-hook Function Efficiency 5-91
Pre-Hook Examples 5-91
Installing the Examples 5-91
Uninstalling the Examples 5-94
Generating Hyperlinks 5-95
Primary Key Hyperlinks 5-95
Composite Primary Keys 5-97
Arbitrary Hyperlinks 5-97
About the related Link Relation 5-99

URL Resolution 5-99

6 REST-Enabled SQL Service

REST-Enabled SQL Service Terminology 6-1
Configuring the REST-Enabled SQL Service 6-2
Using cURL with REST-Enabled SQL Service 6-2
Getting Started with the REST-Enabled SQL Service 6-3
REST-Enabling the Oracle Database Schema 6-3
REST-Enabled SQL Authentication 6-4
REST-Enabled SQL Endpoint 6-4
REST-Enabled SQL Service Examples 6-5
POST Requests Using application/sql Content-Type 6-5
Using a Single SQL Statement 6-6
Using a File with cURL 6-7
Using Multiple SQL Statements 6-8
POST Requests Using application/json Content-Type 6-11
Using a File with cURL 6-11
Specifying the Limit Value in a POST Request for Pagination 6-13
Specifying the Offset Value in a POST Request for Pagination 6-14
Defining Binds in a POST Request 6-16
Specifying Batch Statements in a POST Request 6-20
Example POST Request with DATE and TIMESTAMP Format 6-23
Data Types and Formats Supported 6-25
REST-Enabled SQL Request and Response Specifications 6-29
Request Specification 6-29
Response Specification 6-32
Supported SQL, SQL*Plus, and SQLcl Statements 6-38
Supported SQL Statements 6-38
Supported PL/SQL Statements 6-38
Supported SQL*Plus Statements 6-39
Set System Variables 6-39
Show System Variables 6-40
Supported SQLcl Statements 6-42

7 Migrating from mod_plsql to ORDS

Oracle HTTP Server mod_plsql Authentication 7-1
Example Oracle HTTP Server DAD file 7-2
Mapping mod_plsql Settings to ORDS 7-3
Example ORDS Configuration Files 7-8

Example Configuration File for Basic Authentication 7-8
Example Configuration File for Basic Dynamic Authentication 7-8
Example Configuration file for Custom Authentication 7-9
Example ORDS URL Mapping 7-10
Example ORDS Default Configuration 7-10
ORDS Authentication 7-11
Basic Authentication 7-11
Basic Dynamic Authentication 7-11
Custom Authentication 7-12
ORDS Features 7-12
Request Validation Function 7-13
Pre Process Feature 7-13
Post Process Feature 7-13
File Upload Feature 7-14

8 Oracle REST Data Services PL/SQL Package Reference

ORDS.CREATE_ROLE 8-1
ORDS.CREATE_SERVICE 8-1
ORDS.DEFINE_HANDLER 8-4
ORDS.DEFINE_MODULE 8-6
ORDS.DEFINE_PARAMETER 8-7
ORDS.DEFINE_PRIVILEGE 8-9
ORDS.DEFINE_SERVICE 8-11
ORDS.DEFINE_TEMPLATE 8-14
ORDS.DELETE_MODULE 8-16
ORDS.DELETE_PRIVILEGE 8-16
ORDS.DELETE_ROLE 8-17
ORDS.DROP_REST_FOR_SCHEMA 8-17
ORDS.ENABLE_OBJECT 8-18
ORDS.DROP_REST_FOR_OBJECT 8-19
ORDS.ENABLE_SCHEMA 8-19
ORDS.PUBLISH_MODULE 8-20
ORDS.RENAME_MODULE 8-21
ORDS.RENAME_PRIVILEGE 8-22
ORDS.RENAME_ROLE 8-22
ORDS.SET_MODULE_ORIGINS_ALLOWED 8-23
ORDS.SET_URL_MAPPING 8-24
ORDS.SET_SESSION_DEFAULTS 8-24
ORDS.RESET_SESSION_DEFAULTS 8-25

9 Oracle REST Data Services Administration PL/SQL Package Reference

ORDS_ADMIN.CREATE_ROLE 9-1
ORDS_ADMIN.DEFINE_HANDLER 9-2
ORDS_ADMIN.DEFINE_MODULE 9-4
ORDS_ADMIN.DEFINE_PARAMETER 9-5
ORDS_ADMIN.DEFINE_PRIVILEGE 9-7
ORDS_ADMIN.DEFINE_SERVICE 9-10
ORDS_ADMIN.DEFINE_TEMPLATE 9-13
ORDS_ADMIN.DELETE_MODULE 9-14
ORDS_ADMIN.DELETE_PRIVILEGE 9-15
ORDS_ADMIN.DELETE_ROLE 9-15
ORDS_ADMIN.DROP_REST_FOR_SCHEMA 9-16
ORDS_ADMIN.ENABLE_OBJECT 9-17
ORDS_ADMIN.DROP_REST_FOR_OBJECT 9-18
ORDS_ADMIN.ENABLE_SCHEMA 9-19
ORDS_ADMIN.PUBLISH_MODULE 9-20
ORDS_ADMIN.RENAME_MODULE 9-20
ORDS_ADMIN.RENAME_PRIVILEGE 9-21
ORDS_ADMIN.RENAME_ROLE 9-22
ORDS_ADMIN.SET_MODULE_ORIGINS_ALLOWED 9-23
ORDS_ADMIN.SET_URL_MAPPING 9-23
ORDS_ADMIN.ENABLE_HOUSEKEEPING_JOB 9-24
ORDS_ADMIN.DROP_HOUSEKEEPING_JOB 9-25
ORDS_ADMIN.PERFORM_HOUSEKEEPING 9-25
ORDS_ADMIN.SET_SESSION_DEFAULTS 9-26
ORDS_ADMIN.RESET_SESSION_DEFAULTS 9-26
ORDS_ADMIN.PROVISION_ADMIN_ROLE 9-27
ORDS_ADMIN.PROVISION_RUNTIME_ROLE 9-27
ORDS_ADMIN.UNPROVISION_ ROLES 9-28

10 Implicit Parameters

List of Implicit Parameters 10-1
- About the :body parameter 10-5
- About the :body_text Parameter 10-6
- About the :content_type Parameter 10-6
- About the :current_user Parameter 10-6
- About the :status_code Parameter 10-6
- About the :forward_location Parameter 10-7

About the Pagination Implicit Parameters 10-8
About the :page_offset Parameter 10-9
About the :page_size Parameter 10-9
About the :row_offset Parameter 10-10
About the :row_count Parameter 10-10
About the :fetch_offset Parameter 10-10
About the :fetch_size Parameter 10-10
About Automatic Pagination 10-10
About Manual Pagination 10-11

11 OAUTH PL/SQL Package Reference

OAUTH.CREATE_CLIENT 11-1
OAUTH.DELETE_CLIENT 11-2
OAUTH.GRANT_CLIENT_ROLE 11-3
OAUTH.RENAME_CLIENT 11-4
OAUTH.REVOKE_CLIENT_ROLE 11-4
OAUTH.UPDATE_CLIENT 11-5

12 Enabling ORDS Database API

Basic Setup to Enable ORDS Database API 12-1
Advanced Setup to Enable the ORDS Database API 12-2
Pluggable Database Lifecycle Management 12-3
Disabling PDB Lifecycle Management 12-3
Creating a Default Administrator 12-4
Configuration of Database API Environment Services 12-5
Configuration of Database API with Open Service Broker API Compatible Platforms 12-5

A Oracle REST Data Services Database Type Mappings

Oracle Built-in Types A-1
Handling Structural Database Types A-3
Oracle Geospacial Encoding A-5
Enabling Database Mapping Support A-5

B About the Oracle REST Data Services Configuration Files

Locating Configuration Files B-1
Setting the Location of the Configuration Files B-1
Understanding the Configuration Folder Structure B-1
Understanding the Configuration File Format B-2

Understanding the url-mapping.xml File Format B-2
Understanding Configurable Parameters B-3

C Troubleshooting Oracle REST Data Services

Enabling Detailed Request Error Messages C-1
ORDS User Defined Service C-1
Configuring Application Express Static Resources with Oracle REST Data Services C-13

D Creating an Image Gallery

Before You Begin D-1
About URIs D-1
About Browser Support D-2
Creating an Application Express Workspace D-2
Creating the Gallery Database Table D-2
Creating the Gallery RESTful Service Module D-3
Trying Out the Gallery RESTful Service D-4
Creating the Gallery Application D-5
Trying Out the Gallery Application D-8
Securing the Gallery RESTful Services D-8
Protecting the RESTful Services D-8
Modifying the Application to Use First Party Authentication D-9
Accessing the RESTful Services from a Third Party Application D-11
Creating the Third Party Developer User D-12
Registering the Third Party Application D-12
Acquiring an Access Token D-12
Using an Access Token D-14
About Browser Origins D-15
Configuring a RESTful Service for Cross Origin Resource Sharing D-15
Acquiring a Token Using the Authorization Code Protocol Flow D-15
Registering the Client Application D-16
Acquiring an Authorization Code D-16
Exchanging an Authorization Code for an Access Token D-17
Extending OAuth 2.0 Session Duration D-18
About Securing the Access Token D-19

E Third-Party License Information

jackson-annotations 2.10.3 E-1
jackson-databind 2.10.3 E-5
jackson-dataformat-xml 2.10.3 E-10

jackson-core 2.10.3 E-18
Google Guava 29.0 E-22
Jetty 9.4.28 E-28
Javassist 3.27 E-90
avsc 5.4.21 E-93
babel-polyfill 7.8.7 E-94
CodeMirror 5.53.2 E-134
Dexie.js 3.0.1 E-134
SheetJS 0.15.5 E-139
jQuery 3.5.1 E-144

Index

List of Examples

1-1	Parameters to configure for Application Express and APEX RESTful Services and run in standalone mode	1-24
1-2	Parameters to run in standalone mode using http	1-26
1-3	Parameters to run in standalone mode using https and providing the ssl certificate paths	1-26
1-4		1-29
1-5		1-29
1-6		1-30
2-1	Configuring custom error page for “HTTP 404” status code	2-7
2-2	Using Grant command	2-8
2-3	Using ORDS_ADMIN package method	2-8
2-4	Using REVOKE command	2-8
2-5	Using ORDS_ADMIN package method	2-8
2-6	Using Grant command	2-9
2-7	Using ORDS_ADMIN package method	2-9
2-8	Using REVOKE command	2-9
2-9	Using ORDS_ADMIN package method	2-9
4-1	Configuring ORDS for Application Express	4-10
4-2	Configuring ORDS only	4-10
5-1	Enabling the PL/SQL Function	5-27
5-2	Enabling the PL/SQL Procedure	5-27
5-3	Generating an Endpoint for the Stored Procedure	5-30
5-4	Package Procedure and Function Endpoints	5-30
5-5	Nested JSON Purchase Order with Nested LineItems	5-43
5-6	PL/SQL Handler Code Used for a POST Request	5-44
5-7	GET Handler Code using Oracle REST Data Services Query on Relational Tables for Generating a Nested JSON object	5-49
5-8	PL/SQL API Call for Creating a New test/:id Template and GET Handler in the demo Module	5-50
5-9	Setting the Duser.timezone Java Environment Variable in Standalone Mode	5-54
5-10	Setting the Duser.timezone Java Environment Variable in a Java Application Server	5-55
5-11	Setting Enabled for all Pools	5-87
6-1	Example cURL Command	6-2
6-2	Binds in POST Request	6-16
6-3	Complex Bind in POST Request	6-18
6-4	Batch statements	6-20

6-5	Batch bind values	6-21
6-6	Oracle REST Data services Time Zone Set as Europe/London	6-23
6-7	PL/SQL Statement	6-38
7-1	dads.conf file	7-2
7-2	ords_conf/ords/conf/basic_auth.xml	7-8
7-3	ords_conf/ords/conf/basic_dynamic_auth.xml	7-8
7-4	ords_confs/ords/conf/custom_auth.xml	7-9
7-5	ords_conf/ords/url-mapping.xml	7-10
7-6	ords_conf/ords/default.xml	7-10
7-7	security.requestValidationFunction	7-13
7-8	procedure.preProcess	7-13
7-9	procedure.postProcess	7-13
7-10	Table upload	7-14
7-11	Procedure upload	7-14
7-12	Curl command for file upload	7-14
10-1	Example	10-5
11-1	Example to Add Multiple Privileges	11-6

List of Figures

5-1	Selecting the Enable REST Service Option	5-28
5-2	Auto Enabling the PL/SQL Package Object	5-29
5-3	Adding an Anonymous PL/SQL Block to the Handler for the PUT Method	5-34
5-4	Setting the Bind Parameter l_salarychange to Pass for the PUT Method	5-34
5-5	Obtaining the URL to Call from the Details Tab	5-35
5-6	Displaying the Results from a SQL Query to Confirm the Execution of the PUT Method	5-36
5-7	Creating a Template Definition to Include a Route Pattern for Some Parameters or Bind Variables	5-37
5-8	Adding a SQL Query to the Handler	5-38
5-9	Using Browser to Show the Results of Using a Route Pattern to Send a GET Method with Some Required Parameter Values	5-39
5-10	Using Browser to Show the Results of Using a Query String to Send a GET Method with Some Parameter Name/Value Pairs	5-41
5-11	Complete Response Body in JSON Format	5-48
5-12	Generating Nested JSON Objects	5-52

List of Tables

1-1	Advanced Installation Prompts for Installing and Configuring ORDS	1-11
1-2	Options for Configuring Application Express or Migrating from mod_plsql	1-12
1-3	Enabling Features in ORDS	1-13
1-4	Options for Running in Standalone Mode	1-13
1-5	Parameters for Installing Oracle REST Data Services	1-17
1-6	Parameters for Enabling SQL Developer Web	1-19
1-7	Parameters for Enabling REST-Enabled SQL	1-20
1-8	Parameters for Enabling Database API	1-21
1-9	Parameters for Installing into the CDB	1-21
1-10	Parameters for Configuring Application Express	1-22
1-11	Parameters for Installing Oracle REST Data Services in Standalone Mode	1-25
1-12	Miscellaneous Parameters	1-27
5-1	Parameters for batchload	5-16
7-1	Mappings of mod_plsql Directives to ORDS Settings	7-3
10-1	List of Implicit Parameters	10-1
10-2	Pagination Implicit Parameters	10-9
12-1	Open Service Broker Service Catalog	12-6
B-1	Oracle REST Data Services Configuration Files Parameters	B-3
C-1	List of ORDS user defined service	C-2

Preface

Oracle REST Data Services Installation, Configuration, and Development Guide explains how to install and configure Oracle REST Data Services. (Oracle REST Data Services was called *Oracle Application Express Listener* before Release 2.0.6.)

> **Note:**
>
> Effective with Release 3.0, the title of this book is *Oracle REST Data Services Installation, Configuration, and Development Guide*. The addition of "Development" to the title reflects the fact that material from a previous separate unofficial "Developer's Guide" has been included in this book in Developing Oracle REST Data Services Applications.

Topics:

- Audience
- Documentation Accessibility
- Related Documents
- Conventions

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/`

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.

Changes in This Release for Oracle REST Data Services Installation, Configuration, and Development Guide

This section lists the changes in this document for release 20.2.

Changes in Oracle REST Data Services Release 20.3

This section lists the changes in Oracle REST Data Services for this release.

New Features

Updated the following section with more details:

- See Configuration of Database API with Open Service Broker API Compatible Platforms.

1

Introduction to Oracle REST Data Services

This chapter introduces Oracle REST Data Services and also describes how to install and deploy it. REST stands for Representational State Transfer.

> **Note:**
>
> **Oracle REST Data Services** was called *Oracle Application Express Listener* before Release 2.0.6.

Topics:

- About Oracle REST Data Services
- Understanding the Installation Process
- Configuring and Installing Oracle REST Data Services
- Running in Standalone Mode
- Deploying to Oracle WebLogic Server
- Deploying to Apache Tomcat
- Upgrading Oracle REST Data Services
- Using a Bequeath Connection to Install, Upgrade, Validate, or Uninstall Oracle REST Data Services

About Oracle REST Data Services

Oracle REST Data Services is a Java EE-based alternative for Oracle HTTP Server and `mod_plsql`. The Java EE implementation offers increased functionality including a command line based configuration, enhanced security, file caching, and RESTful web services. Oracle REST Data Services also provides increased flexibility by supporting deployments using Oracle WebLogic Server, Apache Tomcat, and a standalone mode.

The Oracle Application Express architecture requires some form of web server to proxy requests between a web browser and the Oracle Application Express engine. Oracle REST Data Services satisfies this need but its use goes beyond that of Oracle Application Express configurations. Oracle REST Data Services simplifies the deployment process because there is no Oracle home required, as connectivity is provided using an embedded JDBC driver.

Understanding the Installation Process

This section offers an overview of Oracle REST Data Services and provides information about supported Java Platform, Enterprise Edition (Java EE) application servers and system requirements.

Topics:

- Supported Java EE Application Servers
- System Requirements

Supported Java EE Application Servers

Oracle REST Data Services supports the following Java EE application servers:

Application Server	Supported Release
Oracle WebLogic Server	12g Release 2 (12.2.1.3) or later
Apache Tomcat	Release 8.5 or later

Supported Oracle Application Express (APEX) Versions

Oracle REST Data Services supports the currently supported versions of APEX.

> **See Also:**
>
> The Oracle Application Express (Formerly HTML DB) table in the ORACLE INFORMATION-DRIVEN SUPPORT document for supported versions of APEX.

System Requirements

Oracle REST Data Services system requirements are as follows:

- Oracle Database (Enterprise Edition, Standard Edition or Standard Edition One) release 11*g* Release 2 or later, or Oracle Database 11*g* Release 2 Express Edition.
- Oracle Java 8 or later.
- Web browser requirements:
  - Microsoft Internet Explorer 8.0 or later.
  - Mozilla Firefox 3.0 or later.
  - Google Chrome 2.0 or later.

> **Note:**
>
> Oracle Application Express is *not* a prerequisite for using Oracle REST Data Services.
>
> If Oracle Application Express is installed and if RESTful services have been configured during the installation (see the step Configure RESTful Services in Oracle Application Express Installation Guide), then Oracle REST Data Services supports it, including executing the RESTful services defined in Oracle Application Express.

About Installing Oracle REST Data Services

To install Oracle REST Data Services:

1. Download, install, and configure Oracle REST Data Services.
2. Deploy Oracle REST Data Services. Deployment options include:
   - **Standalone Mode**.
   - **Oracle WebLogic Server**.
   - **Apache Tomcat**.

Related Topics

- Running in Standalone Mode
- Deploying to Oracle WebLogic Server

Configuring and Installing Oracle REST Data Services

Before you deploy Oracle REST Data Services, you must install and configure it using a command-line interface.

Topics:

- About Using the Command-Line Interface
- About the Database Users Used by Oracle REST Data Services
- Privileges Required for Oracle REST Data Services
- Downloading, Configuring and Installing Oracle REST Data Services
- Using SQL Developer Oracle REST Data Services Administration (Optional)
- Using OAuth2 in Non-HTTPS Environments

> **See Also:**
>
> To use the Oracle REST API for JSON Data Persistence, you must also install the Oracle REST API. See "Oracle REST API Installation" in *Oracle REST Data Services SODA for REST Developer's Guide*.

About Using the Command-Line Interface

Oracle REST Data Services provides several command line commands. For example, you can configure the location where Oracle REST Data Services stores configuration files, configure the database Oracle REST Data Services uses, and start Oracle REST Data Services in standalone mode.

To display a full list of available commands, go to the directory or folder containing the `ords.war` file and execute the following command:

```
java -jar ords.war help
```

A list of the available commands is displayed. To see instructions on how to use each of these commands, enter help followed by the command name, for example:

```
java -jar ords.war help configdir
```

About the Database Users Used by Oracle REST Data Services

Oracle REST Data Services uses the following database users:

User Name	Required	Description
APEX_PUBLIC_USER	Only if using Oracle REST Data Services with Oracle Application Express	If you use Oracle REST Data Services with Oracle Application Express, this is the database user used when invoking PL/SQL Gateway operations, for example, all Oracle Application Express operations. For information on unlocking the APEX_PUBLIC_USER, see Configure APEX_PUBLIC_USER Account in Oracle Application Express Installation Guide.
APEX_REST_PUBLIC_USER	Only if using RESTful Services defined in Application Express of version 5.0 or above.	The database user used when invoking Oracle Application Express RESTful Services if RESTful Services defined in Application Express workspaces are being accessed
APEX_LISTENER	Only if using RESTful Services defined in Application Express of version 5.0 or above.	The database user used to query RESTful Services definitions stored in Oracle Application Express if RESTful Services defined in Application Express workspaces are being accessed
ORDS_METADATA	Yes	Owner of the PL/SQL packages used for implementing many Oracle REST Data Services capabilities. ORDS_METADATA is where the metadata about Oracle REST Data Services-enabled schemas is stored. It is not accessed directly by Oracle REST Data Services; the Oracle REST Data Services application never creates a connection to the ORDS_METADATA schema. The schema password is set to a random string, connect privilege is revoked, and the password is expired.

User Name	Required	Description
ORDS_PUBLIC_USER	Yes	User for invoking RESTful Services in the Oracle REST Data Services-enabled schemas.

The APEX_<xxx> users are created during the Oracle Application Express installation process.

Privileges Required for Oracle REST Data Services

As part of the Oracle REST Data Services installation, privileges are granted to several users and roles:

- ORDS_RUNTIME_ROLE role
  - ORDS_RUNTIME_ROLE is granted EXECUTE on the following packages if these packages are not granted EXECUTE to PUBLIC:
    - SYS.DBMS_LOB
    - SYS.DBMS_SESSION
    - SYS.DBMS_UTILITY
    - SYS.WPIUTL
  - ORDS_RUNTIME_ROLE is granted the necessary ORDS_METADATA object privileges to determine the repository version and to access the connection pool configurations.
- ORDS_PUBLIC_USER user
  - ORDS_PUBLIC_USER is granted connect to allow connection to the database.
  - ORDS_PUBLIC_USER is granted role, ORDS_RUNTIME_ROLE to allow the user to act as an ORDS runtime user
- ORDS_ADMINISTRATOR_ROLE role
  - ORDS_ADMINISTRATOR_ROLE is granted EXECUTE on ORDS_METADATA.ORDS_ADMIN PL/SQL package.
- PUBLIC
  - PUBLIC is granted EXECUTE on ORDS_METADATA.ORDS_REPVERSION view to allow the repository version to be queried by anyone.
  - PUBLIC is granted SELECT on many ORDS_METADATA views.
  - PUBLIC is granted EXECUTE on ORDS_METADATA PL/SQL packages that are available for developer users.
- ORDS_METADATA schema
  - ORDS_METADATA schema is granted on the following packages if these packages are not granted EXECUTE on PUBLIC:
    - SYS.DBMS_ASSERT
    - SYS.DBMS_LOB
    - SYS.DBMS_OUTPUT

        - SYS.DBMS_SCHEDULER
        - SYS.DBMS_SESSION
        - SYS.DBMS_UTILITY
        - SYS.DEFAULT_JOB_CLASS
        - SYS.HTP
        - SYS.OWA
        - SYS.WPG_DOCLOAD
    - ORDS_METADATA is granted SELECT (11g) or READ (12*c* or later) on the following view if it is not granted SELECT or READ to PUBLIC:
        - SYS.SESSION_PRIVS
    - ORDS_METADATA schema is granted EXECUTE on the following packages:
        - SYS.DBMS_CRYPTO
        - SYS.DBMS_METADATA
    - ORDS_METADATA schema is granted SELECT (11g) or READ (12c or later) on the following views:
        - SYS.DBA_OBJECTS
        - SYS.DBA_ROLE_PRIVS
        - SYS.DBA_TAB_COLUMNS
    - ORDS_METADATA schema is granted SELECT including WITH GRANT OPTION on the following views:
        - SYS.USER_CONS_COLUMNS
        - SYS.USER_CONSTRAINTS
        - SYS.USER_OBJECTS
        - SYS.USER_PROCEDURES
        - SYS.USER_TAB_COLUMNS
        - SYS.USER_TABLES
        - SYS.USER_VIEWS
    - ORDS_METADATA schema is granted the following system privileges:
        - ALTER USER
        - CREATE ANY TRIGGER
        - CREATE JOB
        - CREATE PUBLIC SYNONYM
        - DROP PUBLIC SYNONYM
    - ORDS_METADATA schema is granted the necessary object privileges to migrate Application Express REST data to ORDS_METADATA tables.
    - ORDS_METADATA schema is granted ORDS_ADMINISTRATOR_ROLE, ORDS_RUNTIME_ROLE roles with administrator option.
- PUBLIC is granted SELECT on many ORDS_METADATA tables and views.

- `PUBLIC` is granted `EXECUTE` on PL/SQL packages that are available for users to invoke.
- `ORDS_METADATA` is granted `EXECUTE` on the following packages if these packages are not granted `EXECUTE` to `PUBLIC`:
  - `SYS.DBMS_ASSERT`
  - `SYS.DBMS_LOB`
  - `SYS.DBMS_OUTPUT`
  - `SYS.DBMS_SCHEDULER`
  - `SYS.DBMS_SESSION`
  - `SYS.DBMS_UTILITY`
  - `SYS.DEFAULT_JOB_CLASS`
  - `SYS.HTP`
  - `SYS.OWA`
  - `SYS.WPG_DOCLOAD`
- `ORDS_METADATA` is granted the necessary object privileges to migrate Application Express REST data to `ORDS_METADATA` tables.

Downloading, Configuring and Installing Oracle REST Data Services

The procedures in this topic apply to installing Oracle REST Data Services in a traditional (non-CDB) database.

> **Note:**
>
> You must complete the configuration steps in this topic before deploying to an application server.

To install and configure Oracle REST Data Services:

1. Download the file `ords.`*`version.number`*`.zip` from the Oracle REST Data Services download page.

   Note that the *`version.number`* in the file name reflects the current release number.
2. Unzip the downloaded zip file into a directory (or folder) of your choice:
   - UNIX and Linux: unzip `ords.`*`version.number`*`.zip`
   - Windows: Double-click the file `ords.`*`version.number`*`.zip` in Windows Explorer
3. Choose one of the following installation options:
   - Advanced Installation Using Command-Line Prompts
   - Silent Installation Using a Parameter File
4. You can reinstall or uninstall Oracle REST Data Services if required.

Related Topics

- Using the Multitenant Architecture with Oracle REST Data Services
- About the Database Users Used by Oracle REST Data Services
- If You Want to Reinstall or Uninstall (Remove) Oracle REST Data Services

> **See Also:**
>
> Oracle REST Data Services Downloads

ORDS Installer Privileges Script

This section describes about the script file that provides privileges to the user to install, upgrade, validate and uninstall ORDS.

> **Note:**
>
> This script is used when you do not want to use SYS AS SYSDBA to install, upgrade, validate and uninstall ORDS for Oracle PDB or Oracle 11g.

Starting with ORDS 19.2 release, the Oracle REST Data Services installation archive file contains a script, `ords_installer_privileges.sql` which is located in the installer folder. The script provides the assigned database user the privileges to install, upgrade, validate and uninstall ORDS in Oracle Pluggable Database or Oracle 11g.

Perform the following steps:

1. Using SQLcl or SQL*Plus, connect to Oracle PDB or 11g database with SYSDBA privileges.
2. Execute the following script providing the database user:

```
SQL> @/path/to/installer/ords_installer_privileges.sql exampleuser
SQL> exit
```

You must use the specified database user to install, upgrade, validate and uninstall ORDS.

Advanced Installation Using Command-Line Prompts

You can perform an advanced installation in which you are prompted for the necessary parameter values.

To perform an advanced installation, enter the following command:

```
java -jar ords.war install advanced
```

> **Note:**
>
> If you use a TNS connection to install ORDS, then you must specify the `oracle.net.tns_admin` system property on the command line that contains the folder with the `tnsnames.ora` file.
> To perform installation using TNS connection, enter the following command:
>
> ```
> java -Doracle.net.tns_admin=/path/to/tnsFolder -jar ords.war
> install advanced
> ```
>
> Use the following on-line help command to check for additional options available for the install command: `java –jar ords.war help install`

During installation, Oracle REST Data Services checks if configuration files already exist in the specified configuration folder:

- If configuration files do not exist in that folder, then they are created. For example: `defaults.xml`, `apex_pu.xml` files.
- If configuration files from an earlier release exist in that folder, Oracle REST Data Services checks if `<name>_pu.xml` is present. If it is not present, then you are prompted to enter the password for the ORDS_PUBLIC_USER account. If the configuration files `<name>_al.xml` and `<name>_rt.xml` from Release 2.0.n exist, then they are preserved. (However, in Releases 2.0.n RESTful Services was optional, and therefore the files might not exist in the configuration folder.)
- If multiple configuration files exist from a previous release (examples: `apex.xml`, `apex_al.xml`, `apex_rt.xml`, `sales.xml`, `sales_al.xml`, `sales_rt.xml`, ...), and if `<name>_pu.xml` does not exist, then you are prompted to select the database configuration so that the Oracle REST Data Services schema can be created in that database.

The following shows an example for an advanced installation. In this example, if you accepted the default value as 1 for `Enter 1 if you wish to start in standalone mode or 2 to exit [1]`, the remaining prompts are displayed; and if you will be using Oracle Application Express, then you must specify the `APEX static resources location`.

```
d:\ords> java -jar ords.war install advanced

This Oracle REST Data Services instance has not yet been configured.
Please complete the following prompts

Enter the location to store configuration data: /path/to/config

Specify the database connection type to use.
Enter number for [1] Basic  [2] TNS  [3] Custom URL [1]:
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]:
Enter the database service name:orcl
Enter 1 if you want to verify/install Oracle REST Data Services schema or 2 to
skip this step [1]:
Enter the database password for ORDS_PUBLIC_USER:
Confirm password:
Requires to login with administrator privileges to verify Oracle REST Data
Services schema.
```

```
Enter the administrator username:EXAMPLEUSER
Enter the database password for EXAMPLEUSER:
Confirm password:
Connecting to database user: EXAMPLEUSER url: jdbc:oracle:thin:@//localhost:1521/
orcl

Retrieving information.
Enter the default tablespace for ORDS_METADATA [SYSAUX]:
Enter the temporary tablespace for ORDS_METADATA [TEMP]:
Enter the default tablespace for ORDS_PUBLIC_USER [USERS]:
Enter the temporary tablespace for ORDS_PUBLIC_USER [TEMP]:
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]:
Enter the PL/SQL Gateway database user name [APEX_PUBLIC_USER]:
Enter the database password for APEX_PUBLIC_USER:
Confirm password:
Enter 1 to specify passwords for Application Express RESTful Services database
users (APEX_LISTENER, APEX_REST_PUBLIC_USER) or 2 to skip this step [1]:
Enter the database password for APEX_LISTENER:
Confirm password:
Enter the database password for APEX_REST_PUBLIC_USER:
Confirm password:
Enter a number to select a feature to enable:
   [1] SQL Developer Web  (Enables all features)
   [2] REST Enabled SQL
   [3] Database API
   [4] REST Enabled SQL and Database API
   [5] None
Choose [1]:
Enter 1 if you wish to start in standalone mode or 2 to exit [1]:
Enter the APEX static resources location:/path/to/apex/images
Enter 1 if using HTTP or 2 if using HTTPS [1]:
Enter the HTTP port [8080]:
OR
Enter 1 if using HTTP or 2 if using HTTPS [1]:2
Enter the HTTPS port [8443]:
Enter the SSL hostname:mysslhost
Enter 1 to use the self-signed certificate or 2 if you will provide the SSL
certificate [1]:
```

Descriptions for Advanced Installation Prompts

This section describes the options you can choose while performing advanced installation of Oracle REST Data Services schema.

Table 1-1 Advanced Installation Prompts for Installing and Configuring ORDS

Options	Description

`This Oracle REST Data Services instance has not yet been configured.`
`Please complete the following prompts`
`Enter the location to store configuration data:/path/to/config`
	Specify the location for the ORDS configuration files. If the location does not exist, then it will be created.

`Specify the database connection type to use. Enter number for [1] Basic [2] TNS [3] Custom URL [1]:`	Specify if you want a Basic connection, TNS connection or Custom URL connection
The following example is for a Basic Connection:	
`Enter the name of the database server [localhost]:`	Specify the Oracle database hostname.
`Enter the database listen port [1521]:`	Specify the Oracle database port.
`Enter 1 to specify the database service name, or 2 to specify the database SID [1]:` `Enter the database service name:orcl`	Specify the Oracle database service name, if you choose option 1. Otherwise, if you choose option 2 then, specify the Oracle database SID.
The following example is for a TNS Connection:	
`Enter the TNS Network Alias:orcl`	Specify the TNS network alias identifier.
The following example is for a Custom URL Connection:	
`Enter the Custom JDBC URL: jdbc:oracle:thin:@//localhost:1521/orcl`	Specify the custom url.
`Enter 1 if you want to verify/install Oracle REST Data Services schema` `or 2 to skip this step [1]:`	Specify 1 to install the Oracle REST Data Services schema and create the Oracle REST Data Services proxy user, `ORDS_PUBLIC_USER`.
`Enter the database password for ORDS_PUBLIC_USER:` `Confirm password:`	Specify the proxy user, `ORDS_PUBLIC_USER` and the corresponding password.

Table 1-1 (Cont.) Advanced Installation Prompts for Installing and Configuring ORDS

Options	Description

`Requires to login with administrator privileges to verify Oracle REST Data Services schema.`
`Enter the administrator username:EXAMPLEUSER`
`Enter the database password for EXAMPLEUSER:`
`Confirm password:`
	Specify the user with ORDS Installer privileges or the SYS AS SYSDBA account.
	Specify the password of the user.

`Enter the default tablespace for ORDS_METADATA [SYSAUX]:` `Enter the temporary tablespace for ORDS_METADATA [TEMP]:`	Specify the default tablespace and temporary tablespace for the Oracle REST Data Services schema, `ORDS_METADATA`.
`Enter the default tablespace for ORDS_PUBLIC_USER [USERS]:` `Enter the temporary tablespace for ORDS_PUBLIC_USER [TEMP]`	Specify the default tablespace and temporary tablespace for the Oracle REST Data Services proxy user, `ORDS_PUBLIC_USER`.

Table 1-2 Options for Configuring Application Express or Migrating from mod_plsql

Options	Description

`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]:`
`Enter the PL/SQL Gateway database user name [APEX_PUBLIC_USER]:`
`Confirm password:`
`Enter the database password for APEX_PUBLIC_USER:`
	You can perform one of the following:
	• If you are using Oracle Application Express, then specify the PL/SQL gateway user as `APEX_PUBLIC_USER` and the corresponding database password.
	• If you are migrating from Oracle mod_plsql, then specify the PL/SQL gateway database username and database password.
	• If you are not using either Oracle Application Express or migrating from Oracle mod_plsql, then select 2 to skip this step.

Table 1-2 (Cont.) Options for Configuring Application Express or Migrating from mod_plsql

Options	Description
`Enter 1 to specify passwords for Application Express RESTful Services database user (APEX_LISTENER, APEX_REST_PUBLIC_USER) or 2 to skip this step [1]: Enter the database password for APEX_LISTENER: Confirm password: Enter the database password for APEX_REST_PUBLIC_USER: Confirm password:`	If you have specified APEX_PUBLIC_USER for the PL/SQL Gateway user, then you have the option of using Application Express RESTful Services. Specify 2 if you do not want to use Application Express RESTful Services and skip this step. For Application Express 5.0 and above, option 1 is required. The database users are created using the script `apex_rest_config.sql` provided in the Application Express installation media.

Table 1-3 Enabling Features in ORDS

Options	Description
`Enter a number to select a feature to enable: [1] SQL Developer Web (Enables all features) [2] REST Enabled SQL [3] Database API [4] REST Enabled SQL and Database API [5] None Choose [1]:`	Specify 1 to enable all the features: SQL Developer Web, REST Enabled SQL and Database API. Specify 2 for REST Enabled SQL or 3 for Database API. Specify 4 to enable both REST Enabled SQL and Database API. Refer to "Accessing SQL Developer Web", "REST Enabled SQL Service", and "Enabling ORDS Database API" documentation for more information.

Table 1-4 Options for Running in Standalone Mode

Options	Description
`Enter 1 if you wish to start in standalone mode or 2 to exit [1]:`	Specify 1 to start in standalone mode using the Jetty web server that is bundled with ORDS.
`Enter the APEX static resources location:/path/to/apex/images`	Specify the location of the Application Express images. This prompt will appear if you have specified APEX_PUBLIC_USER for the PL/SQL Gateway user.

Table 1-4 (Cont.) Options for Running in Standalone Mode

Options	Description
`Enter 1 if using HTTP or 2 if using HTTPS [1]: Enter the HTTP port [8080]:`	Specify the HTTP port if you choose 1.
`OR Enter 1 if using HTTP or 2 if using HTTPS [1]:2 Enter the HTTPS port [8443]: Enter the SSL hostname:mysslhost Enter 1 to use the self-signed certificate or 2 if you will provide the SSL certifi cate [1]:`	Specify the HTTPS port and the Secure Socket Layer (SSL) hostname if you choose 2. You have the option of using the self-signed certificate which generates the self-signed certificate automatically.
`OR Enter 1 to use the self-signed certificate or 2 if you will provide the SSL certifi cate [1]:2 Enter the path for the SSL Certificate:/path/to/sslcert Enter the path for the SSL Certificates private key:/path/to/sslcertprivatekey`	Specify the path for the SSL certificate and the path for SSL certificate private key if you choose 2.

Related Topics

- REST-Enabled SQL Service
- About the Oracle REST Data Services Configuration Files
- Starting in Standalone Mode
- Configuring and Installing Oracle REST Data Services
- *Installing Application Express and Configuring Oracle REST Data Services*
- Accessing SQL Developer Web
- Enabling ORDS Database API
  This section describes how to enable the Oracle REST Data Services (ORDS) Database API.

ORDS Parameter File

Oracle REST Data Services can be installed in either simple or silent mode without any user interaction.

You can perform either simple or silent installation for Oracle REST Data Services using the parameters specified in the `params/ords_params.properties` file under the location where you installed Oracle REST Data Services. This is the default Oracle REST Data Services parameter file. You can edit that file to change the default values to reflect your environment and preferences.

You can perform the installation for Oracle REST Data Services using the parameters specified in the `params/ords_params.properties` file under the location where you installed Oracle REST Data Services. This is the default Oracle REST Data Services parameter file. The Oracle REST Data Services parameter file consists of key or value pairs in the format `key=value`.

Alternatively, you have the option of specifying your own Oracle REST Data Services parameter file by including the `--parameterFile` option. If the `--parameterFile` option is not specified, the default Oracle REST Data Services parameter file is used.

Parameters Used in ORDS Parameter File

This section lists the parameters used in ORDS parameter file.

Topics:

- Parameters for Database Connection
- Parameters for Installing Oracle REST Data Services
- Parameters for Installing into the CDB
- Parameters for Configuring Application Express
- Parameters for Running in Standalone Mode
- Miscellaneous Parameters

Parameters for Database Connection

This section lists the database connection parameters that must be specified in the properties file. You can specify a Basic, TNS or Custom URL connection.

Topics:

- Parameters for Basic Connection
- Parameters for TNS Connection
- Parameters for Custom URL Connection

Parameters for Basic Connection

This section lists the parameters that must be specified in the properties file for basic database connection.

For basic connection, you must specify `db.hostname` and `db.port` database connection parameters. In addition, specify either `db.servicename` or `db.sid` parameters. If you are specifying a database connection to an Oracle 12.x PDB, then provide the `db.servicename` parameter.

Key	Type	Description	Example
`db.connectionType`	string	Specifies the connection type. The value is `basic`.	`basic`

Key	Type	Description	Example
db.hostname	string	Specifies the host system for the Oracle database.	myhostname
db.port	numeric	Specifies the database listener port.	1521
db.servicename	string	Specifies the network service name of the database.	orcl.example.com
db.sid	string	Specifies the name of the database.	orcl

Parameters for TNS Connection

This section lists the parameters for TNS connection to install ORDS.

If you use a TNS connection to install ORDS, then you must specify the oracle.net.tns_admin system property on the command line that contains the folder of the tnsnames.ora file. The tnsnames.ora file must exist in that folder. For example:

```
$ java -Doracle.net.tns_admin=/path/to/tnsFolder -jar ords.war install
    simple
```

Key	Type	Description	Example
db.connectionType	string	Specifies the connection type. The value is tns.	tns
db.tnsDirectory	string	Specifies the folder of where the tnsnames.ora file is located.	/path/to/tnsfolder
db.tnsAliasName	string	Specifies the tns alias name that must exist in the tnsnames.ora file.	orcl

Parameters for Custom URL Connection

This section lists the parameters for Custom URL connection to install ORDS.

Key	Type	Description	Example
db.connectionType	string	Specifies the connection type. The value is customurl.	customurl
db.customURL	string	Specifies the custom url.	jdbc:oracle:thin:@//localhost:1521/orcl
db.tnsAliasName	string	Specifies the tns alias name that must exist in the tnsnames.ora file.	orcl

Parameters for Installing Oracle REST Data Services

This section lists the parameters required for installing Oracle REST Data Services schema.

To install Oracle REST Data Services schema, following parameters must be specified:

- Username and password of the user with ORDS Installer privileges or with SYS AS SYSDBA account.
- ORDS_PUBLIC_USER password
- Existing default and temporary tablespaces for the ORDS_METADATA schema and ORDS_PUBLIC_USER.

> **Note:**
>
> If all of the default and temporary tablespace parameters are omitted in the Oracle REST Data Services parameter file, then the Oracle database default and temporary tablespaces are used.

Table 1-5 Parameters for Installing Oracle REST Data Services

Key	Type	Description	Example
`rest.services.ords.add`	boolean	Specifies whether to install the Oracle REST Data Services schema. Set the value to true. Supported values: • true • false (default)	`true`
`user.public.password`	string	Specifies the password for ORDS_PUBLIC_USER.	`password`
`schema.tablespace.default`	string	Specifies the ORDS_METADATA default tablespace. Specify an existing default tablespace.	`SYSAUX`
`schema.tablespace.temp`	string	Specifies the ORDS_METADATA temporary tablespace. Specify an existing temporary tablespace.	`TEMP`
`user.tablespace.default`	string	Specifies the ORDS_PUBLIC_USER default tablespace. Specify an existing default tablespace.	`SYSAUX`

Table 1-5 (Cont.) Parameters for Installing Oracle REST Data Services

Key	Type	Description	Example
`user.tablespace.temp`	string	Specifies the ORDS_PUBLIC_USER temporary tablespace. Specify an existing temporary tablespace.	`TEMP`
`bequeath.connect`	boolean	Specifies whether to connect as bequeath. Supported values: • true • false (default)	`true`

Related Topics

- Using a Bequeath Connection to Install, Upgrade, Validate, or Uninstall Oracle REST Data Services

Parameters for Enabling SQL Developer Web

This section lists the parameters for enabling SQL Developer Web.

Table 1-6 Parameters for Enabling SQL Developer Web

Key	Type	Description	Example
`feature.sdw`	string	Specifies if SQL Developer Web is enabled.	`true`

Note: For enabling SQL Developer Web, the value of re

Table 1-6 (Cont.) Parameters for Enabling SQL Developer Web

Key	Type	Description	Example
		`stEnabledSql.active` must also be `true`.	
		Default value is `false`.	
`restEnabledSql.active`	string	Specifies if REST-Enabled SQL is enabled. Default value is `false`.	`true`
database.api.enabled	string	Specifies if the Database API is enabled. Default value is `false`.	`true`

Parameters for Enabling REST-Enabled SQL

This section describes the parameter for enabling REST-Enabled SQL.

Table 1-7 Parameters for Enabling REST-Enabled SQL

Key	Type	Description	Example
`restEnabledSql.active`	string	Specifies if REST-Enabled SQL is enabled. Default value is `false`.	true

Parameters for Enabling Database API

This section describes the parameter for enabling Database API.

Table 1-8 Parameters for Enabling Database API

Key	Type	Description	Example
`database.api.enabled`	string	Specifies if Database API is enabled. Default value is `false`.	true

Parameters for Installing into the CDB

This section lists the parameters required for installing Oracle REST Data Services into the CDB and all of its PDBs.

Oracle database 12.x provides you the option of installing Oracle REST Data Services in the CDB and all of its PDBs.

> **Note:**
>
> Provide the CDB service name for `db.servicename` or `sid` for `db.sid`.

Table 1-9 Parameters for Installing into the CDB

Key	Type	Description	Example
`pdb.open.asneeded`	boolean	Specifies whether to open all PDBs in read write mode if their status is either closed or read only. If the value is set to true, then the following PDB parameters are ignored: • `pdb.open.readwrite` • `pdb.skip.closed` • `pdb.skip.readonly` Supported values: • true • false (default)	`true`
`pdb.open.readwrite`	string	Specifies the list of PDB service names to open for read write mode if their status is read only.	`PDB1, PDB2, MYPDB`

Table 1-9 (Cont.) Parameters for Installing into the CDB

Key	Type	Description	Example
`pdb.skip.closed`	boolean	Specifies whether to skip PDBs that are closed. Supported values: • true • false (default)	`true`
`pdb.skip.readonly`	boolean	Specifies whether to skip PDBs with read only status. Supported values: • true • false (default)	`true`
`pdb.exclude`	string	Specifies the list of PDB service names to exclude for install.	`PDB3, PDB4, PDB_X`

Related Topics

- Setting Up ORDS in a CDB Environment
  This section describes how to setup Oracle REST Data Services (ORDS) into a multitenant container database (CDB) environment.

Parameters for Configuring Application Express

This section lists the parameters for using Application Express.

Table 1-10 Parameters for Configuring Application Express

Key	Type	Description	Example
`plsql.gateway.add`	boolean	Specifies whether to configure Oracle REST Data Services for Application Express. Set this value to true. Supported values: • true • false (default)	`true`
`db.username`	string	Specifies the PL/SQL gateway username. For Application Express, you must specify APEX_PUBLIC_USER.	`APEX_PUBLIC_USER`

Table 1-10 (Cont.) Parameters for Configuring Application Express

Key	Type	Description	Example
`db.password`	string	Specifies the password for APEX_PUBLIC_USER. The password must match APEX_PUBLIC_USER database password.	`password`
`rest.services.apex.add`	boolean	Specifies whether to configure Oracle REST Data Services for Application Express RESTful Services. Supported values: • true • false (default) Set this value to true if you want to use APEX RESTful Services.	`true`
`user.apex.listener.password`	string	Specifies the password for APEX_LISTENER. If `rest.services.apex.add` is set to true, you must provide a password for APEX_LISTENER. The password must match APEX_LISTENER database password. Otherwise, if `rest.services.apex.add` is set to false, omit this parameter.	`password`
`user.apex.restpublic.password`	string	Specifies the password for APEX_REST_PUBLIC_USER. If `rest.services.apex.add` is set to true, you must provide a password for APEX_REST_PUBLIC_USER. The password must match APEX_REST_PUBLIC_USER database password. Otherwise, if `rest.services.apex.add` is set to false, omit this parameter.	`password`

Example 1-1 Parameters to configure for Application Express and APEX RESTful Services and run in standalone mode

Following example shows parameters to install Oracle REST Data Services, configure for Application Express and APEX RESTful Services and run in standalone mode using http:

> **Note:**
>
> Passwords in the parameter file will be encrypted during installation. The encrypted passwords are stored in the parameter file. For example, `user.public.password=@0585904F6C9B442532D5212962835D00C8`.

```
db.hostname=localhost
db.password=password
db.port=1521
db.servicename=orcl.example.com
db.username=APEX_PUBLIC_USER
plsql.gateway.add=true
rest.services.apex.add=true
rest.services.ords.add=true
schema.tablespace.default=SYSAUX
schema.tablespace.temp=TEMP
standalone.http.port=8080
standalone.mode=true
standalone.static.images=/path/to/images
standalone.use.https=false
user.apex.listener.password=password
user.apex.restpublic.password=password
user.public.password=password
user.tablespace.default=SYSAUX
user.tablespace.temp=TEMP
```

> **See Also:**
>
> - For information on APEX_PUBLIC_USER, refer to section Configure APEX_PUBLIC_USER Account, in Oracle Application Express Installation Guide.
> - For information on APEX_LISTENER and APEX_REST_PUBLIC_USER, refer to section, Configuring RESTful Services with Oracle REST Data Services in Oracle Application Express Installation Guide.

Parameters for Running in Standalone Mode

This section lists parameters for running Oracle REST Data Services in standalone mode.

Table 1-11 Parameters for Installing Oracle REST Data Services in Standalone Mode

Key	Type	Description	Example
`standalone.mode`	boolean	Indicates whether to use the web application server (Jetty) that is included with Oracle REST Data Services. Supported values: • true • false (default)	`true`
`standalone.http.port`	numeric	Specifies the HTTP listener port.	`8080`
`standalone.use.https`	boolean	Specifies whether to use https.	`true`
`standalone.https.port`	numeric	Specifies HTTPS listener port.	`8443`
`standalone.ssl.host`	string	Specifies the Secure Socket Layer (SSL) certificate hostname. You must specify this option if you are using https.	`mysecurehost`
`standalone.use.ssl.cert`	boolean	Specifies whether you will provide the SSL certificate. If this value is set to true, you must specify the `standalone.ssl.cert.path` and `standalone.ssl.key.path`.	`true`
`standalone.ssl.cert.path`	string	Specifies the SSL certificate path. If you are providing the SSL certificate, you must specify the certificate location.	`/path/to/ssl/cert`
`standalone.ssl.key.path`	string	Specifies the SSL certificate key path. If you are providing the SSL certificate, you must specify the certificate key location.	`/path/to/ssl/key`

Table 1-11 (Cont.) Parameters for Installing Oracle REST Data Services in Standalone Mode

Key	Type	Description	Example
`standalone.static.images`	string	Specifies the location of Application Express images. If you are using Application Express, specify the location of Application Express images.	`/path/to/apex/images`

> **Note:**
>
> On Microsoft Windows systems, if you specify an Application Express static images location for `standalone.static.images`, use the backslash character (/) before the colon, and use a forwardslash for the folder separator. For example, `standalone.static.images=d\:/test/apex426/apex/images/`

Example 1-2 Parameters to run in standalone mode using http

Following code snippet shows an example of the list of parameters to specify for installing Oracle REST Data Services and running in standalone mode using http:

```
db.hostname=localhost
db.port=1521
db.servicename=orcl.example.com
plsql.gateway.add=false
rest.services.apex.add=false
rest.services.ords.add=true
schema.tablespace.default=SYSAUX
schema.tablespace.temp=TEMP
standalone.http.port=8080
standalone.mode=true
standalone.use.https=false
user.public.password=password
user.tablespace.default=SYSAUX
user.tablespace.temp=TEMP
```

Example 1-3 Parameters to run in standalone mode using https and providing the ssl certificate paths

Following code snippet shows an example of the list of parameters to specify for installing and running Oracle REST Data Services in standalone mode using https and providing the ssl certificate paths:

```
db.hostname=localhost
db.port=1521
db.servicename=orcl.example.com
plsql.gateway.add=false
rest.services.apex.add=false
```

```
rest.services.ords.add=true
schema.tablespace.default=SYSAUX
schema.tablespace.temp=TEMP
standalone.https.port=8443
standalone.mode=true
standalone.ssl.cert.path=/path/to/ssl/cert
standalone.ssl.host=mysecurehost
standalone.ssl.key.path=/path/to/ssl/key
standalone.use.https=true
standalone.use.ssl.cert=true
user.public.password=password
user.tablespace.default=SYSAUX
user.tablespace.temp=TEMP
```

Related Topics

- Running in Standalone Mode

Miscellaneous Parameters

This section lists some miscellaneous parameters.

Table 1-12 Miscellaneous Parameters

Key	Type	Description	Example
`migrate.apex.rest`	boolean	Specifies whether to migrate APEX RESTful Services definitions to Oracle REST Data Services schema. Supported values: • true • false (default)	`true`

Simple Installation Using a Parameter File

Oracle REST Data Services can be installed in simple mode without any user interaction.

You can perform a simple installation of Oracle REST Data Services using an ORDS parameters file. A simple installation prompts you for the information if the required parameter does not exist in the ORDS parameter file.

Following is an example code snippet for installing Oracle REST Data Services in simple mode:

```
java -jar ords.war install simple
java -jar ords.war install --parameterFile /path/to/
my_params.properties simple

java -jar ords.war install
java -jar ords.war install --parameterFile /path/to/my_params.properties
```

```
java -jar ords.war
```

> **Note:**
>
> Use the following on-line help command to check for additional options available for the install command: `java –jar ords.war help install`

Silent Installation Using a Parameter File

Oracle REST Data Services can be installed in silent mode without any user interaction.

You can perform a silent installation of Oracle REST Data Services using an ORDS parameters file. A silent installation must have the required parameters defined in the parameter file; otherwise, an error occurs.

Following is an example code snippet for installing Oracle REST Data Services in silent mode:

```
java -jar ords.war install --silent
java-jar ords.war install --silent --parameterFile /path/to/
my_params.properties
```

> **Note:**
>
> Use the following on-line help command to check for additional options available for the install command: `java –jar ords.war help install`

Parameters Required for Silent Installation

This section describes the parameters required for installing Oracle REST Data Services in silent mode.

If you want to install Oracle REST Data Services in silent mode, then the required parameters must be defined in the ORDS parameter file.

Parameter Group	Required Parameters	Description
Database Connection	Refer to "Parameters for Database Connection" section for the list of parameters.	Specify Basic, TNS or Custom URL connection.
Installing ORDS	`rest.services.ords.add` Supported values: • `true` • `false`	If `rest.services.ords.add=true`, then refer to "Parameters Used in ORDS Parameter File." section for additional parameters.

Parameter Group	Required Parameters	Description
Configuring for Application Express or PL/SQL Gateway	`plsql.gateway.add` Supported values: • `true` • `false`	If `plsql.gateway.add=true`, then refer to "Parameters for Configuring Application Express" section for additional required parameters.
Running in Standalone Mode	`standalone.mode` Supported values: • true • false	If `standalone.mode=true`, then refer to "Parameters for Running Oracle REST Data Services in Standalone Mode" for additional required parameters.

Related Topics

- Parameters for Database Connection
  This section lists the database connection parameters that must be specified in the properties file. You can specify a Basic, TNS or Custom URL connection.
- Parameters for Installing Oracle REST Data Services
  This section lists the parameters required for installing Oracle REST Data Services schema.
- Parameters Used in ORDS Parameter File
  This section lists the parameters used in ORDS parameter file.
- Parameters for Configuring Application Express
  This section lists the parameters for using Application Express.
- Running in Standalone Mode

Changing Default Configuration from the Command Line

This section describes how you can update the ORDS default configuration file.

The following set property command is used to update the ORDS default configuration file:

```
$ java -jar ords.war set-property <property key> <value>
```

ORDS must be restarted for the changes to take effect.

Example 1-4

Examples of Enabling a Feature.

`default.xml`

```
$ java -jar ords.war set-property feature.sdw true
$ java -jar ords.war set-property restEnabledSql.active true
$ java -jar ords.war set-property database.api.enabled true
```

Example 1-5

The following example updates the properties in the existing `default.xml` file to enable REST-Enable SQL.

```
$ java -jar ords.war set-property restEnabledSql.active true
```

Example 1-6

The following example updates the properties in the existing `default.xml` file to enable Database API.

```
$ java -jar ords.war set-property database.api.enabled true
```

Validating the Oracle REST Data Services Installation

If you want to check that the Oracle REST Data Services installation is valid, go to the directory or folder containing the `ords.war` file and enter the `validate` command in the following format:

```
java -jar ords.war validate [--database <dbname>]
```

> **Note:**
>
> When you install Oracle REST Data Services, it attempts to find the Oracle Application Express (APEX) schema and creates a view. This view joins the relevant tables in the APEX schema to the tables in the Oracle REST Data Services schema. If you install Oracle REST Data Services before APEX, then Oracle REST Data Services cannot find the APEX schema and it creates a stub view in place of the missing APEX tables.
>
> Oracle highly recommends that you install Oracle REST Data Services after APEX to ensure that the APEX objects, which Oracle REST Data Services needs to query, are present. If you install Oracle REST Data Services before APEX, then use the `validate` command to force Oracle REST Data Services to reconstruct the queries against the APEX schema.

If `--database` is specified, *<dbname>* is the pool name that is stored in the Oracle REST Data Services configuration files.

You are prompted for any necessary information that cannot be obtained from the configuration of pool name, such as host, port, SID or service name, and the name and password of the user with ORDS Installer privileges, or SYS AS SYSDBA user.

> **Note:**
>
> If the validate command is run against a CDB, then it will validate the CDB and all of its PDBs.

> **Note:**
>
> If you use a TNS connection to validate ORDS, then you must specify the `oracle.net.tns_admin` system property on the command line that contains the folder with `tnsnames.ora` file.
>
> **Example:**
>
> ```
> $ java -Doracle.net.tns_admin=/path/to/tnsFolder -jar ords.war
> validate
> ```

If You Want to Reinstall or Uninstall (Remove) Oracle REST Data Services

If you want to reinstall Oracle REST Data Services, you must first uninstall the existing Oracle REST Data Services; and before you uninstall, ensure that Oracle REST Data Services is stopped.

Uninstalling Oracle REST Data Services removes the ORDS_METADATA schema, the ORDS_PUBLIC_USER user, and Oracle REST Data Services-related database objects (including public synonyms) if they exist in the database. To uninstall (remove, or deinstall) Oracle REST Data Services, go to the directory or folder containing the `ords.war` file and enter the `uninstall` command as follows:

```
java -jar ords.war uninstall
```

The `uninstall` command prompts you for some necessary information (host, port, SID or service name, username, password).

> **See Also:**
>
> To uninstall Oracle REST Data Services from a CDB, see Using the Multitenant Architecture with Oracle REST Data Services .

> **Note:**
>
> If you use a TNS connection to uninstall ORDS, then you must specify the `oracle.net.tns_admin` system property on the command line that contains the folder with `tnsnames.ora` file.
>
> **Example:**
>
> ```
> $ java -Doracle.net.tns_admin=/path/to/tnsFolder -jar ords.war
> uninstall
> ```

Using SQL Developer Oracle REST Data Services Administration (Optional)

This section describes how to use Oracle SQL Developer to administer Oracle REST Data Services.

> **See Also:**
>
> "Oracle REST Data Services Administration" in *Oracle SQL Developer User's Guide*

Topics:

- About SQL Developer Oracle REST Data Services Administration
- Configuring an Administrator User

About SQL Developer Oracle REST Data Services Administration

Oracle SQL Developer enables you to administer Oracle REST Data Services using a graphical user interface. To take full advantage of these administration capabilities, you must use SQL Developer Release 4.1 or later. Using SQL Developer for Oracle REST Data Services administration is optional.

Using this graphical user interface, you can update the database connections, JDBC settings, URL mappings, RESTful connections, security (allowed procedures, blocked procedures, validation function and virus scanning), Caching, Pre/Post Processing Procedures, Environment, and Excel Settings. Oracle SQL Developer also provides statistical reporting, error reporting, and logging.

> **See Also:**
>
> "Oracle REST Data Services Administration" in *Oracle SQL Developer User's Guide*

Configuring an Administrator User

If you want to be able to administer Oracle REST Data Services using SQL Developer, then you must configure an administrator user as follows:

- Execute the following command:

  ```
  java -jar ords.war user adminlistener "Listener Administrator"
  ```

- Enter a password for the `adminlistener` user.
- Confirm the password for the `adminlistener` user.
- If you are using Oracle REST Data Services without HTTPS, follow the steps listed under the section,**Using OAuth2 in Non-HTTPS Environments**.

When using SQL Developer to retrieve and/or upload an Oracle REST Data Services configuration, when prompted, enter the credentials provided in the preceding list.

Using OAuth2 in Non-HTTPS Environments

RESTful Services can be protected with the OAuth2 protocol to control access to nonpublic data. To prevent data snooping, OAuth2 requires all requests involved in the OAuth2 authentication process to be transported using HTTPS. The default behavior of Oracle REST Data Services is to verify that all OAuth2 related requests have been received using HTTPS. It will refuse to service any such requests received over HTTP, returning an HTTP status code of 403 Forbidden.

This default behavior can be disabled in environments where HTTPS is not available as follows:

1. Locate the folder where the Oracle REST Data Services configuration is stored.
2. Edit the file named `defaults.xml`.
3. Add the following setting to the end of this file just before the `</properties>` tag.

   ```
   <entry key="security.verifySSL">false</entry>
   ```

4. Save the file.
5. Restart Oracle REST Data Services if it is running.

Note that it is only appropriate to use this setting in development or test environments. It is never appropriate to use this setting in production environments because it will result in user credentials being passed in clear text.

> **Note:**
>
> Oracle REST Data Services must be restarted after making configuration changes. See your application server documentation for information on how to restart applications.

Running in Standalone Mode

Although Oracle REST Data Services supports the Java EE application servers, you also have the option of running in standalone mode. This section describes how to run Oracle REST Data Services in standalone mode.

Standalone mode is suitable for development use and is supported in production deployments. Standalone mode, however, has minimal management capabilities when compared to most Java EE application servers and may not have adequate management capabilities for production use in some environments.

Topics:

- Starting in Standalone Mode
- Stopping the Server in Standalone Mode
- Configuring a Doc Root for Non-Application Express Static Resources

Related Topics

- Supported Java EE Application Servers

Starting in Standalone Mode

To launch Oracle REST Data Services in standalone mode:

1. To start Standalone mode, execute the following command:

   ```
   java -jar ords.war standalone
   ```

   If you have not yet completed the standalone configuration, you are prompted to do so.

   > **Tip:**
   >
   > To see help on standalone mode options, execute the following command:
   >
   > ```
   > java -jar ords.war help standalone
   > ```

   > **Note:**
   >
   > If you want to use RESTful services that require secure access, you should use HTTPS.

2. When prompted, specify the location of the folder containing the Oracle Application Express static resources used by Oracle REST Data Services, or press **Enter** if you do not want to specify this location.
3. When prompted select if you want Oracle REST Data Services to generate a self-signed certificate automatically or if you want to provide your own certificate. If you want to use your own certificate, provide the path for the Certificate and DER encoded related private key when prompted.

   If the private key has not already been converted to DER, see section, **Converting a Private Key to DER (Linux and Unix)** before you enter the values here.

   You are only prompted for these values the first time you launch standalone mode.

   > **Note:**
   >
   > Ensure that no other servers are listening on the port you choose. The default port `8080` is commonly used by HTTP or application servers, including the embedded PL/SQL gateway; the default secure port `8443` is commonly used by HTTPS.

Related Topics

- Using OAuth2 in Non-HTTPS Environments
- Converting a Private Key to DER (Linux and Unix)

Converting a Private Key to DER (Linux and Unix)

Usually, you would have created a private key and a Certificate Signing Request before obtaining your signed certificate. The private key needs to be converted into DER in order for Oracle REST Data Services to read it properly.

For example, assume that the original private key was created using the OpenSSL tool with a command similar to either of the following:

```
openssl req -new -newkey rsa:2048 -nodes -keyout yourdomain.key -out
yourdomain.csr
```

or

```
openssl genrsa -out private.em 2048
```

In this case, you must run a command similar to the following to convert it and remove the encryption: openssl pkcs8 -topk8 -inform PEM -outform DER -in yourdomain.key -out yourdomain.der -nocrypt

```
openssl pkcs8 -topk8 -inform PEM -outform DER -in yourdomain.key -out
yourdomain.der -nocrypt
```

After doing this, you can include the path to `yourdomain.der` when prompted by Oracle REST Data Services, or you can modify the following entries in `conf/ords/standalone/standalone.properties`:

```
ssl.cert=<path to yourdomain.crt>
ssl.cert.key=<path to yourdomain.der>
ssl.host=yourdomain
```

Also, ensure that `jetty.secure.port` is set.

Stopping the Server in Standalone Mode

To stop the Oracle REST Data Services server in standalone mode, at a command prompt press **Ctrl+C**.

Configuring a Doc Root for Non-Application Express Static Resources

You can configure a doc root for standalone mode to deploy static resources that are outside the `/i` folder that is reserved for Application Express static resources.

To do so, specify the `--doc-root` parameter with the standalone mode command, as in the following example:

```
java -jar ords.war standalone --doc-root /var/www/html
```

The preceding example makes any resource located within `/var/www/html` available under `http://server:port/`. For example, if the file `/var/www/html/hello.txt` exists, it will be accessible at `http://server:port/hello.txt`.

The value specified for `--doc-root` is stored in `${config.dir}/ords/standalone/standalone.properties` in the `standalone.doc.root` property. If a custom doc root is not specified using `--doc-root`, then the default `doc-root` value of `${config.dir}/ords/standalone/doc_root` is used. Any file placed within this folder will be available at the root context.

This approach has the following features and considerations:

- HTML resources can be addressed without their file extension. For example, if a file named `hello.html` exists in the doc root, it can be accessed at the URI `http://server:port/hello`.
- Attempts to address a HTML resource with its file extension are redirected to the location without an extension. For example, if the URI `http://server:port/hello.html` is accessed, then the client is redirected to `http://server:port/hello`.

  The usual practice is to serve HTML resources without their file extensions, so this feature facilitates that practice, while the redirect handles the case where the resource is addressed with its file extension.
- Index pages for folders are supported. If a folder contains a file named `index.html` or `index.htm`, then that file is used as the index page for the folder. For example, if `/var/www/html` contains `/abc/xyz/index.html`, then accessing `http://server:port/abc/xyz/` displays the contents of `index.html`.
- Addressing a folder without a trailing slash causes a redirect to the URI with a trailing slash. For example, if a client accesses `http://server:port/abc/xyz`, then the server issues a redirect to `http://server:port/abc/xyz/`.
- Resources are generated with weak etags based on the modification stamp of the file and with a Cache Control header that causes the resources to be cached for 1 hour.

Deploying to Oracle WebLogic Server

This section describes how to deploy Oracle REST Data Services on Oracle WebLogic Server. It assumes that you have completed the installation process and are familiar with Oracle WebLogic Server. If you are unfamiliar with domains, managed servers, deployment, security, users and roles, refer to your Oracle WebLogic Server documentation.

Topics:

- About Oracle WebLogic Server
- Downloading, Installing, and Configuring Oracle REST Data Services
- Configuring Oracle Application Express Images
- Launching the Administration Server Console
- Installing the Oracle WebLogic Server Deployment
- Configuring WebLogic to Handle HTTP Basic Challenges Correctly
- Verifying the State and Health of ords and i

About Oracle WebLogic Server

You can download Oracle WebLogic Server from Oracle Technology Network.

To learn more about installing Oracle WebLogic Server, see *Oracle Fusion Middleware Getting Started With Installation for Oracle WebLogic Server* and *Oracle Fusion Middleware Installation Guide for Oracle WebLogic Server*.

> **See Also:**
>
> `weblogic_downloads`

Downloading, Installing, and Configuring Oracle REST Data Services

You must complete this step before deploying Oracle REST Data Services on WebLogic.

Related Topics

- Configuring and Installing Oracle REST Data Services

Configuring Oracle Application Express Images

If you are using Oracle Application Express, you must create a web archive to reference the Oracle Application Express, image files. However, if you are **not** using Oracle Application Express, you may skip the rest of this section about configuring Oracle Application Express images.

Before you begin, you must create a web archive (WAR) file to reference the Oracle Application Express image files. Use the static command to create a web archive file named `i.war`:

```
java -jar ords.war static <apex directory>\images
```

Where:

- *<apex directory>* is the directory location of Oracle Application Express.

This command runs the `static` command contained in the `ords.war` file. It packages the Application Express static images into an archive file named `i.war`.

The created images WAR does not contain the static resources; instead, it references the location where the static resources are stored. Therefore the static resources must be available at the specified path on the server where the WAR is deployed.

> **Tip:**
>
> Use `java -jar ords.war help static` to see the full range of options for the `static` command.

Use the `i.war` file to deploy to WebLogic in the following steps:

1. Launching the Administration Server Console
2. Installing the Oracle WebLogic Server Deployment
3. Configuring WebLogic to Handle HTTP Basic Challenges Correctly

Launching the Administration Server Console

To launch the Administration Server console:

1. Start an Administration Server.
2. Launch the WebLogic Administration Console by typing the following URL in your web browser:

   `http://<host>:<port>/console`

   Where:

   - *<host>* is the DNS name or IP address of the Administration Server.
   - *<port>* is the port on which the Administration Server is listening for requests (port `7001` by default).
3. Enter your WebLogic Administrator username and password.
4. If your domain is in *Production* mode, click the **Lock & Edit** button on the left-pane below the submenu Change Center. If your domain is in *Development* mode, this button does not appear.

Installing the Oracle WebLogic Server Deployment

> **Tip:**
>
> The Oracle REST Data Services files, `ords.war` and `i.war`, must be available before you start this task.

To install the deployment:

1. Go to the WebLogic Server Home Page. Below Domain Configuration, select **Deployments**.

   The Summary of Deployments is displayed.
2. Click **Install**.
3. Specify the location of the `ords.war` file and click **Next**.

   The `ords.war` file is located in the folder where you unzipped the Oracle REST Data Services ZIP file.

> **Tip:**
>
> WebLogic Server determines the context root from the file name of a WAR archive. If you need to keep backward compatibility, so that URLs are of the form *http://server/apex/...* rather than *http://server/ords/...*, then you must rename `ords.war` to `apex.war` before the deployment.

The Install Application assistant is displayed.

4. Select **Install this deployment as an application** and click **Next**.
5. Select the servers and/or clusters to which you want to deploy the application or module and click **Next**.

> **Tip:**
>
> If you have not created additional Managed Servers or clusters, you do not see this assistant page.

6. In the Optional Settings, specify the following:
   a. Name - Enter:

      `ords`

   b. Security - Select the following:

      Custom Roles: Use roles that are defined in the Administration Console; use policies that are defined in the deployment descriptor

   c. Source accessibility - Select:

      Use the defaults defined by the deployment's targets
7. Click **Next**.

   A summary page is displayed.
8. Under Additional configuration, select one of the following:
   - **Yes, take me to the deployment's configuration** - Displays the Configuration page.
   - **No I will review the configuration later** - Returns you to the Summary of Deployments page.
9. Review the summary of configuration settings that you have specified.
10. Click **Finish**.
11. Repeat the previous steps to deploy the `i.war` file.

    In the optional settings, specify the following:

    a. Name - Enter:

       `i`

    b. Security - Select:

       Custom Roles: Use roles that are defined in the Administration Console; use policies that are defined in the deployment descriptor

c. Source Accessibility - Select:

Use the defaults defined by the deployment's targets

12. If your domain is in Production Mode, then on the Change Center click **Activate Changes**.

Related Topics

- Configuring and Installing Oracle REST Data Services
- Configuring Oracle Application Express Images

Configuring WebLogic to Handle HTTP Basic Challenges Correctly

By default WebLogic Server attempts to intercept all HTTP Basic Authentication challenges. This default behavior needs to be disabled for Oracle REST Data Services to function correctly. This is achieved by updating the `enforce-valid-basic-auth-credentials` flag. The WebLogic Server Administration Console does not display the `enforce-valid-basic-auth-credentials` setting. You can use WebLogic Scripting Tool (WLST) commands to check, and edit the value in a running server.

The following WLST commands display the domain settings:

```
connect('weblogic','weblogic','t3://localhost:7001')
cd('SecurityConfiguration')
cd('mydomain')
ls()
```

If the domain settings displayed, contains the following entry:

```
-r--   EnforceValidBasicAuthCredentials             true
```

Then you must set this entry to `false`.

To set the entry to `false`, use the WLST commands as follows:

```
connect('weblogic', 'weblogic', 't3://localhost:7001')
edit()
startEdit()
cd('SecurityConfiguration')
cd('mydomain')
set('EnforceValidBasicAuthCredentials','false')
save()
activate()
disconnect()
exit()
```

> **Note:**
>
> WebLogic Server must be restarted for the new settings to take effect.

In the preceding example:

- `weblogic` is the WebLogic user having administrative privileges
- `weblogic` is the password
- `mydomain` is the domain
- The AdminServer is running on the `localhost` and on port `7001`

Related Topics

- WebLogic Server Command Reference

Verifying the State and Health of ords and i

In the Summary of Deployments, select the **Control** tab and verify that both the `ords` and `i` State are Active and the Health status is OK.

If `ords` and/or `i` are not Active, then enable them. In the Deployments table, select the check box next to `ords` and/or `i`. Click **Start** and select **Servicing all requests** to make them active.

Deploying to Apache Tomcat

This section describes how to deploy Oracle REST Data Services on Apache Tomcat.

Topics:

- About Apache Tomcat
- Downloading, Installing, and Configuring Oracle REST Data Services
- Configuring Oracle Application Express Images
- Installing the Apache Tomcat Deployment

About Apache Tomcat

> **Tip:**
>
> This section assumes that you have completed the installation process and are familiar with Apache Tomcat. If you are unfamiliar with domains, servers, applications, security, users and roles, see your Apache Tomcat documentation.

You can download Apache Tomcat from:

> **See Also:**
>
> apache_tomcat_80

Downloading, Installing, and Configuring Oracle REST Data Services

You must complete this step before deploying Oracle REST Data Services on Apache Tomcat.

Related Topics

- Configuring and Installing Oracle REST Data Services

Configuring Oracle Application Express Images

If you are using Oracle Application Express, you must create a web archive to reference the Oracle Application Express, image files. However, if you are **not** using Oracle Application Express, you may skip the rest of this section about configuring Oracle Application Express images.

To configure Oracle Application Express Images on Apache Tomcat:

- Copy the contents of the <*apex directory*>/images folder to <*Tomcat directory*>/webapps/i/.

  Where:

  – <*apex directory*> is the directory location of the Oracle Application Express distribution.

  – <*Tomcat directory*> is the folder where Apache Tomcat is installed.

Installing the Apache Tomcat Deployment

> **Tip:**
>
> The Oracle REST Data Services file ords.war must be available before you start this task.

To install the Apache Tomcat deployment:

1. Move the ords.war file into the webapps folder where Apache Tomcat is installed.

   > **Tip:**
   >
   > Apache Tomcat determines the context root from the file name of a WAR archive. If you need to keep backward compatibility, so that URLs are of the form *http://server/apex/...* rather than *http://server/ords/...*, then you must rename ords.war to apex.war before moving it into to the webapps folder.

2. Access Oracle Application Express by typing the following URL in your web browser:

   http://<*hostname*>:<*port*>/ords/

Where:

- *<hostname>* is the name of the server where Apache Tomcat is running.
- *<port>* is the port number configured for Apache Tomcat application server.

Related Topics

- Configuring and Installing Oracle REST Data Services
- Configuring Oracle Application Express Images

Upgrading Oracle REST Data Services

If you want to upgrade to a new release of Oracle REST Data Services, you must do the following:

1. Stop the Oracle REST Data Services instance.
   - If you are running Oracle REST Data Services on your application server (such as Oracle WebLogic Server, or Apache Tomcat), stop Oracle REST Data Services.
   - If you are running Oracle REST Data Services in standalone mode, refer to section, **Stopping the Server in Standalone Mode**.
2. Go to the folder where you unzipped the new Oracle REST Data Services release distribution.
3. Enter the following on the command line:

   ```
   java -jar ords.war install advanced
   ```

   or

   ```
   java -jar ords.war
   ```

4. When prompted for the configuration folder, use the configuration folder where the Oracle REST Data Services configuration files are stored. (The configuration location will be stored in the `ords.war` file.)
   - If you specified an existing Oracle REST Data Services configuration folder that contains the configuration files, Oracle REST Data Services will attempt to connect to each database defined in the configuration folder and check the installed version.
   - If you specified an Oracle REST Data Services configuration folder that does not exist, you will be prompted for the database connection information, the ORDS_PUBLIC_USER credentials, and additional configuration information. Oracle REST Data Services will attempt to connect to this database and check the installed version.

When Oracle REST Data Services checks the installed version, it does the following, depending on whether an earlier 3.0.*n* version is already installed in the database.

- If the installed version is an earlier 3.0.*n* version of Oracle REST Data Services, you are prompted for the username and password (user with ORDS Installer privileges or SYS AS SYSDBA) to enable Oracle REST Data Services to apply the in-place upgrade. The in-place upgrade will modify the existing installation to add

the updated schema objects and packages. The existing metadata stored in the Oracle REST Data Services schema will remain intact.

- If Oracle REST Data Services is not already installed in the database (or if you are upgrading from Release 2.0.*n*), you are prompted for the username and password (user with ORDS Installer privileges or SYS AS SYSDBA) to enable Oracle REST Data Services to perform the installation, and you will also be prompted for the default and temporary tablespaces for the `ORDS_METADATA` schema and `ORDS_PUBLIC_USER`.

When the upgrade or installation completes, you can re-deploy the `ords.war` file to your application server or start Oracle REST Data Services in standalone mode.

Related Topics

- Troubleshooting Oracle REST Data Services
- Stopping the Server in Standalone Mode

Using a Bequeath Connection to Install, Upgrade, Validate, or Uninstall Oracle REST Data Services

You can use the bequeath connection to install, upgrade, validate, or uninstall Oracle REST Data Services. The installer will not prompt you for the SYS username and password for the operation

In the parameter file, add the property: `bequeath.connect=true`

Using a bequeath connection for installing, validating, or uninstalling Oracle REST Data Services is supported on Linux and Windows systems for Oracle Database Release 12, but only on Linux systems for Oracle Database Release 11.

The command used must be run by an operating system user that is a member of the DBA group. Example of installing Oracle REST Data Services:

```
java -jar ords.war
```

Bequeath Connection Using Linux

On a Linux system, you must set the following environment variables to use the bequeath connection:

- ORACLE_HOME
- ORACLE_SID
- LD_LIBRARY_PATH (to point to `ORACLE_HOME/lib`)

For Oracle Database Release 11 (but not for Release 12), you must specify the option `-DuseOracleHome=true`. Examples of installing Oracle REST Data Services on a Linux system:

- For Oracle Database Release **11**: `java -DuseOracleHome=true -jar ords.war`
- For Oracle Database Release **12**: `java -jar ords.war`

Authorizing Oracle REST Data Services to Access Oracle Data Guard Protected Users

To access the database schema objects that are protected by an Oracle Data Vault Realm, it is necessary to grant a proxy user authorization to the Oracle REST Data Services Public User.

The following example authorizes the Oracle REST Data Services Public User, `ORDS_PUBLIC_USER` to proxy the database `HR` user:

```
begin
  DBMS_MACADM.AUTHORIZE_PROXY_USER('ORDS_PUBLIC_USER','HR');
end;
/
```

2

Configuring Oracle REST Data Services (Advanced)

This section explains how to configure Oracle REST Data Services for connecting to multiple databases for routing requests, and it refers to other documentation sources for other configuration information.

> **Note:**
>
> Oracle REST Data Services must be restarted after making configuration changes. See your application server documentation for information on how to restart applications.

Topics:

- Configuring Multiple Databases
- Support for Oracle RAC Fast Connection Failover
- Configuring Security, Caching, Pre- and Post Processing, Environment, and Excel Settings
- Configuring REST-Enabled SQL Service Settings
- Configuring the Maximum Number of Rows Returned from a Query
- Configuring the Custom Error Pages
- Developing RESTful Services for Use with Oracle REST Data Services

Configuring Multiple Databases

Oracle REST Data Services supports the ability to connect to more than one database. This section describes different strategies for routing requests to the appropriate database.

Topics:

- About the Request URL
- Configuring Additional Databases
- Routing Based on the Request Path Prefix
- Routing Based on the Request URL Prefix

About the Request URL

Oracle REST Data Services supports a number of different strategies for routing requests to the appropriate database. All of these strategies rely on examining the

request URL and choosing the database based on some kind of match against the URL. It is useful to recap the pertinent portions of a request URL. Consider the following URL:

```
https://www.example.com/ords/sales/f?p=1:1
```

This URL consists of the following sections:

- Protocol: `https`
- Host Name: `www.example.com`
- Context Root: `/ords`

  The context root is the location at which Oracle REST Data Services is deployed on the application server.
- Request Path: `/sales/f?p=1.1`

  This is the portion of the request URL relative to the context root.

For different applications, it may be important to route requests based on certain prefixes in the request path or certain prefixes in the full request URL.

There are two steps to configuring multiple databases:

1. Configuring the database connection information
2. Configuring which requests are routed to which database

Configuring Additional Databases

When you first configure Oracle REST Data Services, you configure a default database connection named: `apex`. You can create additional database connections using the `setup` command.

> **Tip:**
>
> To see full help for the `setup` command type:
>
> ```
> java -jar ords.war help setup
> ```

To create a database connection type the following:

```
java -jar ords.war setup --database <database name>
```

Where:

- *<database name>* is the name you want to give the database connection.

You are prompted to enter the information required to configure the database. After you have configured the additional databases, define the rules for how requests are routed to the appropriate database.

Related Topics

- Configuring and Installing Oracle REST Data Services
- Routing Based on the Request Path Prefix

- Routing Based on the Request URL Prefix

Routing Based on the Request Path Prefix

You create request routing rules using the `map-url` command.

> **Tip:**
>
> To see full help for the `map-url` command type:
>
> ```
> java -jar ords.war help map-url
> ```

If you want to route requests based just on matching a prefix in the request path portion of the URL, use the `map-url` command as follows:

```
java -jar ords.war map-url --type base-path --workspace-id <workspace name>
<path prefix> <database name>
```

Where:

- *<workspace name>* is the name of the Oracle Application Express workspace where RESTful services for this connection are defined. This may be omitted if RESTful Services are not being used.
- *<path prefix>* is the prefix that must occur at the start of the request path.
- *<database name>* is the name of the database connection configured in the previous step.

Related Topics

- Configuring Additional Databases

Example of Routing Based on the Request Path Prefix

Assuming Oracle REST Data Services is deployed on a system named `example.com` at the context path `/ords`, then create the following rule:

```
java -jar ords.war map-url --type base-path --workspace-id sales_rest /sales
sales_db
```

This rule means that any requests matching `https://example.com/ords/sales/...` are routed to the `sales_db` database connection. The `sales_rest` workspace defined within the `sales_db` database is searched for RESTful Services definitions.

The previous rule matches all of the following requests:

```
https://example.com/ords/sales/f?p=1:1
https://example.com/ords/sales/leads/
https://www.example.com/ords/sales/forecasting.report?month=jan  (If
www.example.com resolves to the same system as example.com.)
```

The previous rule does not match of any of the following requests:

```
http://example.com/ords/sales/f?p=1:1  (The protocol is wrong.)
https://example.com:8080/ords/sales/f?p=1:1  (The port is wrong: 443 is default
for https, but don't specify if using default.)
```

```
https://example.com/ords/f?p=1:1  (Missing the /sales prefix.)
https://example.com/pls/sales/leads/  (The context path is wrong.)
```

Routing Based on the Request URL Prefix

If you want to route requests based on a match of the request URL prefix, use the `map-url` command as follows:

```
java -jar ords.war map-url --type base-url --workspace-id <workspace name> <url
prefix> <database name>
```

Where:

- *<workspace name>* is the name of the Oracle Application Express workspace where RESTful services for this connection are defined. This may be omitted if RESTful Services are not being used.
- *<url prefix>* is the prefix with which the request URL must start.
- *<database name>* is the name of the database connection.

Example of Routing Based on the Request URL Prefix

Assuming Oracle REST Data Services is deployed on a system named `example.com` at the context path `/ords`, then create the following rule:

```
java -jar ords.war map-url --type base-url --workspace-id sales_rest https://
example.com/ords/sales sales_db
```

This rule means that any requests matching `https://example.com/ords/sales/...` are routed to the `sales_db` database connection. The `sales_rest` workspace defined within the `sales_db` database is searched for RESTful Services definitions.

The previous rule matches all of the following requests:

```
https://example.com/ords/sales/f?p=1:1
https://example.com/ords/sales/leads/
https://example.com/ords/sales/forecasting.report?month=jan
```

The previous rule does not match of any of the following requests:

```
http://example.com/ords/sales/f?p=1:1  (The protocol is wrong.)
https://example.com:8080/ords/sales/f?p=1:1  (The port is wrong: 443 is default
for https, but don't specify if using default.)
https://example.com/ords/f?p=1:1  (Missing the /sales segment of the base URL.)
https://example.com/pls/sales/leads/  (The context path is wrong.)
https://www.example.com/ords/sales/forecasting.report?month=jan  (The host name
is wrong.)
```

Support for Oracle RAC Fast Connection Failover

Oracle REST Data Services support the Fast Connection Failover (FCF) feature of Oracle Real Application Clusters (Oracle RAC).

Oracle REST Data Services runs with the Universal Connection Pool (UCP) in all the Application Server environments that it supports, such as WebLogic, Tomcat. UCP in turn supports Fast Connection Failover . To enable FCF, Oracle Notification Service (ONS) must to be enabled. To enable ONS, add entries to the list of properties in the

Oracle REST Data Services `defaults.xml` configuration file as shown in the following code snippet:

```
<entry key="jdbc.enableONS">true</entry>
<entry
key= "jdbc.ONSConfig">nodes=racnode1:4200,racnode2:4200\nwalletfile=/
oracle11/onswalletfile</entry>
```

ONS is the messaging facility used to send the Fast Application Notification (FAN) events. When ONS is enabled, Oracle REST Data Services automatically enables FCF. To Enable specific FCF capabilities such as fail over or other advanced FCF capabilities such as load balancing, you need to add entries in the configuration file for the custom connection as shown in the following code snippet:

```
<entry key="db.connectionType">customurl</entry>
<entry key="db.customURL">jdbc:oracle:thin:@(DESCRIPTION=(FAILOVER=ON)
(ADDRESS_LIST=
        (LOAD_BALANCE=ON)(ADDRESS=(PROTOCOL=TCP)
(HOST=prod_scan.example.com)(PORT=1521)))
        (CONNECT_DATA=(SERVICE_NAME=ISPRD)))|</entry>
```

After updating the `defaults.xml` configuration file, Oracle REST Data Services need to be restarted for the changes to take effect.

UCP supports Fast Connection Failover. FCF listens and responds to FAN events to deal with the following two scenarios:

- **Unplanned outages**: When RAC detects an instance failure, it generates a FAN Down event which FCF picks up. FCF then terminates all connections to the failed instance and directs all future requests to the surviving RAC instances.
- **Planned outages**: For instance, when a Database Administrator (DBA) wants to gracefully shut down a RAC instance for performing some maintenance activity. The instance shutdown generates a FAN Planned Down event which FCF picks up. FCF then directs all new requests to other RAC instances and **drains** or allows currently active transactions to complete.

> **Note:**
>
> Long running transactions may need to be terminated forcefully.

Configuring Security, Caching, Pre- and Post Processing, Environment, and Excel Settings

To configure security, caching, pre- and post- processing, environment, and Excel settings, see Using SQL Developer Oracle REST Data Services Administration (Optional).

Configuring REST-Enabled SQL Service Settings

This section explains how to configure the REST- Enabled SQL service.

> **Note:**
>
> Enabling the REST- Enabled SQL service enables authentication against the Oracle REST Data Service enabled database schemas. This makes the database schemas accessible over HTTPS, using the database password. Oracle highly recommends that you provide strong secure database passwords

REST- Enabled SQL service is a feature of Oracle REST Data Service. By default, the REST Enabled SQL service is turned off. To enable the REST- Enabled SQL service and the REST- Enabled SQL Export service, perform the following steps:

1. Locate the folder where the Oracle REST Data Services configuration file is stored.
2. Open the `defaults.xml` file and add: `<entry key="restEnabledSql.active">true</entry>`.
3. Save the file.
4. Restart Oracle REST Data Services.

Configuring the Maximum Number of Rows Returned from a Query

To configure maximum number of rows returned from a query, perform the following steps:

1. Locate the folder where the Oracle REST Data Services configuration file is stored.
2. Open the `defaults.xml` file and update the value of the `misc.pagination.maxRows` parameter:`<entry key=”misc.pagination.maxRows”>1500</entry>`

   > **Note:**
   >
   > The default value for `misc.pagination.maxRows` is 500.

3. Save the file.
4. Restart Oracle REST Data Services.

Configuring the Custom Error Pages

This section explains how to configure a custom error page instead of the error page generated by Oracle REST Data Services.

To configure a custom error page, perform the following steps:

1. Locate the folder where the Oracle REST Data Services configuration file is stored.
2. Open the `defaults.xml` file and update the value of the `error.externalPath` parameter:

   ```
   <entry key="error.externalPath">/path/to/error/pages/folder/</entry>
   ```

   Where:

   - `/path/to/error/pages/folder` is the path to a folder containing files that define the error pages. The files are stored in `{status}.html` format.

     Where, `{status}` is the HTTP status code for which you want to create a custom error page.
3. Save the file.
4. Restart Oracle REST Data Services.

Example 2-1 Configuring custom error page for "HTTP 404" status code

To configure a custom error page for the "HTTP 404 – Not Found" status, perform the following steps:

1. Create a file named `404.html`.
2. Save it under `/usr/local/share/ords/error-pages/` folder.
3. Configure the `error.externalPath` parameter to point to `/usr/local/share/ords/errro-pages/` folder.
4. Save the file.
5. Restart Oracle REST Data Services.

Developing RESTful Services for Use with Oracle REST Data Services

For more information on how to develop RESTful Services for use with Oracle REST Data Services, see Developing Oracle REST Data Services Applications.

Managing ORDS Administrator Privilege

Access to the `ORDS_ADMIN` PL/SQL package is provisioned through the `ORDS_ADMINISTRATOR_ROLE`. This role can be provisioned through the `ORDS_ADMIN` package to create additional ORDS administrators.

Provisioning ORDS_ADMINISTRATOR_ROLE to a User

This section describes how to provision `ORDS_ADMINISTRATOR_ROLE` role to a user.

You can provision `ORDS_ADMINISTRATOR_ROLE` role to a user by using either the database `GRANT` command or through the `ORDS_ADMIN.PROVISION_ADMIN_ROLE` PL/SQL method (as an ORDS Administrator).

Example 2-2 Using Grant command

```
GRANT ORDS_ADMINISTRATOR_ROLE TO HR_ADMIN;
```

Example 2-3 Using ORDS_ADMIN package method

```
BEGIN
  ORDS_ADMIN.PROVISION_ADMIN_ROLE(
    p_user => 'HR_ADMIN');
END;
/
```

Unprovisioning ORDS_ADMINISTRATOR_ROLE from a User

This section describes how to unprovision `ORDS_ADMINISTRATOR_ROLE` from a user.

As an ORDS administrator, you can unprovision `ORDS_ADMINISTRATOR_ROLE` from a user by either using the database `REVOKE` command or through the `ORDS_ADMIN.UNPROVISION_ROLES` PL/SQL method.

Example 2-4 Using REVOKE command

```
REVOKE ORDS_ADMINSTRATOR_ROLE FROM HR_ADMIN;
```

Example 2-5 Using ORDS_ADMIN package method

```
BEGIN
  ORDS_ADMIN.UNPROVISION_ROLES(
    p_user => 'HR_ADMIN',
    p_administrator_role => TRUE);
 END;
 /
```

Managing ORDS Runtime Privilege

The `ORDS_RUNTIME_ROLE` database role allows a user to act as a runtime user. A runtime user can manage and configure the runtime connection resources required by an ORDS service instance. The `ORDS_PUBLIC_USER` is one such database user. When additional runtime users are provisioned, it is possible to configure discrete ORDS service instances with different destination addresses and connection pools but hosted on the same Oracle database container.

It is recommended not to re-use a runtime user for any other purpose as it accumulates the grants necessary to proxy to other users. A runtime user only requires the `CREATE SESSION` privilege in addition to the `ORDS_RUNTIME_ROLE` role.

Provisioning ORDS_RUNTIME_ROLE to a User

This section describes how to provision `ORDS_RUNTIME_ROLE` role to a user.

As an ORDS administrator, you can provision `ORDS_RUNTIME_ROLE` role to a user by using either the database `GRANT` command or through the `ORDS_ADMIN.PROVISION_ADMIN_ROLE` PL/SQL method.

Example 2-6 Using Grant command

```
GRANT ORDS_RUNTIME_ROLE TO ORDS_PUBLIC_USER_2;
```

Example 2-7 Using ORDS_ADMIN package method

```
BEGIN
  ORDS_ADMIN.PROVISION_RUNTIME_ROLE(
    p_user => 'ORDS_PUBLIC_USER_2');
END;
/
```

Unprovisioning ORDS_RUNTIME_ROLE from a User

This section describes how to unprovision the `ORDS_RUNTIME_ROLE` role from a user

As an administrator, you can unprovision the `ORDS_RUNTIME_ROLE` from a user, by either using the database `REVOKE` command or through the `ORDS_ADMIN.UNPROVISION_ROLES` PL/SQL method.

Example 2-8 Using REVOKE command

```
REVOKE ORDS_RUNTIME_ROLE FROM ORDS_RUNTIME_USER_2;
```

Example 2-9 Using ORDS_ADMIN package method

```
BEGIN
  ORDS_ADMIN.UNPROVISION_ROLES(
    p_user => 'ORDS_RUNTIME_USER_2',
    p_runtime_role => TRUE);
 END;
 /
```

3

Installing and Configuring Customer Managed ORDS on Autonomous Database

This section explains how to install and configure Customer Managed Oracle REST Data Services (ORDS) on Autonomous Database.

Topics:

- About Customer Managed Oracle REST Data Services on Autonomous Database
- Downloading Wallet and Verifying Connection to Autonomous Database
- Creating Customer Managed Oracle REST Data Services User Role
- Downloading and Configuring Oracle REST Data Services
- Preparing and Starting ORDS

About Customer Managed Oracle REST Data Services on Autonomous Database

When you provision an Autonomous Database instance, by default Oracle REST Data Services (ORDS) is preconfigured and available for the instance. With the default ORDS, Oracle performs any required configuration, patching, and maintenance. Additionally, you can also configure Autonomous Database to use ORDS running in a customer managed environment.

When you use the default ORDS on Autonomous Database, you cannot modify any of the ORDS configuration options. For example, with the default configuration, the JDBC connection pools have a maximum of 100 connections and the connections for ORDS are preconfigured to use the `LOW` database service. Use a customer managed environment if you want manual control of the configuration and management of Oracle REST Data Services. For example, use this option when your applications require larger connection pools or if you need more control over the ORDS configuration options.

When ORDS runs in a customer managed environment, you are responsible for configuration, patching, and maintenance of ORDS in the customer managed environment. After you configure Autonomous Database to use your customer managed ORDS in addition to the existing autonomously managed ORDS, you can route ORDS HTTPS traffic through your environment. The default Autonomous Database web server and ORDS are still running and ORDS traffic goes to the ORDS running in the customer managed environment. This provides an additional and alternative HTTPS solution for Autonomous Database.

Installing and configuring a customer managed environment for ORDS allows you to run ORDS with configuration options that are not possible using the default Oracle managed ORDS available with Autonomous Database.

Installing and configuring a customer managed environment for ORDS is only supported with Autonomous Database on Shared Exadata Infrastructure.

> **Note:**
>
> - Oracle REST Data Services 19.4.6 or higher is required to use a customer managed environment for ORDS with Autonomous Database.
> - Installing and configuring a customer managed environment for ORDS is only supported with Autonomous Database on Shared Exadata Infrastructure.

Downloading Wallet and Verifying Connection to Autonomous Database

You need to configure ORDS to connect to the Autonomous Database. With Oracle REST Data Services (ORDS) running in a customer managed environment, you need to obtain the Autonomous Database wallet on the system that runs the customer managed ORDS. Perform the following steps to download the wallet and verify the connection to the Autonomous Database:

1. Download the wallet for the Autonomous Database instance. Alternatively you can use the OCI CLI to generate the wallet. See generate-wallet for information on using the CLI.
2. Convert the `wallet.zip` from the previous step to text. The `base64` command encodes binary strings into text representations using the base64 encoding format. For example, run the following command:

   ```
   base64 -w 0 wallet_NAME.zip > wallet_NAME.zip.b64
   ```

   This step prepares the wallet as a text file as required for use in the ORDS configuration (instructions for this follow).
3. Verify that you can connect from the customer managed environment where you are installing and configuring ORDS to your Autonomous Database. For example, using `SQLcl` and the wallet you download in Step 1, verify the connection as follows:
   a. Connect with SQLcl.

> **See Also:**
>
> - Connect with Oracle SQLcl Cloud Connection for Autonomous Data Warehouse environment.
> - Connect with Oracle SQLcl Cloud Connection for Autonomous Transaction Processing environment.

b. View the database services and connect to your Autonomous Database from the customer managed environment.

```
SQL> show tns
TNS_ADMIN set to: /var/folders/4r/path/T/oracle_cloud_config_path

Available TNS Entries
---------------------
dbname_high
dbname_low
dbname_medium


SQL> conn admin@dbname_low
Password? (**********?) *****************
Connected.
SQL>
```

Creating Customer Managed Oracle REST Data Services User Role

To use Autonomous Database with Oracle REST Data Services (ORDS) running in a customer managed environment on your Autonomous Database, you must create a user, grant privileges to the user, and run the procedure `ORDS_ADMIN.PROVISION_RUNTIME_ROLE`.

Perform the following steps to create a user for the ORDS JDBC Connection Pool and prepare the Autonomous Database instance for using Oracle REST Data Services in a customer managed environment:

1. Connect to your Autonomous Database as the ADMIN user.
2. Create a new ORDS user and grant the required privileges to the new user as follows:

   ```
   CREATE USER "ORDS_PUBLIC_USER2" IDENTIFIED BY "password";
   GRANT "CONNECT" TO "ORDS_PUBLIC_USER2";
   ```

   The new user name `ORDS_PUBLIC_USER2` is the recommended user name. This name is not required and you can choose a different user name. If you choose a different user name, then all the corresponding user names in these steps need to use the name you choose, rather than `ORDS_PUBLIC_USER2`.
3. Create a database user with the ORDS Developer role, so that it can act as an ORDS Runtime user. Additional changes to the ORDS configuration are required to use a user other than ORDS_PUBLIC_USER. As the ADMIN user, run the following procedure:

   ```
   BEGIN
        ORDS_ADMIN.PROVISION_RUNTIME_ROLE(
            p_user => 'ORDS_PUBLIC_USER2',
            p_proxy_enabled_schemas => TRUE);
   ```

```
END;
/
```

Following are the parameters:

- `p_user`: The name of the user to be configured.
- `p_proxy_enabled_schemas`: When set to `true`, proxy grants are added for any REST enabled schemas.

Downloading and Configuring Oracle REST Data Services

To use Autonomous Database with Oracle REST Data Services (ORDS) running in a customer managed environment you need to install ORDS.

> **Note:**
>
> Oracle REST Data Services 19.4.6 or higher is required for a customer managed environment with Autonomous Database.

Depending on where you install Oracle REST Data Services for your customer managed environment, do the following:

- If your customer managed environment for Oracle REST Data Services runs in Oracle Cloud Infrastructure, then use an Oracle YUM repository and perform a YUM install of ORDS.
- If your customer managed environment for Oracle REST Data Services runs in some other environment, then download ORDS from the Oracle REST Data Services Download page. See Introduction to Oracle REST Data Services for more information.

1. In the location where ORDS is installed, create an ORDS configuration folder (this creates a folder and sets up the ORDS configuration environment and settings). For example:

   ```
   java -jar ords.war configdir ./ORDSConfig_2
   ```

   On Oracle Cloud Infrastructure with Linux with a YUM repository the ORDS configuration folder is: `/opt/oracle/ords/config`.

2. Edit the ORDS configuration file created in the preceding step `./ords/conf/apex_pu.xml` to add or update the following:

   ```
   <?xml version="1.0" encoding="UTF-8" standalone="no"?>
   <!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/
   properties.dtd">
   <properties>
   <entry key="db.username">ORDS_PUBLIC_USER2</entry>
   <entry key="db.password">!password</entry>
   <entry key="db.wallet.zip.service">dbname_low</entry>
   <entry key="db.wallet.zip"><!
   ```

```
[CDATA[contents_of_wallet_NAME.zip.b64]]></entry>
</properties>
```

Notes:

- The extra "`!`" in front of the password causes the password to be encrypted the next time the ORDS service starts.
- The `db.wallet.zip` entry (*contents_of_wallet_NAME.zip.b64*) includes the contents of the `base64` encoded wallet archive that you prepared previously. Ensure that this text is enclosed in a CDATA block (this is a long string of characters). See Downloading Wallet and Verifying Connection to Autonomous Database for details on how to create the `wallet_NAME.zip.b64` file.
- The `db.username`, specified as *ORDS_PUBLIC_USER2* is the database username you previously defined. See Creating Customer Managed Oracle REST Data Services User Role for more information.

3. Edit `./ords/defaults.xml` as required for your ORDS installation. See Understanding Configurable Parameters for more information.
4. Ensure the plsql gateway is enabled for APEX support:

```
<entry key="plsql.gateway.enabled">true</entry>
```

If you want use ORDS with APEX, then you need to enable the PL/SQL gateway in ORDS.

Preparing and Starting ORDS

To use customer managed Oracle REST Data Services (ORDS) on Autonomous Database, on the system where ORDS runs you need to perform additional configuration steps and then start ORDS.

1. **Note:**

   For ORDS running with APEX, perform all the steps starting from **Step 1**. For ORDS running without APEX, perform the steps starting from **Step 3**.

1. In the location where ORDS is installed, install the APEX images.

```
unzip apex_19version.zip
```

Note:

To use your customer managed ORDS environment for APEX, you need to download the APEX images of the APEX version that is currently pre-deployed in Oracle Autonomous Database. You can download the APEX images from Download APEX Images.

2. Edit the standalone properties to add or edit the static images property:

   ```
   standalone.static.path=/path/to/apex/images
   ```

3. Create a wallet_cache folder, so that ORDS stores the Autonomous Database wallet on this folder and uses it while connecting to JDBC. For example:

   ```
   mkdir wallet_cache
   ```

4. Start ORDS.

   - If your customer managed environment for Oracle REST Data Services runs in Oracle Cloud Infrastructure, then start the ORDS service as follows:

     ```
     /opt/oracle/ords start
     ```

   - If your customer managed environment for Oracle REST Data Services runs in the directory where ORDS is installed, then start the ORDS service as follows:

     ```
     java -Duser.timezone=UTC -Ddb.wallet.cache=wallet_cache -jar
     ords.war standalone --apex-images images --port 8088
     ```

4

Using the Multitenant Architecture with Oracle REST Data Services

This section outlines installing, configuring, upgrading and uninstalling Oracle REST Data Services in a multitenant container database.

- Setting Up ORDS in a CDB Environment
- Setting Up ORDS in an Application Container

Setting Up ORDS in a CDB Environment

This section describes how to setup Oracle REST Data Services (ORDS) into a multitenant container database (CDB) environment.

Oracle Database 12*c* Release 1 (12.1) introduced the multitenant architecture. This database architecture has a multitenant container database (CDB) that includes a root container, `CDB$ROOT`, a seed database, `PDB$SEED`, and multiple pluggable databases (PDBs). A PDB appears to users and applications as if it were a non-CDB. Each PDB is equivalent to a separate database instance in Oracle Database Release 11*g*.

The root container, `CDB$ROOT`, holds common objects that are accessible to every PDB utilizing metadata links or object links. The seed database, `PDB$SEED`, is used when you create a new PDB to seed the new pluggable database. The key benefit of the Oracle Database 12*c* multitenant architecture is that the database resources, such as CPU and memory, can be shared across all of the PDBs. This architecture also enables many databases to be treated as one for tasks such as upgrades or patches, and backups.

The installation process when you have multiple releases is described in the following section:

- Installation Enabling Multiple Releases

> **Note:**
>
> If you want to install directly into a PDB (not connected to Root during installation), see Advanced Installation Using Command-Line Prompts for more information.

Preinstallation Tasks for Oracle REST Data Services CDB Installation

- Ensure that the PDBs are open (not mounted/closed) in read/write mode (except for `PDB$SEED`, which remains in read-only mode). For more information, see Oracle Multitenant Administrator's Guide
- Ensure that the default and temporary tablespaces to be used by the `ORDS_METADATA` schema and the `ORDS_PUBLIC_USER` user exist and that you know

the tablespace names. The installation procedure creates those users, but it does not create the tablespaces.

> **Note:**
>
> ORDS_METADATA and ORDS_PUBLIC_USER are installed in the seed container, and the default and temporary tablespaces exist in PDB$SEED. If these tablespaces do not already exist, then you must create the tablespaces in PDB$SEED. For more information, see Oracle Multitenant Administrator's Guide

Installation Enabling Multiple Releases

This section describes the installation process when you have multiple releases of Oracle REST Data Services and patch sets in the PDBs in a multitenant environment.

When Oracle REST Data Services is installed into a CDB, the proxy user, Oracle REST Data Services public user (ORDS_PUBLIC_USER) is installed in the root container and is a common user. The ORDS_METADATA schema is a local user that contains the metadata for Oracle REST Data Services. Both the ORDS_METADATA schema and the ORDS_PUBLIC_USER are installed in the seed container (PDB$SEED) and all of the pluggable databases.

Since the ORDS_METADATA is installed as a local user, this provides you the flexibility of installing multiple Oracle REST Data Services releases in the pluggable databases.

Command Line Installation

You must provide the SYS AS SYSDBA credentials in the Root (CDB$ROOT) container to perform the installation.

Advanced Installation

This section describes the advanced installation prompts for installing Oracle REST Data Services into a CDB to enable multiple Oracle REST Data Services releases.

To install Oracle REST Data Services into a CDB to enable multiple Oracle REST Data Services releases, perform the following steps:

1. Navigate to the folder where you unzipped the Oracle REST Data Services installation kit.
2. Enter the following command:

   ```
   java -jar ords.war install advanced
   ```

3. When prompted, enter the database connection information for your CDB.

   ```
   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]:
   Enter the database service name:(for example, cdb.example.com)
   ```

4. Verify the Oracle REST Data Services installation.

```
Enter 1 if you want to verify/install Oracle REST Data Services
schema or 2 to skip this step [1]:
```

5. Accept or enter 1 (the default) to install Oracle REST Data Services into the CDB and all of its PDBs.

```
Enter the database password for ORDS_PUBLIC_USER:
Confirm password:
Requires to login with administrator privileges to verify Oracle
REST Data Services schema.

Enter the administrator username: SYS
Enter the database password for SYS AS SYSDBA:
Confirm password:


Retrieving information....
Your database connection is to a CDB.  ORDS common user
ORDS_PUBLIC_USER will be
created in the CDB.  ORDS schema will be installed in the PDBs.
Root CDB$ROOT - create ORDS common user
PDB PDB$SEED - install ORDS 18.2.0.<JulianDay.Time> (mode is READ
ONLY, open for
READ/WRITE)
PDB PDBName1 - install ORDS 18.2.0.<JulianDay.Time>
PDB PDBName2 - install ORDS 18.2.0.<JulianDay.Time>


Enter 1 if you want to install ORDS or 2 to skip this step [1]:
```

6. Press enter to continue with the installation.
7. When prompted, enter additional information as needed. See Advanced Installation Using Command-Line Prompts for more information.

> **Note:**
>
> To use the pluggable mapping feature, see Making All PDBs Addressable by Oracle REST Data Services (Pluggable Mapping) for more information.

Silent Installation

Silent installation reads the properties from the Oracle REST Data Services parameter file.

To perform a silent installation, enter the following command:

```
java –jar ords.war install simple
java –jar ords.war
```

Related Topics

- Advanced Installation Using Command-Line Prompts

Upgrading Oracle REST Data Services in a CDB Environment

When you use a new release of Oracle REST Data Services, upgrading its schema in the CDB and its pluggable databases (PDBs) will occur automatically when you perform a simple or advanced installation.

For example:

```
java -jar ords.war
```

If Oracle REST Data Services is already installed or upgraded, a message displays the Oracle REST Data Services schema version, and you will not be prompted for information.

Migrating Oracle REST Data Services in the CDB to Enable Multiple Releases

This section describes how to migrate Oracle REST Data Services in the CDB to enable multiple releases.

Starting with release 18.2.0 and later, if you have an Oracle REST Data Services schema and `ORDS_METADATA` that is installed in the `CDB$ROOT` container, then during upgrade it will migrate the common `ORDS_METADATA` schema to your PDBs as a local schema. Oracle database 12.1.0.2 and later releases support this change.

Making All PDBs Addressable by Oracle REST Data Services

This section describes how to make all application PDBs in a CDB addressable by ORDS. This step is required for starting ORDS from the CDB.

For more information refer to Making All PDBs Addressable by Oracle REST Data Services (Pluggable Mapping)

Uninstalling Oracle REST Data Services in a CDB Environment

To uninstall Oracle REST Data Services from a CDB, use the `uninstall` command.

For example:

```
java -jar ords.war uninstall
```

Oracle REST Data Services will be removed from the CDB and its pluggable databases (PDBs).

Related Topics

- If You Want to Reinstall or Uninstall (Remove) Oracle REST Data Services

Setting Up ORDS in an Application Container

This section describes how to setup Oracle REST Data Services in an application container.

Starting with ORDS release 20.2.1, Oracle REST Data Services can be installed or upgraded into an application container using the ORDS SQL scripts provided in the `ords.version.number.zip` file.

An application container consists of an application root where the application is defined and one or more PDBs that share the metadata about the application from the application root. You can have multiple application containers within a CDB and each container can have different versions of Oracle REST Data Services. Installing or upgrading Oracle REST Data Services in an application container is done against the application root container. When an application PDB wants to use the upgraded version, it must synchronize with the application root. Oracle REST Data Services continues to run in the application PDB with the existing version until the application PDB synchronizes with the application root.

Topics:

- Prerequisites for Creating ORDS in an Application Container
- Installing ORDS in the Application Root Container
- Creating an Application Seed
- Creating an Application PDB from the Application Seed
- ORDS Configuration Files Setup
- Running ORDS
- Validating ORDS in the Application Root Container
- Upgrading ORDS in the Application Container
- Uninstalling ORDS from the Application Container
- Verifying ORDS in the Application Container

Prerequisites for Creating ORDS in an Application Container

This section describes the prerequisites for installing ORDS in an application container.

Following prerequisites must be met before you install ORDS in an application container:

- Download ORDS version 20.2.1 or later from Oracle REST Data Services Downloads.
- Extract the ORDS SQL scripts.
- To obtain the ORDS SQL scripts, execute the following commands:

```
unzip ords.version.number.zip ords.war
unzip ords.war 'WEB-INF/lib/ords-installer-*.jar'
unzip  'WEB-INF/lib/ords-installer-*.jar' 'db/*'
mv db scripts
```

The ORDS SQL scripts are located in the scripts folder. The scripts folder contains the subdirectories for the install, upgrade, validate, and uninstall SQL scripts. You can run these SQL scripts using SQLcl, SQL*Plus, or SQL Developer.

Creating an Application Root Container

This section describes how to create an application root container.

To create an application root container:

1. Ensure that the current container is in `CDB$ROOT`.
2. Use the AS APPLICATION CONTAINER clause of the `CREATE PLUGGABLE DATABASE` statement to create an application container.
3. Open the application container.

Example:

```
CREATE PLUGGABLE DATABASE ords_app_root1 AS APPLICATION CONTAINER ADMIN
USER admin IDENTIFIED BY <admin_password>

FILE_NAME_CONVERT=('pdbseed', 'ords_app_root1');

ALTER PLUGGABLE DATABASE ords_app_root1 OPEN;
```

> **Note:**
>
> `ords_app_root1` and the `admin` user in the preceding example can be any valid Oracle identifier.
> If Oracle managed files is enabled in the CDB or the `PDB_FILE_NAME_CONVERT` initialization parameter is set, then omit the `FILE_NAME_CONVERT` clause.
>
> The ORDS users, namely `ORDS_PUBLIC_USER` and `ORDS_METADATA`, must not exist in the seed (for example, `pdbseed`) or cloned pdb.

> **See Also:**
>
> Creating an Application Container

Installing ORDS in the Application Root Container

This section describes how to install ORDS in the application root container.

To install ORDS in the application root container, perform the following steps:

1. Connect to the application root container.
2. Run `/path/to/scripts/install/core/ords_app_con_install.sql` command using the following parameters:
   - Log folder (must include the forward slash at the end)
   - Default tablespace for ORDS schema
   - Temporary tablespace for ORDS schema
   - Default tablespace for ORDS proxy user

- Temporary tablespace for ORDS proxy user
- ORDS proxy user password
- Scripts path (requires the fully qualified path to the ORDS scripts)

> **Note:**
>
> The tablespaces must already exist in the database.

```
ALTER SESSION SET CONTAINER = ords_app_root1;

@/path/to/scripts/install/core/ords_app_con_install.sql /path/to/logs/
SYSAUX TEMP SYSAUX TEMP P033w0r6! /path/to/scripts
```

Where:

The `ords_app_con_install.sql` creates an application named ORDS and assigns the application version to the ORDS product version. The product version format is Year.Quarter.Patch.rJulianDay24HRMM (for example, 20.2.1.r2121800).

The preceding script installs ORDS and creates the following:

- The ORDS schema, `ORDS_METADATA`
- The ORDS proxy user, `ORDS_PUBLIC_USER` and
- The related database objects in the application container

> **See Also:**
>
> Verifying ORDS in the Application Container

Creating an Application Seed

This section describes how to create an application seed.

An application seed is used to provision application PDBs with the application root's applications pre-installed.

To create an application seed:

1. Ensure that the current container is in the `CDB$ROOT`.
2. Alter session and set container to the application root.
3. Use the AS SEED clause of the `CREATE PLUGGABLE DATABASE` statement to create an application seed.
4. Sync the ORDS application with the application seed.
5. Compile invalid objects.
6. Open the application seed in a read only mode.

> **Note:**
>
> `ords_app_root1` and the `admin` user in the following example can be any valid Oracle identifier.
> If Oracle managed files is enabled in the CDB or the `PDB_FILE_NAME_CONVERT` initialization parameter is set, then omit the `FILE_NAME_CONVERT` clause.

```
ALTER SESSION SET CONTAINER = ords_app_root1;
CREATE PLUGGABLE DATABASE AS SEED ADMIN USER admin IDENTIFIED BY
<admin_password>
FILE_NAME_CONVERT=('pdbseed', 'ords_app_root1_seed');
ALTER PLUGGABLE DATABASE ords_app_root1$seed open;
ALTER SESSION SET CONTAINER = ords_app_root1$seed;
ALTER PLUGGABLE DATABASE application ORDS sync;
begin
  sys.dbms_utility.compile_schema('ORDS_METADATA', FALSE);
end;
/
ALTER PLUGGABLE DATABASE ords_app_root1$seed close immediate;
ALTER PLUGGABLE DATABASE ords_app_root1$seed open read only;
```

> **See Also:**
>
> Creating an Application Container

Creating an Application PDB from the Application Seed

This section describes how to create an application PDB that is seeded from the application seed

An application PDB is created by issuing the `CREATE PLUGGABLE DATABASE` statement from the application root.

To create an application PDB from the application seed:

1. Ensure that the current container is in `CDB$ROOT`.
2. Alter session and set the container to the application root.
3. Use the `CREATE PLUGGABLE DATABASE` command to create a PDB from the application seed.

> **Note:**
>
> `ords_app_pdb1` and the `admin` user in the following example can be any valid Oracle identifier.
> If Oracle managed files is enabled in the CDB or the `PDB_FILE_NAME_CONVERT` initialization parameter is set, then omit the `FILE_NAME_CONVERT` clause.

```
ALTER SESSION SET CONTAINER=ords_app_root1;
CREATE PLUGGABLE DATABASE ords_app_pdb1 ADMIN USER admin IDENTIFIED BY
<admin password>
FILE_NAME_CONVERT=('ords_app_root1_seed','ords_app_pdb1');
ALTER PLUGGABLE DATABASE ords_app_pdb1 OPEN;
ALTER SESSION SET CONTAINER = ords_app_pdb1;
select app_name, app_version, app_status from dba_applications where
app_name = 'ORDS';
APP_NAME     APP_VERSION         APP_STATUS
--------     ---------------     ------------
ORDS         20.2.1.r2121800     NORMAL
```

> **See Also:**
>
> Creating an Application Container

ORDS Configuration Files Setup

This section describes how to setup the ORDS configuration files:

Topics:

- Specifying the ORDS Configuration Folder
- Creating the ORDS Configuration Files for the Application Root Container
- Making all Application PDBs in an Application Root Container Addressable by ORDS

Specifying the ORDS Configuration Folder

This section describes how to specify the ORDS configuration folder.

The configuration folder contains the ORDS configuration files. If the configuration folder is undefined, then you are prompted for the configuration folder when you execute the setup command.

To specify the location for your ORDS configuration files, use the following command:

```
java -jar ords.war configdir /path/to/config
```

Creating the ORDS Configuration Files for the Application Root Container

This section describes how to create the ORDS configuration files for the application root container.

To create the ORDS configuration files for the application root container, execute the following setup command to create the configuration files:

```
java -jar ords.war setup --configOnly
```

Where, the `--configOnly` option must be specified to create the configuration files. When prompted for the service name, specify the application root servicename.

Example 4-1 Configuring ORDS for Application Express

```
 java -jar ords.war setup --configOnly

    Specify the database connection type to use.
    Enter number for [1] Basic  [2] TNS  [3] Custom URL [1]:
    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]:
    Enter the database service name: ords_app_root1
    Enter the database password for ORDS_PUBLIC_USER:
    Confirm password:
    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]:
    Enter the PL/SQL Gateway database user name [APEX_PUBLIC_USER]:
    Enter the database password for APEX_PUBLIC_USER:
    Confirm password:
    Enter 1 to specify passwords for Application Express RESTful
Services database users (APEX_LISTENER, APEX_REST_PUBLIC_USER) or 2 to
skip this step [1]:
    Enter the database password for APEX_LISTENER:
    Confirm password:
    Enter the database password for APEX_REST_PUBLIC_USER:
    Confirm password:
    Enter a number to select a feature to enable:
      [1] SQL Developer Web  (Enables all features)
      [2] REST Enabled SQL
      [3] Database API
      [4] REST Enabled SQL and Database API
      [5] None
    Choose [1]:
```

Example 4-2 Configuring ORDS only

```
java -jar ords.war setup --configOnly

    Specify the database connection type to use.
    Enter number for [1] Basic  [2] TNS  [3] Custom URL [1]:
```

```
    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]:
    Enter the database service name: ords_app_root1
    Enter the database password for ORDS_PUBLIC_USER:
    Confirm password:
    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
    Enter a number to select a feature to enable:
      [1] SQL Developer Web  (Enables all features)
      [2] REST Enabled SQL
      [3] Database API
      [4] REST Enabled SQL and Database API
      [5] None
    Choose [1]:
```

Making all Application PDBs in an Application Root Container Addressable by ORDS

This section describes how to make all application PDBs in an application root container addressable by ORDS. This step is required for starting ORDS from the application root container.

For more information refer to Making All PDBs Addressable by Oracle REST Data Services (Pluggable Mapping).

Running ORDS

This section lists the different methods you can use to run ORDS after installing ORDS in the application container.

Once you install ORDS in the application container and create the ORDS configuration files, run ORDS using one of the following methods:

- Standalone Mode
- Deploy on Oracle WebLogic Server
- Deploy Oracle REST Data Services on Apache Tomcat

> **See Also:**
>
> - Starting in Standalone Mode
> - Deploying to Oracle WebLogic Server
> - Deploying to Apache Tomcat

Validating ORDS in the Application Root Container

This section describes how to validate ORDS in the application root container.

You can validate an application in an application container. These operations are performed in the application root. The application container propagates the application changes to the application PDBs when the application PDBs synchronize with the application in the application root. The `ords_app_con_validate.sql` script repairs the Oracle REST Data Services schema and verifies if the ORDS schema is valid.

To repair ORDS in the application root:

1. In SQLcl or SQL*Plus, connect to the application root.
2. Run `/path/to/scripts/validate/core/ords_app_con_validate.sql` with the following parameters:
   - Log folder (must include the forward slash at the end)
   - Scripts path (requires the fully qualified path to the ORDS scripts)

```
ALTER SESSION SET CONTAINER = ords_app_root1;

@/path/to/scripts/validate/core/ords_app_con_validate.sql /path/to/
logs/ /path/to/scripts
```

The `ords_app_con_validate.sql` sets the application version to the ORDS product version with suffix "_v_YYMMDD24HRMISS".

For example:

Year.Quarter.Patch.rJulianDay24MI_v_YYMMDD24HRMISS

20.2.0.r1801800_v_200705160015

To synchronize the ORDS application in an application PDB with the latest changes in the application root:

1. In SQLcl or SQL*Plus, ensure that the current container is the application PDB
2. Run the `ALTER PLUGGABLE DATABASE APPLICATION` statement specifying the ORDS application with the SYNC clause.

For example:

```
ALTER SESSION SET CONTAINER = ords_app_pdb1;

ALTER PLUGGABLE DATABASE APPLICATION ORDS SYNC;
```

> **Note:**
>
> When you install ORDS, it attempts to find the Oracle Application Express (APEX) schema and creates a view. This view joins the relevant tables in the APEX schema to the tables in the Oracle REST Data Services schema. If you install Oracle REST Data Services before APEX, then Oracle REST Data Services cannot find the APEX schema and it creates a stub view in place of the missing APEX tables.
>
> Oracle highly recommends that you install Oracle REST Data Services after APEX to ensure that the APEX objects that Oracle REST Data Services needs to query are present.
>
> If you install Oracle REST Data Services before APEX, then use the `ords_app_con_validate.sql` script to force Oracle REST Data Services to reconstruct the queries against the APEX schema.

Upgrading ORDS in the Application Container

This section describes how to upgrade ORDS in the application container.

You can upgrade an application in an application container. These operations are performed in the application root. The application container propagates the application changes to the application PDBs when the application PDBs synchronize with the application in the application root.

Prerequisites:

- ORDS must already be installed in the application container.
- Upgrading ORDS from an earlier release to a new release (for example, ORDS release 20.2.x.x to 20.3.x.x).

To upgrade ORDS in the application root:

1. In SQLcl or SQL*Plus, connect to the application root.
2. Run `/path/to/scripts/upgrade/ords_app_con_upgrade.sql` with the following parameters:
   - Log folder (must include the forward slash at the end)
   - Scripts path (requires the fully qualified path to the ORDS scripts)

> **Note:**
>
> The `ords_app_con_upgrade.sql` script upgrades ORDS in the application root container to the release that you are using. For example, if the ORDS application version is 20.2.1.r2121800, and the ORDS upgrade script is 20.3.0.r2601900, then the script upgrades ORDS to release 20.3.0.r2601900 in the application root container.

To synchronize the ORDS application in an application PDB with the upgrade changes in the application root:

1. In SQLcl or SQL*Plus, ensure that the current container is the application PDB.
2. Run the `ALTER PLUGGABLE DATABASE APPLICATION` statement specifying the ORDS application with the SYNC clause.

```
ALTER SESSION SET CONTAINER = ords_app_pdb1;

ALTER PLUGGABLE DATABASE APPLICATION ORDS SYNC;
```

> **See Also:**
>
> Verifying ORDS in the Application Container

Uninstalling ORDS from the Application Container

This section describes how to uninstall ORDS from the application container.

You can uninstall an application from an application container. These operations are performed in the application root. The application container propagates the application changes to the application PDBs when the application PDBs synchronize with the application in the application root.

To uninstall ORDS from the application root:

1. In SQLcl or SQL*Plus, connect to the application root.
2. Run `/path/to/scripts/uninstall/core/ords_app_con_uninstall.sql` with the following parameters:
   - Log folder (must include the forward slash at the end)
   - Scripts path (requires the fully qualified path to the ORDS scripts)

```
ALTER SESSION SET CONTAINER = ords_app_root1;

@/path/to/scripts/uninstall/core/ords_app_con_uninstall.sql /path/to/
logs/ /path/to/scripts
```

To synchronize the application PDB to uninstall the ORDS application:

1. In SQLcl or SQL*Plus, ensure that the current container is the application PDB.
2. Run the `ALTER PLUGGABLE DATABASE APPLICATION` statement specifying the ORDS application with the SYNC clause.

For example:

```
ALTER SESSION SET CONTAINER = ords_app_pdb1;

ALTER PLUGGABLE DATABASE APPLICATION ORDS SYNC;
```

> **See Also:**
>
> Verifying ORDS in the Application Container

Verifying ORDS in the Application Container

This section describes how to verify ORDS in the application container.

To verify the ORDS for install, upgrade, validate, and uninstall in the application container:

- Manually inspect the following log files for any errors:
  - `Install - ordsinstall_<timestamp>.log`
  - `Upgrade - ordsupgrade_<timestamp>.log`
  - `Validate - ordsvalidate_<timestamp>.log`
  - `Uninstall - ordsuninstall_<timestamp>.log`
- Query `dba_applications` to verify if the ORDS application exists and its application version is the same as the ORDS product version.

  ```
  SQL> select app_name, app_version, app_status from dba_applications
  where app_name = 'ORDS';

      APP_NAME     APP_VERSION          APP_STATUS
      --------     ---------------      ------------
      ORDS         20.2.1.r2121800      NORMAL
  ```

- Query `dba_app_errors` to check for any errors:
  ```
  SQL> select app_name, app_statement, errornum, errormsg from
  dba_app_errors where app_name = 'ORDS';

  no rows selected
  ```

  If you are uninstalling ORDS from the application container, the `APP_STATUS` contains the value `UNINSTALLED`.

Making All PDBs Addressable by Oracle REST Data Services (Pluggable Mapping)

Pluggable mapping refers to the ability to make all PDBs in a CDB root or in an application root container addressable by Oracle REST Data Services. To use this feature, follow the instructions described in this topic.

If the Oracle REST Data Services configuration file includes the `db.serviceNameSuffix` parameter, this indicates that the Oracle REST Data Services pool points to a CDB root or an application root, and that the PDBs connected to that CDB root or an application root should be made addressable by Oracle REST Data Services.

The value of the `db.serviceNameSuffix` parameter must match the value of the `DB_DOMAIN` database initialization parameter, and it must start with a period (.). To set the value of the `db.serviceNameSuffix` parameter:

1. In SQLcl or SQL*Plus, connect to the root as a user with SYSDBA privileges.
2. Check the value of the `DB_DOMAIN` database initialization parameter.

```
SQL> show parameter DB_DOMAIN
```

3. Exit SQLcl or SQL*Plus.

```
SQL> exit
```

4. If the `DB_DOMAIN` value is not empty, then on the command line, enter the command to create the key and value for the `db.serviceNameSuffix` parameter and its `DB_DOMAIN`. This will be used to add this entry to the Oracle REST Data Services configuration file.

```
echo db.serviceNameSuffix=.value-of-DB_DOMAIN > snsuffix.properties
```

For example, if `DB_DOMAIN` is set to `example.com`, enter the following:

```
echo db.serviceNameSuffix=.example.com > snsuffix.properties
```

5. If the `db.serviceNameSuffix` parameter value is not defined, enter a command in the following format to add an entry to the configuration file:

```
java -jar ords.war set-properties --conf pool-name snsuffix.properties
```

Where *pool-name* is one of the following:

- `poolName` for a PL/SQL Gateway configuration
- `poolName_pu` for an Oracle REST Data Services RESTful Services configuration
- `poolName_rt` for an Application Express RESTful Services configuration

Example 1: You want to make PDBs in a CDB root or an application root addressable globally. Specify `defaults` by entering the following command:

```
java -jar ords.war set-properties --conf defaults snsuffix.properties
```

> **Note:**
>
> The approach shown in Example 1 (setting the property for all pools through the `defaults.xml` file) is best for most use cases.

Example 2: You want to make PDBs in a CDB root or an application root addressable for your PL/SQL Gateway, and your pool name is `apex`. Enter the following command:

```
java -jar ords.war set-properties --conf apex snsuffix.properties
```

For example, if the database pointed to by apex has a `DB_DOMAIN` value of `example.com` and contains the two PDBs `pdb1.example.com` and `pdb2.example.com`, the first PDB will be mapped to URLs whose path starts with `/ords/pdb1/`, and the second PDB will be mapped to URLs whose path starts with `/ords/pdb2/`.

Example 3: You want to make PDBs in a CDB root or an application root addressable for your Oracle REST Data Services RESTful Services, and your pool name is `apex_pu`. Enter the following command:

```
java -jar ords.war set-properties --conf apex_pu snsuffix.properties
```

Example 4: You want to make PDBs in a CDB root or an application root addressable for your Application Express RESTful Services and your pool name is `apex_rt`. Enter the following command:

```
java -jar ords.war set-properties --conf apex_rt snsuffix.properties
```

Related Topics

- About the Oracle REST Data Services Configuration Files

5

Developing Oracle REST Data Services Applications

This section explains how to develop applications that use Oracle REST Data Services. It includes guidance and examples.

> **Note:**
>
> If you want to get started quickly, you can try the tutorial in Oracle REST Data Services Quick Start Guide. However, you should then return to this chapter to understand the main concepts and techniques.

> **Note:**
>
> Ensure that you have installed and configured both Oracle Application Express 4.2 or later, and Oracle REST Data Services 3.0 or later, before attempting to follow any of the tutorials and examples.
>
> To use the Oracle REST API for JSON Data Persistence, you must first install the Oracle REST API. See "Oracle REST API Installation" in *Oracle REST Data Services SODA for REST Developer's Guide*.
>
> It is assumed that you are familiar with Oracle Application Express. If you are new to Oracle Application Express, see the Oracle Application Express documentation.

Topics:

- Introduction to Relevant Software
- Getting Started with RESTful Services
- Automatic Enabling of Schema Objects for REST Access (AutoREST)
- Filtering in Queries
- Configuring Secure Access to RESTful Services
- About Oracle REST Data Services User Roles
- Authenticating Against WebLogic Server User Repositories
- Integrating with Existing Group/Role Models
- Using the Oracle REST Data Services PL/SQL API

You may also want to review Creating an Image Gallery , a supplementary extended example that uses Oracle Application Express to build an application.

Introduction to Relevant Software

This section explains some key relevant software for developing applications that use Oracle REST Data Services.

Topics:

- About Oracle Application Express
- About RESTful Web Services

Related Topics

- About Oracle REST Data Services

About Oracle Application Express

Oracle Application Express is a declarative, rapid web application development tool for the Oracle database. It is a fully supported, no cost option available with all editions of the Oracle database. Using only a web browser, you can develop and deploy professional applications that are both fast and secure.

About RESTful Web Services

Representational State Transfer (REST) is a style of software architecture for distributed hypermedia systems such as the World Wide Web. An API is described as RESTful when it conforms to the tenets of REST. Although a full discussion of REST is outside the scope of this document, a RESTful API has the following characteristics:

- Data is modelled as a set of resources. Resources are identified by URIs.
- A small, uniform set of operations are used to manipulate resources (for example, PUT, POST, GET, DELETE).
- A resource can have multiple representations (for example, a blog might have an HTML representation and an RSS representation).
- Services are stateless and since it is likely that the client will want to access related resources, these should be identified in the representation returned, typically by providing hypertext links.

Release 4.2 of Oracle Application Express leverages the capabilities of Oracle REST Data Services to provide developers with an easy to use graphical user interface for defining and testing RESTful Web Services.

Getting Started with RESTful Services

This section introduces RESTful Services, and provides guidelines and examples for developing applications that use RESTful Services.

Topics:

- RESTful Services Terminology
- About Request Path Syntax Requirements
- "Getting Started" Documents Included in Installation

- About cURL and Testing RESTful Services
- Automatic Enabling of Schema Objects for REST Access (AutoREST)
- Manually Creating RESTful Services Using SQL and PL/SQL
- About Working with Dates Using Oracle REST Data Services

Related Topics

- Developing Oracle REST Data Services Applications

RESTful Services Terminology

This section introduces some common terms that are used throughout this document:

- **RESTful service**: An HTTP web service that conforms to the tenets of the RESTful architectural style.
- **Resource module**: An organizational unit that is used to group related resource templates.
- **Resource template**: An individual RESTful service that is able to service requests for some set of URIs (Universal Resource Identifiers). The set of URIs is defined by the URI Pattern of the Resource Template
- **URI pattern**: A pattern for the resource template. Can be either a route pattern or a URI template, although you are encouraged to use route patterns.
- **Route pattern**: A pattern that focuses on decomposing the path portion of a URI into its component parts. For example, a pattern of `/:object/:id?` will match `/emp/101` (matches a request for the item in the `emp` resource with `id` of `101`) and will also match `/emp/` (matches a request for the `emp` resource, because the `:id` parameter is annotated with the `?` modifier, which indicates that the `id` parameter is optional).

  For a detailed explanation of route patterns, see `docs\javadoc\plugin-api\route-patterns.html`, under `<sqldeveloper-install>\ords` and under the location (if any) where you manually installed Oracle REST Data Services.
- **URI template**: A simple grammar that defines the specific patterns of URIs that a given resource template can handle. For example, the pattern `employees/{id}` will match any URI whose path begins with `employees/`, such as `employees/2560`.
- **Resource handler**: Provides the logic required to service a specific HTTP method for a specific resource template. For example, the logic of the GET HTTP method for the preceding resource template might be:

  ```
  select empno, ename, dept from emp where empno = :id
  ```
- **HTTP operation**: HTTP (HyperText Transport Protocol) defines standard methods that can be performed on resources: `GET` (retrieve the resource contents), `POST` (store a new resource), `PUT` (update an existing resource), and `DELETE` (remove a resource).

Related Topics

- About RESTful Web Services

About Request Path Syntax Requirements

To prevent path-based attacks, Oracle REST Data Services performs a number of validation checks on the syntax of the path element of each request URL.

Each path must conform to the following rules:

- Is not empty or whitespace-only
- Does not contain any of the following characters: ?, #, ;, %
- Does not contain the null character (\u0000)
- Does not contain characters in the range: \u0001-\u0031
- Does not end with white space or a period (.)
- Does not contain double forward slash (//) or double back slash(\\)
- Does not contain two or more periods in sequence (.., ..., and so on)
- Total length is {@value #MAX_PATH_LENGTH} characters or less
- Does not match any of the following names (case insensitive), with or without file extensions: CON, PRN, AUX, CLOCK$, NUL, COM0, COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, LPT0, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9

If you intend to auto-REST enable objects, then avoid object names that do not comply with these requirements. For example, do not create a table named #EMPS. If you do want to auto-REST enable objects that have non-compliant names, then you must use an alias that complies with the requirements.

These requirements are applied to the URL decoded form of the URL, to prevent attempted circumvention of percent encodings.

"Getting Started" Documents Included in Installation

When you install Oracle REST Data Services, an examples folder is created with subfolders and files that you may find helpful. The installation folder hierarchy includes this:

```
ords
  conf
  docs
  examples
    soda
    getting-started
      ...
```

In this hierarchy:

- `examples\soda`: Contains sample JSON documents used in some examples included in *Oracle REST Data Services SODA for REST Developer's Guide*.
- `examples\getting-started`: Double-click `index.html` for a short document about how to get started developing RESTful Services using Oracle REST Data Services. This document focuses on using SQL Developer to get started. (SQL Developer is the primary tool for managing Oracle REST Data Services. For example, the ability to auto-enable REST support for schemas and tables is available only in SQL Developer.)

About cURL and Testing RESTful Services

Other sections show the testing of RESTful Services using a web browser. However, another useful way to test RESTful Services is using the command line tool named cURL.

This powerful tool is available for most platforms, and enables you to see and control what data is being sent to and received from a RESTful service.

```
curl -i https://server:port/ords/workspace/hr/employees/7369
```

This example produces a response like the following:

```
HTTP/1.1 200 OK
Server: Oracle-REST-Data-Services/2.0.6.78.05.25
ETag: "..."
Content-Type: application/json
Transfer-Encoding: chunked
Date: Thu, 28 Mar 2014 16:49:34 GMT

{
 "empno":7369,
 "ename":"SMITH",
 "job":"CLERK",
 "mgr":7902,
 "hiredate":"1980-12-17T08:00:00Z",
 "sal":800,
 "deptno":20
}
```

The `-i` option tells cURL to display the HTTP headers returned by the server.

Related Topics

- Exploring the Sample RESTful Services in Application Express (Tutorial)

> **See Also:**
>
> cURL
> The example in this section uses cURL with the services mentioned in Exploring the Sample RESTful Services in Application Express (Tutorial)

Automatic Enabling of Schema Objects for REST Access (AutoREST)

If Oracle REST Data Services has been installed on the system associated with a database connection, you can use the AutoREST feature to conveniently enable or disable Oracle REST Data Services access for specified tables and views in the schema associated with that database connection. Enabling REST access to a table, view or PL/SQL function, procedure or package allows it to be accessed through RESTful services.

AutoREST is a quick and easy way to expose database tables as REST resources. You sacrifice some flexibility and customizability to gain ease of effort. AutoRest lets you quickly expose data but (metaphorically) keeps you on a set of guide rails. For

example, you cannot customize the output formats or the input formats, or do extra validation.

On the other hand, manually created resource modules require you to specify the SQL and PL/SQL to support the REST resources. Using resource modules requires more effort, but offers more flexibility; for example, you can customize what fields are included, do joins across multiple tables, and validate the incoming data using PL/SQL.

So, as an application developer you must make a choice: use the "guide rails" of AutoREST, or create a resource module to do exactly what you need. If you choose AutoREST, you can just enable a table (or set of tables) within a schema.

Note that enabling a schema is not equivalent to enabling all tables and views in the schema. It just means making Oracle REST Data Services aware that the schema exists and that it may have zero or more resources to expose to HTTP. Those resources may be AutoREST resources or resource module resources.

You can automatically enable Oracle REST Data Services queries to access individual database schema objects (tables, views, and PL/SQL) by using a convenient wizard in Oracle SQL Developer. (Note that this feature is only available for Oracle REST Data Services- enabled schemas, not for Oracle Application Express workspaces.)

To enable Oracle REST Data Services access to one or more specified tables or views, you must do the following in SQL Developer:

1. Enable the schema (the one associated with the connection) for REST access.

   Schema level: To enable Oracle REST Data Services access to selected objects (that you specify in the next step) in the schema associated with a connection, right-click its name in the Connections navigator and select **REST Services**, then **Enable REST Services**.

   (To drop support for Oracle REST Data Services access to objects in the schema associated with a connection, right-click its name in the Connections navigator and select **REST Services**, then **Drop REST Services**.)

2. Individually enable REST access for the desired objects.

   Table or view level: To enable Oracle REST Data Services access to a specified table or view, right-click its name in the Connections navigator and select **Enable REST Services**.

For detailed usage information, click the **Help** button in the wizard or dialog box in SQL Developer.

Examples: Accessing Objects Using RESTful Services

This section provides examples of using Oracle REST Data Services queries and other operations against tables and views after you have REST-enabled them.

You can automatically expose table and view objects as RESTful services using SQL Developer. This topic provides examples of accessing these RESTful services.

> **Tip:**
>
> Although these examples illustrate the URL patterns used to access these resources, clients should avoid hard coding knowledge of the structure of these URLs; instead clients should follow the hyperlinks in the resources to navigate between resources. The structure of the URL patterns may evolve and change in future releases.

This topic provides examples of accessing objects using RESTful Services.

- Get Schema Metadata
- Get Object Metadata
- Get Object Data
- Get Table Data Using Paging
- Get Table Data Using Query
- Get Table Row Using Primary Key
- Insert Table Row
- Update/Insert Table Row
- Delete Using Filter
- Post by Batch Load

Get Schema Metadata

This example retrieves a list of resources available through the specified schema alias. It shows RESTful services that are created by automatically enabling a table or view, along with RESTful Services that are created by resource modules.

This example retrieves a list of resources available through the specified schema alias.

Pattern: `GET http://<HOST>:<PORT>/ords/<SchemaAlias>/metadata-catalog/`

Example: `GET http://localhost:8080/ords/ordstest/metadata-catalog/`

Result:

```
{
  "items": [
 {
   "name": "EMP",
   "links": [
  {
    "rel": "describes",
    "href": "http://localhost:8080/ords/ordstest/emp/"
  },
  {
    "rel": "canonical",
    "href": "http://localhost:8080/ords/ordstest/metadata-catalog/emp/",
    "mediaType": "application/json"
  }
   ]
 },
 {
```

```
   "name": "oracle.examples.hello",
   "links": [
  {
    "rel": "describes",
    "href": "http://localhost:8080/ords/ordstest/examples/hello/"
  },
  {
    "rel": "canonical",
    "href": "http://localhost:8080/ords/ordstest/metadata-catalog/examples/
hello/",
    "mediaType": "application/json"
  }
   ]
 }
  ],
  "hasMore": false,
  "limit": 25,
  "offset": 0,
  "count": 2,
  "links": [
 {
   "rel": "self",
   "href": "http://localhost:8080/ords/ordstest/metadata-catalog/"
 },
 {
   "rel": "first",
   "href": "http://localhost:8080/ords/ordstest/metadata-catalog/"
 }
  ]
}
```

The list of resources includes:

- Resources representing tables or views that have been REST enabled.
- Resources defined by resource modules. Note that only resources having a concrete path (that is, not containing any parameters) will be shown. For example, a resource with a path of `/module/some/path/` will be shown, but a resource with a path of `/module/some/:parameter/` will not be shown.

Each available resource has two hyperlinks:

- The link with relation `describes` points to the actual resource.
- The link with relation `canonical` describes the resource.

Get Object Metadata

This example retrieves the metadata (which describes the object) of an individual object. The location of the metadata is indicated by the `canonical` link relation.

Pattern: `GET http://<HOST>:<PORT>/ords/<SchemaAlias>/metadata-catalog/<ObjectAlias>/`

Example: `GET http://localhost:8080/ords/ordstest/metadata-catalog/emp/`

Result:

```
{
    "name": "EMP",
    "primarykey": [
        "empno"
```

```
    ],
    "members": [
        {
            "name": "empno",
            "type": "NUMBER"
        },
        {
            "name": "ename",
            "type": "VARCHAR2"
        },
        {
            "name": "job",
            "type": "VARCHAR2"
        },
        {
            "name": "mgr",
            "type": "NUMBER"
        },
        {
            "name": "hiredate",
            "type": "DATE"
        },
        {
            "name": "sal",
            "type": "NUMBER"
        },
        {
            "name": "comm",
            "type": "NUMBER"
        },
        {
            "name": "deptno",
            "type": "NUMBER"
        }
    ],
    "links": [
        {
            "rel": "collection",
            "href": "http://localhost:8080/ords/ordstest/metadata-catalog/",
            "mediaType": "application/json"
        },
        {
            "rel": "canonical",
            "href": "http://localhost:8080/ords/ordstest/metadata-catalog/emp/"
        },
        {
            "rel": "describes",
            "href": "http://localhost:8080/ords/ordstest/emp/"
        }
    ]
}
```


Get Object Data

This example retrieves the data in the object. Each row in the object corresponds to a JSON object embedded within the JSON array

Pattern: `GET http://<HOST>:<PORT>/ords/<SchemaAlias>/<ObjectAlias>/`

Example: `GET http://localhost:8080/ords/ordstest/emp/`

Result:


```
{
 "items": [
  {
   "empno": 7499,
   "ename": "ALLEN",
   "job": "SALESMAN",
   "mgr": 7698,
   "hiredate": "1981-02-20T00:00:00Z",
   "sal": 1600,
   "comm": 300,
   "deptno": 30,
   "links": [
    {
     "rel": "self",
     "href": "http://localhost:8080/ords/ordstest/emp/7499"
    }
   ]
  },
  ...
  {
   "empno": 7934,
   "ename": "MILLER",
   "job": "CLERK",
   "mgr": 7782,
   "hiredate": "1982-01-23T00:00:00Z",
   "sal": 1300,
   "comm": null,
   "deptno": 10,
   "links": [
    {
     "rel": "self",
     "href": "http://localhost:8080/ords/ordstest/emp/7934"
    }
   ]
  }
 ],
 "hasMore": false,
 "limit": 25,
 "offset": 0,
 "count": 13,
 "links": [
  {
   "rel": "self",
   "href": "http://localhost:8080/ords/ordstest/emp/"
  },
  {
   "rel": "edit",
   "href": "http://localhost:8080/ords/ordstest/emp/"
  },
  {
   "rel": "describedby",
   "href": "http://localhost:8080/ords/ordstest/metadata-catalog/emp/"
  },
  {
   "rel": "first",
   "href": "http://localhost:8080/ords/ordstest/emp/"
  }
 ]
}
```

Get Table Data Using Paging

This example specifies the `offset` and `limit` parameters to control paging of result data.

Pattern: `GET http://<HOST>:<PORT>/ords/<SchemaAlias>/<ObjectAlias>/?offset=<Offset>&limit=<Limit>`

Example: `GET http://localhost:8080/ords/ordstest/emp/?offset=10&limit=5`

Result:

```
{
 "items": [
  {
   "empno": 7900,
   "ename": "JAMES",
   "job": "CLERK",
   "mgr": 7698,
   "hiredate": "1981-12-03T00:00:00Z",
   "sal": 950,
   "comm": null,
   "deptno": 30,
   "links": [
    {
     "rel": "self",
     "href": "http://localhost:8080/ords/ordstest/emp/7900"
    }
   ]
  },
  ...
  {
   "empno": 7934,
   "ename": "MILLER",
   "job": "CLERK",
   "mgr": 7782,
   "hiredate": "1982-01-23T00:00:00Z",
   "sal": 1300,
   "comm": null,
   "deptno": 10,
   "links": [
    {
     "rel": "self",
     "href": "http://localhost:8080/ords/ordstest/emp/7934"
    }
   ]
  }
 ],
 "hasMore": false,
 "limit": 5,
 "offset": 10,
 "count": 3,
 "links": [
  {
   "rel": "self",
   "href": "http://localhost:8080/ords/ordstest/emp/"
  },
  {
   "rel": "edit",
   "href": "http://localhost:8080/ords/ordstest/emp/"
```

```
  },
  {
   "rel": "describedby",
   "href": "http://localhost:8080/ords/ordstest/metadata-catalog/emp/"
  },
  {
   "rel": "first",
   "href": "http://localhost:8080/ords/ordstest/emp/?limit=5"
  },
  {
   "rel": "prev",
   "href": "http://localhost:8080/ords/ordstest/emp/?offset=5&limit=5"
  }
 ]
}
```

Get Table Data Using Query

This example specifies a filter clause to restrict objects returned.

Pattern: `GET http://<HOST>:<PORT>/ords/<SchemaAlias>/<ObjectAlias>/?q=<FilterClause>`

Example: `GET http://localhost:8080/ords/ordstest/emp/?q={"deptno":{"$lte":20}}`

Result:

```
{
 "items": [
  {
   "empno": 7566,
   "ename": "JONES",
   "job": "MANAGER",
   "mgr": 7839,
   "hiredate": "1981-04-01T23:00:00Z",
   "sal": 2975,
   "comm": null,
   "deptno": 20,
   "links": [
    {
     "rel": "self",
     "href": "http://localhost:8080/ords/ordstest/emp/7566"
    }
   ]
  },
  ...
  {
   "empno": 7934,
   "ename": "MILLER",
   "job": "CLERK",
   "mgr": 7782,
   "hiredate": "1982-01-23T00:00:00Z",
   "sal": 1300,
   "comm": null,
   "deptno": 10,
   "links": [
    {
     "rel": "self",
     "href": "http://localhost:8080/ords/ordstest/emp/7934"
    }
```

```
   ]
  }
 ],
 "hasMore": false,
 "limit": 25,
 "offset": 0,
 "count": 7,
 "links": [
  {
   "rel": "self",
   "href": "http://localhost:8080/ords/ordstest/emp/?
q=%7B%22deptno%22:%7B%22%24lte%22:20%7D%7D"
  },
  {
   "rel": "edit",
   "href": "http://localhost:8080/ords/ordstest/emp/?
q=%7B%22deptno%22:%7B%22%24lte%22:20%7D%7D"
  },
  {
   "rel": "describedby",
   "href": "http://localhost:8080/ords/ordstest/metadata-catalog/emp/"
  },
  {
   "rel": "first",
   "href": "http://localhost:8080/ords/ordstest/emp/?
q=%7B%22deptno%22:%7B%22%24lte%22:20%7D%7D"
  }
 ]
}
```

Get Table Row Using Primary Key

This example retrieves an object by specifying its identifying key values.

Pattern: `GET http://<HOST>:<PORT>/ords/<SchemaAlias>/<ObjectAlias>/<KeyValues>`

Where `<KeyValues>` is a comma-separated list of key values (in key order).

Example: `GET http://localhost:8080/ords/ordstest/emp/7839`

Result:

```
{
 "empno": 7839,
 "ename": "KING",
 "job": "PRESIDENT",
 "mgr": null,
 "hiredate": "1981-11-17T00:00:00Z",
 "sal": 5000,
 "comm": null,
 "deptno": 10,
 "links": [
  {
   "rel": "self",
   "href": "http://localhost:8080/ords/ordstest/emp/7839"
  },
  {
   "rel": "edit",
   "href": "http://localhost:8080/ords/ordstest/emp/7839"
  },
```

```
  {
   "rel": "describedby",
   "href": "http://localhost:8080/ords/ordstest/metadata-catalog/emp/item"
  },
  {
   "rel": "collection",
   "href": "http://localhost:8080/ords/ordstest/emp/"
  }
 ]
}
```

Insert Table Row

This example inserts data into the object. The body data supplied with the request is a JSON object containing the data to be inserted.

If the object has a primary key, then there must be an insert trigger on the object that populates the primary key fields. If the table does not have a primary key, then the ROWID of the row will be used as the item's identifier.

If the object lacks a trigger to assign primary key values, then the PUT operation described in next section,**Update/Insert Table Row** should be used instead.

Pattern: `POST http://<HOST>:<PORT>/ords/<SchemaAlias>/<ObjectAlias>/`

Example:

```
curl -i -H "Content-Type: application/json" -X POST -d "{ \"empno\" :7,
\"ename\": \"JBOND\", \"job\":\"SPY\", \"deptno\" :11 }" "http://localhost:8080/
ords/ordstest/emp/
Content-Type: application/json

{ "empno" :7, "ename": "JBOND", "job":"SPY", "deptno" :11 }
```

Result:

```
{
 "empno": 7,
 "ename": "JBOND",
 "job": "SPY",
 "mgr": null,
 "hiredate": null,
 "sal": null,
 "comm": null,
 "deptno": 11,
 "links": [
  {
   "rel": "self",
   "href": "http://localhost:8080/ords/ordstest/emp/7"
  },
  {
   "rel": "edit",
   "href": "http://localhost:8080/ords/ordstest/emp/7"
  },
  {
   "rel": "describedby",
   "href": "http://localhost:8080/ords/ordstest/metadata-catalog/emp/item"
  },
  {
   "rel": "collection",
   "href": "http://localhost:8080/ords/ordstest/emp/"
```

```
  }
 ]
}
```

Update/Insert Table Row

This example inserts or updates (sometimes called an "upsert") data in the object. The body data supplied with the request is a JSON object containing the data to be inserted or updated.

Pattern: `PUT http://<HOST>:<PORT>/ords/<SchemaAlias>/<ObjectAlias>/<KeyValues>`

Example:

```
curl -i -H "Content-Type: application/json" -X PUT -d "{ \"empno\" :7,
\"ename\": \"JBOND\", \"job\":\"SPY\", \"deptno\" :11 }" "http://localhost:8080/
ords/ordstest/emp/7
Content-Type: application/json

{ "empno" :7, "ename": "JBOND", "job":"SPY", "deptno" :11 }
```

Result:

```
{
 "empno": 7,
 "ename": "JBOND",
 "job": "SPY",
 "mgr": null,
 "hiredate": null,
 "sal": null,
 "comm": null,
 "deptno": 11,
 "links": [
  {
   "rel": "self",
   "href": "http://localhost:8080/ords/ordstest/emp/7"
  },
  {
   "rel": "edit",
   "href": "http://localhost:8080/ords/ordstest/emp/7"
  },
  {
   "rel": "describedby",
   "href": "http://localhost:8080/ords/ordstest/metadata-catalog/emp/item"
  },
  {
   "rel": "collection",
   "href": "http://localhost:8080/ords/ordstest/emp/"
  }
 ]
}
```

Delete Using Filter

This example deletes object data specified by a filter clause.

Pattern: `DELETE http://<HOST>:<PORT>/ords/<SchemaAlias>/<ObjectAlias>/?q=<FilterClause>`

Example: `curl -i -X DELETE "http://localhost:8080/ords/ordstest/emp/?q={"deptno":11}"`

Result:

```
{
    "itemsDeleted": 1
}
```

Post by Batch Load

This example inserts object data using the batch load feature. The body data supplied with the request is a CSV file. The behavior of the batch operation can be controlled using the optional query parameters, which are described in Table 5-1.

Pattern: `POST http://<HOST>:<PORT>/ords/<SchemaAlias>/<ObjectAlias>/batchload?<Parameters>`

Parameters:

Table 5-1 Parameters for batchload

Parameter	Description
batchesPerCommit	Sets the frequency for commits. Optional commit points can be set after a batch is sent to the database. The default is every 10 batches. 0 indicates commit deferred to the end of the load. Type: Integer.
batchRows	Sets the number of rows in each batch to send to the database. The default is 50 rows per batch. Type: Integer.
dateFormat	Sets the format mask for the date data type. This format is used when converting input data to columns of type date. Type: String.
delimiter	Sets the field delimiter for the fields in the file. The default is the comma (,).
enclosures	embeddedRightDouble
errors	Sets the user option used to limit the number of errors. If the number of errors exceeds the value specified for `errorsMax` (the service option) or by `errors` (the user option), then the load is terminated. To permit no errors at all, specify 0. To indicate that all errors be allowed (up to errorsMax value), specify UNLIMITED (-1) .
errorsMax	A service option used to limit the number of errors allowed by users. It intended as an option for the service provider and not to be exposed as a user option. If the number of errors exceeds the value specified for `errorsMax` (the service option) or by `errors` (the user option), then the load is terminated. To permit no errors at all, specify 0. To indicate that all errors be allowed, specify UNLIMITED (-1).
lineEnd	Sets the line end (terminator). If the file contains standard line end characters (\r. \r\n or \n), then `lineEnd` does not need to be specified.
lineMax	Sets a maximum line length for identifying lines/rows in the data stream. A `lineMax` value will prevent reading an entire stream as a single line when the incorrect `lineEnd` character is being used. The default is unlimited.
locale	Sets the locale.
responseEncoding	Sets the encoding for the response stream.

Table 5-1 (Cont.) Parameters for batchload

Parameter	Description
responseFormat	Sets the format for response stream. This format determines how messages and bad data will be formatted. Valid values: `RAW`, `SQL`.
timestampFormat	Sets the format mask for the time stamp data type. This format is used when converting input data to columns of type time stamp.
timestampTZFormat	Sets the format mask for the time stamp time zone data type. This format is used when converting input data to columns of type time stamp time zone.
truncate	Indicates if and/or how table data rows should be deleted before the load. `False` (the default) does not delete table data before the load; `True` causes table data to be deleted with the DELETE SQL statement; `Truncate` causes table data to be deleted with the TRUNCATE SQL statement.

Example:

```
POST http://localhost:8080/ords/ordstest/emp/batchload?batchRows=25
Content-Type: text/csv

empno,ename,job,mgr,hiredate,sal,comm,deptno
0,M,SPY MAST,,2005-05-01 11:00:01,4000,,11
7,J.BOND,SPY,0,2005-05-01 11:00:01,2000,,11
9,R.Cooper,SOFTWARE,0,2005-05-01 11:00:01,10000,,11
26,Max,DENTIST,0,2005-05-01 11:00:01,5000,,11
```

Result:

```
#INFO Number of rows processed: 4
#INFO Number of rows in error: 0
#INFO Elapsed time: 00:00:03.939 - (3,939 ms) 0 - SUCCESS: Load processed
without errors
```

Filtering in Queries

This section describes and provides examples of filtering in queries against REST-enabled tables and views.

Filtering is the process of limiting a collection resource by using a per-request dynamic filter definition across multiple page resources, where each page contains a subset of items found in the complete collection. Filtering enables efficient traversal of large collections.

To filter in a query, include the parameter q=*FilterObject*, where *FilterObject* is a JSON object that represents the custom selection and sorting to be applied to the resource. For example, assume the following resource:

```
https://example.com/ords/scott/emp/
```

The following query includes a filter that restricts the ENAME column to "JOHN":

```
https://example.com/ords/scott/emp/?q={"ENAME":"JOHN"}
```

FilterObject Grammar

The `FilterObject` must be a JSON object that complies with the following syntax:

```
FilterObject { orderby , asof, wmembers }
```

The `orderby`, `asof`, and `wmembers` attributes are optional, and their definitions are as follows:

```
orderby
  "$orderby": {orderByMembers}

orderByMembers
    orderByProperty
    orderByProperty , orderByMembers

orderByProperty
    columnName : sortingValue

sortingValue
  "ASC"
  "DESC"
  "-1"
  "1"
   -1
   1

asof
  "$asof": date
  "$asof": "datechars"
  "$asof": scn
  "$asof": +int

wmembers
    wpair
    wpair , wmembers

wpair
    columnProperty
    complexOperatorProperty

columnProperty
    columnName : string
    columnName : number
    columnName : date
    columnName : simpleOperatorObject
columnName : complexOperatorObject
    columnName : [complexValues]

columnName
  "\p{Alpha}[[\p{Alpha}]]([[\p{Alnum}]#$_])*$"

complexOperatorProperty
    complexKey : [complexValues]
    complexKey : simpleOperatorObject

complexKey
  "$and"
  "$or"
```

```
complexValues
    complexValue , complexValues

complexValue
    simpleOperatorObject
    complexOperatorObject
    columnObject

columnObject
    {columnProperty}

simpleOperatorObject
    {simpleOperatorProperty}

complexOperatorObject
    {complexOperatorProperty}

simpleOperatorProperty
   "$eq" : string | number | date
   "$ne" : string | number | date
   "$lt" :  number | date
   "$lte" : number | date
   "$gt" : number | date
   "$gte" : number | date
   "$instr" : string
   "$ninstr" : string
   "$like" : string
   "$null" : null
   "$notnull" : null
   "$between" : betweenValue

betweenValue
    [null , betweenNotNull]
    [betweenNotNull , null]
    [betweenRegular , betweenRegular]

betweenNotNull
    number
    date

betweenRegular
    string
    number
    date
```

Data type definitions include the following:

```
string
       JSONString
number
       JSONNumber
date
       {"$date":"datechars"}
scn
       {"$scn": +int}
```

Where:

```
datechars is an RFC3339 date format in UTC (Z)
```

```
JSONString
          ""
          " chars "
chars
         char
         char chars
char
         any-Unicode-character except-"-or-\-or-control-character
          \"
          \\
           \/
          \b
          \f
          \n
          \r
          \t
          \u four-hex-digits


JSONNumber
    int
    int frac
    int exp
    int frac exp
int
    digit
    digit1-9 digits
    - digit
    - digit1-9 digits
frac
    . digits
exp
    e digits
digits
    digit
    digit digits
e
    e
    e+
    e-
    E
    E+
    E-
```

The `FilterObject` must be encoded according to Section 2.1 of RFC3986.

Examples: FilterObject Specifications

The following are examples of operators in `FilterObject` specifications.

ORDER BY property ($orderby)

Order by with literals

```
{
 "$orderby": {"SALARY":  "ASC","ENAME":"DESC"}
}
```

Order by with numbers

```
{
```

```
 "$orderby": {"SALARY":  -1,"ENAME":  1}
}
```

ASOF property ($asof)

With SCN (Implicit)

```
{
  "$asof": 1273919
}
```

With SCN (Explicit)

```
{
  "$asof": {"$scn": "1273919"}
}
```

With Date (Implicit)

```
{
  "$asof": "2014-06-30T00:00:00Z"
}
```

With Date (Explicit)

```
{
  "$asof": {"$date": "2014-06-30T00:00:00Z"}
}
```

EQUALS operator ($eq)

(Implicit and explicit equality supported._

Implicit (Support String and Dates too)

```
{
 "SALARY": 1000
}
```

Explicit

```
{
 "SALARY": {"$eq": 1000}
}
```

Strings

```
{
 "ENAME": {"$eq":"SMITH"}
}
```

Dates

```
{
  "HIREDATE": {"$date": "1981-11-17T08:00:00Z"}
}
```

NOT EQUALS operator ($ne)

Number

```
{
 "SALARY": {"$ne": 1000}
}
```

String

```
{
 "ENAME": {"$ne":"SMITH"}
}
```

Dates

```
{
  "HIREDATE": {"$ne": {"$date":"1981-11-17T08:00:00Z"}}
}
```

LESS THAN operator ($lt)
(Supports dates and numbers only)

Numbers

```
{
  "SALARY": {"$lt": 10000}
}
```

Dates

```
{
  "SALARY": {"$lt": {"$date":"1999-12-17T08:00:00Z"}}
}
```

LESS THAN OR EQUALS operator ($lte)
(Supports dates and numbers only)

Numbers

```
{
  "SALARY": {"$lte": 10000}
}
```

Dates

```
{
  "HIREDATE": {"$lte": {"$date":"1999-12-17T08:00:00Z"}}
}
```

GREATER THAN operator ($gt)
(Supports dates and numbers only)

Numbers

```
{
  "SALARY": {"$gt": 10000}
}
```

Dates

```
{
  "SALARY": {"$gt": {"$date":"1999-12-17T08:00:00Z"}}
}
```

GREATER THAN OR EQUALS operator ($gte)
(Supports dates and numbers only)

Numbers

```
{
  "SALARY": {"$gte": 10000}
}
```

Dates

```
{
  "HIREDATE": {"$gte": {"$date":"1999-12-17T08:00:00Z"}}
}
```

In string operator ($instr)
(Supports strings only)

```
{
  "ENAME": {"$instr":"MC"}
}
```

Not in string operator ($ninstr)
(Supports strings only)

```
{
  "ENAME": {"$ninstr":"MC"}
}
```

LIKE operator ($like)
(Supports strings. Eescape character not supported to try to match expressions with _ or % characters.)

```
{
  "ENAME": {"$like":"AX%"}
}
```

BETWEEN operator ($between)
(Supports string, dates, and numbers)

Numbers

```
{
  "SALARY": {"$between": [1000,2000]}
}
```

Dates

```
{
  "SALARY": {"$between": [{"$date":"1989-12-17T08:00:00Z"},
{"$date":"1999-12-17T08:00:00Z"}]}
}
```

Strings

```
{
  "ENAME": {"$between": ["A","C"]}
}
```

Null Ranges ($lte equivalent)
(Supported by numbers and dates only)

```
{
  "SALARY": {"$between": [null,2000]}
}
```

Null Ranges ($gte equivalent)
(Supported by numbers and dates only)

```
{
  "SALARY": {"$between": [1000,null]}
}
```

NULL operator ($null)

```
{
  "ENAME": {"$null": null}
}
```

NOT NULL operator ($notnull)

```
{
  "ENAME": {"$notnull": null}
}
```

AND operator ($and)
(Supports all operators, including $and and $or)

Column context delegation
(Operators inside $and will use the closest context defined in the JSON tree.)

```
{
  "SALARY": {"$and": [{"$gt": 1000},{"$lt":4000}]}
}
```

Column context override
(Example: salary greater than 1000 and name like S%)

```
{
  "SALARY": {"$and": [{"$gt": 1000},{"ENAME": {"$like":"S%"}} ] }
}
```

Implicit and in columns

```
{
  "SALARY": [{"$gt": 1000},{"$lt":4000}]
}
```

High order AND

```
(All first columns and or high order operators -- $and and $ors -- defined at
the first level of the JSON will be joined and an implicit AND)
(Example: Salary greater than 1000 and name starts with S or T)

{
  "SALARY": {"$gt": 1000},
  "ENAME": {"$or": [{"$like":"S%"}, {"$like":"T%"}]}
}
```

Invalid expression (operators $lt and $gt lack column context)

```
{
   "$and": [{"$lt": 5000},{"$gt": 1000}]
}
```

Valid alternatives for the previous invalid expression

```
{
   "$and": [{"SALARY": {"$lt": 5000}}, {"SALARY": {"$gt": 1000}}]
}

{
   "SALARY": [{"$lt": 5000},{"$gt": 1000}]
}

{
   "SALARY": {"$and": [{"$lt": 5000},{"$gt": 1000}]}
}
```

OR operator ($or)
(Supports all operators including $and and $or)

Column context delegation
(Operators inside $or will use the closest context defined in the JSON tree)

```
{
  "ENAME": {"$or": [{"$eq":"SMITH"},{"$eq":"KING"}]}
}
```

Column context override
(Example: name starts with S or salary greater than 1000)

```
{
  "SALARY": {"$or": [{"$gt": 1000},{"ENAME": {"$like":"S%"}} ] }
}
```

Auto PL/SQL

This section explains how PL/SQL is made available through HTTP(S) for Remote Procedure call (RPC).

The auto PL/SQL feature uses a standard to provide consistent encoding and data transfer in a stateless web service environment. Using this feature, you can enable Oracle Database stored PL/SQL functions and procedures at package level through Oracle REST Data Services, similar to how you enable the views and tables.

Auto Enabling PL/SQL Subprograms

Oracle REST Data Services supports auto enabling of the following PL/SQL objects, based on their catalog object identifier:

- PL/SQL Procedure
- PL/SQL Function
- PL/SQL Package

The functions, and procedures within the PL/SQL package cannot be individually enabled as they are named objects within a PL/SQL package object. Therefore, the granularity level enables the objects at the package level. This granularity level enables to expose all of its public functions and procedures.

If you want to *only* enable a subset of functions and procedures, then you must create a separate delegate package and enable it to expose only that subset of functions and procedures.

> **Note:**
>
> Overloaded package functions and procedures are not supported.

Method and Content Type Supported for Auto Enabling PL/SQL Objects

This section discusses the method and content-type supported by this feature.

The auto enabling of the PL/SQL Objects feature supports POST as the HTTP method. In POST method, input parameters are encoded in the payload and output parameters are decoded from the response.

> **Note:**
>
> The standard data CRUD to HTTP method mappings are not applicable as this feature provides an RPC-style interaction.

The content-type supported is `application/json`.

Auto-Enabling the PL/SQL Objects

This section explains how to auto-enable the PL/SQL objects through Oracle REST Data Services.

You can enable the PL/SQL objects in one of the following ways:

- Auto-Enabling Using the PL/SQL API
- Auto-Enabling the PL/SQL Objects Using SQL Developer

Auto-Enabling Using the PL/SQL API

You can enable a PL/SQL object using the Oracle REST Data Services PL/SQL API.

To enable the PL/SQL package, use the Oracle REST Data Services PL/SQL API as shown in following sample code snippet:

```
BEGIN
  ords.enable_object(
    p_enabled => TRUE,
    p_schema => 'MY_SCHEMA',
    p_object => 'MY_PKG',
    p_object_type => 'PACKAGE',
    p_object_alias => 'my_pkg',
    p_auto_rest_auth => FALSE);
   commit;
END;
/
```

Example 5-1 Enabling the PL/SQL Function

To enable the PL/SQL function, use the Oracle REST Data Services PL/SQL API as shown in following sample code snippet:

```
BEGIN
  ords.enable_object(
    p_enabled => TRUE,
    p_schema => 'MY_SCHEMA',
    p_object => 'MY_FUNC',
    p_object_type => 'FUNCTION',
    p_object_alias => 'my_func',
    p_auto_rest_auth => FALSE);

    commit;
END;
/
```

Example 5-2 Enabling the PL/SQL Procedure

To enable the PL/SQL procedure, use the Oracle REST Data Services PL/SQL API as shown in following sample code snippet:

```
BEGIN
  ords.enable_object(
    p_enabled => TRUE,
    p_schema => 'MY_SCHEMA',
    p_object => 'MY_PROC',
    p_object_type => 'PROCEDURE',
    p_object_alias => 'my_proc',
    p_auto_rest_auth => FALSE);

    commit;
END;
/
```

Auto-Enabling the PL/SQL Objects Using SQL Developer

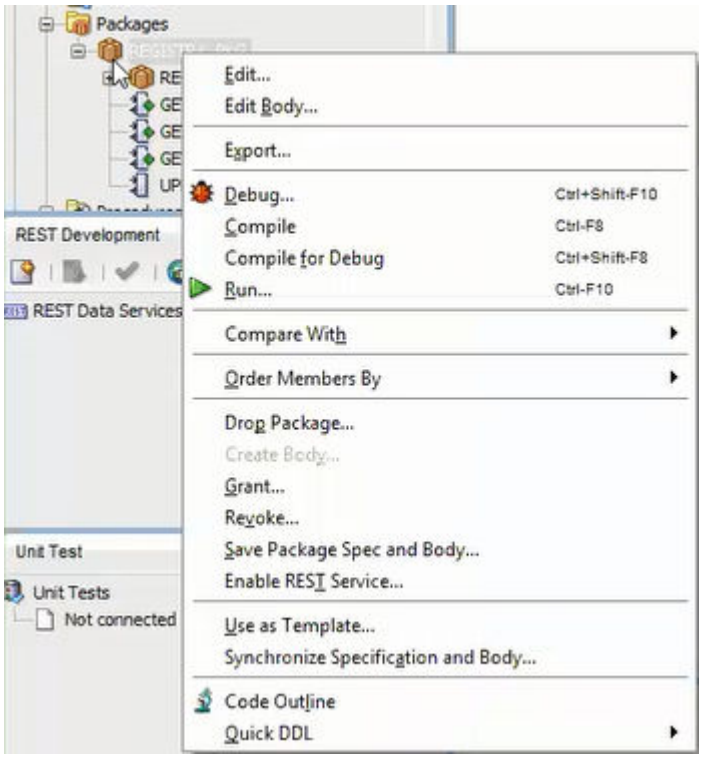
This section describes how to enable the PL/SQL objects using SQL Developer 4.2 and above.

To enable the PL/SQL objects (for example, package) using SQL Developer, perform the following steps:

> **Note:**
>
> You can now enable, packages, functions and procedures. However, the granularity of enabling is either at the whole package level, standalone function level, or at the standalone procedure level.

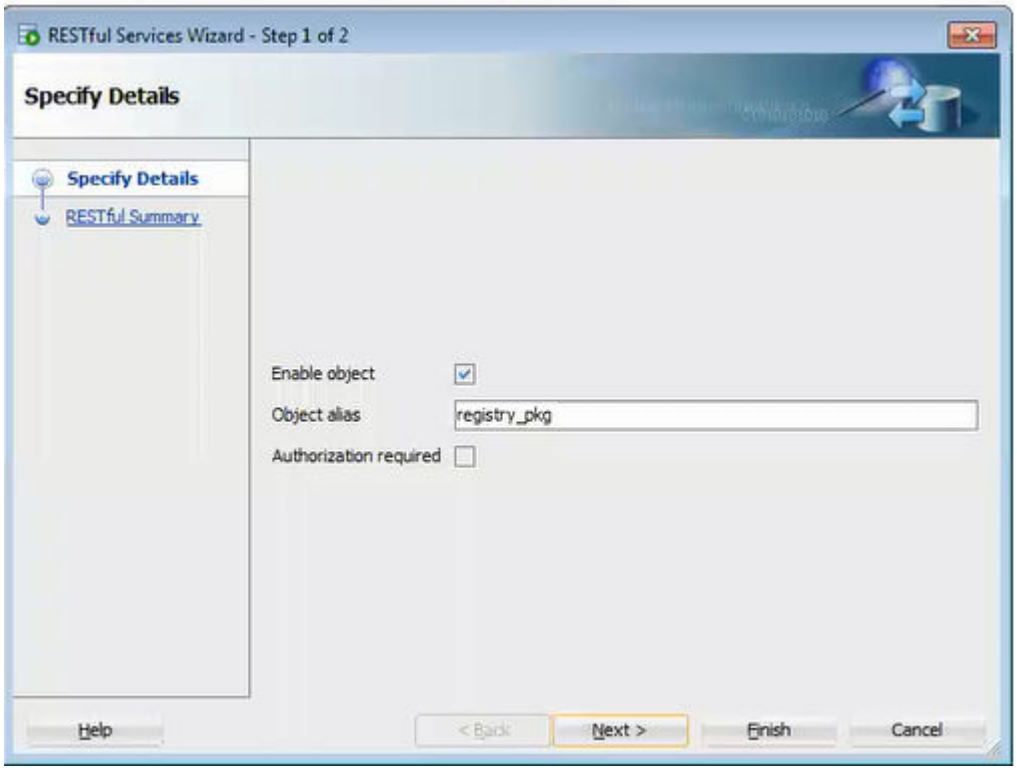
1. In SQL Developer, right-click on a package as shown in the following figure:

Figure 5-1 Selecting the Enable REST Service Option



2. Select **Enable RESTful Services** to display the following wizard page:

Figure 5-2 Auto Enabling the PL/SQL Package Object



- **Enable object**: Enable this option (that is, enable REST access for the package).
- **Object alias**: Accept `registry_pkg` for the object alias.
- **Authorization required**: For simplicity, disable this option.
- On the RESTful Summary page of the wizard, click **Finish**.

Generating the PL/SQL Endpoints

HTTP endpoints are generated dynamically per request for the enabled database objects. Oracle REST Data Services uses the connected database catalog to generate the endpoints using a query.

The following rules apply for all the database objects for generating the HTTP endpoints:

- All names are converted to lowercase
- An endpoint is generated if it is not already allocated

Stored Procedure and Function Endpoints

The function or procedure name is generated into the URL in the same way as tables and views in the same namesspace.

Example 5-3 Generating an Endpoint for the Stored Procedure

```
CREATE OR REPLACE PROCEDURE MY_SCHEMA.MY_PROC IS
BEGIN
  NULL;
END;
```

Following endpoint is generated:

```
http://localhost:8080/ords/my_schema/my_proc/
```

Example 5-4 Package Procedure and Function Endpoints

The package, function, and procedure endpoints are generated with package name as a parent. Endpoints for functions and procedures that are not overloaded or where the lowercase name is not already in use are generated.

If you have a package, MY_PKG as defined in the following code snippet:

```
CREATE OR REPLACE  PACKAGE MY_SCHEMA.MY_PKG AS
  PROCEDURE MY_PROC;
  FUNCTION MY_FUNC RETURN VARCHAR2;
  PROCEDURE MY_PROC2;
  PROCEDURE "my_proc2";
  PROCEDURE MY_PROC3(P1 IN VARCHAR);
  PROCEDURE MY_PROC3(P2 IN NUMBER);
END MY_PKG;
```

Then the following endpoints are generated:

```
http://localhost:8080/ords/my_schema/my_pkg/MY_PROC
http://localhost:8080/ords/my_schema/my_pkg/MY_FUNC
```

> **Note:**
>
> Endpoints for the procedure `my_proc2` is not generated because its name is not unique when the name is converted to lowercase, and endpoints for the procedure `my_proc3` is not generated because it is overloaded.

Resource Input Payload

The input payload is a JSON document with values adhering to the REST standard.

The payload should contain a name/value pair for each IN or IN OUT parameter as shown in the following code snippet:

```
{
  "p1": "abc",
  "p2": 123,
  "p3": null
}
```

> **Note:**
>
> Where there are no IN or IN OUT parameters, an empty JSON body is required as shown in the following code snippet:
>
> ```
> {
>
> }
> ```

Oracle REST Data Services uses the database catalog metadata to unmarshal the JSON payload into Oracle database types, which is ready to be passed to the database through JDBC.

Resource Payload Response

When the PL/SQL object is executed successfully, it returns a JSON body.

The JSON body returned, contains all OUT and IN OUT output parameter values. Oracle REST Data Services uses the database catalog metadata to marshal the execution of the result back into JSON as shown in the following code snippet:

```
{
  "p3" : "abc123",
  "p4" : 1
}
```

Where there are no OUT or IN OUT parameters, an empty JSON body is returned as shown in the following code snippet:

```
{

}
```

Function Return Value

The return value of functions do not have an associated name.

As the return value of functions do not have an associated name, the name `"~ret"` is used as shown in the following code snippet:

```
{
  "~ret" : "abc123"
}
```

Manually Creating RESTful Services Using SQL and PL/SQL

This section describes how to manually create RESTful Services using SQL and PL/SQL and shows how to use a JSON document to pass parameters to a stored procedure in the body of a REST request.

This section includes the following topics:

- About Oracle REST Data Services Mechanisms for Passing Parameters
- Using SQL/JSON Database Functions

About Oracle REST Data Services Mechanisms for Passing Parameters

This section describes the main mechanisms that Oracle REST Data Services supports for passing parameters using REST HTTP to handlers that are written by the developer:

- Using JSON to Pass Parameters

  You can use JSON in the body of REST requests, such as the `POST` or `PUT` method, where each parameter is a JSON name/value pair.

- Using Route Patterns to Pass Parameters

  You can use route patterns for required parameters in the URI to specify parameters for REST requests such as the `GET` method, which does not have a body, and in other special cases.

- Using Query Strings for Optional Parameters

  You can use query strings for optional parameters in the URI to specify parameters for REST requests, such as the `GET` method, which does not have a body, and in other special cases.

Prerequisite Setup Tasks To Be Completed Before Performing Tasks for Passing Parameters

This prerequisite setup information assumes you have completed steps 1 and 2 in **Getting Started with RESTful Services** section, where you have REST-enabled the `ordstest` schema and `emp` database table (Step 1) and created and tested the RESTful service from a SQL query (Step 2). You must complete these two steps before performing the tasks about passing parameters described in the subsections that follow.

Related Topics

- Getting Started with RESTful Services

Using JSON to Pass Parameters

This section shows how to use a JSON document to pass parameters to a stored procedure in the body of a REST request, such as `POST` or `PUT` method, where each parameter is a name/value pair. This operation performs an update on a record, which in turn returns the change to the record as an `OUT` parameter.

Perform the following steps:

1. **Note:**

   The following stored procedure performs an update on an existing record in the `emp` table to promote an employee by changing any or all of the following: job, salary, commission, department number, and manager. The stored procedure returns the salary change as an `OUT` parameter.

```
create or replace procedure promote ( l_empno IN number,
l_job IN varchar2,
           l_mgr IN number, l_sal IN number,  l_comm IN
number,  l_deptno IN number,
           l_salarychange OUT number)
        is
            oldsalary   number;
        begin
            select nvl(e.sal, 0)into oldsalary FROM emp e
                    where e.empno = l_empno;
            update emp e set
                e.job = nvl(l_job, e.job),
                e.mgr = nvl(l_mgr, e.mgr),
                e.sal = nvl(l_sal, e.sal),
                e.comm = nvl(l_comm, e.comm),
                e.deptno = nvl(l_deptno, e.deptno)
                        where e.empno = l_empno;
            l_salarychange := nvl(l_sal, oldsalary) -
oldsalary;
        end;
```

   As a privileged `ordstest` user, connect to the `ordstest` schema and create the `promote` stored procedure.

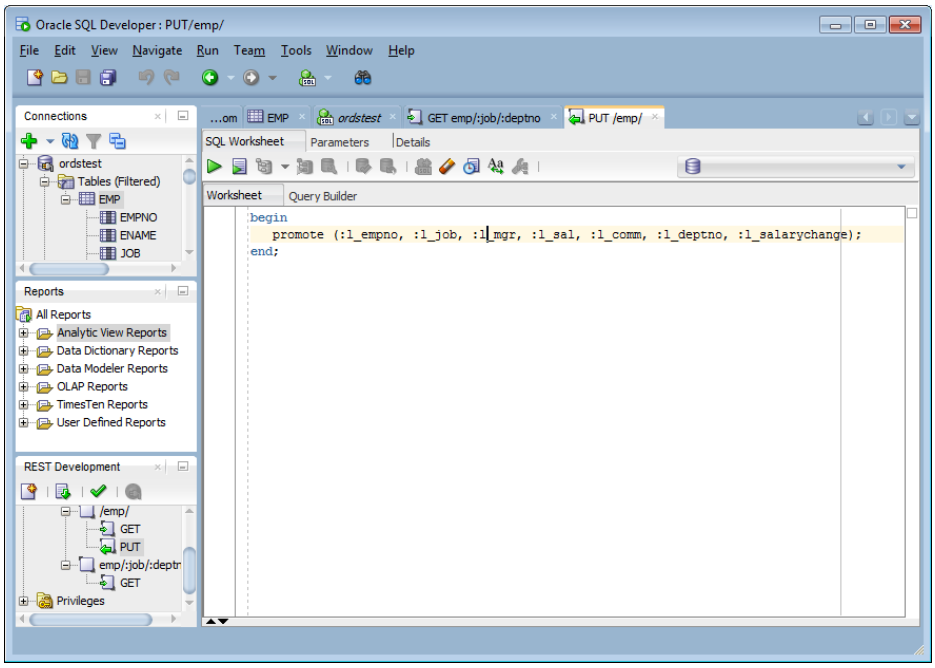
2. Perform the following steps to setup a handler for a `PUT` request on the `emp` resource to pass parameters in the body of the `PUT` method in a JSON document to the `promote` stored procedure.

   a. Using Oracle SQL Developer, in the REST Development section, right click on the `emp` template and select **Add Handler** for the `PUT` method.

   b. In the **Create Resource Handler** dialog, click the green plus symbol to add the MIME type `application/json` and then click **Apply** to send it a JSON document in the body of the `PUT` method.

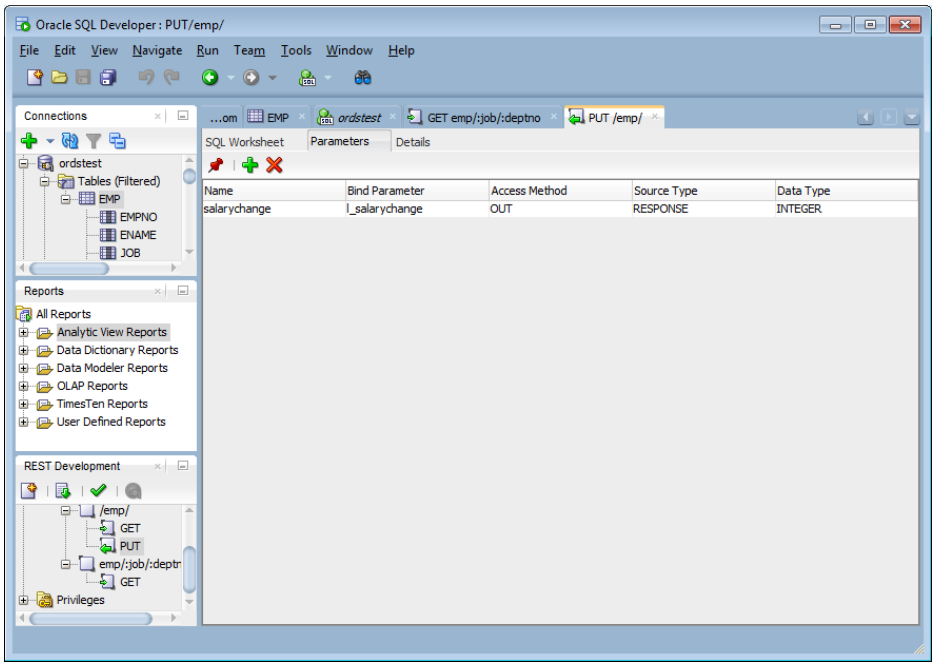
   c. Using the SQL Worksheet, add the following anonymous PL/SQL block: `begin promote (:l_empno, :l_job, :l_mgr, :l_sal, :l_comm, :l_deptno, :l_salarychange); end;` as shown in the following figure.

Figure 5-3 Adding an Anonymous PL/SQL Block to the Handler for the PUT Method



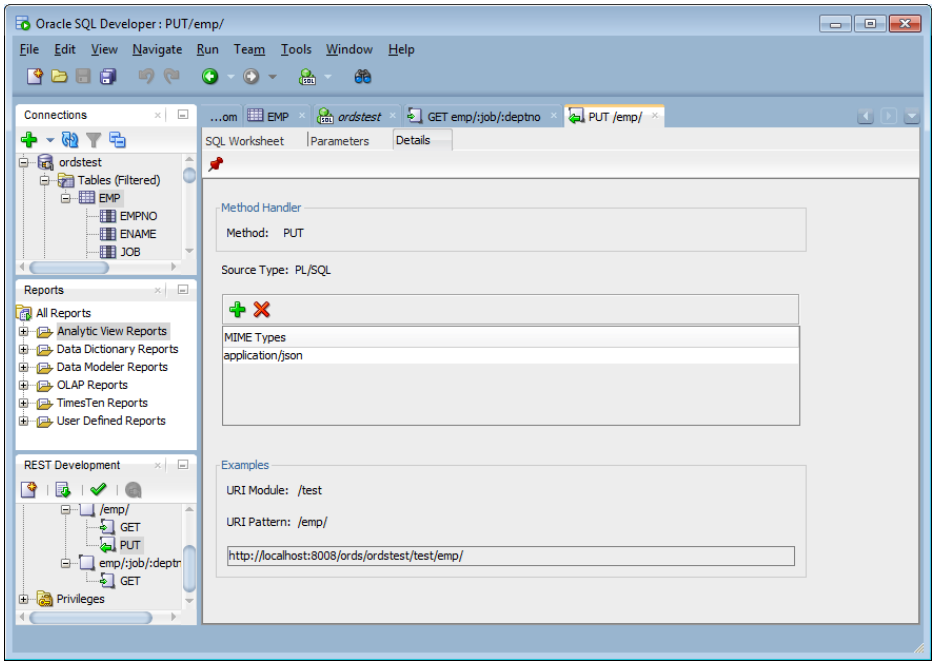
d. Click the **Parameters** tab to set the **Bind Parameter** as `l_salarychange` , the **Access Method** as an `OUT` parameter, the **Source Type** as `RESPONSE`, and **Data Type** as `INTEGER` as shown in the following figure. This is the promote procedure's output which is an integer value equal to the change in salary in a JSON name/value format.

Figure 5-4 Setting the Bind Parameter l_salarychange to Pass for the PUT Method



e. Click the **Details** tab to get the URL to call as shown in the **Examples** section of the following figure. Copy this URL to your clipboard.

Figure 5-5 Obtaining the URL to Call from the Details Tab



f. Right click on the `test` module to upload the module. Do not forget this step.

3. To test the RESTful service, execute the following cURL command in the command prompt:`curl -i -H "Content-Type: application/json" -X PUT -d "{ \"l_empno\" : 7499, \"l_sal\" : 9999, \"l_job\" : \"Director\", \"l_comm\" : 300}`

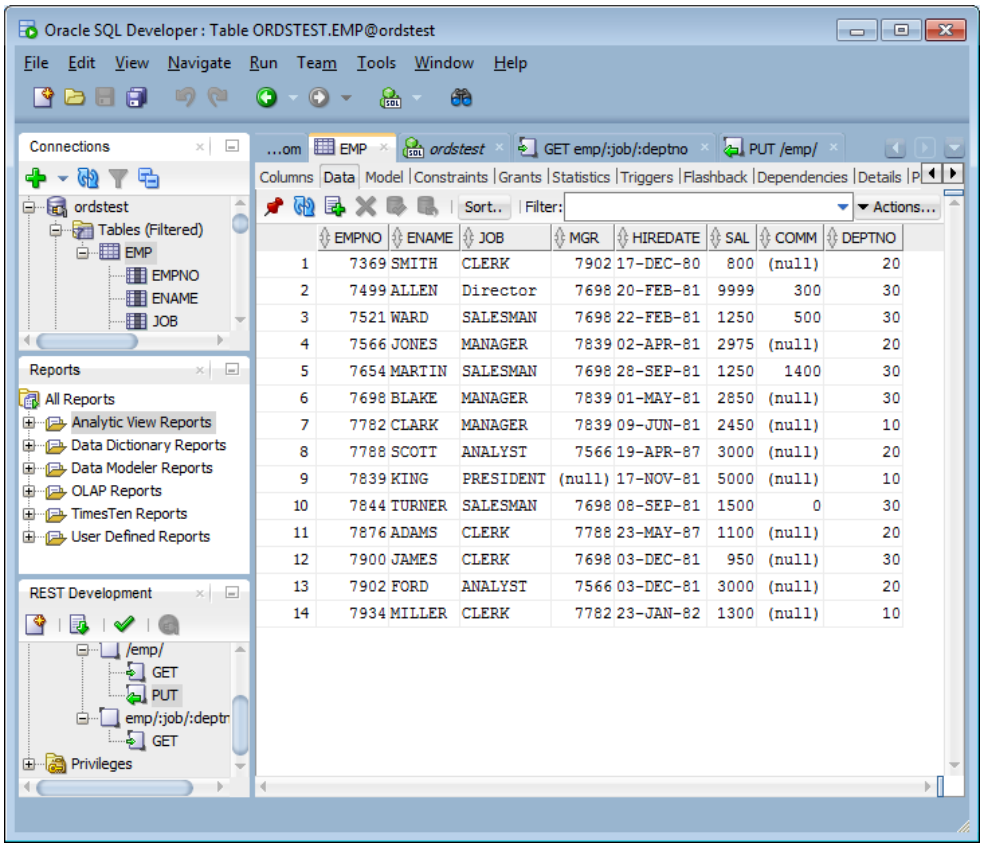
> **Note:**
>
> You can also use any REST client available to test the RESTful service.

The cURL command returns the following response:

```
HTTP/1.1 200 OK
Content-Type: application/json Transfer-Encoding: chunked
{"salarychange":8399}
```

4. In SQL Developer SQL Worksheet, perform the following `SELECT` statement on the `emp` table: `SELECT * from emp` to see that the `PUT` method was executed, then select the **Data** tab to display the records for the `EMP` table.

Figure 5-6 Displaying the Results from a SQL Query to Confirm the Execution of the PUT Method



Note:

- All parameters are optional. If you leave out a name/value pair for a parameter in your JSON document, the parameter is set to `NULL`.
- The name/value pairs can be arranged in any order in the JSON document. JSON allows much flexibility in this regard in the JSON document.
- Only one level of JSON is supported. You can not have nested JSON objects or arrays.

Using Route Patterns to Pass Parameters

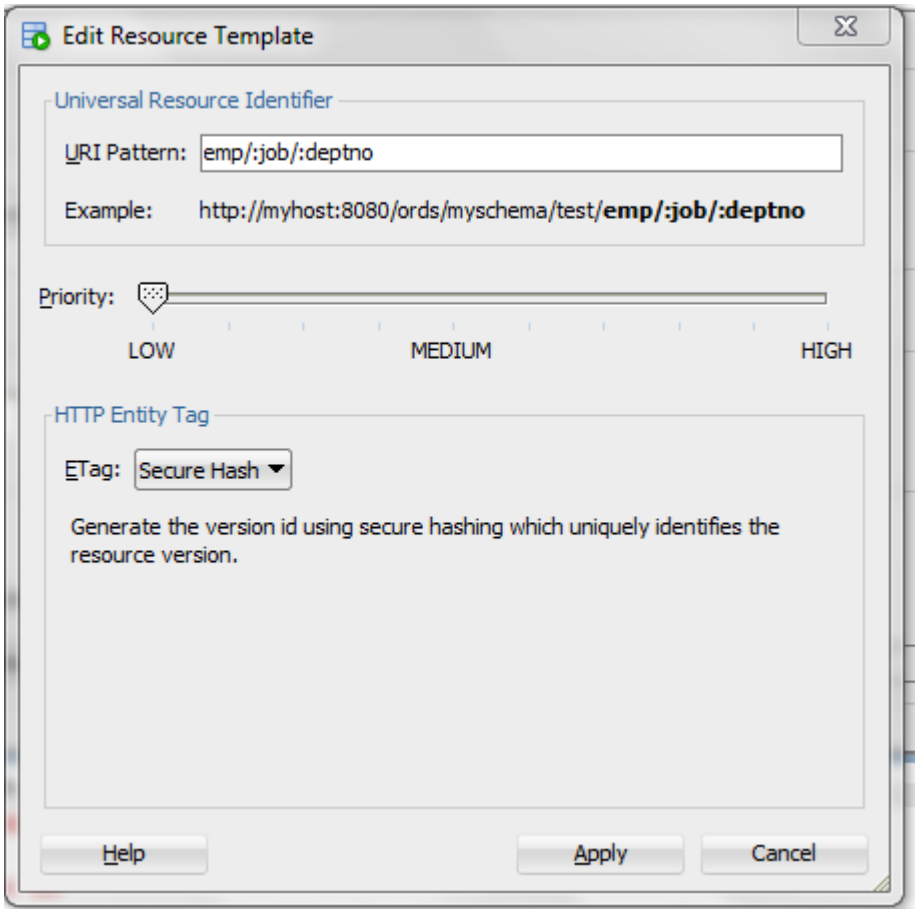
This section describes how to use route patterns in the URI to specify parameters for REST requests, such as with the `GET` method, which does not have a body.

First create a `GET` method handler for a query on the `emp` table that has many bind variables. These steps use a route pattern to specify the parameter values that are required.

Perform the following steps to use a route pattern to send a `GET` method with some required parameter values:

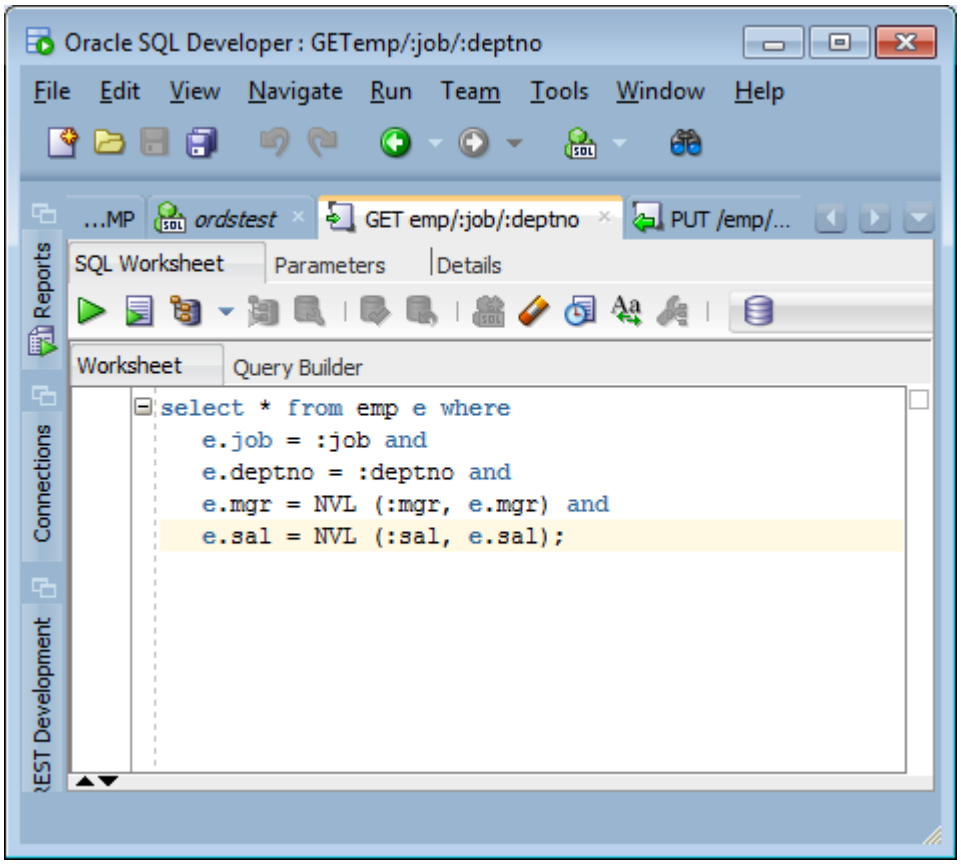
1. In SQL Developer, right click on the test module and select **Add Template** to create a new template that calls `emp`; however, in this case the template definition includes a route pattern for the parameters or bind variables that is included in the URI rather than in the body of the method. To define the required parameters, use a route pattern by specifying a `/:` before the `job` and `deptno` parameters. For example, for the URI pattern, enter: `emp/:job/:deptno` as shown in the following figure.

Figure 5-7 Creating a Template Definition to Include a Route Pattern for Some Parameters or Bind Variables



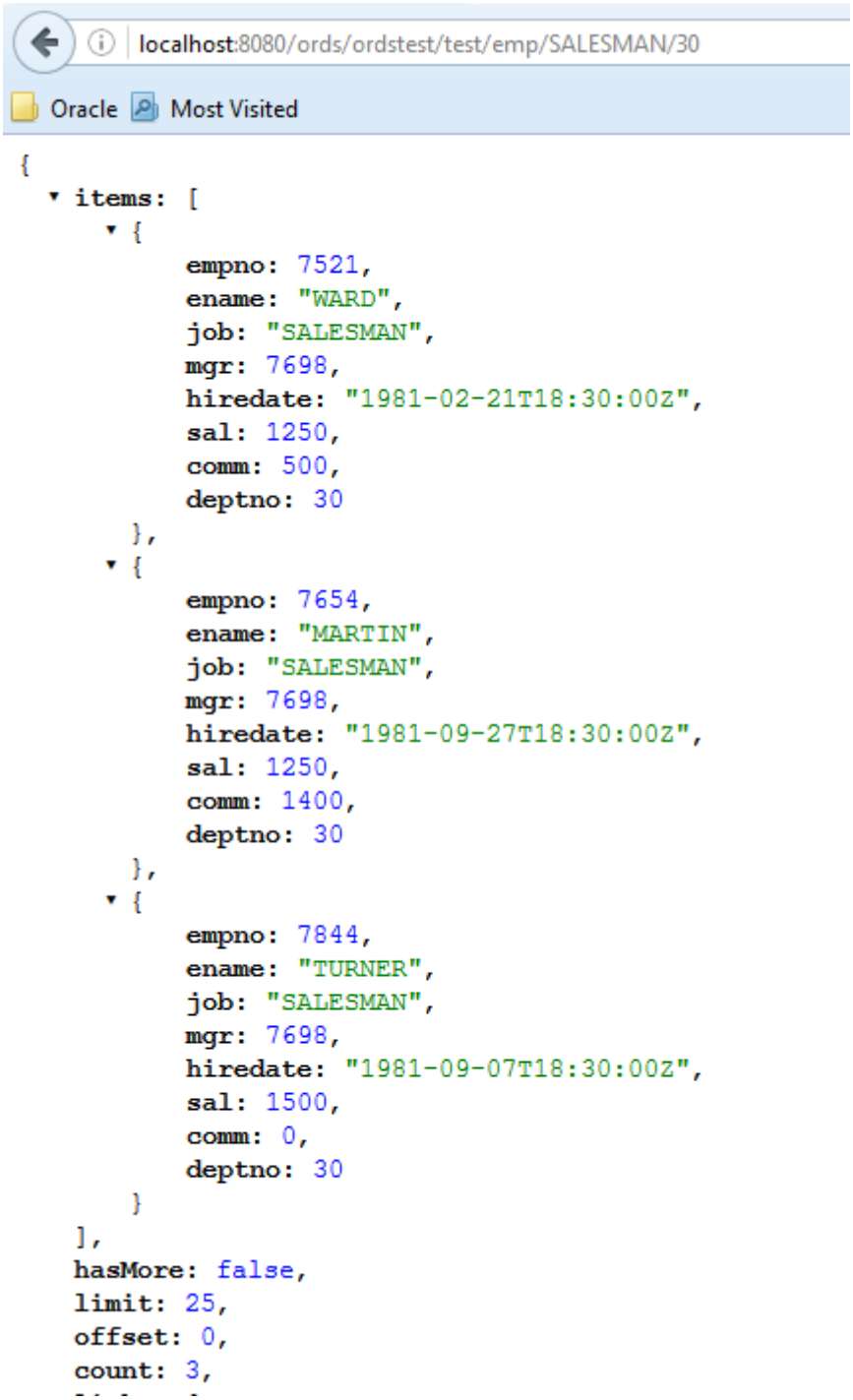
2. Click **Next** to go to **REST Data Services — Step 2 of 3**, and click **Next** to go to **REST Data Services — Step 3 of 3**, then click **Finish** to complete the template.
3. Right click on the `emp/:job/:deptno` template and select `Add Handler` for the `GET` method.
4. Right click on the `GET` method to open the handler.
5. Add the following query to the SQL Worksheet: `select * from emp e where e.job = :job and e.deptno = :deptno and e.mgr = NVL (:mgr, e.mgr) and e.sal = NVL (:sal, e.sal);` as also shown in the following figure.

Figure 5-8 Adding a SQL Query to the Handler



6. Click the **Details** tab to get the URL to call. Copy this URL to your clipboard.
7. Right click on the `test` module to upload the module. Do not forget this step.
8. Test the REST endpoint. In a web browser enter the URL:`http://localhost:8080/ords/ordstest/test/emp/SALESMAN/30` as shown in the following figure.

Figure 5-9 Using Browser to Show the Results of Using a Route Pattern to Send a GET Method with Some Required Parameter Values



The query returns 3 records for the salesmen named Ward, Martin, and Turner.

> **See Also:**
>
> To learn more about Route Patterns see this document in the Oracle REST Data Services distribution at `docs/javadoc/plugin-api/route-patterns.html` and this document **Oracle REST Data Services Route Patterns**

Using Query Strings for Optional Parameters

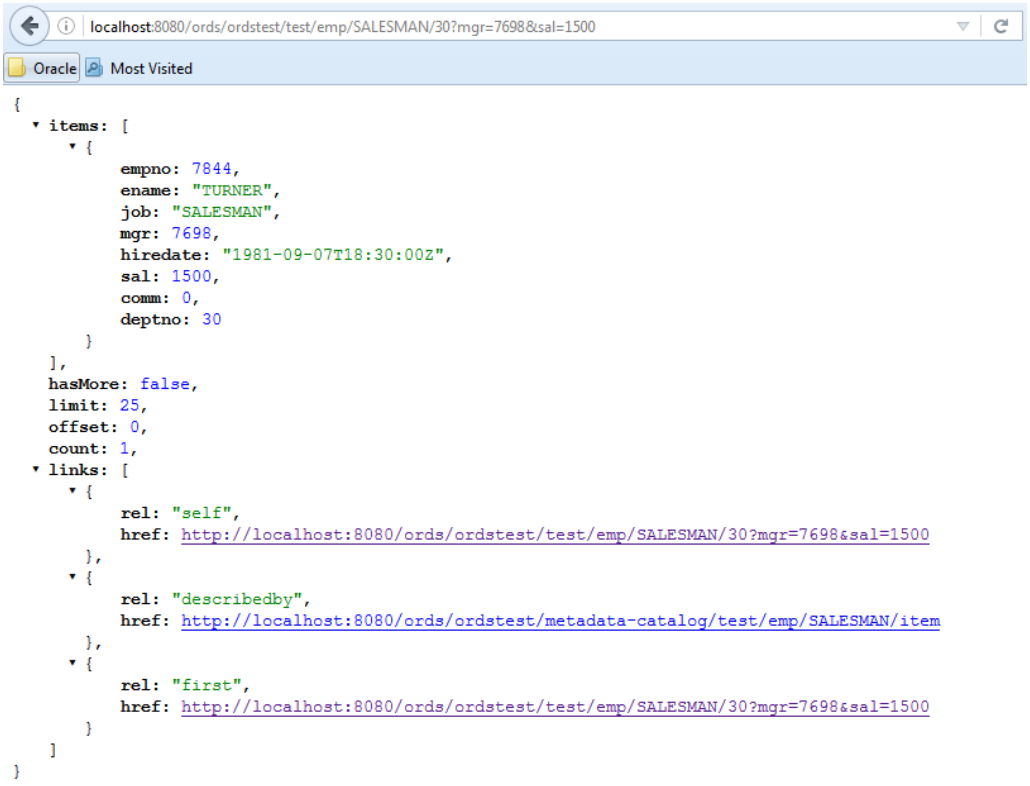
This section describes how to use query strings in the URI to specify parameters for REST requests like the GET method, which does not have a body. You can use query strings for any of the other optional bind variables in the query as you choose.

The syntax for using query strings is: `?parm1=value1&parm2=value2 … &parmN=valueN`.

For example, to further filter the query: `http://localhost:8080/ords/ordstest/test/emp/SALESMAN/30`, to use a query string to send a GET method with some parameter name/value pairs, select employees whose `mgr` (manager) is `7698` and whose `sal` (salary) is `1500` by appending the query string `?mgr=7698&sal=1500` to the URL as follows: `http://localhost:8080/ords/ordstest/test/emp/SALESMAN/30?mgr=7698&sal=1500`.

To test the endpoint, in a web browser enter the following URL: http://localhost:8080/ords/ordstest/test/emp/SALESMAN/30?mgr=7698&sal=1500 as shown in the following figure:

Figure 5-10 Using Browser to Show the Results of Using a Query String to Send a GET Method with Some Parameter Name/Value Pairs



The query returns one record for the salesman named Turner in department 30 who has a salary of 1500 and whose manager is 7698.

Note the following points:

- It is a good idea to URL encode your parameter values. This may not always be required; however, it is the safe thing to do. This prevents the Internet from transforming something, for example, such as a special character in to some other character that may cause a failure. Your REST client may provide this capability or you can search the Internet for the phrase `url encoder` to find tools that can do this for you.
- Never put a backslash at the end of your parameter list in the URI; otherwise, you may get a `404 Not Found` error.

> **See Also:**
>
> To gain more experience using JSON to pass parameter values, see **Lab 4 of the ORDS Oracle By Example (OBE)** and **Database Application Development Virtual Image**.

Using SQL/JSON Database Functions

This section describes how to use the SQL/JSON database functions available in Oracle Database 12c Release 2 (12.2) to map the nested JSON objects to and from the hierarchical relational tables.

This section includes the following topics:

- Inserting Nested JSON Objects into Relational Tables
- Generating Nested JSON Objects from Hierachical Relational Data

Inserting Nested JSON Objects into Relational Tables

This section explains how to insert JSON objects with nested arrays into multiple, hierarchical relational tables.

The two key technologies used to implement this functionality are as follows:

- The `:body` bind variable that Oracle REST Data Services provides to deliver JSON and other content in the body of POST and other REST calls into PL/SQL REST handlers
- JSON_TABLE and other SQL/JSON operators provided in Oracle Database 12c Release 2 (12.2)

Some of the advantages of using these technologies for inserting data into relational tables are as follows:

- Requirements for implementing this functionality are very minimal. For example, installation of JSON parser software is not required
- You can use simple, declarative code that is easy to write and understand when the JSON to relational mapping is simple
- Powerful and sophisticated capabilities to handle more complex mappings. This includes:
  - Mechanisms for mapping NULLS and boolean values
  - Sophisticated mechanisms for handling JSON. JSON evolves over time. Hence, the mapping code must be able to handle both the older and newer versions of the JSON documents.

    For example, simple scalar values may evolve to become JSON objects containing multiple scalars or nested arrays of scalar values or objects. SQL/JSON operators that return the scalar value can continue to work even when the simple scalar is embedded within these more elaborate structures. A special mechanism, called the **Ordinality Column**, can be used to determine the structure from where the value was derived.

> **See Also:**
>
> The following pages for more information on JSON_TABLE and other SQL/JSON operators and on Ordinality Column mechanism:
>
> - json_db
> - ordinality_column

Usage of the :body Bind Variable

This section provides some useful tips for using the `:body` bind variable.

Some of the useful tips for using the `:body` bind variable are as follows:

- The `:body` bind variable can be accessed, or de-referenced, only once. Subsequent accesses return a NULL value. So, you must first assign the `:body` bind variable to the local `L_PO` variable before using it in the two JSON_Table operations.
- The `:body` bind variable is a BLOB datatype and you can assign it only to a BLOB variable.

> **Note:**
>
> Since `L_PO` is a BLOB variable, you must use the `FORMAT JSON` phrase after the expression in the JSON_TABLE function. section for more information.

The `:body` bind variable can be used with other types of data such as image data.

> **See Also:**
>
> - Creating an Image Gallery for a working example of using `:body` bind variable with image data .
> - Database SQL Language Reference

Example of JSON Purchase Order with Nested LineItems

This section shows an example that takes the JSON Purchase Order with Nested LineItems and inserts it into a row of the PurchaseOrder table and rows of the LineItem table.

Example 5-5 Nested JSON Purchase Order with Nested LineItems

```
{"PONumber"        : 1608,
  "Requestor"      : "Alexis Bull",
  "CostCenter"     : "A50",
  "Address"        : {"street"  : "200 Sporting Green",
                      "city"    : "South San Francisco",
```

```
                        "state"   : "CA",
                        "zipCode" : 99236,
                        "country" : "United States of America"},
  "LineItems"      : [ {"ItemNumber" : 1,
                         "Part"       : {"Description" : "One Magic
Christmas",
                                         "UnitPrice"   : 19.95,
                                         "UPCCode"     : 1313109289},
                         "Quantity"   : 9.0},
                        {"ItemNumber" : 2,
                         "Part"       : {"Description" : "Lethal Weapon",
                                         "UnitPrice"   : 19.95,
                                         "UPCCode"     : 8539162892},
                         "Quantity"   : 5.0}]}'
```

Table Definitions for PurchaseOrder and LineItems Tables

This section provides definitions for the **PurchaseOrder** and **LineItem** tables.

The definitions for the **PurchaseOrder** and the **LineItems** tables are as follows:

```
CREATE TABLE PurchaseOrder (
     PONo NUMBER (5),
     Requestor VARCHAR2 (50),
     CostCenter VARCHAR2 (5),
     AddressStreet VARCHAR2 (50),
     AddressCity VARCHAR2 (50),
     AddressState VARCHAR2 (2),
     AddressZip VARCHAR2 (10),
     AddressCountry VARCHAR2 (50),
     PRIMARY KEY (PONo));

CREATE TABLE LineItem (
     PONo NUMBER (5),
     ItemNumber NUMBER (10),
     PartDescription VARCHAR2 (50),
     PartUnitPrice NUMBER (10),
     PartUPCCODE NUMBER (10),
     Quantity NUMBER (10),
     PRIMARY KEY (PONo,ItemNumber));
```

PL/SQL Handler Code for a POST Request

This section gives an example PL/SQL handler code for a POST request. The handler code is used to insert a purchase order into a row of the PurchaseOrder table and rows of the LineItem table.

Example 5-6 PL/SQL Handler Code Used for a POST Request

```
Declare
  L_PO     BLOB;

Begin
  L_PO := :body;
```

```
INSERT INTO PurchaseOrder
      SELECT * FROM json_table(L_PO  FORMAT JSON, '$'
         COLUMNS (
           PONo            Number    PATH '$.PONumber',
           Requestor       VARCHAR2  PATH '$.Requestor',
           CostCenter      VARCHAR2  PATH '$.CostCenter',
           AddressStreet   VARCHAR2  PATH '$.Address.street',
           AddressCity     VARCHAR2  PATH '$.Address.city',
           AddressState    VARCHAR2  PATH '$.Address.state',
           AddressZip      VARCHAR2  PATH '$.Address.zipCode',
           AddressCountry  VARCHAR2  PATH '$.Address.country'));

INSERT INTO LineItem
SELECT * FROM json_table(L_PO  FORMAT JSON, '$'
         COLUMNS (
           PONo  Number PATH '$.PONumber',
           NESTED             PATH '$.LineItems[*]'
             COLUMNS (
               ItemNumber        Number   PATH '$.ItemNumber',
               PartDescription   VARCHAR2   PATH '$.Part.Description',
               PartUnitPrice     Number   PATH '$.Part.UnitPrice',
               PartUPCCode       Number   PATH '$.Part.UPCCode',
               Quantity          Number   PATH '$.Quantity')));
commit;
end;
```

Creating the REST API Service to Invoke the Handler

This section explains how to create the REST API service to invoke the handler, using the Oracle REST Data Services.

To setup the REST API service, a URI is defined to identify the resource the REST calls will be operating on. The URI is also used by Oracle REST Data Services to route the REST HTTP calls to specific handlers. The general format for the URI is as follows:

```
<server>:<port>/ords/<schema>/<module>/<template>/<parameters>
```

Here, `<server>:<port>` is where the Oracle REST Data Service is installed. For testing purposes, you can use **demo** and **test** in place of **module** and **template** respectively in the URI. Modules are used to group together related templates that define the resources the REST API will be operating upon.

To create the REST API service, use one of the following methods:

- Use the Oracle REST Data Services PL/SQL API to define the REST service and a handler for the POST insert. Then connect to the `jsontable` schema on the database server that contains the PurchaseOrder and LineItem tables.

> **Note:**
>
> JSON_TABLE and other SQL/JSON operators use single quote so these must be escaped. For example, every single quote (') must be replaced with double quotes (").

- Use the Oracle REST Data Services, REST Development pane in SQL Developer to define the REST service.

Defining the REST Service and Handler using PL/SQL API

This section shows how to define the REST Service and Handler for the POST insert using the Oracle REST Data Services PL/SQL API.

You can alternatively use the Oracle REST Data Services REST development pane in SQL Developer to create the modules, templates and handlers.

```
BEGIN
  ORDS.ENABLE_SCHEMA(
      p_enabled             => TRUE,
      p_schema              => 'ORDSTEST',
      p_url_mapping_type    => 'BASE_PATH',
      p_url_mapping_pattern => 'ordstest',
      p_auto_rest_auth      => FALSE);

  ORDS.DEFINE_MODULE(
      p_module_name    => 'demo',
      p_base_path      => '/demo/',
      p_items_per_page =>  25,
      p_status         => 'PUBLISHED',
      p_comments       => NULL);
  ORDS.DEFINE_TEMPLATE(
      p_module_name    => 'demo',
      p_pattern        => 'test',
      p_priority       => 0,
      p_etag_type      => 'HASH',
      p_etag_query     => NULL,
      p_comments       => NULL);
  ORDS.DEFINE_HANDLER(
      p_module_name    => 'demo',
      p_pattern        => 'test',
      p_method         => 'POST',
      p_source_type    => 'plsql/block',
      p_items_per_page =>  0,
      p_mimes_allowed  => '',
      p_comments       => NULL,
      p_source         => '
declare
    L_PO BLOB := :body;
begin

INSERT INTO PurchaseOrder
      SELECT * FROM json_table(L_PO  FORMAT JSON, ''$''
         COLUMNS (
```

```
            PONo                  Number           PATH ''$.PONumber'',
            Requestor            VARCHAR2  PATH ''$.Requestor'',
            CostCenter           VARCHAR2   PATH ''$.CostCenter'',
            AddressStreet       VARCHAR2  PATH ''$.Address.street'',
            AddressCity          VARCHAR2  PATH ''$.Address.city'',
            AddressState         VARCHAR2  PATH ''$.Address.state'',
            AddressZip            VARCHAR2  PATH ''$.Address.zipCode'',
            AddressCountry     VARCHAR2  PATH ''$.Address.country''));

INSERT INTO LineItem
SELECT * FROM json_table(L_PO  FORMAT JSON, ''$''
         COLUMNS (
           PONo  Number PATH ''$.PONumber'',
           NESTED                       PATH ''$.LineItems[*]''
             COLUMNS (
               ItemNumber       Number      PATH ''$.ItemNumber'',
               PartDescription  VARCHAR2    PATH ''$.Part.Description'',
               PartUnitPrice      Number     PATH ''$.Part.UnitPrice'',
               PartUPCCode     Number     PATH ''$.Part.UPCCode'',
               Quantity              Number    PATH ''$.Quantity'')));

commit;
end;'
      );


  COMMIT;
END;
```

Related Topics

- Using the Oracle REST Data Services PL/SQL API
- About Oracle REST Data Services Mechanisms for Passing Parameters
- Oracle REST Data Services PL/SQL Package Reference

Generating Nested JSON Objects from Hierachical Relational Data

This section explains how to query the relational tables in hierarchical (parent/child) relationships and return the data in a nested JSON format using the Oracle REST Data Services.

The two key technologies used to implement this functionality are as follows:

- The new SQL/JSON functions available with Oracle Database 12c Release 2 (12.2). You can use `json_objects` for generating JSON objects from the relational tables, and `json_arrayagg`, for generating nested JSON arrays from nested (child) relational tables.
- The Oracle REST Data Services media source type used for enabling the REST service handler to execute a SQL query that in turn returns the following types of data:
  - The HTTP Content-Type of the data, which in this case is **application/json**
  - The JSON data returned by the `json_object`

Some of the advantages of using this approach are as follows:

- Requirements for implementing this functionality is very minimal. For example, installation of JSON parser software is not required.
- Simple, declarative coding which is easy to write and understand which makes the JSON objects to relational tables mapping simple.
- Powerful and sophisticated capabilities to handle more complex mappings. This includes mechanisms for mapping NULLS and boolean values.

  For example, a NULL in the Oracle Database can be converted to either the absence of the JSON element or to a JSON NULL value. The Oracle Database does not store Boolean types but the SQL/JSON functions allow string or numeric values in the database to be mapped to Boolean TRUE or FALSE values.

Bypassing JSON Generation for Relational Data

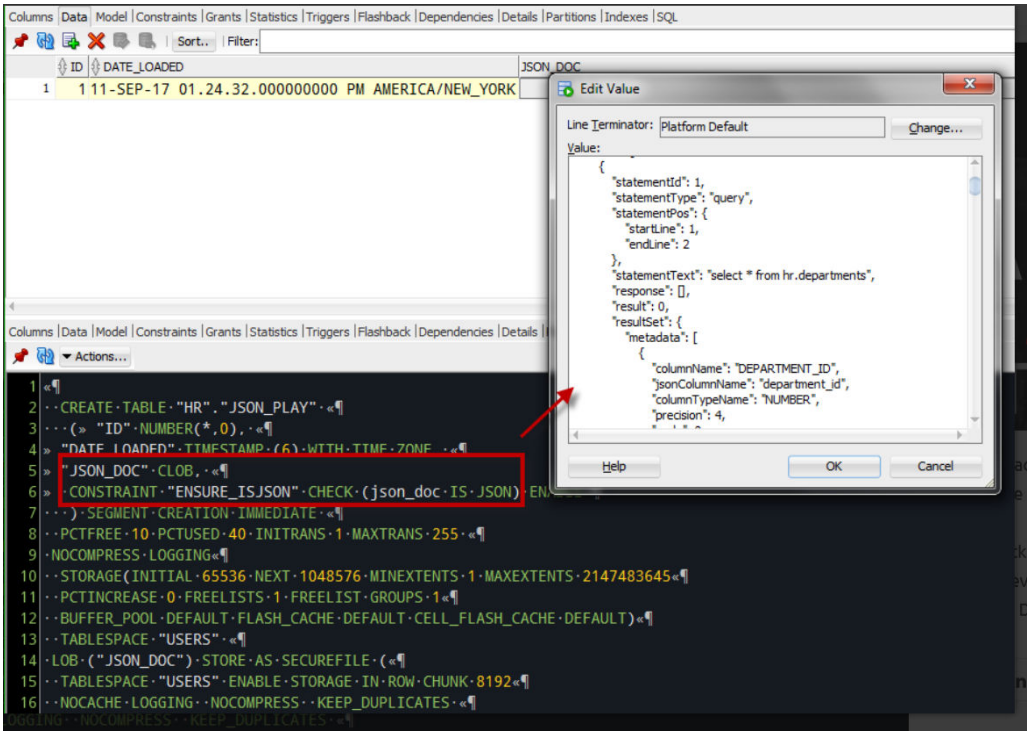
This section describes and provides solutions for handling responses that are already in a JSON format.

ORDS auto-formats your SQL or PL/SQL results and response to a JSON format before returning to your application. However, in some cases, the complete response body or part of it is already in a JSON format. Following are two such use cases:

Use Case 1: When the response is already in a JSON format

Following figure shows an example where the complete response is already in a JSON format:

Figure 5-11 Complete Response Body in JSON Format



You must adjust your GET query text to include "application/json" before including the JSON itself as shown in the following example GET query:

```
Select 'application/json',
            upper(json_doc)
   from json_play
```

The Media resource in this case is application/json and the browser handles it similar to a BLOB or a PDF.

Use Case 2: One or more columns of the response is already in a JSON format.

If one or more columns are in a JSON format, then such columns in the source query need to be aliased to indicate that the attribute must not be converted to a JSON format.

For example:

```
Select id,
                jsons "{}jsons"
           from table_with_json
```

The alias text is used to name the nested JSON document attribute.

> **See Also:**
>
> ORDS: Returning Raw {JSON}

Example to Generate Nested JSON Objects from the Hierachical Relational Tables

This section describes how to query or GET the data we inserted into the PurchaseOrder and LineItem relational tables in the form of nested JSON purchase order.

Example 5-7 GET Handler Code using Oracle REST Data Services Query on Relational Tables for Generating a Nested JSON object

```
SELECT 'application/json', json_object('PONumber' VALUE po.PONo,
       'Requestor' VALUE po.Requestor,
       'CostCenter' VALUE po.CostCenter,
       'Address' VALUE
           json_object('street' VALUE po.AddressStreet,
                 'city' VALUE po.AddressCity,
                 'state' VALUE po.AddressState,
                 'zipCode' VALUE po.AddressZip,
                 'country' VALUE po.AddressCountry),
        'LineItems' VALUE (select json_arrayagg(
            json_object('ItemNumber' VALUE li.ItemNumber,
                   'Part' VALUE
                     json_object('Description' VALUE li.PartDescription,
                                 'UnitPrice' VALUE li.PartUnitPrice,
                                 'UPCCode' VALUE li.PartUPCCODE),
                   'Quantity' VALUE li.Quantity))
```

```
                    FROM LineItem li WHERE po.PONo = li.PONo))
               FROM PurchaseOrder po
                 WHERE po.PONo = :id
```

PL/SQL API Calls for Defining Template and GET Handler

This section provides an example of Oracle REST Data Services PL/SQL API call for creating a new template in the module created.

Example 5-8 PL/SQL API Call for Creating a New `test/:id` Template and GET Handler in the `demo` Module

```
Begin
ords.define_template(
 p_module_name => 'demo',
 p_pattern => 'test/:id');

ords.define_handler(
 p_module_name => 'demo',
 p_pattern => 'test/:id',
 p_method  => 'GET',
 p_source_type => ords.source_type_media,
 p_source => '

   SELECT ''application/json'', json_object(''PONumber'' VALUE po.PONo,
          ''Requestor'' VALUE po.Requestor,
          ''CostCenter'' VALUE po.CostCenter,
          ''Address'' VALUE
              json_object(''street'' VALUE po.AddressStreet,
                          ''city'' VALUE po.AddressCity,
                          ''state'' VALUE po.AddressState,
                          ''zipCode'' VALUE po.AddressZip,
                          ''country'' VALUE po.AddressCountry),
          ''LineItems'' VALUE (select json_arrayagg(
              json_object(''ItemNumber'' VALUE li.ItemNumber,
                     ''Part'' VALUE
                       json_object(''Description'' VALUE
li.PartDescription,
                                   ''UnitPrice'' VALUE li.PartUnitPrice,
                                   ''UPCCode'' VALUE li.PartUPCCODE),
                     ''Quantity'' VALUE li.Quantity))
                     FROM LineItem li WHERE po.PONo = li.PONo))
                FROM PurchaseOrder po
                  WHERE po.PONo = :id '
   );

Commit;
End;
```

Testing the RESTful Services

This section shows how to test the **POST** and **GET** RESTful Services to access the Oracle database and get the results in a JSON format.

This section includes the following topics:

- Insertion of JSON Object into the Database
- Generating JSON Object from the Database

Insertion of JSON Object into the Database

This section shows how to test insertion of JSON purchase order into the database.

URI Pattern: `http://<HOST>:<PORT>/ords/<SchemaAlias>/<module>/<template>`

Example:

Method: POST

URI Pattern: `http://localhost:8080/ords/ordstest/demo/test/`

To test the RESTful service, create a file such as `po1.json` with the following data for PONumber 1608 :

```
{"PONumber"        : 1608,
 "Requestor"       : "Alexis Bull",
 "CostCenter"      : "A50",
 "Address"         : {"street"  : "200 Sporting Green",
                                "city"    : "South San Francisco",
                                "state"   : "CA",
                                "zipCode" : 99236,
                                "country" : "United States of
America"},
      "LineItems"  : [ {"ItemNumber" : 1,
                           "Part"     : {"Description" : "One Magic
Christmas",
                                         "UnitPrice"   : 19.95,
                                         "UPCCode"     : 1313109289},
                                         "Quantity"   : 9.0},
                                         {"ItemNumber" : 2,
                                           "Part"       :
{"Description" : "Lethal Weapon",

"UnitPrice"   : 19.95,

"UPCCode"     : 8539162892},
                                           "Quantity"   : 5.0}]}'
```

Then, execute the following cURL command in the command prompt:

```
curl -i -H "Content-Type: application/json" -X POST -d @po1.json "http://
localhost:8080/ords/ordstest/demo/test/"
```

The cURL command returns the following response:

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
```

Generating JSON Object from the Database

This section shows the results of a GET method to fetch the JSON object from the database..

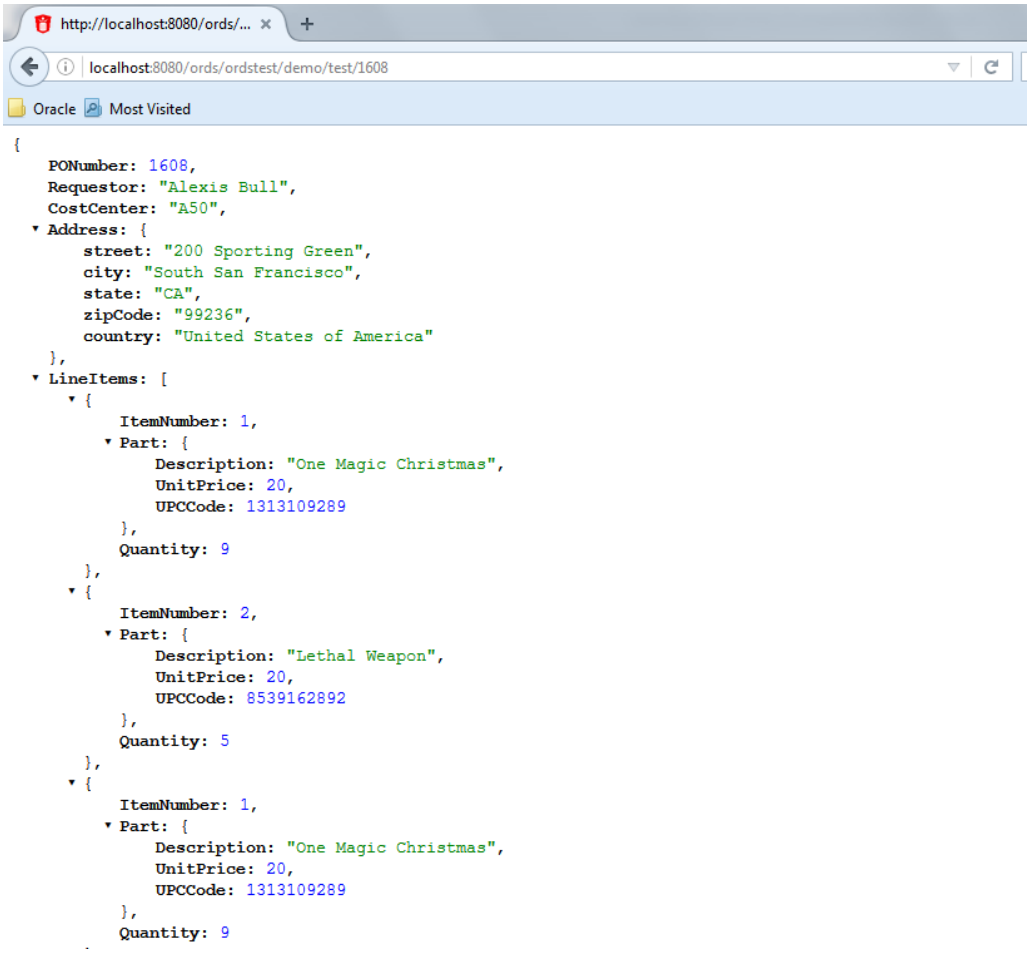
Method: `GET`

URI Pattern: `http://<HOST>:<PORT>/ords/<SchemaAlias>/<module>/<template>/<parameters>`

Example:

To test the RESTful service, in a web browser, enter the URL http://localhost:8080 / ords/ordstest/demo/test/1608 as shown in the following figure:

Figure 5-12 Generating Nested JSON Objects



About Working with Dates Using Oracle REST Data Services

Oracle REST Data Services enables developers to create REST interfaces to Oracle Database, Oracle Database 12c JSON Document Store as quickly and easily as possible. When working with Oracle Database, developers can use the AutoREST feature for tables or write custom modules using SQL and PL/SQL routines for more complex operations.

Oracle REST Data Services uses the RFC3339 standard for encoding dates in strings. Typically, the date format used is dd-mmm-yyyy, for example, 15-Jan-2017. Oracle REST Data Services automatically converts JSON strings in the specified format to Oracle date data types when performing operations such as inserting or updating

values in Oracle Database. When converting back to JSON strings, Oracle REST Data Services automatically converts Oracle date data types to the string format.

> **Note:**
>
> Oracle Database supports a date data type while JSON does not support a date data type.

This section includes the following topics:

- About Datetime Handling with Oracle REST Data Services
- About Setting the Time Zone

> **See Also:**
>
> The following page for more information, including details on how time and time zones are handled jsao_io_dates

About Datetime Handling with Oracle REST Data Services

As data arrives from a REST request, Oracle REST Data Services may parse ISO 8601 strings and convert them to the `TIMESTAMP` data type in Oracle Database. This occurs with AutoREST (`POST` and `PUT`) as well as with bind variables in custom modules. Remember that `TIMESTAMP` does not support time zone related components, so the `DATETIME` value is set to the time zone Oracle REST Data Services uses during the conversion process.

When constructing responses to REST requests, Oracle REST Data Services converts `DATETIME` values in Oracle Database to ISO 8601 strings in Zulu. This occurs with AutoREST (`GET`) and in custom modules that are mapped to SQL queries (GET). In the case of `DATE` and `TIMESTAMP` data types, which do not have time zone related components, the time zone is assumed to be that in which Oracle REST Data Services is running and the conversion to Zulu is made from there.

Here are some general recommendations when working with Oracle REST Data Services for REST (that is, not APEX):

- Ensure that Oracle REST Data Services uses the appropriate time zone as per the data in the database (for example, the time zone you want dates going into the database).
- Do not alter NLS settings (that is, the time_zone) mid-stream.

Note that while ISO 8601 strings are mentioned, Oracle REST Data Services actually supports strings. RFC3339 strings are a conformant subset of ISO 8601 strings. The default format returned by `JSON.stringify(date)` is supported.

> **WARNING:**
>
> It is important to keep the time zone that Oracle REST Data Services uses in sync with the session time zone to prevent issues with implicit data conversion to `TIMESTAMP WITH TIME ZONE` or `TIMESTAMP WITH LOCAL TIME ZONE`. Oracle REST Data Services does this automatically by default but developers can change the session time zone with an `ALTER SESSION` statement.

> **See Aslo:**
>
> rfc3339_date_time_format

About Setting the Time Zone

When Oracle REST Data Services is started, the JVM it runs in obtains and caches the time zone Oracle REST Data Services uses for various time zone conversions. By default, the time zone is obtained from the operating system (OS), so an easy way to change the time zone Oracle REST Data Services uses is to change the time zone of the OS and then restart Oracle REST Data Services or the application server on which it is running. Of course, the instructions for changing the time zone vary by the operating system.

If for any reason you do not want to use the same time zone as the OS, it is possible to override the default using the Java environment variable `Duser.timezone`. Exactly how that variable is set depends on whether you are running in standalone mode or in a Java application server. The following topics show some examples.

Standalone Mode

When running Oracle REST Data Services in standalone mode, it is possible to set Java environment variables by specifying them as command line options before the `-jar` option.

Example 5-9 Setting the Duser.timezone Java Environment Variable in Standalone Mode

The following code example shows how to set the timezone in standalone mode on the command line.

```
$ java -Duser.timezone=America/New_York -jar ords.war standalone
```

Java Application Server — Tomcat 8

In a Java application server, Tomcat 8, and possibly earlier and later versions too, it is possible to set the time zone using the environment variable `CATALINA_OPTS`. The recommended way to do this is not to modify the `CATALINA_BASE/bin/catalina.sh` directly, but instead to set environment variables by creating a script named `setenv.sh` in `CATALINA_BASE/bin`.

Example 5-10 Setting the Duser.timezone Java Environment Variable in a Java Application Server

The following code example shows the contents of the `setenv.sh` script for setting the timezone in a Java Application server — Tomcat 8.

```
CATALINA_TIMEZONE="-Duser.timezone=America/New_York"
CATALINA_OPTS="$CATALINA_OPTS $CATALINA_TIMEZONE
```

Exploring the Sample RESTful Services in Application Express (Tutorial)

Oracle highly recommends to develop Oracle REST Data Services application using SQL Developer because it supports the most recent Oracle REST Data Services releases, that is, 3.0.X. Application Express provides a tutorial that is useful for learning some basic concepts of REST and Oracle REST Data Services. However, the tutorial uses the earlier Oracle REST Data Services releases, that is, 2.0.X. Following are some of the useful tips discussed on how to use the tutorial:

If your Application Express instance is configured to automatically add the sample application and sample database objects to workspaces, then a sample resource module named: `oracle.example.hr` will be visible in the list of Resource Modules. If that resource module is not listed, then you can click the **Reset Sample Data** task on the right side of the RESTful Services Page to create the sample resource module.

1. Click on `oracle.example.hr` to view the Resource Templates and Resource Handlers defined within the module. Note how the module has a URI prefix with the value: `hr/`. This means that all URIs serviced by this module will start with the characters `hr/`.
2. Click on the resource template named `employees/{id}`. Note how the template has a URI Template with the value: `employees/{id}`. This means that all URIs starting with `hr/employees/` will be serviced by this Resource Template.

   The HTTP methods supported by a resource template are listed under the resource template. In this case, the only supported method is the `GET` method.
3. Click on the `GET` Resource Handler for `hr/employees/{id}` to view its configuration.

   The **Source Type** for this handler is `Query One Row`. This means that the resource is expected to be mapped to a single row in the query result set. The Source for this handler is:

   ```
   select * from emp
            where empno = :id
   ```

   Assuming that the `empno` column is unique, the query should only produce a single result (or no result at all if no match is found for `:id`). To try it out, press the **Test** button. The following error message should be displayed:

   400 - Bad Request - Request path contains unbound parameters: id

   If you look at the URI displayed in the browser, it will look something like this:

   ```
   https://server:port/ords/workspace/hr/employees/{id}
   ```

   where:

- `server` is the DNS name of the server where Oracle Application Express is deployed
- `port` is the port the server is listening on
- `workspace` is the name of the Oracle Application Express workspace you are logged into

Note the final part of the URI: `hr/employees/{id}`. The error message says that this is not a valid URI, the problem is that you did not substitute in a concrete value for the parameter named `{id}`. To fix that, press the browser **Back** button, then click **Set Bind Variables**.

4. For the bind variable named `:id`, enter the value `7369`, and press **Test**.

   A new browser window appears displaying the following JSON (JavaScript Object Notation):

   ```
   {
    "empno":7369,
    "ename":"SMITH",
    "job":"CLERK",
    "mgr":7902,
    "hiredate":"1980-12-17T08:00:00Z",
    "sal":800,
    "deptno":20
   }
   ```

   Note also the URI displayed in the browser for this resource:

   ```
   https://server:port/ords/workspace/hr/employees/7369
   ```

   The `{id}` URI Template parameter is bound to the SQL `:id` bind variable, and in this case it has been given the concrete value of `7369`, so the query executed by the RESTful Service becomes:

   ```
   select * from emp
            where empno = 7369
   ```

   The results of this query are then rendered as JSON as shown above.

   > **Tip:**
   >
   > Reading JSON can be difficult. To make it easier to read, install a browser extension that *pretty prints* the JSON. For example, Mozilla Firefox and Google Chrome both have extensions:
   >
   > - jsonview_firefox
   > - json_formatter_chrome

   Now see what happens when you enter the URI of a resource that does not exist.

5. On the Set Bind Variables page, change the value of `:id` from `7369` to `1111`, and press **Test**.

   As before, a new window pops up, but instead of displaying a JSON resource, it displays an error message reading:

   ```
   404 - Not Found
   ```

This is the expected behavior of this handler: when a value is bound to `:id` that does not exist in the `emp` table, the query produces no results and consequently the standard HTTP Status Code of `404 - Not Found` is returned.

So, you have a service that will provide information about individual employees, if you know the ID of an employee, but how do you discover the set of valid employee ids?

6. Press **Cancel** to return to the previous page displaying the contents of the Resource Module.
7. Click on the template named `employees/`.

   The following steps look at the resource it generates, and later text will help you understand its logic.
8. Click on the GET handler beneath `employees/`, and click **Test**.

   A resource similar to the following is displayed (If you haven't already done so, now would be a good time to install a JSON viewer extension in your browser to make it easier to view the output):

```
{
 "next":
  {"$ref":
    "https://server:port/ords/workspace/hr/employees/?page=1"},
 "items": [
  {
   "uri":
    {"$ref":
      "https://server:port/ords/workspace/hr/employees/7369"},
   "empno": 7369,
   "ename": "SMITH"
  },
  {
   "uri":
    {"$ref":
      "https://server:port/ords/workspace/hr/employees/7499"},
   "empno": 7499,
   "ename": "ALLEN"
  },
  ...
  {
   "uri":
    {"$ref":
      "https://server:port/ords/workspace/hr/employees/7782"},
   "empno": 7782,
   "ename": "CLARK"
  }
 ]
}
```

   This JSON document contains a number of things worth noting:

   - The first element in the document is named `next` and is a URI pointing to the next page of results. (An explanation of how paginated results are supported appears in later steps)
   - The second element is named `items` and contains a number of child elements. Each child element corresponds to a row in the result set generated by the query.

- The first element of each child element is named `uri` and contains a URI pointing to the service that provides details of each employee. Note how the latter part of the URI matches the URI Template: `employees/{id}`. In other words, if a client accesses any of these URIs, the request will be serviced by the `employees/{id}` RESTful service previously discussed.

So, this service addresses the problem of identifying valid employee IDs by generating a resource that lists all valid employee resources. The key thing to realize here is that it does not do this by just listing the ID value by itself and expecting the client to be able to take the ID and combine it with prior knowledge of the `employees/{id}` service to produce an employee URI; instead, it lists the URIs of each employee.

Because the list of valid employees may be large, the service also breaks the list into smaller pages, and again uses a URI to tell the client where to find the next page in the results.

To see at how this service is implemented, continue with the next steps.

9. Press the **Back** button in your browser to return to the `GET` handler definition.

Note the Source Type is `Query`, this is the default Source Type, and indicates that the resource can contain zero or more results. The Pagination Size is `7`, which means that there will be seven items on each page of the results. Finally, the Source for the handler looks like this:

```
select empno "$uri", empno, ename from (
 select emp.*,
        row_number() over (order by empno) rn
        from emp
 ) tmp
 where
  rn between :row_offset and :row_count
```

In this query:

- The first line states that you want to return three columns. The first column is the employee id: `empno`, but aliased to a column name of `$uri` (to be explained later), the second column is again the employee ID, and the third column is the employee name, `ename`.
- Columns in result sets whose first character is $ (dollar sign) are given special treatment. They are assumed to denote columns that must be transformed into URIs, and these are called Hyperlink Columns. Thus, naming columns with a leading $ is a way to generate hyperlinks in resources.

  When a Hyperlink Column is encountered, its value is prepended with the URI of the resource in which the column is being rendered, to produce a new URI. For example, recall that the URI of this service is `https://server:port/ords/workspace/hr/employees/`. If the value of `empno` in the first row produced by the this service's query is `7369`, then the value of `$uri` becomes: `https://server:port/ords/workspace/hr/employees/7369`.
- JSON does not have a URI data type, so a convention is needed to make it clear to clients that a particular value represents a URI. Oracle REST Data Services uses the JSON Reference proposal, which states that any JSON object containing a member named `$ref`, and whose value is a string, is a URI. Thus, the column: `$uri` and its value: `https://server:port/ords/workspace/hr/employees/7369` is transformed to the following JSON object:

```
{"uri":
     {"$ref":
      "https://server:port/ords/workspace/hr/employees/7369"
     }
    }
```

- The inner query uses the `row_number()` analytical function to count the number of rows in the result set, and the outer WHERE clause constrains the result set to only return rows falling within the desired page of results. Oracle REST Data Services defines two implicit bind parameters, `:row_offset` and `:row_count`, that always contain the indicies of the first and last rows that should be returned in a given page's results.

  For example, if the current page is the first page and the pagination size is `7`, then the value of `:row_offset` will be `1` and the value of `:row_count` will be `7`.

To see a simpler way to do both hyperlinks and paged results, continue with the following steps.

10. Click on the `GET` handler of the `employeesfeed/` resource template.

    Note that the Source Type of this handler is `Feed` and Pagination Size is `25`.

11. Change the pagination size to `7`, and click **Apply Changes**.

    The Source of the handler is just the following:

    ```
    select empno, ename from emp
                       order by deptno, ename
    ```

    As you can see, the query is much simpler than the previous example; however, if you click **Test**, you will see a result that is very similar to the result produced by the previous example.

    - The `Feed` Source Type is an enhanced version of the `Query` Source Type that automatically assumes the first column in a result set should be turned into a hyperlink, eliminating the need to alias columns with a name starting with `$`. In this example, the `empno` column is automatically transformed into a hyperlink by the `Feed` Source Type.
    - This example demonstrates the ability of Oracle REST Data Services to automatically paginate result sets if a Pagination Size of greater than zero is defined, and the query does *not* explicitly dereference the `:row_offset` or `:row_count` bind parameters. Because both these conditions hold true for this example, Oracle REST Data Services enhances the query, wrapping it in clauses to count and constrain the number and offset of rows returned. Note that this ability to automatically paginate results also applies to the `Query` Source Type.

> **See Also:**
>
> json_ref

Configuring Secure Access to RESTful Services

This section describes how to configure secure access to RESTful Services

RESTful APIs consist of resources, each resource having a unique URI. A set of resources can be protected by a privilege. A privilege defines the set of roles, at least one of which an authenticated user must possess to access a resource protected by a privilege.

Configuring a resource to be protected by a particular privilege requires creating a privilege mapping. A privilege mapping defines a set of patterns that identifies the resources that a privilege protects.

Topics:

- Authentication
- About Privileges for Accessing Resources
- About Users and Roles for Accessing Resources
- About the File-Based User Repository
- Tutorial: Protecting and Accessing Resources

Authentication

Users can be authenticated through first party cookie-based authentication or third party OAuth 2.0-based authentication

Topics:

- First Party Cookie-Based Authentication
- Third Party OAuth 2.0-Based Authentication

First Party Cookie-Based Authentication

A first party is the author of a RESTful API. A first party application is a web application deployed on the same web origin as the RESTful API. A first party application is able to authenticate and authorize itself to the RESTful API using the same cookie session that the web application is using. The first party application has full access to the RESTful API.

Third Party OAuth 2.0-Based Authentication

A third party is any party other than the author of a RESTful API. A third party application cannot be trusted in the same way as a first party application; therefore, there must be a mediated means to selectively grant the third party application limited access to the RESTful API.

The OAuth 2.0 protocol defines flows to provide conditional and limited access to a RESTful API. In short, the third party application must first be registered with the first party, and then the first party (or an end user of the first party RESTful service) approves the third party application for limited access to the RESTful API, by issuing the third party application a short-lived access token.

> **See Also:**
>
> `ietf_rfc6749`

Two-Legged and Three-Legged OAuth Flows

Some flows in OAuth are defined as two-legged and others as three-legged.

Two-legged OAuth flows involve two parties: the party calling the RESTful API (the third party application), and the party providing the RESTful API. Two-legged flows are used in server to server interactions where an end user does not need to approve access to the RESTful API. In OAuth 2.0 this flow is called the client credentials flow. It is most typically used in business to business scenarios.

Three-legged OAuth flows involve three parties: the party calling the RESTful API, the party providing the RESTful API, and an end user party that owns or manages the data to which the RESTful API provides access. Three-legged flows are used in client to server interactions where an end user must approve access to the RESTful API. In OAuth 2.0 the authorization code flow and the implicit flow are three-legged flows. These flows are typically used in business to consumer scenarios.

For resources protected by three-legged flows, when an OAuth client is registering with a RESTful API, it can safely indicate the protected resources that it requires access to, and the end user has the final approval decision about whether to grant the client access. However for resources protected by two-legged flows, the owner of the RESTful API must approve which resources each client is authorized to access.

About Privileges for Accessing Resources

A privilege for accessing resources consists of the following data:

- Name: The unique identifier for the Privilege. This value is required.
- Label: The name of the privilege presented to an end user when the user is being asked to approve access to a privilege when using OAuth. This value is required if the privilege is used with a three-legged OAuth flow.
- Description: A description of the purpose of the privilege. It is also presented to the end user when the user is being asked to approve access to a privilege. This value is required if the privilege is used with a three-legged OAuth flow.
- Roles: A set of role names associated with the privilege. An authenticated party must have at least one of the specified roles in order to be authorised to access resources protected by the privilege. A value is required, although it may be an empty set, which indicates that a user must be authenticated but that no specific role is required to access the privilege.

For two-legged OAuth flows, the third party application (called a *client* in OAuth terminology) must possess at least one of the required roles.

For three-legged OAuth flows, the end user that approves the access request from the third party application must possess at least one of the required roles.

Related Topics

- Two-Legged and Three-Legged OAuth Flows

About Users and Roles for Accessing Resources

A privilege enumerates a set of roles, and users can possess roles. but where are these Roles defined? What about the users that possess these roles? Where are they defined?

A privilege enumerates a set of roles, and users can possess roles. Oracle REST Data Services delegates the task of user management to the application server on which Oracle REST Data Services is deployed. Oracle REST Data Services is able to authenticate users defined and managed by the application server and to identify the roles and groups to which the authenticated user belongs. It is the responsibility of the party deploying Oracle REST Data Services on an application server to also configure the user repository on the application server.

Because an application server can be configured in many ways to define a user repository or integrate with an existing user repository, this document cannot describe how to configure a user repository in an application server. See the application server documentation for detailed information.

About the File-Based User Repository

Oracle REST Data Services provides a a simple file-based user repository mechanism. However, this user repository is only intended for the purposes of demonstration and testing, and is not supported for production use.

See the command-line help for the user command for more information on how to create a user in this repository:

```
java -jar ords.war help user
```

Format:

```
java -jar ords.war user <user> <roles>
```

Arguments:

- `<user>` is the user ID of the user.
- `<roles>` is the list of roles (zero or more) that the user has.

Related Topics

- Tutorial: Protecting and Accessing Resources

Tutorial: Protecting and Accessing Resources

This tutorial demonstrates creating a privilege to protect a set of resources, and accessing the protected resource with the following OAuth features:

- Client credentials
- Authorization code
- Implicit flow

It also demonstrates access the resource using first-party cookie-based authentication.

Topics:

- OAuth Flows and When to Use Each
- Assumptions for This Tutorial
- Steps for This Tutorial

OAuth Flows and When to Use Each

This topic explains when to use various OAuth flow features.

Use *first party cookie-based authentication* when accessing a RESTful API from a web application hosted on the same origin as the RESTful API.

Use the *authorization code* flow when you need to permit third party web applications to access a RESTful API and the third party application has its own web server where it can keep its client credentials secure. This is the typical situation for most web applications, and it provides the most security and best user experience, because the third party application can use refresh tokens to extend the life of a user session without having to prompt the user to reauthorize the application.

Use the *implicit flow* when the third party application does not have a web server where it can keep its credentials secure. This flow is useful for third party single-page-based applications. Because refresh tokens cannot be issued in the Implicit flow, the user will be prompted more frequently to authorize the application.

Native mobile or desktop applications should use the authorization code or implicit flows. They will need to display the sign in and authorization prompts in a web browser view, and capture the access token from the web browser view at the end of the authorization process.

Use the *client credentials* flow when you need to give a third party application direct access to a RESTful API without requiring a user to approve access to the data managed by the RESTful API. The third party application must be a server-based application that can keep its credentials secret. The client credentials flow *must not* be used with a native application, because the client credentials can *always* be discovered in the native executable.

Assumptions for This Tutorial

This tutorial assumes the following:

- Oracle REST Data Services is deployed at the following URL: `https://example.com/ords/`
- A database schema named ORDSTEST has been enabled for use with Oracle REST Data Services, and its RESTful APIs are exposed under: `https://example.com/ords/ordstest/`
- The ORDSTEST schema contains a database table named EMP, which was created as follows:

```
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)
  );
```

- The resources to be protected are located under: `https://example.com/ords/ordstest/examples/employees/`

Steps for This Tutorial

Follow these steps to protect and access a set of resources.

1. **Enable the schema**. Connect to the ORDSTEST schema and execute the following PL/SQL statements;

```
begin
  ords.enable_schema;
  commit;
end;
```

2. **Create a resource.** Connect to the ORDSTEST schema and execute the following PL/SQL statements:

```
begin
ords.create_service(
      p_module_name => 'examples.employees' ,
      p_base_path  => '/examples/employees/',
      p_pattern =>  '.' ,
      p_items_per_page => 7,
      p_source  =>  'select * from emp order by empno desc');
commit;
end;
```

The preceding code creates the `/examples/employees/` resource, which you will protect with a privilege in a later step.

You can verify the resource by executing following cURL command:

```
curl -i https://example.com/ords/ordstest/examples/employees/
```

The result should be similar to the following (edited for readability):

```
Content-Type: application/json
Transfer-Encoding: chunked

{
 "items":
   [

{"empno":7934,"ename":"MILLER","job":"CLERK","mgr":7782,"hiredate":"1982-01-2
3T00:00:00Z","sal":1300,"comm":null,"deptno":10},
    ...
   ],
 "hasMore":true,
 "limit":7,
 "offset":0,
 "count":7,
 "links":
   [
    {"rel":"self","href":"https://example.com/ords/ordstest/examples/
employees/"},
    {"rel":"describedby","href":"https://example.com/ords/ordstest/metadata-
catalog/examples/employees/"},
    {"rel":"first","href":"https://example.com/ords/ordstest/examples/
employees/"},
    {"rel":"next","href":"https://example.com/ords/ordstest/examples/
employees/?offset=7"}
```

```
   ]
}
```

3. **Create a privilege.** While connected to the ORDSTEST schema, execute the following PL/SQL statements:

```
begin
  ords.create_role('HR Administrator');

  ords.create_privilege(
      p_name => 'example.employees',
      p_role_name => 'HR Administrator',
      p_label => 'Employee Data',
      p_description => 'Provide access to employee HR data');
  commit;
end;
```

The preceding code creates a role and a privilege, which belong to the ORDSTEST schema.

- The role name must be unique and must contain printable characters only.
- The privilege name must be unique and must conform to the syntax specified by the OAuth 2.0 specification, section 3.3 for scope names.
- Because you will want to use this privilege with the three-legged authorization code and implicit flows, you must provide a label and a description for the privilege. The label and description are presented to the end user during the approval phase of three-legged flows.
- The values should be plain text identifying the name and purpose of the privilege.

You can verify that the privilege was created correctly by querying the USER_ORDS_PRIVILEGES view.

```
select id,name from user_ords_privileges where name = 'example.employees';
```

The result should be similar to the following:

```
ID
NAME



----- -----------------
10260 example.employees
```

The ID value will vary from database to database, but the NAME value should be as shown.

4. **Associate the privilege with resources.** While connected to the ORDSTEST schema, execute the following PL/SQL statements:

```
begin
 ords.create_privilege_mapping(
      p_privilege_name => 'example.employees',
      p_pattern => '/examples/employees/*');
  commit;
end;
```

The preceding code associates the `example.employees` privilege with the resource pattern `/examples/employees/`.

You can verify that the privilege was created correctly by querying the USER_ORDS_PRIVILEGE_MAPPINGS view.

```
select privilege_id, name, pattern from user_ords_privilege_mappings;
```

The result should be similar to the following:

```
PRIVILEGE_ID NAME                 PATTERN
------------ -------------------- --------------------
10260        example.employees    /examples/employees/*
```

The PRIVILEGE_ID value will vary from database to database, but the NAME and PATTERN values should be as shown.

You can confirm that the `/examples/employees/` resource is now protected by the `example.employees` privilege by executing the following cURL command:

```
curl -i https://example.com/ords/ordstest/examples/employees/
```

The result should be similar to the following (reformatted for readability):

```
HTTP/1.1 401 Unauthorized
Content-Type: text/html
Transfer-Encoding: chunked

<!DOCTYPE html>
<html>
...
</html>
```

You can confirm that the protected resource can be accessed through first party authentication, as follows.

a. **Create an end user.** Create a test user with the HR Administrator role, required to access the `examples.employees` privilege using the file-based user repository. Execute the following command at a command prompt

   ```
   java -jar ords.war user "hr_admin" "HR Administrator"
   ```

   When prompted for the password, enter and confirm it.

b. **Sign in as the end user.** Enter the following URL in a web browser:

   ```
   https://example.com/ords/ordstest/examples/employees/
   ```

   On the page indicating that access is denied, click the link to sign in.

   Enter the credentials registered for the HR_ADMIN user, and click Sign In.

   Confirm that the page redirects to `https://example.com/ords/ordstest/examples/employees/` and that the JSON document is displayed.

5. **Register the OAuth client.** While connected to the ORDSTEST schema, execute the following PL/SQL statements:

```
begin
 oauth.create_client(
      p_name => 'Client Credentials Example',
      p_grant_type => 'client_credentials',
      p_privilege_names => 'example.employees',
      p_support_email => 'support@example.com');
 commit;
end;
```

The preceding code registers a client named `Client Credentials Example`, to access the `examples.employees` privilege using the client credentials OAuth flow.

You can verify that the client was registered and has requested access to the `examples.employees` privilege by executing the following SQL statement:

```
select client_id,client_secret from user_ords_clients where name = 'Client
Credentials Example';
```

The result should be similar to the following:

```
CLIENT_ID                      CLIENT_SECRET
------------------------------ ------------------------
o_CZBVkEMN23tTB-IddQsQ..       4BJXceufbmTki-vruYNLIg..
```

The CLIENT_ID and CLIENT_SECRET values represent the secret credentials for the OAuth client. These values must be noted and kept secure. You can think of them as the userid and password for the client application.

6. **Grant the OAuth client a required role.** While connected to the ORDSTEST schema, execute the following PL/SQL statements:

```
begin
 oauth.grant_client_role(
     'Client Credentials Example',
     'HR Administrator');
 commit;
end;
```

The preceding code registers a client named `Client Credentials Example`, to access the `examples.employees` privilege using the client credentials OAuth flow.

You can verify that the client was granted the role by executing the following SQL statement:

```
select * from user_ords_client_roles where client_name = 'Client Credentials
Example';
```

The result should be similar to the following:

```
 CLIENT_ID CLIENT_NAME                ROLE_ID   ROLE_NAME
---------- -------------------------- --------  ---------------------
     10286 Client Credentials Example    10222   HR Administrator
```

7. **Obtain an OAuth access token using client credentials.**

The OAuth protocol specifies the HTTP request that must be used to create an access token using the client credentials flow[rfc6749-4.4.].

The request must be made to a well known URL, called the token endpoint. For Oracle REST Data Services the path of the token endpoint is always oauth/token, relative to the root path of the schema being accessed. The token endpoint for this example is:

```
https://example.com/ords/ordstest/oauth/token
```

Execute the following cURL command:

```
curl -i --user clientId:clientSecret --data "grant_type=client_credentials"
https://example.com/ords/ordstest/oauth/token
```

In the preceding command, replace `clientId` with the CLIENT_ID value in USER_ORDS_CLIENTS for `Client Credentials Example`, and replace

clientSecret with the CLIENT_SECRET value shown in USER_ORDS_CLIENTS for Client Credentials Example. The output should be similar to the following:

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

{
 "access_token": "2YotnFZFEjr1zCsicMWpAA",
 "token_type":   "bearer",
 "expires_in":3600
}
```

In the preceding output, the access token is of type bearer, and the value is specified by the access_token field. This value will be different for every request. The expires_in value indicates the number of seconds until the access token expires; in this case the token will expire in one hour (3600 seconds).

8. **Access a protected resource using the access token.** Execute the following cURL command:

```
curl -i -H"Authorization: Bearer accessToken" https://example.com/ords/
ordstest/examples/employees/
```

In the preceding command, replace accessToken with the value of the access_token field shown in the preceding step. The output should be similar to the following:

```
Content-Type: application/json
Transfer-Encoding: chunked

{
 "items":
   [

{"empno":7934,"ename":"MILLER","job":"CLERK","mgr":7782,"hiredate":"1982-01-2
3T00:00:00Z","sal":1300,"comm":null,"deptno":10},
    ...
   ],
 "hasMore":true,
 "limit":7,
 "offset":0,
 "count":7,
 "links":
   [
    {"rel":"self","href":"https://example.com/ords/ordstest/examples/
employees/"},
    {"rel":"describedby","href":"https://example.com/ords/ordstest/metadata-
catalog/examples/employees/"},
    {"rel":"first","href":"https://example.com/ords/ordstest/examples/
employees/"},
    {"rel":"next","href":"https://example.com/ords/ordstest/examples/
employees/?offset=7"}
   ]
}
```

9. **Register the client for authorization code.** While connected to the ORDSTEST schema, execute the following PL/SQL statements:

```
begin
 oauth.create_client(
    p_name => 'Authorization Code Example',
    p_grant_type => 'authorization_code',
```

```
    p_owner => 'Example Inc.',
    p_description => 'Sample for demonstrating Authorization Code Flow',
    p_redirect_uri => 'http://example.org/auth/code/example/',
    p_support_email => 'support@example.org',
    p_support_uri => 'http://example.org/support',
    p_privilege_names => 'example.employees'
    );
 commit;
end;
```

The preceding code registers a client named `Authorization Code Example`, to access the `examples.employees` privilege using the authorization code OAuth flow. For an actual application, a URI must be provided to redirect back to with the authorization code, and a valid support email address must be supplied; however, this example uses fictitious data and the sample `example.org` web service.

You can verify that the client is now registered and has requested access to the `examples.employees` privilege by executing the following SQL statement:

```
select id, client_id, client_secret from user_ords_clients where name =
'Authorization Code Example';
```

The result should be similar to the following:

```
        ID CLIENT_ID                      CLIENT_SECRET
---------- ------------------------------ ------------------------------
     10060 IGHso4BRgrBC3Jwg0Vx_YQ.. GefAsWv8FJdMSB30Eg6lKw..
```

To grant access to the privilege, an end user must approve access. The CLIENT_ID and CLIENT_SECRET values represent the secret credentials for the OAuth client. These values must be noted and kept secure. You can think of them as the userid and password for the client application.

10. **Obtain an OAuth access token using an authorization code.** This major step involves several substeps. (You must have already created the HR_ADMIN end user in a previous step.)

    a. **Obtain an OAuth authorization code.**

       The end user must be prompted (via a web page) to sign in and approve access to the third party application. The third party application initiates this process by directing the user to the OAuth Authorization Endpoint. For Oracle REST Data Services, the path of the authorization endpoint is always `oauth/auth`, relative to the root path of the schema being accessed. The token endpoint for this example is:

       ```
       https://example.com/ords/ordstest/oauth/auth
       ```

       The OAuth 2.0 protocol specifies that the Authorization request URI must include certain parameters in the query string:

       The `response_type` parameter must have a value of `code`.

       The `client_id` parameter must contain the value of the applications client identifier. This is the `client_id` value determined in a previous step.

       The `state` parameter must contain a unique unguessable value. This value serves two purposes: it provides a way for the client application to uniquely identify each authorization request (and therefore associate any application specific state with the value; think of the value as the application's own session identifier); and it provides a means for the client application to protect

against Cross Site Request Forgery (CSRF) attacks. The `state` value will be returned in the redirect URI at the end of the authorization process. The client must confirm that the value belongs to an authorization request initiated by the application. If the client cannot validate the state value, then it should assume that the authorization request was initiated by an attacker and ignore the redirect.

To initiate the Authorization request enter the following URL in a web browser:

```
https://example.com/ords/ordstest/oauth/auth?
response_type=code&client_id=cliendId&state=uniqueRandomValue
```

In the preceding URI, replace `clientId` with the value of the CLIENT_ID column that was noted previously, and replace `uniqueRandromValue` with a unique unguessable value. The client application must remember this value and verify it against the `state` parameter returned as part of the redirect at the end of the authorization flow.

If the `client_id` is recognized, then a sign in prompt is displayed. Enter the credentials of the HR_ADMIN end user, and click Sign In; and on the next page click Approve to cause a redirect to redirect URI specified when the client was registered. The redirect URI will include the authorization code in the query string portion of the URI. It will also include the same `state` parameter value that the client provided at the start of the flow. The redirect URI will look like the following:

```
http://example.org/auth/code/example/?
code=D5doeTSIDgbxWiWkPl9UpA..&state=uniqueRandomValue
```

The client application must verify the value of the `state` parameter and then note the value of the `code` parameter, which will be used in to obtain an access token.

b. Obtain an OAuth access token.

After the third party application has an authorization code, it must exchange it for an access token. The third party application's server must make a HTTPS request to the Token Endpoint. You can mimic the server making this request by using a cURL command as in the following example:

```
curl --user clientId:clientSecret --data
"grant_type=authorization_code&code=authorizationCode" https://
example.com/ords/ordstest/oauth/token
```

In the preceding command, replace `clientId` with the value of the CLIENT_ID shown in USER_ORDS_CLIENTS for `Authorization Code Example`, replace `clientSecret` with the value of the CLIENT_SECRET shown in USER_ORDS_CLIENTS for `Authorization Code Example`, and replace `authorizationCode` with the value of the authorization code noted in a previous step (the value of the `code` parameter).

The result should be similar to the following:

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

{
 "access_token": "psIGSSEXSBQyib0hozNEdw..",
 "token_type":   "bearer",
 "expires_in":3600,
```

```
 "refresh_token": "aRMg7AdWPuDvnieHucfV3g.."
}
```

In the preceding result, the access token is specified by the `access_token` field, and a refresh token is specified by the `refresh_token` field. This refresh token value can be used to extend the user session without requiring the user to reauthorize the third party application.

c. Access a protected resource using the access token.

After the third party application has obtained an OAuth access token, it can use that access token to access the protected `/examples/employees/` resource:

```
curl -i -H"Authorization: Bearer accessToken" https://example.com/ords/
ordstest/examples/employees/
```

In the preceding command, `accessToken` with the value of the `access_token` field shown in a previous step.

The result should be similar to the following:

```
Content-Type: application/json
Transfer-Encoding: chunked

{
 "items":
   [

{"empno":7934,"ename":"MILLER","job":"CLERK","mgr":7782,"hiredate":"1982-
01-23T00:00:00Z","sal":1300,"comm":null,"deptno":10},
    ...
   ],
 "hasMore":true,
 "limit":7,
 "offset":0,
 "count":7,
 "links":
   [
    {"rel":"self","href":"https://example.com/ords/ordstest/examples/
employees/"},
    {"rel":"describedby","href":"https://example.com/ords/ordstest/
metadata-catalog/examples/employees/"},
    {"rel":"first","href":"https://example.com/ords/ordstest/examples/
employees/"},
    {"rel":"next","href":"https://example.com/ords/ordstest/examples/
employees/?offset=7"}
   ]
}
```

d. Extend the session using a refresh token.

At any time, the third party application can use the refresh token value to generate a new access token with a new lifetime. This enables the third party application to extend the user session at will. To do this, the third party application's server must make an HTTPS request to the Token Endpoint. You can mimic the server making this request by using a cURL command as in the following example:

```
curl --user clientId:clientSecret --data
“grant_type=refresh_token&refresh_token=refreshToken" https://
example.com/ords/ordstest/oauth/token
```

In the preceding command, replace `clientId` with the value of the CLIENT_ID shown in USER_ORDS_CLIENTS for `Client Credentials Client`, replace `clientSecret` with the value of the CLIENT_SECRET shown in USER_ORDS_CLIENTS for `Client Credentials Client`, and replace `refreshToken` with the value of `refresh_token` obtained in a previous step.

The result should be similar to the following:

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

{
 "access_token":  "psIGSSEXSBQyib0hozNEdw..",
 "token_type":    "bearer",
 "refresh_token": "aRMg7AdWPuDvnieHucfV3g..",
 "expires_in":    3600
}
```

In the preceding result, the access token is specified by the `access_token` field, a new refresh token is specified by the `refresh_token` field. This refresh token value can be used to extend the user session without requiring the user to reauthorize the third party application. (Note that the previous access token and refresh token are now invalid; the new values must be used instead.)

11. **Register the client for implicit flow.** While connected to the ORDSTEST schema, execute the following PL/SQL statements:

```
begin
 oauth.create_client(
    p_name => 'Implicit Example',
    p_grant_type => 'implicit',
    p_owner => 'Example Inc.',
    p_description => 'Sample for demonstrating Implicit Flow',
    p_redirect_uri => 'http://example.org/implicit/example/',
    p_support_email => 'support@example.org',
    p_support_uri => 'http://example.org/support',
    p_privilege_names => 'example.employees'
    );
 commit;
end;
```

The preceding code registers a client named `Implicit Example` to access the `examples.employees` privilege using the implicit OAuth flow. For an actual application, a URI must be provided to redirect back to with the authorization code, and a valid support email address must be supplied; however, this example uses fictitious data and the sample `example.org` web service.

You can verify that the client is now registered and has requested access to the `examples.employees` privilege by executing the following SQL statement:

```
select id, client_id, client_secret from user_ords_clients where name =
'Implicit Example';
```

The result should be similar to the following:

```
        ID CLIENT_ID CLIENT_SECRET
---------- ------------------------------ ------------------------------
     10062 7Qz--bNJpFpv8qsfNQpS1A..
```

To grant access to the privilege, an end user must approve access.

12. **Obtain an OAuth access token using implicit flow.** (You must have already created the HR_ADMIN end user in a previous step.)

    The end user must be prompted (via a web page) to sign in and approve access to the third party application. The third party application initiates this process by directing the user to the OAuth Authorization Endpoint. For Oracle REST Data Services, the path of the authorization endpoint is always `oauth/auth`, relative to the root path of the schema being accessed. The token endpoint for this example is:

    ```
    https://example.com/ords/ordstest/oauth/auth
    ```

    The OAuth 2.0 protocol specifies that the Authorization request URI must include certain parameters in the query string:

    The `response_type` parameter must have a value of `token`.

    The `client_id` parameter must contain the value of the applications client identifier. This is the `client_id` value determined in a previous step.

    The `state` parameter must contain a unique unguessable value. This value serves two purposes: it provides a way for the client application to uniquely identify each authorization request (and therefore associate any application specific state with the value; think of the value as the application's own session identifier); and it provides a means for the client application to protect against Cross Site Request Forgery (CSRF) attacks. The `state` value will be returned in the redirect URI at the end of the authorization process. The client must confirm that the value belongs to an authorization request initiated by the application. If the client cannot validate the state value, then it should assume that the authorization request was initiated by an attacker and ignore the redirect.

    To initiate the Authorization request enter the following URL in a web browser:

    ```
    https://example.com/ords/ordstest/oauth/auth?
    response_type=token&client_id=cliendId&state=uniqueRandomValue
    ```

    In the preceding URI, replace `clientId` with the value of the CLIENT_ID column that was noted previously, and replace `uniqueRandromValue` with a unique unguessable value. The client application must remember this value and verify it against the `state` parameter returned as part of the redirect at the end of the authorization flow.

    If the `client_id` is recognized, then a sign in prompt is displayed. Enter the credentials of the HR_ADMIN end user, and click Sign In; and on the next page click Approve to cause a redirect to redirect URI specified when the client was registered. The redirect URI will include the access token in the query string portion of the URI. It will also include the same `state` parameter value that the client provided at the start of the flow. The redirect URI will look like the following:

    ```
    http://example.org/auth/code/example/
    #access_token=D5doeTSIDgbxWiWkPl9UpA..&type=bearer&expires_in=3600&state=uniq
    ueRandomValue
    ```

    The client application must verify the value of the `state` parameter and then note the value of the access token.

13. **Access a protected resource using an access token.** Execute the following cURL command:

    ```
    curl -i -H "Authorization: Bearer accessToken" https://example.com/ords/
    ordstest/examples/employees/
    ```

In the preceding command, replace `accessToken` with the value of the `access_token` field shown in the preceding step. The output should be similar to the following:

```
Content-Type: application/json
Transfer-Encoding: chunked

{
 "items":
   [

{"empno":7934,"ename":"MILLER","job":"CLERK","mgr":7782,"hiredate":"1982-01-2
3T00:00:00Z","sal":1300,"comm":null,"deptno":10},
    ...
   ],
 "hasMore":true,
 "limit":7,
 "offset":0,
 "count":7,
 "links":
   [
    {"rel":"self","href":"https://example.com/ords/ordstest/examples/
employees/"},
    {"rel":"describedby","href":"https://example.com/ords/ordstest/metadata-
catalog/examples/employees/"},
    {"rel":"first","href":"https://example.com/ords/ordstest/examples/
employees/"},
    {"rel":"next","href":"https://example.com/ords/ordstest/examples/
employees/?offset=7"}
   ]
}
```

Related Topics

- Using the Oracle REST Data Services PL/SQL API

About Oracle REST Data Services User Roles

Oracle REST Data Services defines a small number of predefined user roles:

- `RESTful Services` - This is the default role associated with a protected RESTful service.
- `OAuth2 Client Developer` - Users who want to register OAuth 2.0 applications must have this role.
- `oracle.dbtools.autorest.any.schema` - Users who want to access all AutoREST services.
- `SQL Developer` - Users who want to use Oracle SQL Developer to develop RESTful services must have this role.
- `SODA Developer` - This is the default role that is required to access the SODA REST API. For more information about this role, see *Oracle REST Data Services SODA for REST Developer's Guide*.
- `Listener Administrator` - Users who want to administrate an Oracle REST Data Services instance through Oracle SQL Developer must have this role. Typically, only users created through the `java -jar ords.war user` command will have this role.

Because the `Listener Administrator` role enables a user to configure an Oracle REST Data Services instance, and therefore has the capability to affect all Application Express workspaces served through that instance, Application Express users are not permitted to acquire the `Listener Administrator` role.

Topics:

- About Oracle Application Express Users and Oracle REST Data Services Roles
- Controlling RESTful Service Access with Roles

About Oracle Application Express Users and Oracle REST Data Services Roles

By default, Oracle Application Express users do not have any of the Oracle REST Data Services predefined user roles. This means that, by default, Application Express users cannot:

- Invoke protected RESTful Services
- Register OAuth 2.0 applications
- Use Oracle SQL Developer to develop RESTful services.

This applies to all Application Express users, including Application Express developers and administrators. It is therefore important to remember to follow the steps below to add Application Express users to the appropriate user groups, so that they can successfully perform the above actions.

Topics:

- Granting Application Express Users Oracle REST Data Services Roles
- Automatically Granting Application Express Users Oracle REST Data Services Roles

Granting Application Express Users Oracle REST Data Services Roles

To give an Application Express User any of the roles above, the user must be added to the equivalent Application Express user group. For example, to give the `RESTEASY_ADMIN` user the `RESTful Services` role, follow these steps:

1. Log in to the `RESTEASY` workspace as a `RESTEASY_ADMIN`.
2. Navigate to **Administration** and then **Manage Users and Groups**.
3. Click the Edit icon to the left of the `RESTEASY_ADMIN` user.
4. For **User Groups**, select `RESTful Services`.
5. Click **Apply Changes**.

Automatically Granting Application Express Users Oracle REST Data Services Roles

Adding Application Express users to the appropriate user groups can be an easily overlooked step, or can become a repetitive task if there are many users to be managed.

To address these issues, you can configure Oracle REST Data Services to automatically grant Application Express users a predefined set of RESTful Service roles by modifying the `defaults.xml` configuration file.

In that file, Oracle REST Data Services defines three property settings to configure roles:

- `apex.security.user.roles` - A comma separated list of roles to grant ordinary users, that is, users who are not developers or administrators.
- `apex.security.developer.roles` - A comma separated list of roles to grant users who have the `Developer` account privilege. `Developers` also inherit any roles defined by the `apex.security.user.roles` setting.
- `apex.security.administrator.roles` - A comma separated list of roles to grant users who have the `Administrator` account privilege. `Administrators` also inherit any roles defined by the `apex.security.user.roles` and `apex.security.developer.roles` settings.

For example, to automatically give all users the `RESTful Services` privilege and all developers and administrators the `OAuth2 Client Developer` and `SQL Developer` roles, add the following to the `defaults.xml` configuration file:

```
<!-- Grant all Application Express Users the ability
        to invoke protected RESTful Services -->
<entry key="apex.security.user.roles">RESTful Services</entry>
<!-- Grant Application Express Developers and Administrators the ability
        to register OAuth 2.0 applications and use Oracle SQL Developer
        to define RESTful Services -->
<entry key="apex.security.developer.roles">
   OAuth2 Client Developer, SQL Developer</entry>
```

Oracle REST Data Services must be restarted after you make any changes to the `defaults.xml` configuration file.

Controlling RESTful Service Access with Roles

The built-in `RESTful Service` role is a useful default for identifying users permitted to access protected RESTful services.

However, it will often also be necessary to define finer-grained roles to limit the set of users who may access a specific RESTful service.

Topics:

- About Defining RESTful Service Roles
- Associating Roles with RESTful Privileges

About Defining RESTful Service Roles

A RESTful Service **role** is an Application Express user group. To create a user group to control access to the Gallery RESTful Service, follow these steps. (

1. Log in to the `RESTEASY` workspace as a workspace administrator.
2. Navigate to **Administration** and then **Manage Users and Groups**.
3. Click the **Groups** tab.

4. Click **Create User Group**.
5. For **Name**, enter `Gallery Users`.
6. Click **Create Group**.

Associating Roles with RESTful Privileges

After a user group has been created, it can be associated with a RESTful privilege. To associate the Gallery Users role with the `example.gallery` privilege, follow these steps.

1. Navigate to **SQL Workshop** and then **RESTful Services**.
2. In the Tasks section, click **RESTful Service Privileges**.
3. Click **Gallery Access**.
4. For **Assigned Groups**, select `Gallery Users`.
5. Click **Apply Changes**.

With these changes, users must have the Gallery Users role to be able to access the Gallery RESTful service.

> **See Also:**
>
> The steps here use the image gallery application in Creating an Image Gallery as an example.

Authenticating Against WebLogic Server User Repositories

Oracle REST Data Services can use APIs provided by WebLogic Server to verify credentials (username and password) and to retrieve the set of groups and roles that the user is a member of.

This section walks through creating a user in the built-in user repositories provided by WebLogic Server, and verifying the ability to authenticate against that user.

This document does not describe how to integrate WebLogic Server with the many popular user repository systems such as LDAP repositories, but Oracle REST Data Services can authenticate against such repositories after WebLogic Server has been correctly configured. See your application server documentation for more information on what user repositories are supported by the application server and how to configure access to these repositories.

Topics:

- Authenticating Against WebLogic Server

Authenticating Against WebLogic Server

Authenticating a user against WebLogic Server involves the following major steps:

1. Creating a WebLogic Server User
2. Verifying the WebLogic Server User

Creating a WebLogic Server User

To create a sample WebLogic Server user, follow these steps:

1. Start WebLogic Server if it is not already running
2. Access the WebLogic Server Administration Console (typically `http://server:7001/console`), enter your credentials.
3. In the navigation tree on the left, click the **Security Realms** node
4. If a security realm already exists, go to the next step. If a security realm does not exist, create one as follows:
   a. Click **New**.
   b. For **Name**, enter `Test-Realm`, then click **OK**.
   c. Click **Test-Realm**.
   d. Click the **Providers** tab.
   e. Click **New**, and enter the following information:

      Name: `test-authenticator`

      Type: `DefaultAuthenticator`
   f. Restart WebLogic Server if you are warned that a restart is necessary.
   g. Click **Test-Realm**.
5. Click the **Users and Groups** tab.
6. Click **New**, and enter the following information:
   - **Name**: `3rdparty_dev2`
   - **Password**: Enter and confirm the desired password for this user.
7. Click **OK**.
8. Click the **Groups** tab.
9. Click **New**., and enter the following information:
   - **Name**: OAuth2 Client Developer (case sensitive)
10. Click **OK**.
11. Click the **Users** tab.
12. Click the **3rdparty_dev2** user.
13. Click the **Groups** tab.
14. In the **Chosen** list, add `OAuth2 Client Developer` .
15. Click **Save**.

You have created a user named `3rdparty_dev2` and made it a member of a group named `OAuth2 Client Developer`. This means the user will acquire the `OAuth2 Client Developer` role, and therefore will be authorized to register OAuth 2.0 applications.

Now verify that the user can be successfully authenticated.

Verifying the WebLogic Server User

To verify that the WebLogic Server user created can be successfully authenticated, follow these steps:

1. In your browser, go to a URI in the following format:

   `https://server:port/ords/resteasy/ui/oauth2/clients/`

2. Enter the credentials of the `3rdparty_dev2` user, and click **Sign In**.

The OAuth 2.0 Client Registration page should be displayed, with no applications listed. If this page is displayed, you have verified that authentication against the WebLogic Server user repository is working.

However, if the sign-on prompt is displayed again with the message `User is not authorized to access resource`, then you made mistake (probably misspelling the Group List value).

Integrating with Existing Group/Role Models

The examples in other sections demonstrate configuring the built-in user repositories of WebLogic Server. In these situations you have full control over how user groups are named. If a user is a member of a group with the exact same (case sensitive) name as a role, then the user is considered to have that role.

However, when integrating with existing user repositories, RESTful service developers will often not have any control over the naming and organization of user groups in the user repository. In these situations a mechanism is needed to map from existing "physical" user groups defined in the user repository to the "logical" roles defined by Oracle REST Data Services and/or RESTful Services.

In Oracle REST Data Services, this group to role mapping is performed by configuring a configuration file named `role-mapping.xml`.

Topics:

- About `role-mapping.xml`

About `role-mapping.xml`

`role-mapping.xml` is a Java XML Properties file where each property key defines a pattern that matches against a set of user groups, and each property value identifies the roles that the matched user group should be mapped to. It must be located in the same folder as the `defaults.xml` configuration file. The file must be manually created and edited.

Consider this example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
 <entry key="webdevs">RESTful Services</entry>
</properties>
```

This role mapping is straightforward, stating that any user who is a member of a group named: `webdevs` is given the role `RESTful Services`, meaning that all members of the `webdevs` group can invoke `RESTful Services`.

A mapping can apply more than one role to a group. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
 <entry key="webdevs">RESTful Services, SQL Developer</entry>
</properties>
```

This rule gives members of the `webdevs` group both the `RESTful Services` and `SQL Developer` roles.

Topics:

- Parameterizing Mapping Rules
- Dereferencing Parameters
- Indirect Mappings

Parameterizing Mapping Rules

Having to explicitly map from each group to each role may not be scalable if the number of groups or roles is large. To address this concern, you can parameterize rules. Consider this example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
 <entry key="{prefix}.webdevs">RESTful Services</entry>
</properties>
```

This example says that any group name that ends with `.webdevs` will be mapped to the `RESTful Services` role. For example, a group named: `HQ.webdevs` would match this rule, as would a group named: `EAST.webdevs`.

The syntax for specifying parameters in rules is the same as that used for URI Templates; the parameter name is delimited by curly braces ({}).

Dereferencing Parameters

Any parameter defined in the group rule can also be dereferenced in the role rule. Consider this example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
 <entry key="cn={userid},ou={group},dc=MyDomain,dc=com">{group}</entry>
</properties>
```

This example maps the organizational unit component of an LDAP distinguished name to a role. It says that the organizational unit name maps directly to a role with same name. Note that it refers to a `{userid}` parameter but never actually uses it; in effect, it uses `{userid}` as a wildcard flag.

For example, the distinguished name `cn=jsmith,ou=Developers,dc=MyDomain,dc=com` will be mapped to the logical role named `Developers`.

Indirect Mappings

To accomplish the desired role mapping, it may sometimes be necessary to apply multiple intermediate rules. Consider this example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
 <entry key="cn={userid},ou={group},dc=example,dc=com">{group}</entry>
 <entry key="{prefix},ou={group},dc=acquired,dc=com">{group}</entry>
 <entry key="Developers">RESTful Services, SQL Developer</entry>
</properties>
```

This example maps the organizational unit component of an LDAP distinguished name to some roles. Complicating matters is the fact that users can come from two different organizations, resulting in differing distinguishing name patterns.

- Users from `example.com` always have a single common name (CN) identifying their user id, followed by the organizational unit (OU) and the domain name (DC). For example: `cn=jsmith,ou=Developers,dc=example,dc=com`.
- Users from `acquired.com` have varying numbers of common name (CN) prefixes, but the organizational unit is the field you are interested in. For example: `cn=ProductDev,cn=abell,ou=Engineering,dc=acquired,dc=com`.
- Both organizations identify software engineers with `ou=Developers`.

You want to map engineers in both organizations to the `RESTful Services` and `SQL Developer` roles.

- The first rule maps engineers in the `example.com` organization to the intermediate `Developers` role.
- The second rule maps engineers in the `acquired.com` organization to the intermediate `Developers` role.
- The final rule maps from the intermediate `Developers` role to the `RESTful Services` and `SQL Developer` roles.

Integrating Oracle REST Data Services and WebLogic Server

Oracle REST Data Services (ORDS) recommends that for complex or enterprise user identity integrations, customers can leverage the capabilities of WebLogic server. WebLogic server has a rich and diverse set of capabilities to integrate with existing enterprise identity solutions. When Oracle REST Data Services is deployed on the WebLogic server, it can leverage the capabilities of WebLogic server to get secure access to ORDS based RESTful Services.

Once ORDS is configured to work with WebLogic server, the WebLogic server can provide the authenticated user identity and roles. Based on the memberships of the user role, ORDS authorizes access to the protected RESTful Services.

Configuring ORDS to Integrate with WebLogic Server

This section explains how to configure ORDS to work with WebLogic server.

To configure ORDS to work with WebLogic server, run the following command to prepare the `ords.war` file to integrate with WebLogic server:

```
java -jar ords.war oam-config
```

Run the following command to get help on the `oam-config` command:

```
java -jar ords.war help oam-config
```

Using the oam-config Command

The `oam-config` command re-configures the `web.xml` deployment descriptor in the `ords.war` file that helps the WebLogic server to pass any established user identity to ORDS.

After executing the preceding command, the `ords.war` file must be re-deployed to the WebLogic server.

Determining the Identity and Roles of the User

ORDS uses APIs provided by WebLogic server to retrieve the WLSUser and WLSGroup for the established user identity.

ORDS treats the WLSGroup to be equivalent to the role that the user possesses. For example, if a user or users belongs to a WLSGroup named "Sales Assistant", then ORDS considers such user to have a role named "Sales Assistant".

Retrieving the Authenticated User Information

The user visits the single sign-on login form and obtains a cookie or an access token that asserts the identity and roles. The cookie or the token is then passed to the WebLogic server. The WebLogic server is configured to validate the cookie or token and then map it to a specific user to determine what roles the user possesses. The WebLogic Server performs this operation before passing the request to ORDS. Once ORDS receives the request, it calls the APIs provided by WebLogic server to retrieve the WLSUser and WLSGroup to retrieve the information of the user identity and roles from the WebLogic server.

Related Topics

- Oracle WebLogic APIs
- API to retrieve the WLSUser
- API to retrieve the WLSGroup

Using the Oracle REST Data Services PL/SQL API

Oracle REST Data Services has a PL/SQL API (application programming interface) that you can use as an alternative to the SQL Developer graphical interface for many operations. The available subprograms are included in the following PL/SQL packages:

- Oracle REST Data Services, documented in Oracle REST Data Services PL/SQL Package Reference

- OAUTH, documented in OAUTH PL/SQL Package Reference

To use the Oracle REST Data Services PL/SQL API, you must first:

- Install Oracle REST Data Services in the database that you will use to develop RESTful services.
- Enable one or more database schemas for REST access.

Topics:

- Creating a RESTful Service Using the PL/SQL API
- Testing the RESTful Service

Related Topics

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

Creating a RESTful Service Using the PL/SQL API

You can create a RESTful service by connecting to a REST-enabled schema and using the ORDS.CREATE_SERVICE procedure.

The following example creates a simple "Hello-World"-type service:

```
begin
 ords.create_service(
      p_module_name => 'examples.routes' ,
      p_base_path   => '/examples/routes/',
      p_pattern     => 'greeting/:name',
      p_source => 'select ''Hello '' || :name || '' from '' ||
nvl(:whom,sys_context(''USERENV'',''CURRENT_USER'')) "greeting" from dual');
 commit;
end;
/
```

The preceding example does the following:

- Creates a resource module named examples.routes.
- Sets the base path (also known as the URI prefix) of the module to `/examples/routes/`.
- Creates a resource template in the module, with the route pattern `greeting/:name`.
- Creates a GET handler and sets its source as a SQL query that forms a short greeting:
  - `GET` is the default value for the `p_method` parameter, and it is used here because that parameter was omitted in this example.
  - `COLLECTION_FEED` is the default value for the `p_method` parameter, and it is used here because that parameter was omitted in this example
- An optional parameter named `whom` is specified.

Related Topics

- ORDS.CREATE_SERVICE

Testing the RESTful Service

To test the RESTful service that you created, start Oracle REST Data Services if it is not already started:

```
java -jar ords.war
```

Enter the URI of the service in a browser. The following example displays a "Hello" greeting to `Joe`, by default from the current user because no `whom` parameter is specified.:

```
http://localhost:8080/ords/ordstest/examples/routes/greeting/Joe
```

In this example:

- Oracle REST Data Services is running on localhost and listening on port 8080.
- Oracle REST Data Services is deployed at the context-path `/ords`.
- The RESTful service was created by a database schema named `ordstest`.
- Because the URL does not include the optional whom parameter, the :whom bind parameter is bound to the null value, which causes the query to use the value of the current database user (`sys_context(''USERENV'',''CURRENT_USER'')`) instead.

If you have a JSON viewing extension installed in your browser, you will see a result like the following:

```
{
 "items": [
  {
   "greeting": "Hello Joe from ORDSTEST"
  }
 ],
 "hasMore": false,
 "limit": 25,
 "offset": 0,
 "count": 1,
 "links": [
  {
   "rel": "self",
   "href": "http://localhost:8080/ords/ordstest/examples/routes/greeting/"
  },
  {
   "rel": "describedby",
   "href": "http://localhost:8080/ords/ordstest/metadata-catalog/examples/routes/
greeting/"
  },
  {
   "rel": "first",
   "href": "http://localhost:8080/ords/ordstest/examples/routes/greeting/Joe"
  }
 ]
}
```

The next example is like the preceding one, except the optional parameter `whom` is specified to indicate that the greeting is from `Jane`.

```
http://localhost:8080/ords/ordstest/examples/routes/greeting/Joe?whom=Jane
```

This time, the result will look like the following:

```
{
 "items": [
  {
   "greeting": "Hello Joe from Jane"
  }
 ],
 "hasMore": false,
 "limit": 25,
 "offset": 0,
 "count": 1,
 "links": [
  {
   "rel": "self",
   "href": "http://localhost:8080/ords/ordstest/examples/routes/greeting/"
  },
  {
   "rel": "describedby",
   "href": "http://localhost:8080/ords/ordstest/metadata-catalog/examples/routes/
greeting/"
  },
  {
   "rel": "first",
   "href": "http://localhost:8080/ords/ordstest/examples/routes/greeting/Joe"
  }
 ]
}
```

Notice that in this result, what follows "from" is `Jane` and not `ORDSTEST`, because the `:whom` bind parameter was bound to the `Jane` value.

Oracle REST Data Services Database Authentication

This section describes how to use the database authentication feature to provide basic authentication for PL/SQL gateway calls.

Database authentication feature is similar to dynamic basic authentication provided by mod-plsql where the user is prompted for the database credentials to authenticate and authorize access to PL/SQL stored procedures.

Installing Sample Database Scripts

This section describes how to install the sample database scripts.

The unzipped Oracle REST Data Services installation kit contains the sample database scripts that create a basic demo scenario for the database authentication.

The following code snippet shows how to install the sample database schema:

```
db_auth $ cd sql/
sql $ sql sys as sysdba

SQLcl: Release Release 18.1.1 Production on Fri Mar 23 14:03:18 2018

Copyright (c) 1982, 2018, Oracle.  All rights reserved.

Password? (**********?) ******
```

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


SQL> @install <chosen-password>
```

> **Note:**
>
> - You need to adjust the SQLcl connect string and the user credentials to suit your environment. For this demo scenario, SQLcl connects to the database with service name `orcl`
> - `<chosen-password>` is the password you assigned to `EXAMPLE_USER1` and `EXAMPLE_USER2` database users. Make a note of this password value for later reference.

The sample database schema creates the following database users:

- **SAMPLE_PLSQL_APP**: A database schema where the protected `SAMPLE_PROC` will be installed.
- **EXAMPLE_USER1**: A database user granted with execute privilege on `SAMPLE_PLSQL_APP.SAMPLE_PROC` procedure.
- **EXAMPLE_USER2**: A second database user granted with execute privilege on `SAMPLE_PLSQL_APP.SAMPLE_PROC` procedure.

Enabling the Database Authentication

This section describes how to enable the database authentication feature.

To enable the database authentication feature, do one of the following:

- For fresh installation of Oracle REST Data Services, update the `/u01/ords/params/ords_params` properties file with the following entry:

  ```
  jdbc.auth.enabled=true
  ```

- For existing Oracle REST Data Services installation, run the following commands:

  ```
  cd /u01/ords
  $JAVA_HOME/bin/java -jar ords.war set-property jdbc.auth.enabled
  true
  ```

This setting is applicable to PL/SQL gateway pools (for example, `apex.xml`), it does not apply to other pool types such as the `ORDS_PUBLIC_USER` pool (for example, `apex_pu.xml`).

> **Note:**
>
> The `jdbc.auth.enabled` setting can be configured per database pool. Alternatively, it can be configured in `defaults.xml` file so that it is enabled for all pools.

Example 5-11 Setting Enabled for all Pools

This example code snippet shows how `jdbc.auth.enabled` setting is enabled for all pools.

```
ords $ java -jar ords.war set-property jdbc.auth.enabled true
Mar 23, 2018 2:23:49 PM oracle.dbtools.rt.config.setup.SetProperty
execute
INFO: Modified: /tmp/cd/ords/defaults.xml, setting: jdbc.auth.enabled =
true
```

After you update the configuration settings, restart the Oracle REST Data Services for the changes to take effect.

Configuring the Request Validation Function

This section describes how to temporarily disable the request validation function.

If you want to invoke only a whitelisted set of stored procedures in the database through the PL/SQL gateway, then you must configure Oracle REST Data Services to use a request validation function (especially when you are using Oracle Application Express).

The demo sample procedure used for testing the database authentication feature is not whitelisted, so you must temporarily disable the request validation function.

To disable the request validation function, perform the following steps:

1. Locate the folder where the Oracle REST Data Services configuration file is stored.
2. Open the `defaults.xml` file.
3. Look for `security.requestValidationFunction` entry and remove it from the file.
4. Save the file.
5. Restart Oracle REST Data Services, if it is already running.

> **Note:**
>
> In production environment, you must use a custom request validation function that whitelists the stored procedures you want to access for your application

Testing the Database Authenticated User

This section describes how to test if the database user is authenticated.

Assuming that Oracle REST Data Service is running in a standalone mode on local host and on port 8080, access the following URL in your web browser:

```
http://localhost:8080/ords/sample_plsql_app.sample_proc
```

The browser prompts you to enter credentials. Enter `example_user1` for user name and enter the password value you noted while installing the sample schema.

The browser displays 'Hello EXAMPLE_USER1!' to demonstrate that the database user was authenticated and the identity of the user was propagated to the database through the OWA CGI variable named `REMOTE_USER`..

Uninstalling the Sample Database Schema

To uninstall the database schema, run the commands as shown in the following code snippet:

```
db_auth $ cd sql/
sql $ sql sys as sysdba

SQLcl: Release Release 18.1.1 Production on Fri Mar 23 14:03:18 2018

Copyright (c) 1982, 2018, Oracle.  All rights reserved.

Password? (**********?) ******
Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit
Production
SQL> @uninstall
```

Overview of Pre-hook Functions

This section explains how to use PL/SQL based pre-hook functions that are invoked prior to an Oracle REST Data Services (ORDS) based REST call.

A pre-hook function is typically used to implement application logic that needs to be applied across all REST endpoints of an application. For example a pre-hook enables the following functionality:

- **Configure application specific database session state**: Configure the session to support a VPD policy.
- **Custom authentication and authorization**: As the pre-hook is invoked prior to dispatching the REST service, it is used to inspect the request headers and determine the user who is making the request, and also find if that user is authorized to make the request.
- **Auditing or metrics gathering**: To track information regarding the REST APIs invoked.

Topics:

- Configuring the Pre-hook Function
- Using a Pre-hook Function
- Processing of a Request
- Identity Assertion of a User
- Aborting Processing of a Request
- Ensuring Pre-hook is Executable
- Exceptions Handling by Pre-hook Function
- Pre-hook Function Efficiency
- Pre-Hook Examples

Configuring the Pre-hook Function

This section describes how to configure a pre-hook function.

The pre-hook function is configured using `procedure.rest.preHook` setting. The value of this setting must be the name of a stored PL/SQL function.

Using a Pre-hook Function

This section explains how the pre-hook function is used.

A pre-hook must be a PL/SQL function with no arguments and must return a `BOOLEAN` value. The function must be executable by the database user to whom the request is mapped. For example, if the request is mapped to an ORDS enabled schema, then that schema must be granted the execute privilege on the pre-hook function (or to `PUBLIC`).

If the function returns `true`, then it indicates that the normal processing of the request must continue. If the function returns `false`, then it indicates that further processing of the request must be aborted.

ORDS invokes a pre-hook function in an OWA (Oracle Web Agent) that is a PL/SQL Gateway Toolkit environment. This means that the function can introspect the request headers and the OWA CGI environment variables, and use that information to drive its logic. The function can also use the OWA PL/SQL APIs to generate a response for the request (for example, in a case where the pre-hook function needs to abort further processing of the request, and provide its own response).

Processing of a Request

The pre-hook function must return `true` if it determines that the processing of a request must continue. In such cases, any OWA response produced by the pre-hook function is ignored (except for cases as detailed in the section Identity Assertion of a User), and the REST service is invoked as usual.

Identity Assertion of a User

This section describes how pre-hook function can make assertions about the identity of the user.

When continuing processing, a pre-hook can make assertions about the identity and the roles assigned to the user who is making the request. This information is used in the processing of the REST service. A pre-hook function can determine this by setting one or both of the following OWA response headers.

- `X-ORDS-HOOK-USER`: Identifies the user making the request, the value is bound to the `:current_user` implicit parameter and the `REMOTE_IDENT` OWA CGI environment variable.
- `X-ORDS-HOOK-ROLES`: Identifies the roles assigned to the user. This information is used to determine the authorization of the user to access the REST service. If this header is present then `X-ORDS-HOOK-USER` must also be present.

> **Note:**
>
> `X-ORDS-HOOK-USER` and `X-ORDS-HOOK-ROLES` headers are not included in the response of the REST service. These headers are only used internally by ORDS to propagate the user identity and roles.
>
> Using these response headers, a pre-hook can integrate with the role based access control model of ORDS. This enables the application developer to build rich integrations with third party authentication and access control systems.

Aborting Processing of a Request

This section explains how the pre-hook function aborts the processing of a request.

If a pre-hook determines that the processing of the REST service should not continue, then the function must return `false` value. This value indicates to ORDS that further processing of the request must not be attempted.

If the pre-hook does not produce any OWA output, then ORDS generates a `403 Forbidden` error response page. If the pre-hook produces any OWA response, then ORDS returns the OWA output as the response. This enables the pre-hook function to customize the response that client receives when processing of the REST service is aborted.

Ensuring Pre-hook is Executable

If a schema cannot invoke a pre-hook function, then ORDS generates a `503 Service Unavailable` response for *any* request against that schema. Since a pre-hook has been configured, it would not be safe for ORDS to continue processing the request without invoking the pre-hook function. It is very important that the pre-hook function is executable by all ORDS enabled schemas. If the pre-hook function is not executable, then the REST services defined in those schemas will not be available.

Exceptions Handling by Pre-hook Function

When a pre-hook raises an error condition, for example, when a run-time error occurs, a `NO DATA FOUND` exception is raised. In such cases, ORDS cannot proceed with processing of the REST service as it would not be secure. ORDS inteprets any

exception raised by the pre-hook function as a signal that the request is forbidden and generates a `403 Forbidden` response, and does not proceed with invoking the REST service. Therefore, if the pre-hook raises an unexpected exception, it forbids access to that REST service. It is highly recommended that all pre-hook functions must have a robust exception handling block so that any unexpected error conditions are handled appropriately and do not make REST Services unavailable.

Pre-hook Function Efficiency

A pre-hook function is invoked for every REST service call. Therefore, the pre-hook function must be designed to be efficient. If a pre-hook function is inefficient, then it has a negative effect on the performance of the REST service call. Invoking the pre-hook involves at least one additional database round trip. It is critical that the ORDS instance and the database are located close together so that the round-trip latency overhead is minimized.

Pre-Hook Examples

This section provides some sample PL/SQL functions that demonstrate different ways in which the pre-hook functionality can be leveraged.

Source code for the examples provided in the following sections is included in the unzipped Oracle REST Data Services distribution archive `examples/pre_hook/sql` sub-folder.

Installing the Examples

This section describes how to install the pre-hook examples.

To install the pre-hook examples, execute `examples/pre_hook/sql/install.sql` script. The following code snippet shows how to install the examples using Oracle SQLcl command line interface:

```
pre_hook $ cd examples/pre_hook/sql/
sql $ sql sys as sysdba

SQLcl: Release Release 18.1.1 Production on Fri Mar 23 14:03:18 2018

Copyright (c) 1982, 2018, Oracle.  All rights reserved.

Password? (**********?) ******
Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit
Production


SQL> @install <chosen-password>
```

- You need to adjust the SQLcl connect string and the user credentials to suit your environment. For these demo scenarios, SQLcl connects to the database with service name `orcl`.
- `<chosen-password>` is the password you assigned to the `PRE_HOOK_TEST` database user. Make a note of this password value for later reference.

- The `examples/pre_hook/sql/install.sql` command creates the following two databases schemas:
  - The `PRE_HOOK_DEFNS` schema where the pre-hook function is defined along with a database table named `custom_auth_users`, where user identities are stored. This table is populated with a single user `joe.bloggs@example.com`, whose password is the value assigned for `<chosen-password>`.
  - The `PRE_HOOK_TESTS` schema where ORDS based REST services that are used to demonstrate the pre-hooks are defined.

Example: Denying all Access

The simplest pre-hook is one that unilaterally denies access to any REST Service.

To deny access to any REST service, the function must always return `false` as shown in the following code snippet:

```
create or replace function deny_all_hook return boolean as
begin
 return false;
end;
/
grant execute on deny_all_hook to public;
```

Where:

- The `deny_all_hook` pre-hook function always returns `false` value.
- Execute privilege is granted to all users. So, any ORDS enabled schema can invoke this function

Configuring ORDS

To enable `deny_all_hook` pre-hook function, perform the following steps:

1. Locate the folder where the Oracle REST Data Services configuration file is stored.
2. Open the `defaults.xml` file and add:

   ```
   <entry key="procedure.rest.preHook">pre_hook_defns.deny_all_hook</
   entry>
   ```

3. Save the file.
4. Restart Oracle REST Data Services.

Try it out

The install script creates an ORDS enabled schema and a REST service which can be accessed at the following URL (assuming ORDS is deployed on `localhost` and listening on port `8080`) :

```
http://localhost:8080/ords/pre_hook_tests/prehooks/user
```

Access the URL in a browser. You should get a response similar to the following:

```
403 Forbidden
```

This demonstrates that the `deny_all_hook` pre-hook function was invoked and it prevented the access to the REST service by returning a `false` value.

Example: Allowing All Access

Modify the source code of the `deny_all_hook` pre-hook function to allow access to all REST service requests as shown in the following code snippet:

```
create or replace function deny_all_hook return boolean as
begin
 return true;
end;
/
```

Try it out

Access the following test URL in a browser:

```
http://localhost:8080/ords/pre_hook_tests/prehooks/user
```

The response should include JSON similar to the following code snippet:

```
{
 "authenticated_user": "no user authenticated"
}
```

> **Note:**
>
> The REST service executes because the pre-hook function authorized it.

Related Topics

- Identity Assertion of a User
  This section describes how pre-hook function can make assertions about the identity of the user.

Example: Asserting User Identity

The following code snippet demonstrates how the pre-hook function makes assertions about the user identity and the roles they possess:

```
create or replace function identity_hook return boolean as
begin
 if custom_auth_api.authenticate_owa then
  custom_auth_api.assert_identity;
  return true;
 end if;
```

```
 custom_auth_api.prompt_for_basic_credentials('Test Custom Realm');
 return false;
end;
```

The pre-hook delegates the task of authenticating the user to the `custom_auth_api.authenticate_owa` function. If the function indicates that the user is authenticated, then it invokes the `custom_auth_api.assert_identity` procedure to propagate the user identity and roles to ORDS.

Configuring ORDS

To enable pre-hook function, perform the following steps:

1. Locate the folder where the Oracle REST Data Services configuration file is stored.
2. Open the `defaults.xml` file and add:

   ```
   <entry key="procedure.rest.preHook">pre_hook_defns.identity_hook</
   entry></entry>
   ```

3. Save the file.
4. Restart Oracle REST Data Services.

Try it out

The install script creates an ORDS enabled schema and a REST service that can be accessed at the following URL (assuming ORDS is deployed on localhost and listening on port 8080):

```
http://localhost:8080/ords/pre_hook_tests/prehooks/user
```

In a web browser access the preceding URL.

> **Note:**
>
> The first time you access the URL, the browser will prompt you to enter your credentials. Enter the user name as `joe.bloggs@example.com` and for the password, use the value you assigned for `<chosen-password>` when you executed the install script. Click the link to sign in.

In response a JSON document is displayed with the JSON object in it.

```
{"authenticated_user":"joe.bloggs@example.com"}
```

Uninstalling the Examples

This section explains how to uninstall the examples.

The following code snippet shows how to uninstall the examples:

```
pre_hook $ cd sql/
sql $ sql sys as sysdba
```

```
SQLcl: Release Release 18.1.1 Production on Fri Mar 23 14:03:18 2018

Copyright (c) 1982, 2018, Oracle.  All rights reserved.

Password? (**********?) ******
Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit
Production


SQL> @uninstall
```

Generating Hyperlinks

Oracle REST Data Services (ORDS) provides a mechanism to transform relational result sets into JSON representations, and provides hyperlinks that automatically paginates the result set to allow navigation between the pages of the result set.

For many use cases, it is required to treat certain columns in the result set as hyperlinks. ORDS provides the following simple yet powerful mechanisms for adding hyperlinks to REST resources:

- **Primary Key Hyperlinks**: A column with the reserved alias `$.id` identifies the primary key column of a single row in the result set. Such column values are used to form a hyperlink that points to a child resource of the current resource that provides specific details about that particular row in the result set.
- **Arbitrary Hyperlinks**: A column whose alias starts with the reserved character `$` is treated as a hyperlink. The subsequent characters in the column alias indicates the link relation type.

Primary Key Hyperlinks

This section describes how to add primary key hyperlinks.

Typically, when you are modelling a REST API, you need to model the Resource Collection Pattern that enumerates the hyperlinks to the other resources.

In a simple use case, a query is against a single table that contains a single column with primary key that is used to identify each row. The collection resource provides summary information of each row, and provides a self link for each row. The self link points to the resource that provides more detailed information about the row. For example, if we use the `EMP` table, we can define a service as shown in the following code snippet:

```
begin
    ords.define_service(
        p_module_name => 'links.example',
        p_base_path => 'emp-collection/',
        p_pattern => '.',
        p_source => 'select empno "$.id", empno id, ename employee_name
from emp order by empno ename';
```

```
    commit;
end;
```

Where:

- The reserved value '.' is used for the p_pattern value. This indicates the path of the resource template in the base path of the resource module, emp-collection/ in this example.
- The EMPNO column is aliased as $.id, to produce a hyperlink.

Following code snippet shows the output produced after invoking the preceding service:

```
{
    "items": [{
        "id": 7369,
        "employee_name": "SMITH",
        "links": [{
            "rel": "self",
            "href": "http://localhost:8080/ords/ordstest/emp-collection/
7369"
        }]
    },
    ...
    ],
    "hasMore": false,
    "limit": 25,
    "offset": 0,
    "count": 14,
    "links": [{
        "rel": "self",
        "href": "http://localhost:8080/ords/ordstest/emp-collection/"
    }, {
        "rel": "describedby",
        "href": "http://localhost:8080/ords/ordstest/metadata-catalog/
emp-collection/"
    }, {
        "rel": "first",
        "href": "http://localhost:8080/ords/ordstest/emp-collection/"
    }]
}
```

Observe that the value of EMPNO column is concatenated with the URL of the service to produce a new hyperlink with relation self. The value is not simply concatenated, it is resolved using the algorithm specified in RFC3986. Therefore, Oracle REST Data Services (ORDS) can take the value of the column, and apply the resolution algorithm to produce a new absolute URL.

> **See Also:**
>
> Section 5 of rfc3986

If you attempt to navigate to this URL, it results in a 404 HTTP status because a resource handler for that endpoint has not yet been defined. The following code snippet shows a sample resource handler:

```
begin
  ords.define_template(
      p_module_name    => 'links.example',
      p_pattern        => ':id');
  ords.define_handler(
      p_module_name    => 'links.example',
      p_pattern        => ':id',
      p_source_type    => ords.source_type_collection_item,
      p_source         => 'select emp.empno "$.id", emp.* from emp
where empno = :id');
  commit;
end;
```

Composite Primary Keys

This section describes the support for composite primary keys.

If multiple columns in a query form the primary key of a row, then each of those columns must be aliased by `$.id`. ORDS combines such values to form the relative path of the item URL.

Related Topics

- Route Patterns Specification

Arbitrary Hyperlinks

This section describes how to create hyperlinks to point to a resource one level up in the heirarchy.

Rich hypermedia documents have many different hyperlinks. ORDS provides a mechanism to turn any column value into a hyperlink. Any column whose alias starts with the `$` character is treated as a hyperlink. The following example code snippet shows how an employee resource can provide a hyperlink to their manager:

```
begin
  ords.define_handler(
      p_module_name    => 'links.example',
      p_pattern        => ':id',
      p_source_type    => ords.source_type_collection_item,
      p_source         => 'select emp.empno "$.id", emp.*, emp.mgr
"$related" from emp where empno = :id');commit;end;
```

ORDS treats the column named `$related` to a hyperlink and the column value is treated as a path relative to the containing base URI of the resource. Similar to how `$.id` column value is transformed into an absolute URI by applying the algorithm specified in RFC 3986.

> **See Also:**
>
> Section 5.2 of rfc3986.

The following example code snippet shows the updated employee resource:

```
{
    "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/ordstest/emp-collection/
7369"
    }, {
        "rel": "describedby",
        "href": "http://localhost:8080/ords/ordstest/metadata-catalog/
emp-collection/item"
    }, {
        "rel": "collection",
        "href": "http://localhost:8080/ords/ordstest/emp-collection/"
    }, {
        "rel": "related",
        "href": "http://localhost:8080/ords/ordstest/emp-collection/
7902"
    }]
}
```

Note that the new `related` link points to the manager resource of the employee. The manager resource in turn has a related link that points to their manager, and so on up the management chain until you reach employee number 7839 who is the president of the company and whose `mgr` column is `null`. If the column value is null, then ORDS will not create a hyperlink.

```
{
    "empno": 7839,
    "ename": "KING",
    "job": "PRESIDENT",
    "mgr": null,
    "hiredate": "1981-11-17T00:00:00Z",
    "sal": 5000,
    "comm": null,
    "deptno": 10,
    "links": [{
        "rel": "self",
        "href": "http://localhost:8080/ords/ordstest/emp-collection/
7839"
```

```
    }, {
        "rel": "describedby",
        "href": "http://localhost:8080/ords/ordstest/metadata-catalog/
emp-collection/item"
    }, {
        "rel": "collection",
        "href": "http://localhost:8080/ords/ordstest/emp-collection/"
    }]
}
```

About the related Link Relation

This section explains the use of existing registered link relation types instead of extension link relation types.

As per RFC 8288 Section 2.1.2, any extension link relation must be an URI and not a simple value. This means that a link relation such as `manager` is not a legal link relation according to the specification. A custom link relation type will reduce interoperability. If your application uses a non-registered link relation type, then only a few clients will be able to understand the custom link relation type. Conversely, if you use registered link relation types, then more clients can navigate to your link relations. Oracle recommends using existing registered link relation types instead of extension link relation types.

Related Topics

- rfc8288

URL Resolution

This section describes how ORDS resolves column values using URI resolution algorithm.

Related Topics

- rfc3986

Child Paths

This section describes how to use the relative paths to refer to the child resources.

Following code snippet shows the use of relative paths to refer to child resources:

```
select'child/resource'"$related" from dual
```

Assuming that the base URL of the containing resource is `https://example.com/ords/some_schema_alias/some/resource`, then the link is as shown in the following code snippet:

```
{
 "rel": "related",
 "href": "https://example.com/ords/some_schema_alias/some/child/
resource"
}
```

Ancestor Paths

This section provides examples to show how ORDS lets you use `../` and `./` syntax to refer to parent paths of the current resource.

Following is an example code snippet:

```
select'../'"$up", './'"$self" from dual
```

Assuming the base URL of the containing resource is `https://example.com/ords/some_schema_alias/some/collection/`, then the links will be as shown in the following code snippet:

```
{
 "rel": "up",
 "href": "https://example.com/ords/some_schema_alias/some/"
},
{
 "rel": "self",
 "href": "https://example.com/ords/some_schema_alias/some/collection/"
}
```

Absolute URLs

This section provides examples for the absolute paths.

A hyperlink value can be an absolute path or a fully qualified URL as shown in the following code snippet:

```
select'/cool/stuff'"$related", 'https://oracle.com/rest'"$related" from
dual
```

Assuming the base URL of the containing resource is, `https://example.com/ords/some_schema_alias/some/collection/` the links will be as shown in the following code snippet:

```
{
 "rel": "related",
 "href": "https://example.com/cool/stuff"
},
{
 "rel": "related",
 "href": "https://oracle.com/rest"
}
```

You can have multiple links for the same link relation.

Context Root Relative Paths

This section provides example for the context root relative path.

The context root relative path is the URL of the root resource of an ORDS enabled schema.

The following code snippet shows the context root path for the example discussed in the preceding sections:

```
https://example.com/ords/some_schema_alias/
```

ORDS provides the following syntax to express the resource paths relative to the URL:

```
select'^/another/collection/'"$related"from dual
```

Assuming the base URL of the containing resource is `https://example.com/ords/some_schema_alias/some/collection/`, the link is as shown in the following code snippet:

```
{
 "rel": "related",
 "href": "https://example.com/ords/some_schema_alias/another/collection"
}
```

Any path starting with `^/1` is resolved relative to the context root path.

Dynamic Paths

This section describes how you can have dynamic values for the hyperlinks.

Examples provided in the preceding sections use literal values for the hyperlinks. The hyperlink value can be completely dynamic, formed from any value that is a string (or can be automatically converted to a string). For example, instead of pointing directly to the employee resource, for managers only, you can point to a more specialized resource that can show additional information such as the total number of reports. The GET handler can be redefined for the `emp-collection` or `:id` resource as shown in the following code snippet:

```
begin
  ords.define_handler(
      p_module_name    => 'links.example',
      p_pattern        => ':id',
      p_source_type    => ords.source_type_collection_item,
      p_source         => 'select emp.empno "$.id", emp.*,
decode(emp.mgr, null, null, '^/managers/' || emp.mgr) "$related" from
emp where empno = :id');
     commit;
end;
```

Where:

- The value of the `$related` column is formed from `^/managers/: emp.mgr` unless the value of `emp.mgr` is `null`. In such a case, a `null` value is substituted that causes ORDS not to generate the hyperlink.

The following code snippet shows the updated employee resource:

```
{
    "empno": 7566,
    "ename": "JONES",
    "job": "MANAGER",
    "mgr": 7839,
    "hiredate": "1981-04-01T23:00:00Z",
    "sal": 2975,
    "comm": null,
    "deptno": 20,
    "links": [{
        "rel": "self",
        "href": "http://localhost:8080/ords/ordstest/emp-collection/
7566"
    }, {
        "rel": "describedby",
        "href": "http://localhost:8080/ords/ordstest/metadata-catalog/
emp-collection/item"
    }, {
        "rel": "collection",
        "href": "http://localhost:8080/ords/ordstest/emp-collection/"
    }, {
        "rel": "related",
        "href": "http://localhost:8080/ords/ordstest/managers/7839"
    }]
}
```

Note:

The `related` link now points to the dynamically generated path, that is, to the `managers/:id` resource.

6

REST-Enabled SQL Service

The REST-Enabled SQL service is a HTTPS web service that provides access to the Oracle Database SQL engine. You can POST SQL statements to the service. The service then runs the SQL statements against Oracle Database and returns the result to the client in a JSON format.

Statically defined RESTful services use predefined SQL statements that are useful when you need a fixed and repeatable service. The REST- Enabled SQL service enables you to define SQL statements dynamically and run them against the database without predefined SQL statements. This makes your data more accessible over REST.

Typical Use Case: Your Oracle Database is in the cloud and you want to make it available through a REST API over HTTPS.

Predefined REST APIs provide common operations such as returning the results of reports and providing an API for updating common tables in your database. There is a need for client developers to run their own queries or queries that can only be written at run time. In these cases, a REST- Enabled SQL service is useful.

> **Note:**
>
> If you have Oracle REST Data Services installed and if you do not have SQL*Net (JDBC, OCI) to establish an network connection to Oracle Database, then a REST-Enabled SQL service provides an easy mechanism to query and run SQL, SQL*Plus, and SQLcl statements against the REST-enabled Oracle Database schema.

Topics:

- REST-Enabled SQL Service Terminology
- Configuring the REST-Enabled SQL Service
- Using cURL with REST-Enabled SQL Service
- Getting Started with the REST-Enabled SQL Service

REST-Enabled SQL Service Terminology

This section introduces some common terms that are used throughout this document.

- **REST- Enabled SQL service**: A HTTPS web service that provides SQL access to the database. SQL statements can be posted to the service, and the results are returned in a JSON format to the client.
- **HTTPS**: Hyper Text Transfer Protocol Secure (HTTPS) is the secure version of **HTTP**, the protocol over which data is sent between your browser and the website to which you are connected. The ***'S'*** stands for secure. It means that

all communications between your browser and Oracle REST Data Services are encrypted.

- **cURL**: cURL is a command-line tool used to transfer data. It is free and open source software that can be downloaded from the following location: curl_haxx.
- **SQL*Net (or Net8)**: SQL*Net is the networking software of Oracle that enables remote data access between programs and Oracle Database.

Configuring the REST-Enabled SQL Service

By default, the REST- Enabled SQL service is turned off. To configure REST- Enabled SQL service settings, see Configuring REST-Enabled SQL Service Settings.

Using cURL with REST-Enabled SQL Service

This section explains how to use cURL commands to access the REST-Enabled SQL service.

You can use the HTTPS POST method to access the REST-Enabled SQL service. To access the REST-Enabled SQL service, you can use the command-line tool named cURL. This powerful tool is available for most platforms, and enables you to connect and control the data that you send to and receive from a REST-Enabled SQL service.

Example 6-1 Example cURL Command

Request: `curl -i -X POST --user ORDSTEST:ordstest --data-binary "select sysdate from dual" -H "Content-Type: application/sql" -k https://localhost:8088/ords/ordstest/_/sql`

Where:

- The `-i` option displays the HTTP headers returned by the server.
- The `-k` option enables cURL to proceed and operate even for server connections that are otherwise considered to be insecure.

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Frame-Options: SAMEORIGIN
Transfer-Encoding: chunked
{
    "env":{
        "defaultTimeZone":"Europe/London"
    },
    "items":[
        {
            "statementId":1,
            "statementType":"query",
            "statementPos":{
                "startLine":1,
                "endLine":2
            },
            "statementText":"select sysdate from dual",
            "response":[
```

```
            ],
            "result":0,
            "resultSet":{
                "metadata":[
                    {
                        "columnName":"SYSDATE",
                        "jsonColumnName":"sysdate",
                        "columnTypeName":"DATE",
                        "precision":0,
                        "scale":0,
                        "isNullable":1
                    }
                ],
                "items":[
                    {
                        "sysdate":"2017-07-21T08:06:44Z"
                    }
                ],
                "hasMore":false,
                "limit":1500,
                "offset":0,
                "count":1
            }
        }
    ]
}
```


Getting Started with the REST-Enabled SQL Service

The REST- Enabled SQL service is provided only through HTTPS POST method.

Topics:

- REST-Enabling the Oracle Database Schema
- REST-Enabled SQL Authentication
- REST-Enabled SQL Endpoint

REST-Enabling the Oracle Database Schema

You must REST-enable the Oracle database schema on which you want to use the REST- Enabled SQL service. To REST-enable the Oracle Database schema, you can use SQL Developer or the PL/SQL API.

The following code snippet shows how to REST-enable the Oracle Database schema `ORDSTEST`:

```
SQL> CONNECT ORDSTEST/*****;
Connected
SQL> exec ords.enable_schema;
anonymous block completed
SQL> commit;
```

```
Commit complete.
SQL>
```

Related Topics

- Auto-Enabling Using the PL/SQL API

REST-Enabled SQL Authentication

This section explains how to authenticate the schema on which you want to use the REST-Enabled SQL service.

Before using the REST-Enabled SQL service, you must authenticate using the SQL Developer role.

The Following are the different types of authentications available:

- **First Party Authentication (Basic Authentication)**: For this authentication, create a user in Oracle REST Data Services with the **SQL Developer** role. This Oracle REST Data Services user will be able to run SQL for any Oracle database schema that is REST-enabled.
- **Schema Authentication**: For this authentication, use the Oracle Database schema name in uppercase and the Oracle database schema password (for example, `HR` and `HRPassword`). This type of user will be able to run SQL for the specified schema. It will be given the SQL Developer role by Oracle REST Data Services.
- **OAuth 2 Client Credentials**: For this authentication, perform the following steps to grant the SQL Developer role to the client in Oracle REST Data Services:
  1. Create a client using `OAUTH.create_client`.
  2. Grant the **SQL Developer** role to the client.
  3. Acquire the access token using the `client_id` and `client_secret` of the client.
  4. Specify the access token in subsequent REST-Enabled SQL requests.

REST-Enabled SQL Endpoint

This section shows the format or pattern used to access the REST- Enabled SQL service.

If Oracle REST Data Services is running in a Java EE Application Server, then the REST-Enabled SQL service is only accessible through HTTPS. If Oracle REST Data Services is running in standalone mode, then Oracle REST Data Services can be configured to use HTTPS. The examples in this document use this configuration.

The following example URL locates the REST-Enabled SQL service for the specified schema alias:

Pattern: `https://<HOST>/ords/<SchemaAlias>/_/sql`

Example: `https://host/ords/ordstest/_/sql`

Where: The default port is 443

Content Type and Payload Data Type Supported

The HTTPS POST request consists of the following:

- Header Content-Type
  - `application/sql`: for SQL statements
  - `application/json`: for JSON documents
- Payload data type
  - **SQL**: SQL, PL/SQL, SQL*Plus, SQLcl statements
  - **JSON document**: A JSON document with SQL statements and other options such as bind variables

REST-Enabled SQL Service Examples

This section provides different HTTPS POST request examples that use Oracle REST Data Services standalone setup with secure HTTPS access.

The payload data of the HTTPS POST request message can be in one of the following formats:

- POST Requests Using application/sql Content-Type
- POST Requests Using application/json Content-Type

POST Requests Using application/sql Content-Type

For POST requests with `Content-Type` as `application/sql` , the payload is specified using SQL, SQL*Plus, and SQLcl statements. The payload can be a single line statement, multiple line statements, or a file that consists of multiline statements as shown in the following examples:

- Using a Single SQL Statement
- Using Multiple SQL Statements
- Using a File with cURL

> **Note:**
>
> While evaluating your SQL/PLSQL statements, if you see an error message 555 with the following message, then ensure that you have correctly formed your SQL/PLSQL statement:
> " 555 User Defined Resource Error
>
> The request could not be processed because an error occurred whilst attempting to evaluate the SQL statement associated with this resource.Please check the SQL statement is correctly formed and executes without error"

Using a Single SQL Statement

The following example uses Schema Authentication to run a single SQL statement against the `demo` Oracle Database schema:

Request:

```
curl -i -X POST --user DEMO:demo --data-binary "select sysdate from
dual" -H "Content-Type: application/sql" -k https://localhost:8088/ords/
demo/_/sql
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Frame-Options: SAMEORIGIN
Transfer-Encoding: chunked

{
    "env":{
        "defaultTimeZone":"Europe/London"
    },
    "items":[
        {
            "statementId":1,
            "statementType":"query",
            "statementPos":{
                "startLine":1,
                "endLine":2
            },
            "statementText":"select sysdate from dual",
            "response":[

            ],
            "result":0,
            "resultSet":{
                "metadata":[
                    {
                        "columnName":"SYSDATE",
                        "jsonColumnName":"sysdate",
                        "columnTypeName":"DATE",
                        "precision":0,
                        "scale":0,
                        "isNullable":1
                    }
                ],
                "items":[
                    {
                        "sysdate":"2017-07-21T08:06:44Z"
                    }
                ],
                "hasMore":false,
                "limit":1500,
                "offset":0,
                "count":1
```

```
            }
        }
    ]
}
```

Where:

- `DEMO` is the Oracle Database schema name.
- `demo` is the Oracle Database schema password.
- `select sysdate from dual` is the SQL statement that will run in the `DEMO` Oracle Database schema.
- `Content-Type: application/sql` is the content type. Only `application/sql` and `application/json` are supported.
- `https://localhost:8088/ords/demo/_/sql` is the location of the REST- Enabled SQL service for the `demo` Oracle Database schema.

Using a File with cURL

For multiline SQL statements, using a file as payload data in requests is useful.

File: `simple_query.sql`

```
SELECT 10
FROM dual;
```

Request:

```
curl -i -X POST --user DEMO:demo --data-binary "@simple_query.sql" -H
"Content-Type: application/sql" -k https://localhost:8088/ords/demo/_/sql
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Frame-Options: SAMEORIGIN
Transfer-Encoding: chunked

{
    "env":{
        "defaultTimeZone":"Europe/London"
    },
    "items":[
        {
            "statementId":1,
            "statementType":"query",
            "statementPos":{
                "startLine":1,
                "endLine":1
            },
            "statementText":"SELECT 10 FROM dual",
            "response":[
```

```
            ],
            "result":0,
            "resultSet":{
                "metadata":[
                    {
                        "columnName":"10",
                        "jsonColumnName":"10",
                        "columnTypeName":"NUMBER",
                        "precision":0,
                        "scale":-127,
                        "isNullable":1
                    }
                ],
                "items":[
                    {
                        "10":10
                    }
                ],
                "hasMore":false,
                "limit":1500,
                "offset":0,
                "count":1
            }
        }
    ]
}
```


Using Multiple SQL Statements

You can run one or more statements in each POST request. Statements are separated similar to Oracle Database SQL*Plus script syntax, such as, end of line for SQL*Plus statements, a semi colon for SQL statements, and forward slash for PL/SQL statements.

File: **`script.sql`**:

```
CREATE TABLE T1 (col1 INT);
DESC T1
INSERT INTO T1 VALUES(1);
SELECT * FROM T1;
BEGIN
INSERT INTO T1 VALUES(2);
END;
/
SELECT * FROM T1;
```

Request:`curl -i -X POST --user DEMO:demo --data-binary "@script.sql" -H "Content-Type: application/sql" -k https://localhost:8088/ords/demo/_/sql`

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Frame-Options: SAMEORIGIN
```

```
Transfer-Encoding: chunked

{
    "env":{
        "defaultTimeZone":"Europe/London"
    },
    "items":[
        {
            "statementId":1,
            "statementType":"ddl",
            "statementPos":{
                "startLine":1,
                "endLine":1
            },
            "statementText":"CREATE TABLE T_EXAMPLE1 (col1 INT)",
            "response":[
                "\nTable T_EXAMPLE1 created.\n\n"
            ],
            "result":0
        },
        {
            "statementId":2,
            "statementType":"sqlplus",
            "statementPos":{
                "startLine":2,
                "endLine":2
            },
            "statementText":"DESC T_EXAMPLE1",
            "response":[
                "Name Null\n Type \n---- ----- ---------- \nCOL1
NUMBER(38) \n"
            ],
            "result":0
        },
        {
            "statementId":3,
            "statementType":"dml",
            "statementPos":{
                "startLine":3,
                "endLine":3
            },
            "statementText":"INSERT INTO T_EXAMPLE1 VALUES(1)",
            "response":[
                "\n1 row inserted.\n\n"
            ],
            "result":1
        },
        {
            "statementId":4,
            "statementType":"query",
            "statementPos":{
                "startLine":4,
                "endLine":4
            },
            "statementText":"SELECT * FROM T_EXAMPLE1",
```

```
            "response":[

            ],
            "result":1,
            "resultSet":{
                "metadata":[
                    {
                        "columnName":"COL1",
                        "jsonColumnName":"col1",
                        "columnTypeName":"NUMBER",
                        "precision":38,
                        "scale":0,
                        "isNullable":1
                    }
                ],
                "items":[
                    {
                        "col1":1
                    }
                ],
                "hasMore":false,
                "limit":1500,
                "offset":0,
                " count":1
            }
        },
        {
            "statementId":5,
            "statementType":"plsql",
            "statementPos":{
                "startLine":5,
                "endLine":8
            },
            "statementText":"BEGIN\n INSERT INTO T_EXAMPLE1
VALUES(2);\nEND;",
            "response":[
                "\nPL\/SQL procedure successfully completed.\n\n"
            ],
            "result":1
        },
        {
            "statementId":6,
            "statementType":"query",
            "statementPos":{
                "startLine":9,
                "endLine":9
            },
            "statementText":"SELECT * FROM T_EXAMPLE1",
            "response":[

            ],
            "result":1,
            "resultSet":{
                "metadata":[
                    {
```

```
                        "columnName":"COL1",
                        "jsonColumnName":"col1",
                        "columnTypeName":"NUMBER",
                        "precision":38,
                        "scale":0,
                        "isNullable":1
                    }
                ],
                "items":[
                    {
                        "col1":1
                    },
                    {
                        "col1":2
                    }
                ],
                "hasMore":false,
                "limit":1500,
                "offset":0,
                "count":2
            }
        },
        {
            "statementId":7,
            "statementType":"ddl",
            "statementPos":{
                "startLine":10,
                "endLine":10
            },
            "statementText":"DROP TABLE T_EXAMPLE1",
            "response":[
                "\nTable T_EXAMPLE1 dropped.\n\n"
            ],
            "result":1
        }
    ]
}
```


POST Requests Using application/json Content-Type

Using a JSON document as the payload enables you to define more complex requests as shown in the following sections:

- Using a File with cURL
- Specifying the Limit Value in a POST Request for Pagination
- Specifying the Offset Value in a POST Request for Pagination
- Defining Binds in a POST Request

Using a File with cURL

The following example posts a JSON document (within the `simple_query.json` file) to the REST-Enabled SQL service.

File: `simple_query.json`

```
{ "statementText":"SELECT TO_DATE('01-01-1976','dd-mm-yyyy') FROM dual;"}
```

Request: `curl -i -X POST --user DEMO:demo --data-binary "@simple_query.json" -H "Content-Type: application/json" -k https://localhost:8088/ords/demo/_/sql`

Where:

- The `statementText` holds the SQL statement or statements.
- The `Content-Type` is `application/json`.

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Frame-Options: SAMEORIGIN
Transfer-Encoding: chunked
{
    "env":{
        "defaultTimeZone":"Europe/London"
    },
    "items":[
        {
            "statementId":1,
            "statementType":"query",
            "statementPos":{
                "startLine":1,
                "endLine":1
            },
            "statementText":"SELECT TO_DATE('01-01-1976','dd-mm-yyyy')
FROM dual",
            "response":[
             ],
            "result":0,
            "resultSet":{
                "metadata":[
                    {
                        "columnName":"TO_DATE('01-01-1976','DD-MM-
YYYY')",
                        "jsonColumnName":"to_date('01-01-1976','dd-mm-
yyyy')",
                        "columnTypeName":"DATE",
                        "precision":0,
                        "scale":0,
                        "isNullable":1
                    }
                ],
                "items":[
                    {
                        "to_date('01-01-1976','dd-mm-
yyyy')":"1976-01-01T00:00:00Z"
                    }
                ],
                "hasMore":false,
```

```
                "limit":1500,
                "offset":0,
                "count":1
            }
        }
    ]
}
```

Specifying the Limit Value in a POST Request for Pagination

You can specify the `limit` value in a POST JSON request for the pagination of a large result set returned from a query.

File: `limit.json`

```
{
 "statementText": "
 WITH data(r) AS (
 SELECT 1 r FROM dual
 UNION ALL
 SELECT r+1 FROM data WHERE r < 100
 )
 SELECT r FROM data;",
 "limit": 5
}
```

Request: `curl -i -X POST --user DEMO:demo --data-binary "@limit.json" -H "Content-Type: application/json" -k https://localhost:8088/ords/demo/_/sql`

Where: The `limit` is the maximum number of rows returned from a query.

> **Note:**
>
> The maximum number of rows returned from a query is based on the `misc.pagination.maxRows` value set in `default.xml` file.

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Frame-Options: SAMEORIGIN
Transfer-Encoding: chunked
 {
    "env":{
        "defaultTimeZone":"Europe/London"
    },
    "items":[
        {
            "statementId":1,
            "statementType":"query",
            "statementPos":{
                "startLine":1,
```

```
                "endLine":1
            },
            "statementText":" WITH data(r) AS ( SELECT 1 r FROM dual
UNION ALL SELECT r+1 FROM data WHERE r < 100 ) SELECT r FROM data",
            "response":[
             ],
            "result":0,
            "resultSet":{
                "metadata":[
                    {
                        "columnName":"R",
                        "jsonColumnName":"r",
                        "columnTypeName":"NUMBER",
                        "precision":0,
                        "scale":-127,
                        "isNullable":1
                    }
                ],
                "items":[
                    {
                        "r":1
                    },
                    {
                        "r":2
                    },
                    {
                        "r":3
                    },
                    {
                        "r":4
                    },
                    {
                        "r":5
                    }
                ],
                "hasMore":true,
                "limit":5,
                "offset":0,
                "count":5
            }
        }
    ]
}
```


Related Topics

- Configuring the Maximum Number of Rows Returned from a Query

Specifying the Offset Value in a POST Request for Pagination

You can specify the `offset` value in a POST JSON request. This value specifies the first row that must be returned and is used for pagination of the result set returned from a query.

File: `offset_limit.json`

```
{
 "statementText": "
 WITH data(r) AS (
 SELECT 1 r FROM dual
 UNION ALL
 SELECT r+1 FROM data WHERE r < 100
 )
 SELECT r FROM data;",
 "offset": 25,
 "limit": 5
}
```

Request: `curl -i -X POST --user DEMO:demo --data-binary "@offset_limit.json" -H "Content-Type: application/json" -k https://localhost:8088/ords/demo/_/sql`

Where: `offset` is the first row to be returned in the result set. Typically, this is used to provide the pagination for a large result set that returns the **next** page of rows in the result set.

> **Note:**
>
> Each request made to the REST-Enabled SQL service is performed in its own transaction, which means that you cannot ensure that the rows returned will match the previous request. To avoid these risks, queries that need pagination should use the ORDER BY clause on a primary key.

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Frame-Options: SAMEORIGIN
Transfer-Encoding: chunked
 {
    "env":{
        "defaultTimeZone":"Europe/London"
    },
    "items":[
        {
            "statementId":1,
            "statementType":"query",
            "statementPos":{
                "startLine":1,
                "endLine":1
            },
            "statementText":" WITH data(r) AS ( SELECT 1 r FROM dual
UNION ALL SELECT r+1 FROM data WHERE r < 100 ) SELECT r FROM data",
            "response":[
            ],
            "result":0,
```

```
            "resultSet":{
                "metadata":[
                    {
                        "columnName":"R",
                        "jsonColumnName":"r",
                        "columnTypeName":"NUMBER",
                        "precision":0,
                        "scale":-127,
                        "isNullable":1
                    }
                ],
                "items":[
                    {
                        "r":26
                    },
                    {
                        "r":27
                    },
                    {
                        "r":28
                    },
                    {
                        "r":29
                    }
                    {
                        "r":30
                    }
                ],
                "hasMore":true,
                "limit":5,
                "offset":25,
                "count":5
            }
        }
    ]
}
```


Defining Binds in a POST Request

You can define binds in JSON format. This functionality is useful when calling procedures and functions that use binds as the parameters.

Example 6-2 Binds in POST Request

File: `binds.json`

```
{
 "statementText": "CREATE PROCEDURE TEST_OUT_PARAMETER (V_PARAM_IN INT
IN, V_PARAM_OUT INT OUT) AS BEGIN V_PARAM_OUT := V_PARAM_IN + 10; END;
/
EXEC TEST_OUT_PARAMETER(:var1, :var2)",
 "binds":[
 {"name":"var1","data_type":"NUMBER","value":10},
 {"name":"var2","data_type":"NUMBER","mode":"out"}
```

```
 ]
}
```

Request: `curl -i -X POST --user DEMO:demo --data-binary "@binds.json" -H "Content-Type: application/json" -k https://localhost:8088/ords/demo/_/sql`

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Frame-Options: SAMEORIGIN
Transfer-Encoding: chunked
 {
    "env":{
        "defaultTimeZone":"Europe/London"
    },
    "items":[
        {
            "statementId":1,
            "statementType":"plsql",
            "statementPos":{
                "startLine":1,
                "endLine":2
            },
            "statementText":"CREATE PROCEDURE TEST_OUT_PARAMETER
(V_PARAM_IN IN INT, V_PARAM_OUT OUT INT) AS BEGIN V_PARAM_OUT :=
V_PARAM_IN + 10; END;",
            "response":[
                "\nProcedure TEST_OUT_PARAMETER compiled\n\n"
            ],
            "result":0,
            "binds":[
                {
                    "name":"var1",
                    "data_type":"NUMBER",
                    "value":10
                },
                {
                    "name":"var2",
                    "data_type":"NUMBER",
                    "mode":"out",
                    "result":null
                }
            ]
        },
        {
            "statementId":2,
            "statementType":"sqlplus",
            "statementPos":{
                "startLine":3,
                "endLine":3
            },
            "statementText":"EXEC TEST_OUT_PARAMETER(:var1, :var2)",
            "response":[
                "\nPL\/SQL procedure successfully completed.\n\n"
```

```
            ],
            "result":0,
            "binds":[
                {
                    "name":"var1",
                    "data_type":"NUMBER",
                    "value":10
                },
                {
                    "name":"var2",
                    "data_type":"NUMBER",
                    "mode":"out",
                    "result":20
                }
            ]
        }
    ]
}
```

Example 6-3 Complex Bind in POST Request

Filecomplex_bind_example.json

```
{
    "statementText":"
declare
 type t is table of number index by binary_integer;
 l_in  t    := :IN;
 l_out t;
 begin
   for i in 1..l_in.count loop
  l_out(i) := l_in(i) * 2;
   end loop;
   :L_OUT := l_out;
 end;
",
    "binds":[
        {
            "name":"IN",
            "data_type":"PL/SQL TABLE",
            "type_name":"",
            "type_subname":"",
            "type_components":[
                {
                    "data_type":"NUMBER"
                }
            ],
            "value":[
                2,
                4,
                7
            ]
        },
        {
            "name":"L_OUT",
```

```
            "data_type":"PL/SQL TABLE",
            "type_name":"",
            "type_subname":"",
            "type_components":[
                {
                    "data_type":"NUMBER"
                }
            ],
            "mode":"out"
        }
    ]
}
```

Request: `curl -i -X POST --user DEMO:demo --data-binary "@complex_bind_example.json" -H "Content-Type: application/json" -k https://localhost:8088/ords/demo/_/sql`

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Frame-Options: SAMEORIGIN
Transfer-Encoding: chunked
{
    "env":{
        "defaultTimeZone":"Europe/London"
    },
    "items":[
        {
            "statementId":1,
            "statementType":"plsql",
            "statementPos":{
                "startLine":2,
                "endLine":12
            },
            "statementText":"declare \n type t is table of number index
by binary_integer;   \n l_in  t    := :IN;   \n l_out t;  \n begin
\n   for i in 1..l_in.count loop   \n  l_out(i) := l_in(i) * 2;  \n
end loop;   \n   :L_OUT := l_out;   \n end;",
            "response":[
            ],
            "result":1,
            "binds":[
                {
                    "name":"IN",
                    "data_type":"PL/SQL TABLE",
                    "type_components":[
                        {
                            "data_type":"NUMBER"
                        }
                    ],
                    "type_name":"",
                    "type_subname":"",
                    "value":[
                        2,
```

```
                    4,
                    7
                ]
            },
            {
                "name":"L_OUT",
                "data_type":"PL/SQL TABLE",
                "mode":"out",
                "type_components":[
                    {
                        "data_type":"NUMBER"
                    }
                ],
                "type_name":"",
                "type_subname":"",
                "result":[
                    4,
                    8,
                    14
                ]
            }
        ]
    }
  ]
}
```

Specifying Batch Statements in a POST Request

This section shows the examples with batch statements and batch bind values in a POST request.

Example 6-4 Batch statements

File: batch_example.json

```
{
    "statementText":[
        "insert into adhoc_table_simple values(1)",
        "insert into adhoc_table_simple values(2)",
        "delete from adhoc_table_simple"
    ]
}
```

Request :`curl -i -X POST --user DEMO:demo --data-binary "@batch_example.json" -H "Content-Type: application/json" -k https://localhost:8088/ords/demo/_/sql`

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Frame-Options: SAMEORIGIN
```

```
Transfer-Encoding: chunked
{
    "env":{
        "defaultTimeZone":"Europe/London"
    },
    "items":[
        {
            "statementId":1,
            "statementType":"dml",
            "statementPos":{
                "startLine":0,
                "endLine":0
            },
            "statementText":[
                "insert into adhoc_table_simple values(1)",
                "insert into adhoc_table_simple values(2)",
                "delete from adhoc_table_simple"
            ],
            "response":[
                "\n1 row inserted.\n\n",
                "\n1 row inserted.\n\n",
                "\n2 rows inserted.\n\n"
            ],
            "result":[
                1,
                1,
                2
            ]
        }
    ]
}
```

Example 6-5 Batch bind values

File: batch_bind_example.json

```
{
    "statementText":"INSERT INTO ADHOC_TABLE_DATE VALUES(?,?)",

    "binds":[
        {
            "index":1,
            "data_type":"NUMBER",
            "batch":true,
            "value":[
                3,
                6,
                9,
                13,
                17
            ]
        },
        {
            "index":2,
```

```
            "data_type":"DATE",
            "batch":true,
            "value":[
                "2017-02-21T06:12:20Z",
                "2017-02-21T06:12:20Z",
                "2017-02-21T06:12:20Z",
                "2017-02-21T06:12:20Z",
                "2017-02-21T06:12:20Z"
            ]
        }
    ]
}
```

Request: `curl -i -X POST --user DEMO:demo --data-binary "@batch_bind_example.json" -H "Content-Type: application/json" -k https://localhost:8088/ords/demo/_/sql`

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Frame-Options: SAMEORIGIN
Transfer-Encoding: chunked
{
    "env":{
        "defaultTimeZone":"Europe/London"
    },
    "items":[
        {
            "statementId":1,
            "statementType":"dml",
            "statementPos":{
                "startLine":1,
                "endLine":2
            },
            "statementText":"INSERT INTO ADHOC_TABLE_DATE VALUES(?,?)",
            "response":[
                "\n1 row inserted.\n\n",
                "\n1 row inserted.\n\n",
                "\n1 row inserted.\n\n",
                "\n1 row inserted.\n\n",
                "\n1 row inserted.\n\n"
            ],
            "result":[
                1,
                1,
                1,
                1,
                1
            ],
            "binds":[
                {
                    "index":1,
                    "data_type":"NUMBER",
                    "batch":true,
```

```
                    "value":[
                        3,
                        6,
                        9,
                        13,
                        17
                    ]
                },
                {
                    "index":2,
                    "data_type":"DATE",
                    "batch":true,
                    "value":[
                        "2017-02-21T06:12:20Z",
                        "2017-02-21T06:12:20Z",
                        "2017-02-21T06:12:20Z",
                        "2017-02-21T06:12:20Z",
                        "2017-02-21T06:12:20Z"
                    ]
                }
            ]
        }
    ]
}
```


Example POST Request with DATE and TIMESTAMP Format

Example 6-6 Oracle REST Data services Time Zone Set as Europe/London

Oracle Database DATE and TIMESTAMP data types do not have a time zone associated with them. The DATE and TIMESTAMP values are associated with the time zone of the application. Oracle REST Data Services and the REST- Enabled SQL service return values in a JSON format. The standard for JSON is to return date and timestamp values using the UTC Zulu format. Oracle REST Data Services and the REST- Enabled SQL service return Oracle Database DATE and TIMESTAMP values in the Zulu format using the time zone in which Oracle REST Data Services is running.

Oracle recommends running Oracle REST Data Services using the UTC time zone to make this process easier.

File: `date.json`

```
{
    "statementText":"SELECT TO_DATE('2016-01-01 10:00:03','yyyy-mm-
dd hh24:mi:ss' ) winter, TO_DATE('2016-07-01 10:00:03','yyyy-mm-dd
hh24:mi:ss' ) summer FROM dual;"
}
```

Request: `curl -i -X POST --user DEMO:demo --data-binary "@date.json" -H "Content-Type: application/json" -k https://localhost:8088/ords/demo/_/sql`

Response:

> **Note:**
>
> In this example, both DATE values are specified as 10 a.m. The "summer" value is returned as 9 a.m. Zulu time. This is due to British Summer Time.

```
HTTP/1.1 200 OK
Date: Wed, 26 Jul 2017 14:59:27 GMT
Content-Type: application/json
X-Frame-Options: SAMEORIGIN
Transfer-Encoding: chunked
Server: Jetty(9.2.21.v20170120)
{
    "env":{
        "defaultTimeZone":"Europe/London"
    },
    "items":[
        {
            "statementId":1,
            "statementType":"query",
            "statementPos":{
                "startLine":1,
                "endLine":1
            },
            "statementText":"SELECT TO_DATE('2016-01-01 10:00:03','yyyy-
mm-dd hh24:mi:ss' ) winter, TO_DATE('2016-07-01 10:00:03','yyyy-mm-dd
hh24:mi:ss' ) summer FROM dual",
            "response":[
             ],
            "result":0,
            "resultSet":{
                "metadata":[
                    {
                        "columnName":"WINTER",
                        "jsonColumnName":"winter",
                        "columnTypeName":"DATE",
                        "precision":0,
                        "scale":0,
                        "isNullable":1
                    },
                    {
                        "columnName":"SUMMER",
                        "jsonColumnName":"summer",
                        "columnTypeName":"DATE",
                        "precision":0,
                        "scale":0,
                        "isNullable":1
                    }
                ],
                "items":[
                    {
                        "winter":"2016-01-01T10:00:03Z",
                        "summer":"2016-07-01T09:00:03Z"
                    }
```

```
                ],
                "hasMore":false,
                "limit":1500,
                "offset":0,
                "count":1
            }
        }
    ]
}
```

Data Types and Formats Supported

The following code snippet shows the different data types and the formats supported:

```
{

"statementText":"SELECT ?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,
?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,? FROM dual",
    "binds":[
        {
            "index":1,
            "data_type":"NUMBER",
            "value":1233
        },
        {
            "index":2,
            "data_type":"NUMERIC",
            "value":123
        },
        {
            "index":3,
            "data_type":"DECIMAL",
            "value":123
        },
        {
            "index":4,
            "data_type":"DEC",
            "value":123
        },
        {
            "index":5,
            "data_type":"NUMBER",
            "value":123
        },
        {
            "index":6,
            "data_type":"INTEGER",
            "value":123
        },
        {
            "index":7,
            "data_type":"INT",
            "value":123
        },
```

```
{
    "index":8,
    "data_type":"SMALLINT",
    "value":123
},
{
    "index":9,
    "data_type":"FLOAT",
    "value":123
},
{
    "index":10,
    "data_type":"DOUBLE PRECISION",
    "value":123
},
{
    "index":11,
    "data_type":"REAL",
    "value":123
},
{
    "index":12,
    "data_type":"BINARY_FLOAT",
    "value":123
},
{
    "index":13,
    "data_type":"BINARY_DOUBLE",
    "value":123
},
{
    "index":14,
    "data_type":"CHAR",
    "value":"abc"
},
{
    "index":15,
    "data_type":"CHARACTER",
    "value":"abc"
},
{
    "index":16,
    "data_type":"VARCHAR",
    "value":"abc"
},
{
    "index":17,
    "data_type":"VARCHAR2",
    "value":"abc"
},
{
    "index":18,
    "data_type":"CHAR VARYING",
    "value":"abc"
},
```

```
{
    "index":19,
    "data_type":"CHARACTER VARYING",
    "value":"abc"
},
{
    "index":20,
    "data_type":"NCHAR",
    "value":"abc"
},
{
    "index":21,
    "data_type":"NATIONAL CHAR",
    "value":"abc"
},
{
    "index":22,
    "data_type":"NATIONAL CHARACTER",
    "value":"abc"
},
{
    "index":23,
    "data_type":"NVARCHAR",
    "value":"abc"
},
{
    "index":24,
    "data_type":"NVARCHAR2",
    "value":"abc"
},
{
    "index":25,
    "data_type":"NCHAR VARYING",
    "value":"abc"
},
{
    "index":26,
    "data_type":"NATIONAL CHAR VARYING",
    "value":"abc"
},
{
    "index":27,
    "data_type":"NATIONAL CHARACTER VARYING",
    "value":"abc"
},
{
    "index":28,
    "data_type":"DATE",
    "value":"01-Jan-2016"
},
{
    "index":29,
    "data_type":"TIMESTAMP",
    "value":"1976-02-01T00:00:00Z"
},
```

```
{
    "index":30,
    "data_type":"TIMESTAMP",
    "value":"1976-02-01T00:00:00Z"
},
{
    "index":31,
    "data_type":"TIMESTAMP WITH LOCAL TIME ZONE",
    "value":"1976-02-01T00:00:00Z"
},
{
    "index":32,
    "data_type":"TIMESTAMP WITH TIME ZONE",
    "value":"1976-02-01T00:00:00Z"
},
{
    "index":33,
    "data_type":"INTERVALYM",
    "value":"P10Y10M"
},
{
    "index":34,
    "data_type":"INTERVAL YEAR TO MONTH",
    "value":"P10Y10M"
},
{
    "index":35,
    "data_type":"INTERVAL YEAR(2) TO MONTH",
    "value":"P10Y10M"
},
{
    "index":36,
    "data_type":"INTERVALDS",
    "value":"P11DT10H10M10S"
},
{
    "index":37,
    "data_type":"INTERVAL DAY TO SECOND",
    "value":"P11DT10H10M10S"
},
{
    "index":38,
    "data_type":"INTERVAL DAY(2) TO SECOND(6)",
    "value":"P11DT10H10M10S"
},
{
    "index":39,
    "data_type":"ROWID",
    "value":1
},
{
    "index":40,
    "data_type":"RAW",
    "value":"AB"
},
```

```
            {
                "index":41,
                "data_type":"LONG RAW",
                "value":"AB"
            },
            {
                "index":42,
                "data_type":"CLOB",
                "value":"clobvalue"
            },
            {
                "index":43,
                "data_type":"NCLOB",
                "value":"clobvalue"
            },
            {
                "index":45,
                "data_type":"LONG",
                "value":"A"
            }
        ]

}
```

REST-Enabled SQL Request and Response Specifications

The following sections provide REST-Enabled SQL request and response specifications:

- Request Specification
- Response Specification

Request Specification

Request Specification for application/sql

The body of the request is in plain UTF8 text. Statements can be separated by their usual SQL*Plus terminator.

Specification for application/json

JSONPath	Type	Description	Example	Default Value	Possible Values
`$.statementText`	String	Specifies the SQL statements to execute.	`"select 1 from dual"`	Not applicable	Not applicable

JSONPath	Type	Description	Example	Default Value	Possible Values
`$.statementText`	Array	Specifies batch DML statements using an array. One DML statement is specified per string in an array.	`[ "insert into test1 values(1)", "update test1 set col1=2" ]`	Not applicable	Not applicable
`$.offset`	Number	Specifies the number of rows to offset the query result. This is used for pagination of the result set returned from a query.	`25`	`0`	Between 0 to `misc.pagination.maxRows`.
`$.limit`	Number	Specifies the maximum number of rows returned from a query. Values greater than the value of the `misc.pagination.maxRows` property, specified in the `default.xml`, is ignored.	`500`	`misc.pagination.maxRows`	Between 0 to `misc.pagination.maxRows`.
`$.binds`	Array	Specifies an array of objects specifying the bind information.	`"binds": [ { "name": "mybind1", "data_type":"NUMBER", "mode":"out" }, { "name":"mybind2", "data_type":"NUMBER", "value":7 } ]`	Not applicable	Not applicable
`$.binds[*].name`	String	Specifies the name of the bind, when you are using named notation.	`"mybind"`	Not applicable	Not applicable
`$.binds[*].index`	Number	Specifies the index of bind, when you are using positional notation.	`1`	Not applicable	`Between 1 to n`

JSONPath	Type	Description	Example	Default Value	Possible Values
`$.binds[*].data_type`	String	Specifies Oracle data type of the bind.	`"NUMBER"`	Not applicable	For more information, refer to Oracle Built-in Types
`$.binds[*].value`	Any value	Specifies the value of the bind.	`"value to insert"`	`null`	Can be one of the following data-types: • Number • String • Array For more information, refer to Oracle Built-in Types
`$.binds[*].mode`	String	Specifies the mode in which the bind is used.	`"out"`	`"in"`	`[ "in" , "inout", "out" ]`
`$.binds[*].batch`	Boolean	Specifies whether or not you want to perform a batch bind. If you want to perform a batch bind, then set the value to `true`. If the value is set to `true`, then `$binds[*]` must consist of an array of values.	`true`	`false`	`[ true, false ]`
`$.binds[*].type_name`	String	Required when you are using `$binds[*].data_type = "PL/SQL TABLE"` Currently, only an empty string is accepted as the value.	`""`	Not applicable	Not applicable
`$.binds[*].type_subname`	String	Required when you are using `$binds[*].data_type = "PL/SQL TABLE"` Currently, only an empty string is accepted as the value.	`""`	Not applicable	Not applicable

JSONPath	Type	Description	Example	Default Value	Possible Values
`$.binds[*].type_components`	Array	Specifies an array of data types in the PL/SQL TABLE Required when you are using `$binds[*].data_type = "PL/SQL TABLE"`	`[{"data_type":"NUMBER"}]`	Not applicable	Not applicable
`$.binds[*].type_components[*].data_type`	String	Specifies Oracle data type of a column in the PL/SQL TABLE. Required when you are using `$binds[*].data_type = "PL/SQL TABLE"`	`"NUMBER"`	Not applicable	For more information, refer to Oracle Built-in Types

Response Specification

JSONPath	Data type	Description	Example Values	Possible values
`$.env`	Object	Specifies the information about the Oracle REST Data Services environment.	Not applicable	Not applicable
`$.env.defaultTimeZone`	String	Specifies the timezone in which Oracle REST Data Services server is running on.	`"Europe/London"`	Not applicable
`$.items`	Array	Specifies that there is one item for each statement executed.	Not applicable	Not applicable
`$.items[*].statementId`	Number	Specifies the sequence number of the statement.	`1`	Not applicable
`$.items[*].statementType`	String	Specifies the type of statement.	`"query"`	["query" , "dml", "ddl", "plsql" , "sqlplus" , "ignore", "transaction-control", "session-control", "system-control", "jdbc", "other"]

JSONPath	Data type	Description	Example Values	Possible values
`$.items[*].statementPos`	Object	Specifies information about the position of a specified statement.	Not applicable	Not applicable
`$.items[*].statementPos.startLine`	Number	Specifies start line of the statement.	Not applicable	Not applicable
`$.items[*].statementPos.endLine`	Number	Specifies end line of the statement.	Not applicable	Not applicable
`$items[*].statementText`	String	Specifies the SQL statement to be executed.	`"select 1 from dual"`	Not applicable
`$items[*].statementText`	Array	Specifies batch DML statements can be specified using an array. One DML statement specified per string in an array.	`[ "insert into test1 values(1) ","update test1 set col1=2" ]`	Not applicable
`$.items[*].response`	Array	Specifies array of Strings. The response generated when running the statement.	`[ "\n1 row inserted.\n\n" ]`	Not applicable
`$.items[*].result`	Number	Specifies the result generated when running the statement. For DML statements, this will be the number of rows affected.	`5`	Not applicable
`$.items[*].result`	Array	Specifies the result generated when running each of the batch statements. For DML statements, this will be the number of rows affected.	`[ 1, 1, 2 ]`	Not applicable
`$.items[*].resultSet`	Object	Specifies information about the result set generated from a query.	Not applicable	Not applicable

JSONPath	Data type	Description	Example Values	Possible values
`$.items[*].resultSet.metadata`	Array	Specifies each object in the array provides information about the metadata of a column.	Not applicable	Not applicable
`$.items[*].resultSet.metadata[*].columnName`	String	Specifies the name of the column used in the Oracle Database.	Not applicable	Not applicable
`$.items[*].resultSet.metadata[*].jsonColumnName`	String	Specifies the name of the column used in `$.items[*].resultSet.items[*].<columnname>`	Not applicable	Not applicable
`$.items[*].resultSet.metadata[*].columnTypeName`	String	Specifies the Oracle Database data type of the column.	Not applicable	Not applicable
`$.items[*].resultSet.metadata[*].precision`	Number	Specifies the precision of the column.	Not applicable	Not applicable
`$.items[*].resultSet.metadata[*].scale`	Number	Specifies the scale of the column.	Not applicable	Not applicable
`$.items[*].resultSet.metadata[*].isNullable`	Number	Specifies whether the column is nullable or not. 0, if the column is not nullable. 1, if the column is nullable.	Not applicable	Not applicable
`$.items[*].resultSet.items`	Array	Specifies the list of all rows returned in the result set.	Not applicable	Not applicable
`$.items[*].resultSet.items[*].<columnname>`	Any type	Specifies the value of a particular column and row in the result set.	Not applicable	Not applicable

JSONPath	Data type	Description	Example Values	Possible values
`$.items[*].resultSet.hasMore`	Boolean	Specifies whether result set has more rows. Value is set to `true` if the result set has more rows, otherwise set to `false`. The rows in the result set depend on `misc.pagination.maxRows` value configured in `defaults.xml` file or as specified in the request.	`false`	[true , false]
`$.items[*].resultSet.count`	Number	Specifies the number of rows returned.	Not applicable	Not applicable
`$.items[*].resultSet.offset`	Number	Specifies the number of rows to offset the query result. This is used for pagination of the result set returned from a query.	25	Between 0 to `misc.pagination.maxRows`
`$.items[*].resultSet.limit`	Number	Specifies the maximum number of rows returned from a query. Values greater than `misc.pagination.maxRows` value specified in `default.xml` file are ignored.	`500`	Between 0 to `misc.pagination.maxRows`

JSONPath	Data type	Description	Example Values	Possible values
`$.items[*].binds`	Array	Specifies an array of objects specifying the bind information.	`"binds": [ { "name":"mybind1", "data_type":"NUMBER", "mode":"out" }, { "name": "mybind2", "data_type":"NUMBER", "value":7 } ]`	Not applicable
`$.items[*].binds[*].name`	String	Specifies the name of the bind, when you are using named notation.	`"mybind"`	Not applicable
`$.items[*].binds[*].index`	Number	specifies iIndex of bind, when you are using positional notation.	`1`	`1 - n`
`$.items[*].binds[*].data_type`	String	Specifies the Oracle data type of the bind.	`"NUMBER"`	For more information, refer to Oracle Built-in Types
`$.items[*].binds[*].value`	Any type	Specifies the value of the bind.	`"value to insert"`	Can be one of the following data types: • Number • String • Array For more information, refer to Oracle Built-in Types
`$.items[*].binds[*].result`	Any type	Specifies the result of an OUT bind.	Not applicable	Not applicable
`$.items[*].binds[*].mode`	String	Specifies the mode in which the bind is used.	`"out"`	`[ "in" , "inout", "out" ]`

JSONPath	Data type	Description	Example Values	Possible values
`$.items[*].binds[*].batch`	Boolean	Specifies whether or not you want to perform a batch bind. If you want to perform a batch bind, then set the value to `true`. If a batch bind is to be performed, then the value is set to `true`. If the value is set to `true`, then `$binds[*]` value must be an array of values.	`true`	`[ true, false ]`
`$.items[*].binds[*].type_name`	String	Required when using `$binds[*].data_type = "PL/SQL TABLE"`. Currently, only an empty string is accepted as the value.	`""`	Not applicable
`$.items[*].binds[*].type_subname`	String	Required when using `$binds[*].data_type = "PL/SQL TABLE"`. Currently, only an empty string is accepted as the value.	`""`	Not applicable
`$.items[*].binds[*].type_components`	Array	Array of data types in the PL/SQL TABLE Required when using `$binds[*].data_type = "PL/SQL TABLE"`.	`[{"data_type":"NUMBER"}]`	Not applicable
`$.items[*].binds[*].type_components[*].data_type`	String	The Oracle data type of a column in the PL/SQL TABLE. Required when using `$binds[*].data_type = "PL/SQL TABLE"`	`"NUMBER"`	For more information, refer to Oracle Built-in Types

Supported SQL, SQL*Plus, and SQLcl Statements

This section lists all the supported SQL, SQL*Plus and SQLcl statements for REST-Enabled SQL service.

Topics

- Supported SQL Statements
- Supported PL/SQL Statements
- Supported SQL*Plus Statements
- Supported SQLcl Statements

Supported SQL Statements

This section describes the SQL statements that the REST- Enabled SQL service supports.

REST- Enabled SQL service supports all SQL commands. If the specified Oracle Database schema has the appropriate privileges, then you can run them. Oracle REST Data Services makes all queries into in-line views before execution to provide pagination support. Queries are made in-line irrespective of the format in which you provide the query. All the other nonquery SQL statements are executed as they are.

In-line views have the following limitations:

- All column names in a query must be unique because the views and in-line views cannot have ambiguous column names.
- Cursor expressions are not displayed in view or in-line views.
- WITH FUNCTION clause is not supported in in-line views.

Related Topics

- SQL_statements_ref

Supported PL/SQL Statements

The REST- Enabled SQL service supports PL/SQL statements and blocks.

Example 6-7 PL/SQL Statement

```
DECLARE v_message VARCHAR2(100) := 'Hello World';
BEGIN
  FOR i IN 1..3 LOOP
    DBMS_OUTPUT.PUT_LINE (v_message);
  END LOOP;
END;
/
```

Related Topics

- plsql_block

Supported SQL*Plus Statements

This section lists all the SQL*Plus statements that the REST- Enabled SQL service supports.

REST- Enabled SQL service supports most of the SQL*Plus statements except those statements that are related to formatting. The specific Oracle Database schema must have the appropriate privileges to run the SQL*Plus statemments.

The following is a list of supported SQL*Plus statements:

- `SET system_variable value`

  > **Note:**
  >
  > `system_variable` and `value` represent one of the clauses described in Set System Variables section.

- `/ (slash)`
- `DEF[INE] [variable] | [variable = text]`
- `DESC[RIBE] {[schema.]object[@connect_identifier]}`
- `EXEC[UTE] statement`
- `HELP | ? [topic]`
- `PRINT [variable ...]`
- `PRO[MPT] [text]`
- `REM[ARK]`
- `SHO[W] [option]`
- `TIMI[NG] [START text | SHOW | STOP]`
- `UNDEF[INE] variable ...`
- `VAR[IABLE] [variable [type][=value]]`

Related Topics

- sqlplus_commands

Set System Variables

The following is a list of possible values for `system_variable` and `value`:

> **Note:**
>
> The command `SET CMDS[EP] {; | c | ON | OFF}` is obsolete.

- `SET APPI[NFO]{ON | OFF | text}`
- `SET AUTOP[RINT] {ON | OFF}`

- SET AUTOT[RACE] {ON | OFF | TRACE[ONLY]} [EXP[LAIN]] [STAT[ISTICS]]
- SET BLO[CKTERMINATOR] {. | c | ON | OFF}
- SET CMDS[EP] {; | c | ON | OFF}
- SET COLINVI[SIBLE] [ON | OFF]
- SET CON[CAT] {. | c | ON | OFF}
- SET COPYC[OMMIT] {0 | n}
- SET DEF[INE] {& | c | ON | OFF}
- SET DESCRIBE [DEPTH {1 | n | ALL}] [LINENUM {ON | OFF}] [INDENT {ON | OFF}]
- SET ECHO {ON | OFF}
- SET ERRORL[OGGING] {ON | OFF} [TABLE [schema.]tablename] [TRUNCATE] [IDENTIFIER identifier]
- SET ESC[APE] {\ | c | ON | OFF}
- SET FEED[BACK] {6 | n | ON | OFF | ONLY}]
- SET SERVEROUT[PUT] {ON | OFF} [SIZE {n | UNL[IMITED]}] [FOR[MAT] {WRA[PPED] | WOR[D_WRAPPED] | TRU[NCATED]}]
- SET SHOW[MODE] {ON | OFF}
- SET SQLBL[ANKLINES] {ON | OFF}
- SET SQLP[ROMPT] {SQL> | text}
- SET TI[ME] {ON | OFF}
- SET TIMI[NG] {ON | OFF}
- SET VER[IFY] {ON | OFF}

Related Topics

- set-system_var_summary

Show System Variables

This section lists the possible values for `option` which is either a term or a clause used in the `SHO[W] option` command.

The following is a list of possible values for the `option` variable:

> **Note:**
>
> The commands `SHOW CMDSEP` and `SHOW DESCR[IBE]` are obsolete.

- SHOW system_variable
- SHOW EDITION
- SHOW ERR[ORS] [{ ANALYTIC VIEW | ATTRIBUTE DIMENSION | HIERARCHY | FUNCTION | PROCEDURE | PACKAGE | PACKAGE BODY | TRIGGER | VIEW | TYPE | TYPE BODY | DIMENSION | JAVA CLASS } [schema.]name]

- SHOW PDBS
- SHOW SGA
- SHOW SQLCODE
- SHOW COLINVI[SIBLE]
- SHOW APPIN[FO]
- SHOW AUTOT[RACE]
- SHOW BINDS
- SHOW BLO[CK TERMINATOR]
- SHOW CMDSEP
- SHOW COPYTYPECHECK
- SHOW COPYCOMMIT
- SHOW DEFINE
- SHOW DEFINES
- SHOW DESCR[IBE]
- SHOW ECHO
- SHOW EDITION
- SHOW ERRORL[OGGING]
- SHOW ESC[APE]
- SHOW FEEDBACK
- SHOW CONCAT
- SHOW SHOW[MODE]
- SHOW RECYC[LEBIN]
- SHOW RELEASE
- SHOW SQLBL[ANKLINES]
- SHOW SCAN
- SHOW SERVEROUT[PUT]
- SHOW SPACE
- SHOW TABLES
- SHOW TIMI[NG]
- SHOW USER
- SHOW VER[IFY]
- SHOW XQUERY

Related Topics

- show_command

Supported SQLcl Statements

This section lists the SQLcl statements that the REST- Enabled SQL service supports.

REST- Enabled SQL service supports some of the SQLcl statements. The specific Oracle Database schema must have the appropriate privileges to run the SQLcl statements.

The following is a list of supported SQLcl statements:

- `CTAS`
- `DDL`
- `SET DDL`

7

Migrating from mod_plsql to ORDS

This chapter demonstrates how a mod_plsql application is migrated to Oracle REST Data Services (ORDS).

Oracle REST Data Services is a Java EE-based alternative for Oracle HTTP Server and mod_plsql. An Oracle HTTP Server mod_plsql application can be migrated to ORDS by defining new ORDS configuration files. The mod_plsql database resources such as before procedures, after procedures, request validation functions, owa_custom packages, doc upload procedures and doc tables require no change when you are migrating to ORDS.

Topics:

- Oracle HTTP Server mod_plsql Authentication
- Example Oracle HTTP Server DAD file
- Mapping mod_plsql Settings to ORDS
- Example ORDS Configuration Files
- Example ORDS URL Mapping
- Example ORDS Default Configuration
- ORDS Authentication
- ORDS Features

Oracle HTTP Server mod_plsql Authentication

Oracle HTTP Server mod_plsql applications are configured in a database access descriptor (DAD) file.

The following example mod_plsql application provides the methods to authenticate the requests against the Oracle Database:

- **Basic authentication**: The username and password are stored in the DAD file and so the end user is not required to log in. This method is useful for web pages that provide public information.
- **Basic dynamic authentication**: The users provide credentials in a browser HTTP basic authentication dialog box. The only way to log out is to close all the instances of the browser.
- **Custom authentication**: Enables applications to invoke a user-written authentication function to authenticate the users within the application and not at the database level.

Related Topics

- Oracle HTTP Server mod_plsql

Example Oracle HTTP Server DAD file

This section provides an example Oracle HTTP Server DAD file.

The following `dads.conf` file includes three locations demonstrating the basic, basic dynamic and custom authentications and the following directives:

- `PlsqlBeforeProcedure`
- `PlsqlAfterProcedure`
- `PlsqlRequestValidationFunction`
- `PlsqlDocumentTablename`
- `PlsqlDocumentProcedure`

Example 7-1 dads.conf file

```
#
================================================================================
====
#                     mod_plsql DAD Configuration File
#
================================================================================
====
<Location /pls/basic_auth>
  SetHandler pls_handler
  Order deny,allow
  Allow from all
  AllowOverride                     None
  PlsqlDatabaseUsername             PRIVILEGED_USER
  PlsqlDatabasePassword             passwordF0R$0RD5Example
  PlsqlDatabaseConnectString        oracle-ee:1521:ORCLPDB1
ServiceNameFormat
  PlsqlAuthenticationMode           Basic
  PlsqlBeforeProcedure              sample_plsql_app_metadata.beforeProc
  PlsqlAfterProcedure               sample_plsql_app_metadata.afterProc
  PlsqlRequestValidationFunction
sample_plsql_app_metadata.validationFunc
  PlsqlDocumentTablename            privileged_user.doc_table
  PlsqlDocumentProcedure            privileged_user.upload
</Location>
<Location /pls/basic_dynamic_auth>
  SetHandler pls_handler
  Order deny,allow
  Allow from all
  AllowOverride                     None
  PlsqlDatabaseConnectString        oracle-ee:1521:ORCLPDB1
ServiceNameFormat
  PlsqlAuthenticationMode           Basic
  PlsqlBeforeProcedure              sample_plsql_app_metadata.beforeProc
  PlsqlAfterProcedure               sample_plsql_app_metadata.afterProc
  PlsqlRequestValidationFunction
sample_plsql_app_metadata.validationFunc
```

```
</location>
<Location /pls/custom_auth>
  SetHandler pls_handler
  Order deny,allow
  Allow from all
  AllowOverride                     None
  PlsqlDatabaseUsername             PRIVILEGED_USER
  PlsqlDatabasePassword             passwordF0R$0RD5Example
  PlsqlDatabaseConnectString        oracle-ee:1521:ORCLPDB1
ServiceNameFormat
  PlsqlAuthenticationMode           CustomOwa
  PlsqlBeforeProcedure              sample_plsql_app_metadata.beforeProc
  PlsqlAfterProcedure               sample_plsql_app_metadata.afterProc
  PlsqlRequestValidationFunction
sample_plsql_app_metadata.validationFunc
</location>
```

Mapping mod_plsql Settings to ORDS

This section shows the mappings of mod_plsql settings to ORDS.

ORDS allows you to specify configuration files that are similar to a location defined in an Oracle HTTP Server mod_plsql DAD file. Each configuration file is defined in `ords_conf/ords/conf` directory and the configuration file is then mapped to a particular URL using the `ords_conf/ords/url-mapping.xml` file. ORDS provides the following configurable parameters that can be used when migrating mod_plsql directives:

Table 7-1 Mappings of mod_plsql Directives to ORDS Settings

mod_plsql Setting	ORDS Setting	Description
`PlsqlDatabaseUserName`	`db.username`	Specifies the username to use to log in to the database. ORDS and mod_plsql are equivalent.
`PlsqlDatabasePassword`	`db.password`	Specifies the password to use to log in to the database. ORDS and mod_plsql are equivalent.
`PlsqlDatabaseConnectString`	Multiple Settings such as: • `db.hostname` • `db.port` • `db.servicename` • `db.sid`	Specifies the connection to an Oracle database. ORDS and mod_plsql are equivalent.

Table 7-1 (Cont.) Mappings of mod_plsql Directives to ORDS Settings

mod_plsql Setting	ORDS Setting	Description
`PlsqlAuthenticationMode`	`security.requestAuthenticationFunction`	Specifies the authentication mode to use to allow access. When `security.requestAuthenticationFunction` is not specified, ORDS behavior is same as `Basic` mode of mod_plsql. When `security.requestAuthenticationFunction` is specified, ORDS can perform the same action as example dad directive `PlsqlAuthenticationMode CustomOwa`of mod_plsql. Example ORDS equivalent configuration parameter: `<entry key="security.requestAuthenticationFunction">privileged_user.owa_custom.authorize</entry>` ORDS and mod_plsql are equivalent.
`PlsqlBeforeProcedure`	`procedure.preProcess`	Specifies the procedure to be invoked before calling the requested procedure. ORDS and mod_plsql are equivalent.
`PlsqlAfterProcedure`	`procedure.postProcess`	Specifies the procedure to be invoked after calling the requested procedure. ORDS and mod_plsql are equivalent.
`PlsqlRequestValidationFunction`	`security.requestValidationFunction`	Specifies an application-defined PL/SQL function that can allow or disallow further processing of the requested procedure. ORDS and mod_plsql are equivalent.
`PlsqlDocumentTablename`	`owa.docTable`	Specifies the table in the database to which all documents are uploaded. ORDS and mod_plsql are equivalent.

Table 7-1 (Cont.) Mappings of mod_plsql Directives to ORDS Settings

mod_plsql Setting	ORDS Setting	Description
`PlsqlDocumentProcedure`	N/A	Specifies the procedure to call when a document download is initiated. In ORDS the document procedure is the requested resource. It is not defined in the configuration file. ORDS and mod_plsql are equivalent.
`PlsqlDocumentPath`	N/A	ORDS has no equivalent.
`PlsqlDefaultPage`	`misc.defaultPage`	Specifies the default procedure to call if none is specified in the URL. ORDS and mod_plsql are equivalent.
`PlsqlErrorStyle`	`debug.printDebugToScreen`	Specifies the error reporting mode for mod_plsql errors. `debug.printDebugToScreen` is equivalent to `PlsqlErrorStyle DebugStyle`, otherwise there is no equivalent. ORDS and mod_plsql are equivalent.
`PlsqlExclusionList`	`security.exclusionList`	Specifies a pattern for procedures, packages, or schema names which are forbidden to be directly run from a browser. See Understanding Configurable Parameters. ORDS and mod_plsql are equivalent.
`PlsqlIdleSessionCleanupInterval`	`jdbc.InactivityTimeout`	Specifies the time (in minutes) in which the idle database sessions should be closed and cleaned. Value can be 0 to N seconds. Where, 0 (default) means that the idle connections are not removed from pool. ORDS and mod_plsql are equivalent.

Table 7-1 (Cont.) Mappings of mod_plsql Directives to ORDS Settings

mod_plsql Setting	ORDS Setting	Description
`PlsqlMaxRequestsPerSession`	`jdbc.MaxConnectionReuseCount`	Specifies the maximum number of requests a pooled database connection should service before it is closed and re-opened. Default value is 1000. ORDS and mod_plsql are equivalent.
`PlsqlInfoLogging`	N/A	See Understanding Configurable Parameters.
`PlsqlLogDirectory`	N/A	See Understanding Configurable Parameters.
`PlsqlLogEnable`	N/A	See Understanding Configurable Parameters.
`PlsqlSessionStateManagement`	N/A	Specifies how package and session state should be cleaned up at the end of each request. ORDS always performs: `dbms_session.modify_package_state(dbms_session.reinitialize)` at the end of each request.
`PlsqlAlwaysDescribeProcedure`	N/A	Specifies whether the mod_plsql application should describe a procedure before trying to run it. ORDS always describes procedure on first access, and then the definition is cached. Changes in signature are detected and recached.
`PlsqlConnectionValidation`	N/A	Specifies the mechanism the mod_plsql module should use to detect terminated connections in its connection pool. ORDS always validates connections on borrow.
`PlsqlFetchBufferSize`	N/A	Specifies the number of rows of content to fetch from the database for each trip, using either `owa_util.get_page` or `owa_util.get_page_raw`. ORDS materializes results as a 32K VARCHAR or CLOB if results are greater than 32K, so not applicable.

Table 7-1 (Cont.) Mappings of mod_plsql Directives to ORDS Settings

mod_plsql Setting	ORDS Setting	Description
`PlsqlNLSLanguage`	N/A	Specifies the NLS_LANG variable. ORDS, Java, and JDBC use unicode.
`PlsqlTransferMode`	N/A	`PlsqlTransferMode` specifies the transfer mode for data from the database back to the mod_plsql application. ORDS always uses unicode.
`PlsqlBindBucketLengths`	N/A	Specifies the rounding size to use while binding the number of elements in a collection bind. Rarely used in mod_plsql, and JDBC has no equivalent concept.
`PlsqlBindBucketWidths`	N/A	Specifies the rounding size to use while binding the number of elements in a collection bind. Rarely used in mod_plsql and JDBC has no equivalent concept.
`PlsqlCacheCleanupTime`	N/A	ORDS has no equivalent.
`PlsqlDMSEnable`	N/A	ORDS does not support DMS.
`PlsqlSessionCookieName`	N/A	ORDS does not offer session management for PL/SQL Gateway calls.
`PlsqlCacheDirectory`	N/A	ORDS has no equivalent.
`PlsqlCacheEnable`	N/A	ORDS has no equivalent.
`PlsqlCacheMaxAge`	N/A	ORDS has no equivalent.
`PlsqlCacheMaxSize`	N/A	ORDS has no equivalent.
`PlsqlCacheTotalSize`	N/A	ORDS has no equivalent.
`PlsqlCGIEnvironmentList`	N/A	ORDS has no equivalent.
`PlsqlConnectionTimeout`	N/A	ORDS has no equivalent.
`PlsqlPathAlias`	N/A	ORDS has no equivalent.
`PlsqlPathAliasProcedure`	N/A	ORDS has no equivalent.
`PlsqlUploadAsLongRaw`	N/A	ORDS has no equivalent.

Example ORDS Configuration Files

The following sections show how the example mod_plsql application can be migrated to ORDS.

Topics:

- Example Configuration File for Basic Authentication
- Example Configuration File for Basic Dynamic Authentication
- Example Configuration file for Custom Authentication

Example Configuration File for Basic Authentication

Example 7-2 ords_conf/ords/conf/basic_auth.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
    <comment>Saved on Wed Jul 25 10:22:37 UTC 2018</comment>
    <entry key="db.username">PRIVILEGED_USER</entry>
    <entry key="db.password">!passwordF0R$0RD5Example</entry>
    <!-- Example url -->
    <!-- See url-mapping.xml -->
    <!-- http://localhost:8086/ords/pls/basic_auth/
sample_plsql_app.sample_public_proc-->
    <!-- http://localhost:8086/ords/pls/basic_auth/
sample_plsql_app.privileged_public_proc-->
    <entry
key="procedure.postProcess">sample_plsql_app_metadata.afterProc</entry>
    <entry
key="procedure.preProcess">sample_plsql_app_metadata.beforeProc</entry>
    <entry
key="security.requestValidationFunction">sample_plsql_app_metadata.valid
ationFunc</entry>
    <entry key="owa.docTable">sample_plsql_app.doc_table</entry>
</properties>
```

Example Configuration File for Basic Dynamic Authentication

Example 7-3 ords_conf/ords/conf/basic_dynamic_auth.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
    <comment>Saved on Wed Jul 25 10:22:37 UTC 2018</comment>
    <!--  NOTE THAT IF THIS USER HAS EXECUTE PRIVILEGE ON THE RESOURCE
THEN jdbc.auth.enabled IS IGNORED -->
    <!--  IF THIS USER DOES NOT HAVE EXECUTE PRIVILEGE ON THE RESOURCE
THEN jdbc.auth.enabled IS INVOKED AND THE CREDENTIALS OF A PRIVILEGED
```

```
USER HAS TO BE PROVIDED-->
    <entry key="db.username">NON_PRIVILEGED_USER</entry>
    <entry key="db.password">!passwordF0R$0RD5Example</entry>
    <entry key="jdbc.auth.enabled">true</entry>
    <!-- Example url -->
    <!-- See url-mapping.xml -->
    <!-- INVOKE jdbc.auth.enabled :   http://localhost:8086/ords/pls/
basic_dynamic_auth/sample_plsql_app.sample_privileged_proc -->
    <!-- IGNORE jdbc.auth.enabled :   http://localhost:8086/ords/pls/
basic_dynamic_auth/sample_plsql_app.sample_public_proc   -->
    <!-- Because jdbc.auth.enabled is ignored when referencing the
sample_public_app, the beforeProc,afterProc and validationFunc must be
accessible by NON_PRIVILEGED_USER -->
    <!-- The following objects are executed by the same credentials
used to access the resource -->
    <!-- If the resource can be accessed by the db.username then that
connection is used to access these methods -->
    <!-- If the resource cannot be accessed by the db.username then
jdbc.auth.enabled is invoked and those credentials as used to access
these methods -->
    <entry
key="procedure.postProcess">sample_plsql_app_metadata.afterProc</entry>
    <entry
key="procedure.preProcess">sample_plsql_app_metadata.beforeProc</entry>
    <entry
key="security.requestValidationFunction">sample_plsql_app_metadata.valid
ationFunc</entry>
</properties>
```

Example Configuration file for Custom Authentication

Example 7-4 ords_confs/ords/conf/custom_auth.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
    <comment>Saved on Wed Jul 25 10:22:37 UTC 2018</comment>
    <entry key="db.username">PRIVILEGED_USER</entry>
    <entry key="db.password">!passwordF0R$0RD5Example</entry>
    <!-- Example url -->
    <!-- See url-mapping.xml -->
    <!-- http://localhost:8086/ords/pls/custom_auth/
sample_plsql_app.sample_proc -->
    <!-- privileged_user.owa_custom.authorize requires the following as
the custom login -->
    <entry
key="procedure.postProcess">sample_plsql_app_metadata.afterProc</entry>
    <entry
key="procedure.preProcess">sample_plsql_app_metadata.beforeProc</entry>
    <entry
key="security.requestValidationFunction">sample_plsql_app_metadata.valid
ationFunc</entry>
    <entry
```

```
key="security.requestAuthenticationFunction">privileged_user.owa_custom.
authorize</entry>
</properties>
```

Example ORDS URL Mapping

This section shows the example mapping between base-path url and the configuration files.

Example 7-5 ords_conf/ords/url-mapping.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<pool-config xmlns="http://xmlns.oracle.com/apex/pool-config">
    <pool name="basic_auth" base-path="/pls/basic_auth"
updated="2018-07-17T20:52:29.045Z" />
    <pool name="basic_dynamic_auth" base-path="/pls/basic_dynamic_auth"
updated="2018-07-17T20:52:29.045Z" />
    <pool name="custom_auth" base-path="/pls/custom_auth"
updated="2018-07-17T20:52:29.045Z" />
</pool-config>
```

Example ORDS Default Configuration

This section shows the example default configuration setting for ORDS.

The `default.xml` file provides the database connection details used by all configurations.

> **Note:**
>
> To turn off procedure validation caching, set `security.maxEntries` value to 0. This is necessary to emulate Oracle HTTP Server mod_plsql.

Example 7-6 ords_conf/ords/default.xml

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
   <!-- by default security.maxEntries = 2000 which means 2000
procedures validity will be cached-->
   <!-- this is fine for applications like apex where the validation of
a procedure does not change  -->
   <!-- for applications migrating from mod_plsql the cache should be
disabled so that procedures validity is determined for each request -->
   <!-- this is done by setting security.maxentries to 0 -->
   <entry key="security.maxEntries">0</entry>
   <entry key="db.hostname">oracle-ee</entry>
   <entry key="db.port">1521</entry>
   <entry key="db.servicename">orclpdb1</entry>
</properties>
```

ORDS Authentication

ORDS has the ability to perform HTTP Basic Authentication by providing a one to one mapping from mod_plsql. In ORDS more secure methods of authentication are available.

Topics:

- Basic Authentication
- Basic Dynamic Authentication
- Custom Authentication

Related Topics

- Developing Oracle REST Data Services Applications

Basic Authentication

This section describes the basic authentication implemented using ORDS.

The database credentials are specified in the ORDS configuration file. The `db.username` must have the required privileges to access the resources.

> **Note:**
>
> The entry `security.requestAuthenticationFunction` is not specified.

Basic Dynamic Authentication

This section describes how basic dynamic authentication is implemented using ORDS.

A default `db.username` and `db.password` must be specified in ORDS configuration file when providing basic dynamic authentication for accessing the resources.

The resources that cannot be accessed using this type of authentication can be accessed if the following conditions are satisfied:

- The value for `<entry key="jdbc.auth.enabled">true</entry>` entry must be `true`..
- The `security.requestAuthenticationFunction` entry must not be specified.
- When ORDS response prompts a Basic HTTP Authentication dialog box in a browser, the credentials provided by the user must have the required privileges, then the resource is made available.

> **Note:**
>
> If the credentials are provided through the browser HTTP authentication dialog box, then the only way to log out is to close all the instances of the browser.

Custom Authentication

This section describes how custom authentication is implemented using ORDS.

A function is specified to perform the custom authentication. This function has access to the owa variables. Resources are only available if the following function returns a `TRUE` value:

```
<entry
key="security.requestAuthenticationFunction">privileged_user.owa_custom.au
thorize</entry>
```

The authentication function must have signature as shown in the following code snippet:

```
/**
* OWA_CUSTOM used in mod_plsql when the following is used in the dad
configuration file
  PlsqlAuthenticationMode       Custom
  In ORDS environment this can reside in any schema as long as the
connection has execute privileges
  In mod_plsql this has to reside in the connections schema as you
cannot specify the name of the schema,package or function
  ex: PlsqlAuthenticationMode           CustomOwa
*/
CREATE OR REPLACE PACKAGE OWA_CUSTOM  AS
/**
 * Response:
  >IF Failed
   WWW-Authenticate in response header
   Authorization Required
   You are not authorized to access the requested resource. Check the
supplied credentials (e.g., username and password).
 */
FUNCTION authorize RETURN BOOLEAN;
END OWA_CUSTOM ;
/
```

ORDS Features

This section describes the ORDS features that are useful when you are migrating from a mod_plsql application to ORDS.

Topics:

- Request Validation Function

- Pre Process Feature
- Post Process Feature
- File Upload Feature

Request Validation Function

This section explains the use of request validation function.

The request validation function restricts the access to resources. The request validation function is provided with the name of the resource being requested and returns `TRUE` or `FALSE` value in response.

If the request validation function returns a `FALSE` value, then ORDS terminates the request.

Example 7-7 security.requestValidationFunction

```
<entry
key="security.requestValidationFunction">sample_plsql_app_metadata.valid
ationFunc</entry>
```

You can choose any name for the validation function. However, the signature must be in the following format:

```
CREATE OR REPLACE FUNCTION validationfunc(procedure_name VARCHAR2) RETURN
BOOLEAN IS.
```

Pre Process Feature

This section describes the `procedure.preProcess` ORDS configuration parameter.

The `procedure.preProcess` ORDS configuration parameter allows a comma delimited list of procedures that are executed before the requested resource.

Example 7-8 procedure.preProcess

Following example code snippet shows a use case for logging in:

```
<entry key="procedure.preProcess">sample_plsql_app_metadata.beforeProc</
entry>
```

Post Process Feature

This section describes the `procedure.postProcess` ORDS configuration parameter.

The `procedure.postProcess` ORDS configuration parameter allows a comma delimited list of procedures that are executed after the requested resource.

Example 7-9 procedure.postProcess

Following example code snippet shows a use case for logging out:

```
<entry key="procedure.postProcess">sample_plsql_app_metadata.afterProc</
entry>
```

File Upload Feature

This section describes the ORDS file upload feature.

The ORDS configuration parameter `owa.docTable`, defines the table name where the uploaded files persist.

Example 7-10 Table upload

```
CREATE TABLE DOC_TABLE (
     NAME               VARCHAR(256)   UNIQUE NOT NULL,
     MIME_TYPE          VARCHAR(128),
     DOC_SIZE           NUMBER,
     DAD_CHARSET        VARCHAR(128),
     LAST_UPDATED       DATE,
     CONTENT_TYPE       VARCHAR(128),
     CONTENT            LONG RAW,
     BLOB_CONTENT       BLOB );
```

Example 7-11 Procedure upload

You can choose to have any name for the upload function. However, the signature must match the following POST request:

```
--The parameters of the procedure should match the parameters of the
request
--The procedure is called after ORDS performs the file upload/insert.
--This procedure can rollback the file INSERT as it is in the same
transaction as the INSERT
CREATE OR REPLACE PROCEDURE upload (filename VARCHAR2 DEFAULT NULL)
```

Example 7-12 Curl command for file upload

```
curl -i -X POST -F 'filename=@helloworld.txt' "http://localhost:8086/
ords/pls/basic_auth/example_user1.upload
```

8

Oracle REST Data Services PL/SQL Package Reference

The Oracle REST Data Services PL/SQL package contains subprograms (procedures and functions) for developing RESTful services using Oracle REST Data Services.

Related Topics

- Using the Oracle REST Data Services PL/SQL API

ORDS.CREATE_ROLE

Format

```
ORDS.CREATE_ROLE(
   p_role_name IN sec_roles.name%type);
```

Description

CREATE_ROLE creates an Oracle REST Data Services role with the specified name.

Parameters

p_role_name
Name of the role.

Usage Notes

After the role is created, it can be associated with any Oracle REST Data Services privilege.

Examples

The following example creates a role.

```
EXECUTE ORDS.CREATE_ROLE(p_role_name=>'Tickets User');
```

ORDS.CREATE_SERVICE

> **Note:**
>
> ORDS.CREATE_SERVICE is deprecated. Use ORDS.DEFINE_SERVICE instead.

Format

```
ORDS.CREATE_SERVICE(
   p_module_name        IN ords_modules.name%type,
   p_base_path          IN ords_modules.uri_prefix%type,
   p_pattern            IN ords_templates.uri_template%type,
   p_method             IN ords_handlers.method%type DEFAULT 'GET',
   p_source_type        IN ords_handlers.source_type%type
                           DEFAULT ords.source_type_collection_feed,
   p_source             IN ords_handlers.source%type,
   p_items_per_page     IN ords_modules.items_per_page%type DEFAULT 25,
   p_status             IN ords_modules.status%type DEFAULT 'PUBLISHED',
   p_etag_type          IN ords_templates.etag_type%type DEFAULT 'HASH',
   p_etag_query         IN ords_templates.etag_query%type DEFAULT NULL,
   p_mimes_allowed      IN ords_handlers.mimes_allowed%type DEFAULT NULL,
   p_module_comments    IN ords_modules.comments%type DEFAULT NULL,
   p_template_comments  IN ords_modules.comments%type DEFAULT NULL,
   p_handler_comments   IN ords_modules.comments%type DEFAULT NULL);
```

Description

Creates a new RESTful service.

Parameters

p_module_name
The name of the RESTful service module. Case sensitive. Must be unique.

p_base_path
The base of the URI that is used to access this RESTful service. Example: `hr/` means that all URIs starting with `hr/` will be serviced by this resource module.

p_pattern
A matching pattern for the resource template. For example, a pattern of `/objects/:object/:id?` will match `/objects/emp/101` (matches a request for the item in the `emp` resource with `id` of 101) and will also match `/objects/emp/` (matches a request for the `emp` resource, because the `:id` parameter is annotated with the `?` or question mark modifier, which indicates that the `id` parameter is optional).

p_method
The HTTP method to which this handler will respond. Valid values: `GET` (retrieves a representation of a resource), `POST` (creates a new resource or adds a resource to a collection), `PUT` (updates an existing resource), `DELETE` (deletes an existing resource).

p_source_type
The HTTP request method for this handler. Valid values:

- `source_type_collection_feed`. Executes a SQL query and transforms the result set into an Oracle REST Data Services Standard JSON representation. Available when the HTTP method is GET. Result Format: JSON
- `source_type_collection_item`. Executes a SQL query returning one row of data into a Oracle REST Data Services Standard JSON representation. Available when the HTTP method is GET. Result Format: JSON

- `source_type_media`. Executes a SQL query conforming to a specific format and turns the result set into a binary representation with an accompanying HTTP Content-Type header identifying the Internet media type of the representation. Result Format: Binary
- `source_type_plsql`. Executes an anonymous PL/SQL block and transforms any OUT or IN/OUT parameters into a JSON representation. Available only when the HTTP method is DELETE, PUT, or POST. Result Format: JSON
- `source_type_query || source_type_csv_query`. Executes a SQL query and transforms the result set into either an Oracle REST Data Services legacy JavaScript Object Notation (JSON) or CSV representation, depending on the format selected. Available when the HTTP method is GET. Result Format: JSON or CSV
- `source_type_query_one_row`. Executes a SQL query returning one row of data into an Oracle REST Data Services legacy JSON representation. Available when the HTTP method is GET. Result Format: JSON
- `source_type_feed`. Executes a SQL query and transforms the results into a JSON Feed representation. Each item in the feed contains a summary of a resource and a hyperlink to a full representation of the resource. The first column in each row in the result set must be a unique identifier for the row and is used to form a hyperlink of the form: `path/to/feed/{id}`, with the value of the first column being used as the value for `{id}`. The other columns in the row are assumed to summarize the resource and are included in the feed. A separate resource template for the full representation of the resource should also be defined. Result Format: JSON

p_source
The source implementation for the selected HTTP method.

p_items_per_page
The default pagination for a resource handler HTTP operation GET method, that is, the number of rows to return on each page of a JSON format result set based on a database query. Default: NULL (defers to the resource module setting).

p_status
The publication status. Valid values: 'PUBLISHED' (default) or 'NOT_PUBLISHED'.

p_etag_type
A type of entity tag to be used by the resource template. An entity tag is an HTTP Header that acts as a version identifier for a resource. Use entity tag headers to avoid retrieving previously retrieved resources and to perform optimistic locking when updating resources. Valid values: 'HASH' or 'QUERY' or 'NONE'.

- HASH - Known as Secure HASH: The contents of the returned resource representation are hashed using a secure digest function to provide a unique fingerprint for a given resource version.
- QUERY - Manually define a query that uniquely identifies a resource version. A manually defined query can often generate an entity tag more efficiently than hashing the entire resource representation.
- NONE - Do not generate an entity tag.

p_etag_query
A query that is used to generate the entity tag.

p_mimes_allowed
A comma-separated list of MIME types that the handler will accept. Applies to PUT and POST only.

p_module_comments
Comment text.

p_template_comments
Comment text.

p_handler_comments
Comment text.

Usage Notes

Creates a resource module, template, and handler in one call.

This procedure is deprecated. Use ORDS.DEFINE_SERVICE instead.

Examples

The following example creates a simple service.

```
BEGIN
  ORDS.CREATE_SERVICE(
    p_module_name => 'my.tickets',
    p_base_path => '/my/tickets/',
    p_pattern => '.',
    p_source => 'select t.id "$.id", t.id, t.title from tickets t' ||
                ' where t.owner = :current_user order by t.updated_on desc'
  );
END;
/
```

ORDS.DEFINE_HANDLER

Format

```
ORDS.DEFINE_HANDLER(
p_module_name      IN ords_modules.name%type,
p_pattern          IN ords_templates.uri_template%type,
p_method           IN ords_handlers.method%type DEFAULT 'GET',
p_source_type      IN ords_handlers.source_type%type
DEFAULT ords.source_type_collection_feed,
p_source           IN ords_handlers.source%type,
p_items_per_page   IN ords_handlers.items_per_page%type DEFAULT NULL,
p_mimes_allowed    IN ords_handlers.mimes_allowed%type DEFAULT NULL,
p_comments         IN ords_handlers.comments%type DEFAULT NULL);
```

Description

DEFINE_HANDLER defines a module handler. If the handler already exists, then the handler and any existing handlers will be replaced by this definition; otherwise, a new handler is created.

Parameters

p_module_name
Name of the owning RESTful service module. Case sensitive.

p_pattern
Matching pattern for the owning resource template.

p_method
The HTTP method to which this handler will respond. Valid values: `GET` (retrieves a representation of a resource), `POST` (creates a new resource or adds a resource to a collection), `PUT` (updates an existing resource), `DELETE` (deletes an existing resource).

p_source_type
The HTTP request method for this handler. Valid values:

- `source_type_collection_feed`. Executes a SQL query and transforms the result set into an Oracle REST Data Services Standard JSON representation. Available when the HTTP method is GET. Result Format: JSON
- `source_type_collection_item`. Executes a SQL query returning one row of data into a Oracle REST Data Services Standard JSON representation. Available when the HTTP method is GET. Result Format: JSON
- `source_type_media`. Executes a SQL query conforming to a specific format and turns the result set into a binary representation with an accompanying HTTP Content-Type header identifying the Internet media type of the representation. Result Format: Binary
- `source_type_plsql`. Executes an anonymous PL/SQL block and transforms any OUT or IN/OUT parameters into a JSON representation. Available only when the HTTP method is DELETE, PUT, or POST. Result Format: JSON
- `source_type_query || source_type_csv_query`. Executes a SQL query and transforms the result set into either an Oracle REST Data Services legacy JavaScript Object Notation (JSON) or CSV representation, depending on the format selected. Available when the HTTP method is GET. Result Format: JSON or CSV
- `source_type_query_one_row`. Executes a SQL query returning one row of data into an Oracle REST Data Services legacy JSON representation. Available when the HTTP method is GET. Result Format: JSON
- `source_type_feed`. Executes a SQL query and transforms the results into a JSON Feed representation. Each item in the feed contains a summary of a resource and a hyperlink to a full representation of the resource. The first column in each row in the result set must be a unique identifier for the row and is used to form a hyperlink of the form: `path/to/feed/{id}`, with the value of the first column being used as the value for `{id}`. The other columns in the row are assumed to summarize the resource and are included in the feed. A separate resource template for the full representation of the resource should also be defined. Result Format: JSON

p_source
The source implementation for the selected HTTP method.

p_items_per_page
The default pagination for a resource handler HTTP operation GET method, that is, the number of rows to return on each page of a JSON format result set based on a database query. Default: NULL (defers to the resource module setting).

p_mimes_allowed
Comma-separated list of MIME types that the handler will accept. Applies to PUT and POST only.

p_comments
Comment text.

Usage Notes

Only one handler for each HTTP method (source type) is permitted.

Examples

The following example defines a POST handler to the `/my/tickets/` resource to accept new tickets.

```
BEGIN
  ORDS.DEFINE_HANDLER(
    p_module_name => 'my.tickets',
    p_pattern => '.',
    p_method  => 'POST',
    p_mimes_allowed => 'application/json',
    p_source_type => ords.source_type_plsql,
    p_source => '
      declare
        l_owner varchar2(255);
        l_payload blob;
        l_id number;
      begin
        l_payload := :body;
        l_owner := :owner;
        if ( l_owner is null ) then
          l_owner := :current_user;
        end if;
        l_id := ticket_api.create_ticket(
          p_json_entity => l_payload,
          p_author => l_owner
        );
        :location := ''./'' || l_id;
        :status := 201;
      end;
      '
  );
END;
/
```

ORDS.DEFINE_MODULE

Format

```
ORDS.DEFINE_MODULE(
   p_module_name     IN ords_modules.name%type,
   p_base_path       IN ords_modules.uri_prefix%type,
   p_items_per_page  IN ords_modules.items_per_page%type DEFAULT 25,
```

```
   p_status          IN ords_modules.status%type DEFAULT 'PUBLISHED',
   p_comments        IN ords_modules.comments%type DEFAULT NULL);
```

Description

DEFINE_MODULE defines a resource module. If the module already exists, then the module and any existing templates will be replaced by this definition; otherwise, a new module is created.

Parameters

p_module_name
Name of the owning RESTful service module. Case sensitive.

p_base_path
The base of the URI that is used to access this RESTful service. Example: `hr/` means that all URIs starting with `hr/` will be serviced by this resource module.

p_items_per_page
The default pagination for a resource handler HTTP operation GET method, that is, the number of rows to return on each page of a JSON format result set based on a database query. Default: 25.

p_status
Publication status. Valid values: `PUBLISHED` (default) or `NOT_PUBLISHED`.

p_comments
Comment text.

Usage Notes

(None.)

Examples

The following example creates a simple module.

```
BEGIN
  ORDS.DEFINE_MODULE(
    p_module_name => 'my.tickets',
    p_base_path => '/my/tickets/'
  );
END;
/
```

ORDS.DEFINE_PARAMETER

Format

```
ORDS.DEFINE_PARAMETER(
   p_module_name        IN ords_modules.name%type,
   p_pattern            IN ords_templates.uri_template%type,
   p_method             IN ords_handlers.method%type,
   p_name               IN ords_parameters.name%type ,
   p_bind_variable_name IN ords_parameters.bind_variable_name%type
                           DEFAULT NULL,
   p_source_type        IN ords_parameters.source_type%type DEFAULT 'HEADER',
   p_param_type         IN ords_parameters.param_type%type DEFAULT 'STRING',
```

```
  p_access_method      IN ords_parameters.access_method%type DEFAULT 'IN',
  p_comments           IN ords_parameters.comments%type DEFAULT NULL);
```

Description

DEFINE_PARAMETER defines a module handler parameter. If the parameter already exists, then the parameter will be replaced by this definition; otherwise, a new parameter is created.

Parameters

p_module_name
Name of the owning RESTful service module. Case sensitive.

p_pattern
Matching pattern for the owning resource template.

p_method
The owning handler HTTP Method. Valid values: `GET` (retrieves a representation of a resource), `POST` (creates a new resource or adds a resource to a collection), `PUT` (updates an existing resource), `DELETE` (deletes an existing resource).

p_name
The name of the parameter, as it is named in the URI Template or HTTP Header. Used to map names that are not valid SQL parameter names.

p_bind_variable_name
The name of the parameter, as it will be referred to in the SQL. If NULL is specified, then the parameter is unbound.

p_source_type
The type that is identified if the parameter originates in the URI Template or a HTTP Header. Valid values: `HEADER`, `RESPONSE`, `URI`.

p_param_type
The native type of the parameter. Valid values: `STRING`, `INT`, `DOUBLE`, `BOOLEAN`, `LONG`, `TIMESTAMP`.

p_access_method
The parameter access method. Indicates if the parameter is an input value, output value, or both. Valid values: `IN`, `OUT`, `INOUT`.

p_comments
Comment text.

Usage Notes

All parameters must have unique names and variable names for the same handler.

Examples

The following example defines an outbound parameter on the POST handler to store the location of the created ticket.

```
BEGIN
  ORDS.DEFINE_PARAMETER(
    p_module_name => 'my.tickets',
    p_pattern => '.',
```

```
    p_method => 'POST',
    p_name => 'X-APEX-FORWARD',
    p_bind_variable_name => 'location',
    p_source_type => 'HEADER',
    p_access_method => 'OUT'
  );
END;
/
```

The following example defines an outbound parameter on the POST handler to store the HTTP status of the operation.

```
BEGIN
  ORDS.DEFINE_PARAMETER(
    p_module_name => 'my.tickets',
    p_pattern => '.',
    p_method => 'POST',
    p_name => 'X-APEX-STATUS-CODE',
    p_bind_variable_name => 'status',
    p_source_type => 'HEADER',
    p_access_method => 'OUT'
  );
END;
/
```

ORDS.DEFINE_PRIVILEGE

Format

```
ORDS.DEFINE_PRIVILEGE(
   p_privilege_name    IN sec_privileges.name%type,
   p_roles             IN owa.vc_arr,
   p_patterns          IN owa.vc_arr,
   p_modules           IN owa.vc_arr,
   p_label             IN sec_privileges.label%type DEFAULT NULL,
   p_description       IN sec_privileges.description%type DEFAULT NULL,
   p_comments          IN sec_privileges.comments%type DEFAULT NULL);
or
ORDS.DEFINE_PRIVILEGE(
   p_privilege_name    IN sec_privileges.name%type,
   p_roles             IN owa.vc_arr,
   p_patterns          IN owa.vc_arr,
   p_label             IN sec_privileges.label%type DEFAULT NULL,
   p_description       IN sec_privileges.description%type DEFAULT NULL,
   p_comments          IN sec_privileges.comments%type DEFAULT NULL);
or
ORDS.DEFINE_PRIVILEGE(
   p_privilege_name    IN sec_privileges.name%type,
   p_roles             IN owa.vc_arr,
   p_label             IN sec_privileges.label%type DEFAULT NULL,
   p_description       IN sec_privileges.description%type DEFAULT NULL,
   p_comments          IN sec_privileges.comments%type DEFAULT NULL);
```

Description

DEFINE_PRIVILEGE defines an Oracle REST Data Services privilege. If the privilege already exists, then the privilege and any existing patterns and any associations with modules and roles will be replaced by this definition; otherwise, a new privilege is created.

Parameters

p_privilege_name
Name of the privilege. No spaces allowed.

p_roles
The names of the roles, at least one of which the privilege requires. May be empty, in which case the user must be authenticated but does not require any specific role; however, must not be null. Unauthenticated users will be denied access.

p_patterns
A list of patterns.

p_modules
A list of module names referencing modules created for the current schema.

p_label
Name of this security constraint as displayed to an end user. May be null.

p_description
A brief description of the purpose of the resources protected by this constraint.

p_comments
Comment text.

Usage Notes

`p_roles`, `p_patterns`, and `p_modules` do not accept null values. If no value is to be passed, then either choose the appropriate procedure specification or pass an empty `owa.vc_arr` value.

Examples

The following example creates a privilege connected to roles, patterns, and modules:

```
DECLARE
  l_priv_roles owa.vc_arr;
  l_priv_patterns owa.vc_arr;
  l_priv_modules owa.vc_arr;
BEGIN
  l_priv_roles(1) := 'Tickets User';
  l_priv_patterns(1) := '/my/*';
  l_priv_patterns(2) := '/comments/*';
  l_priv_patterns(3) := '/tickets_feed/*';
  l_priv_patterns(4) := '/tickets/*';
  l_priv_patterns(5) := '/categories/*';
  l_priv_patterns(6) := '/stats/*';

  l_priv_modules(1) := 'my.tickets';

  ords.create_role('Tickets User');

  ords.define_privilege(
    p_privilege_name     => 'tickets.privilege',
    p_roles              => l_priv_roles,
    p_patterns           => l_priv_patterns,
    P_modules            => l_priv_modules,
    p_label              => 'Task Ticketing Access',
    p_description        => 'Provides the ability to create, ' ||
```

```
                            'update and delete tickets ' ||
                            'and post comments on tickets'
  );
END;
/
```

The following example creates a privilege connected to roles and patterns:

```
DECLARE
  l_priv_roles owa.vc_arr;
  l_priv_patterns owa.vc_arr;
BEGIN
  l_priv_roles(1) := 'Tickets User';
  l_priv_patterns(1) := '/my/*';
  l_priv_patterns(2) := '/comments/*';
  l_priv_patterns(3) := '/tickets_feed/*';
  l_priv_patterns(4) := '/tickets/*';
  l_priv_patterns(5) := '/categories/*';
  l_priv_patterns(6) := '/stats/*';

  ords.create_role('Tickets User');

  ords.define_privilege(
    p_privilege_name     => 'tickets.privilege',
    p_roles              => l_priv_roles,
    p_patterns           => l_priv_patterns,
    p_label              => 'Task Ticketing Access',
    p_description        => 'Provides the ability to create, ' ||
                            'update and delete tickets ' ||
                            'and post comments on tickets'
  );
END;
/
```

The following example creates a privilege connected to roles:

```
DECLARE
  l_priv_roles owa.vc_arr;
BEGIN
  l_priv_roles(1) := 'Tickets User';

  ords.create_role('Tickets User');

  ords.define_privilege(
    p_privilege_name     => 'tickets.privilege',
    p_roles              => l_priv_roles,
    p_label              => 'Task Ticketing Access',
    p_description        => 'Provides the ability to create, ' ||
                            'update and delete tickets ' ||
                            'and post comments on tickets'
  );
END;
/
```

ORDS.DEFINE_SERVICE

Format

```
ORDS.DEFINE_SERVICE(
   p_module_name        IN ords_modules.name%type,
```

```
    p_base_path          IN ords_modules.uri_prefix%type,
    p_pattern            IN ords_templates.uri_template%type,
    p_method             IN ords_handlers.method%type DEFAULT 'GET',
    p_source_type        IN ords_handlers.source_type%type
                              DEFAULT ords.source_type_collection_feed,
    p_source             IN ords_handlers.source%type,
    p_items_per_page     IN ords_modules.items_per_page%type DEFAULT 25,
    p_status             IN ords_modules.status%type DEFAULT 'PUBLISHED',
    p_etag_type          IN ords_templates.etag_type%type DEFAULT 'HASH',
    p_etag_query         IN ords_templates.etag_query%type DEFAULT NULL,
    p_mimes_allowed      IN ords_handlers.mimes_allowed%type DEFAULT NULL,
    p_module_comments    IN ords_modules.comments%type DEFAULT NULL,
    p_template_comments  IN ords_modules.comments%type DEFAULT NULL,
    p_handler_comments   IN ords_modules.comments%type DEFAULT NULL);
```

Description

DEFINE_SERVICE defines a resource module, template, and handler in one call. If the module already exists, then the module and any existing templates will be replaced by this definition; otherwise, a new module is created.

Parameters

p_module_name
Name of the RESTful service module. Case sensitive. Must be unique.

p_base_path
The base of the URI that is used to access this RESTful service. Example: `hr/` means that all URIs starting with `hr/` will be serviced by this resource module.

p_pattern
A matching pattern for the resource template. For example, a pattern of `/objects/:object/:id?` will match `/objects/emp/101` (matches a request for the item in the `emp` resource with `id` of 101) and will also match `/objects/emp/`. (Matches a request for the `emp` resource, because the `:id` parameter is annotated with the `?` modifier, which indicates that the `id` parameter is optional.)

p_method
The HTTP Method to which this handler will respond. Valid values: `GET` (retrieves a representation of a resource), `POST` (creates a new resource or adds a resource to a collection), `PUT` (updates an existing resource), `DELETE` (deletes an existing resource).

p_source_type
The HTTP request method for this handler. Valid values:

- `source_type_collection_feed`. Executes a SQL query and transforms the result set into an Oracle REST Data Services Standard JSON representation. Available when the HTTP method is GET. Result Format: JSON
- `source_type_collection_item`. Executes a SQL query returning one row of data into a Oracle REST Data Services Standard JSON representation. Available when the HTTP method is GET. Result Format: JSON
- `source_type_media`. Executes a SQL query conforming to a specific format and turns the result set into a binary representation with an accompanying HTTP Content-Type header identifying the Internet media type of the representation. Result Format: Binary

- `source_type_plsql`. Executes an anonymous PL/SQL block and transforms any OUT or IN/OUT parameters into a JSON representation. Available only when the HTTP method is DELETE, PUT, or POST. Result Format: JSON
- `source_type_query || source_type_csv_query`. Executes a SQL query and transforms the result set into either an Oracle REST Data Services legacy JavaScript Object Notation (JSON) or CSV representation, depending on the format selected. Available when the HTTP method is GET. Result Format: JSON or CSV
- `source_type_query_one_row`. Executes a SQL query returning one row of data into an Oracle REST Data Services legacy JSON representation. Available when the HTTP method is GET. Result Format: JSON
- `source_type_feed`. Executes a SQL query and transforms the results into a JSON Feed representation. Each item in the feed contains a summary of a resource and a hyperlink to a full representation of the resource. The first column in each row in the result set must be a unique identifier for the row and is used to form a hyperlink of the form: `path/to/feed/{id}`, with the value of the first column being used as the value for `{id}`. The other columns in the row are assumed to summarize the resource and are included in the feed. A separate resource template for the full representation of the resource should also be defined. Result Format: JSON

p_source
The source implementation for the selected HTTP method.

p_items_per_page
The default pagination for a resource handler HTTP operation GET method, that is, the number of rows to return on each page of a JSON format result set based on a database query. Default: NULL (defers to the resource module setting).

p_status
Publication status. Valid values: `PUBLISHED` (default) or `NOT_PUBLISHED`.

p_etag_type
A type of entity tag to be used by the resource template. An entity tag is an HTTP Header that acts as a version identifier for a resource. Use entity tag headers to avoid retrieving previously retrieved resources and to perform optimistic locking when updating resources. Valid values are `HASH`, `QUERY`, `NONE`:

- `HASH` (known as Secure HASH): The contents of the returned resource representation are hashed using a secure digest function to provide a unique fingerprint for a given resource version.
- `QUERY`: Manually define a query that uniquely identifies a resource version. A manually defined query can often generate an entity tag more efficiently than hashing the entire resource representation.
- `NONE`: Do not generate an entity tag.

p_etag_query
Query that is used to generate the entity tag.

p_mimes_allowed
Comma-separated list of MIME types that the handler will accept. Applies to PUT and POST only.

p_module_comments
Comment text.

p_template_comments
Comment text.

p_handler_comments
Comment text.

Usage Notes

Creates a resource module, template, and handler in one call.

Use this procedure instead of the deprecated ORDS.CREATE_SERVICE procedure.

Examples

The following example defines a REST service that retrieves the current user's tickets.

```
BEGIN
  ORDS.DEFINE_SERVICE(
    p_module_name => 'my.tickets',
    p_base_path => '/my/tickets/',
    p_pattern => '.',
    p_source => 'select t.id "$.id", t.id, t.title from tickets t' ||
                ' where t.owner = :current_user order by t.updated_on desc'
  );
END;
/
```

The following example defines a REST service that retrieves tickets filtered by category.

```
BEGIN
  ORDS.DEFINE_SERVICE(
    p_module_name => 'by.category',
    p_base_path => '/by/category/',
    p_pattern => ':category_id',
    p_source => 'select ''../../my/tickets/'' ||
                  t.id "$.id", t.id, t.title' ||
                ' from tickets t, categories c, ticket_categories tc' ||
                ' where c.id = :category_id and c.id = tc.category_id and' ||
                ' tc.ticket_id = t.id order by t.updated_on desc'
  );
END;
/
```

ORDS.DEFINE_TEMPLATE

Format

```
ORDS.DEFINE_TEMPLATE(
   p_module_name  IN ords_modules.name%type,
   p_pattern      IN ords_templates.uri_template%type,
   p_priority     IN ords_templates.priority%type DEFAULT 0,
   p_etag_type    IN ords_templates.etag_type%type DEFAULT 'HASH',
   p_etag_query   IN ords_templates.etag_query%type DEFAULT NULL,
   p_comments     IN ords_templates.comments%type DEFAULT NULL);
```

Description

DEFINE_TEMPLATE defines a resource template. If the template already exists, then the template and any existing handlers will be replaced by this definition; otherwise, a new template is created.

Parameters

p_module_name
Name of the owning RESTful service module. Case sensitive.

p_pattern
A matching pattern for the resource template. For example, a pattern of `/objects/:object/:id?` will match `/objects/emp/101` (matches a request for the item in the `emp` resource with `id` of 101) and will also match `/objects/emp/`. (Matches a request for the `emp` resource, because the `:id` parameter is annotated with the `?` modifier, which indicates that the `id` parameter is optional.)

p_priority
The priority for the order of how the resource template should be evaluated: 0 (low priority. the default) through 9 (high priority).

p_etag_type
A type of entity tag to be used by the resource template. An entity tag is an HTTP Header that acts as a version identifier for a resource. Use entity tag headers to avoid retrieving previously retrieved resources and to perform optimistic locking when updating resources. Valid values are `HASH`, `QUERY`, `NONE`:

- `HASH` (known as Secure HASH): The contents of the returned resource representation are hashed using a secure digest function to provide a unique fingerprint for a given resource version.
- `QUERY`: Manually define a query that uniquely identifies a resource version. A manually defined query can often generate an entity tag more efficiently than hashing the entire resource representation.
- `NONE`: Do not generate an entity tag.

p_etag_query
Query that is used to generate the entity tag.

p_comments
Comment text.

Usage Notes

he resource template pattern must be unique with a resource module.

Examples

The following example defines a resource for displaying ticket items.

```
BEGIN
  ORDS.DEFINE_TEMPLATE(
    p_module_name => 'my.tickets',
    p_pattern => '/:id'
  );
```

```
END;
/
```

ORDS.DELETE_MODULE

Format

```
ORDS.DELETE_MODULE(
   p_module_name  IN ords_modules.name%type);
```

Description

DELETE_MODULE deletes a resource module.

Parameters

p_module_name
Name of the owning RESTful service module. Case sensitive.

Usage Notes

If the module does not already exist or is accessible to the current user, then no exception is raised.

Examples

The following example deletes a resource module.

```
EXECUTE ORDS.DELETE_MODULE(p_module_name=>'my.tickets');
```

ORDS.DELETE_PRIVILEGE

Format

```
ORDS.DELETE_PRIVILEGE(
   p_name  IN sec_privileges.name%type);
```

Description

DELETE_PRIVILEGE deletes a provilege.

Parameters

p_name
Name of the privilege.

Usage Notes

If the privilege does not already exist or is not accessible to the current user, then no exception is raised.

Examples

The following example deletes a privilege.

```
EXECUTE ORDS.DELETE_PRIVILEGE(p_name=>'tickets.privilege');
```

ORDS.DELETE_ROLE

Format

```
ORDS.DELETE_ROLE(
   p_role_name IN sec_roles.name%type);
```

Description

DELETE_ROLE deletes the named role.

Parameters

p_name
Name of the role.

Usage Notes

This will also delete any association between the role and any privileges that reference the role.

No exception is produced if the role does not already exist.

Examples

The following example deletes a role.

```
EXECUTE ORDS.DELETE_ROLE(p_role_name=>'Tickets User');
```

ORDS.DROP_REST_FOR_SCHEMA

Format

```
ORDS.DROP_REST_FOR_SCHEMA(
   p_schema ords_schemas.parsing_schema%type DEFAULT NULL);
```

Description

DROP_REST_FOR_SCHEMA deletes all auto-REST Oracle REST Data Services metadata for the associated schema.

Parameters

p_schema
Name of the schema.

Usage Notes

This procedure effectively "undoes" the actions performed by the `ORDS.Enable_Schema` procedure.

Examples

The following example deletes all auto-REST Oracle REST Data Services metadata for the TICKETS schema.

```
EXECUTE ORDS.DROP_REST_FOR_SCHEMA('tickets');
```

Related Topics

- ORDS.ENABLE_SCHEMA

ORDS.ENABLE_OBJECT

Format

```
ORDS.ENABLE_OBJECT(
   p_enabled         IN boolean DEFAULT TRUE,
   p_schema          IN ords_schemas.parsing_schema%type DEFAULT NULL,
   p_object          IN ords_objects.parsing_object%type,
   p_object_type     IN ords_objects.type%type DEFAULT 'TABLE',
   p_object_alias    IN ords_objects.object_alias%type DEFAULT NULL,
   p_auto_rest_auth  IN boolean DEFAULT NULL);
```

Description

ENABLE_OBJECT enables Oracle REST Data Services access to a specified table or view in a schema.

Parameters

p_enabled
`TRUE` to enable access; `FALSE` to disable access.

p_schema
Name of the schema for the table or view.

p_object
Name of the table or view.

p_object_type
Type of the object: `TABLE` (default) or `VIEW`.

p_object_alias
Alias of the object.

p_auto_rest_auth
Controls whether Oracle REST Data Services should require user authorization before allowing access to the Oracle REST Data Services metadata for this object. If this value is `TRUE`, then the service is protected by the following roles:

- `oracle.dbtools.autorest.any.schema`
- `oracle.dbtools.role.autorest.<SCHEMANAME>.<OBJECTNAME>`

Usage Notes

Only database users with the DBA role can enable/access to objects that they do now own.

Examples

The following example enables a table named CATEGORIES.

```
EXECUTE ORDS.ENABLE_OBJECT(p_object=>'CATEGORIES');
```

The following example enables a view named TICKETS_FEED.

```
BEGIN
  ORDS.ENABLE_OBJECT(
    p_object => 'TICKETS_FEED',
    p_object_type => 'VIEW'
  );
END;
/
```

ORDS.DROP_REST_FOR_OBJECT

Format

```
ORDS.DROP_REST_FOR_OBJECT(
   p_object ords_objects.parsing_object%type);
```

Description

DROP_REST_FOR_OBJECT deletes all auto-REST Oracle REST Data Services metadata for the associated schema object.

Parameters

p_object
Name of the table or view.

Usage Notes

This procedure effectively "undoes" the actions performed by the `ORDS.ENABLE_OBJECT` procedure.

Examples

The following example deletes all auto-REST Oracle REST Data Services metadata for the curent user CATEGORIES table.

```
BEGIN
  ORDS.DROP_REST_FOR_OBJECT(
    p_object=>'CATEGORIES'
  );
END;
/
```

ORDS.ENABLE_SCHEMA

Format

```
ORDS.ENABLE_SCHEMA
   p_enabled             IN boolean DEFAULT TRUE,
   p_schema              IN ords_schemas.parsing_schema%type DEFAULT NULL,
   p_url_mapping_type    IN ords_url_mappings.type%type DEFAULT 'BASE_PATH',
   p_url_mapping_pattern IN ords_url_mappings.pattern%type DEFAULT NULL,
   p_auto_rest_auth      IN boolean DEFAULT NULL);
```

Description

ENABLE_SCHEMA enables Oracle REST Data Services to access the named schema.

Parameters

p_enabled
`TRUE` to enable Oracle REST Data Services access; `FALSE` to disable Oracle REST Data Services access.

p_schema
Name of the schema. If the `p_schema` parameter is omitted, then the current schema is enabled.

p_url_mapping_type
URL Mapping type: `BASE_PATH` or `BASE_URL`.

p_url_mapping_pattern
URL mapping pattern.

p_auto_rest_auth
For a schema, controls whether Oracle REST Data Services should require user authorization before allowing access to the Oracle REST Data Services metadata catalog of this schema.

Usage Notes

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

Examples

The following example enables the current schema.

```
EXECUTE ORDS.ENABLE_SCHEMA;
```

ORDS.PUBLISH_MODULE

Format

```
ORDS.PUBLISH_MODULE(
   p_module_name  IN ords_modules.name%type,
   p_status       IN ords_modules.status%type DEFAULT 'PUBLISHED');
```

Description

PUBLISH_MODULE changes the publication status of an Oracle REST Data Services resource module.

Parameters

p_module_name
Current name of the RESTful service module. Case sensitive.

p_status
Publication status. Valid values: `PUBLISHED` (default) or `NOT_PUBLISHED`.

Usage Notes

(None.)

Examples

The following example publishes a previously defined module named `my.tickets`.

```
EXECUTE ORDS.PUBLISH_MODULE(p_module_name=>'my.tickets');
```

ORDS.RENAME_MODULE

Format

```
ORDS.RENAME_MODULE(
   p_module_name    IN ords_modules.name%type,
   p_new_name       IN ords_modules.name%type DEFAULT NULL,
   p_new_base_path  IN ords_modules.uri_prefix%type DEFAULT NULL);
```

Description

RENAME_MODULE lets you change the name or the base path, or both, of an Oracle REST Data Services resource module.

Parameters

p_module_name
Current name of the RESTful service module. Case sensitive.

p_new_name
New name to be assigned to the RESTful service module. Case sensitive. If this parameter is null, the name is not changed.

p_new_base_path
The base of the URI to be used to access this RESTful service. Example: hr/ means that all URIs starting with hr/ will be serviced by this resource module. If this parameter is null, the base path is not changed.

Usage Notes

Both the new resource module name and the base path must be unique within the enabled schema.

Examples

The following example renames resource module `my.tickets` to `old.tickets`.

```
BEGIN
  ORDS.RENAME_MODULE(
    p_module_name =>'my.tickets',
    p_new_name=>'old.tickets',
    p_new_base_path=>'/old/tickets/');
END;
/
```

ORDS.RENAME_PRIVILEGE

Format

```
ORDS.RENAME_PRIVILEGE(
   p_name      IN sec_privileges.name%type,
   p_new_name  IN sec_privileges.name%type);
```

Description

RENAME_PRIVILEGE renames a privilege.

Parameters

p_name
Current name of the privilege.

p_new_name
New name to be assigned to the privilege.

Usage Notes

(None.)

Examples

The following example renames the privilege `tickets.privilege` to `old.tickets.privilege`.

```
BEGIN
  ORDS.RENAME_PRIVILEGE(
    p_name =>'tickets.privilege',
    p_new_name=>'old.tickets.privilege');
END;
/
```

ORDS.RENAME_ROLE

Format

```
ORDS.RENAME_ROLE(
   p_role_name  IN sec_roles.name%type,
   p_new_name   IN sec_roles.name%type);
```

Description

RENAME_ROLE renames a role.

Parameters

p_role_name
Current name of the role.

p_new_name
New name to be assigned to the role.

Usage Notes

`p_role_name` must exist.

Examples

The following example renames an existing role.

```
BEGIN
  ORDS.RENAME_ROLE(
    p_role_name=>'Tickets User',
    p_new_name=>'Legacy Tickets User');
END;
/
```

ORDS.SET_MODULE_ORIGINS_ALLOWED

Format

```
ORDS.SET_MODULE_ORIGINS_ALLOWED
   p_module_name     IN ords_modules.name%type,
   p_origins_allowed IN sec_origins_allowed_modules.origins_allowed%type);
```

Description

SET_MODULE_ORIGINS_ALLOWED configures the allowed origins for a resource module. Any existing allowed origins will be replaced.

Parameters

p_module_name
Name of the resource module.

p_origins_allowed
A comma-separated list of URL prefixes. If the list is empty, any existing origins are removed.

Usage Notes

To indicate no allowed origins for a resource module (and remove any existing allowed origins), specify an empty `p_origins_allowed` value.

Examples

The following restricts the resource module `my.tickets` to two specified origins.

```
BEGIN
  ORDS.SET_MODULE_ORIGINS_ALLOWED(
    p_module_name     => 'my.tickets',
    p_origins_allowed => 'http://example.com,https://example.com');
END;
/
```

ORDS.SET_URL_MAPPING

Format

```
ORDS.SET_URL_MAPPING
   p_schema             IN ords_schemas.parsing_schema%type DEFAULT NULL,
   p_url_mapping_type   IN ords_url_mappings.type%type,
   p_url_mapping_pattern IN ords_url_mappings.pattern%type);
```

Description

SET_URL_MAPPING configures how the specified schema is mapped to request URLs.

Parameters

p_schema
Name of the schema to map. The default is the schema of the current user.

p_url_mapping_type
URL Mapping type: `BASE_PATH` or `BASE_URL`.

p_url_mapping_pattern
URL mapping pattern.

Usage Notes

Only DBA users can update the mapping of a schema other than their own.

Examples

The following example creates a `BASE_PATH` mapping for the current user.

```
BEGIN
  ORDS.SET_URL_MAPPING(
    p_url_mapping_type    => 'BASE_PATH',
    p_url_mapping_pattern => 'https://example.com/ords/ticketing'
  );
END;
/
```

ORDS.SET_SESSION_DEFAULTS

Format

```
ORDS.SET_SESSION_DEFAULTS(
       p_runtime_user IN varchar2);
```

Description

Set defaults that apply for the duration of the database session.

Parameters

p_schema
Name of the schema to map. The default is the schema of the current user.

p_runtime_user
Sets a runtime user as the target when you REST enable or disable the schemas. Otherwise all runtime users are targeted.

Usage Notes

`NULL` values have no effect. Use `RESET_SESSION_DEFAULTS` to reset values and start again.

Examples

The following example sets the `HR` user as the only grantee target for the “connect through” proxy privilege when a schema is REST enabled or disabled:

```
BEGIN
  ORDS.SET_SESSION_DEFAULTS(
    p_runtime_user => 'HR');
END;
/
```

ORDS.RESET_SESSION_DEFAULTS

Format

```
ORDS.RESET_SESSION_DEFAULTS;
```

Description

Reset session defaults back to the initial values.

Parameters

None.

Usage Notes

Use the `SET_SESSION_DEFAULTS` function to set the default values that are reset using this function.

Examples

The following example resets all the session default values:

```
BEGIN
      ORDS.RESET_SESSION_DEFAULTS;
END;
/
```

9

Oracle REST Data Services Administration PL/SQL Package Reference

The Oracle REST Data Services (ORDS) ADMIN PL/SQL package contains subprograms (procedures and functions) for developing and administering the RESTful services using Oracle REST Data Services for a privileged user.

Before a database user can invoke the `ORDS_ADMIN` package, they must be granted the `ORDS_ADMINISTRATOR_ROLE` database role.

The following example grants the `ORDS_ADMINISTRATOR_ROLE` role to the `ADMIN` user:

```
GRANT ORDS_ADMINSTRATOR_ROLE TO ADMIN;
```

The `ORDS_ADMIN` package is identical to the ORDS package except for the `AUTHID CURRENT_USER` right, without the deprecated methods and a `p_schema` parameter for every method where the target schema must be specified and some additional methods.

Related Topics

- Oracle REST Data Services PL/SQL Package Reference

ORDS_ADMIN.CREATE_ROLE

Format

```
ORDS_ADMIN.CREATE_ROLE(
   p_schema    IN ords_schemas.parsing_schema%type,
   p_role_name IN sec_roles.name%type);
```

Description

CREATE_ROLE creates an Oracle REST Data Services role with the specified name.

Parameters

p_schema
Name of the schema. This parameter is mandatory.

p_role_name
Name of the role.

Usage Notes

After the role is created, it can be associated with any Oracle REST Data Services privilege.

Examples

The following example creates a role.

```
BEGIN
  ORDS_ADMIN.CREATE_ROLE(
    p_schema => 'tickets',
    p_role_name => 'Tickets User'
  };
END;
/
```

ORDS_ADMIN.DEFINE_HANDLER

Format

```
ORDS_ADMIN.DEFINE_HANDLER(
p_schema           IN ords_schemas.parsing_schema%type,
p_module_name      IN ords_modules.name%type,
p_pattern          IN ords_templates.uri_template%type,
p_method           IN ords_handlers.method%type DEFAULT 'GET',
p_source_type      IN ords_handlers.source_type%type
DEFAULT ords_admin.source_type_collection_feed,
p_source           IN ords_handlers.source%type,
p_items_per_page   IN ords_handlers.items_per_page%type DEFAULT NULL,
p_mimes_allowed    IN ords_handlers.mimes_allowed%type DEFAULT NULL,
p_comments         IN ords_handlers.comments%type DEFAULT NULL);
```

Description

DEFINE_HANDLER defines a module handler. If the handler already exists, then the handler and any existing handlers will be replaced by this definition; otherwise, a new handler is created.

Parameters

p_schema
Name of the schema. This parameter is mandatory.

p_module_name
Name of the owning RESTful service module. Case sensitive.

p_pattern
Matching pattern for the owning resource template.

p_method
The HTTP method to which this handler will respond. Valid values: `GET` (retrieves a representation of a resource), `POST` (creates a new resource or adds a resource to a collection), `PUT` (updates an existing resource), `DELETE` (deletes an existing resource).

p_source_type
The HTTP request method for this handler. Valid values:

- `source_type_collection_feed`. Executes a SQL query and transforms the result set into an Oracle REST Data Services Standard JSON representation. Available when the HTTP method is GET. Result Format: JSON
- `source_type_collection_item`. Executes a SQL query returning one row of data into a Oracle REST Data Services Standard JSON representation. Available when the HTTP method is GET. Result Format: JSON
- `source_type_media`. Executes a SQL query conforming to a specific format and turns the result set into a binary representation with an accompanying HTTP Content-Type header identifying the Internet media type of the representation. Result Format: Binary
- `source_type_plsql`. Executes an anonymous PL/SQL block and transforms any OUT or IN/OUT parameters into a JSON representation. Available only when the HTTP method is DELETE, PUT, or POST. Result Format: JSON
- `source_type_query || source_type_csv_query`. Executes a SQL query and transforms the result set into either an Oracle REST Data Services legacy JavaScript Object Notation (JSON) or CSV representation, depending on the format selected. Available when the HTTP method is GET. Result Format: JSON or CSV
- `source_type_query_one_row`. Executes a SQL query returning one row of data into an Oracle REST Data Services legacy JSON representation. Available when the HTTP method is GET. Result Format: JSON
- `source_type_feed`. Executes a SQL query and transforms the results into a JSON Feed representation. Each item in the feed contains a summary of a resource and a hyperlink to a full representation of the resource. The first column in each row in the result set must be a unique identifier for the row and is used to form a hyperlink of the form: `path/to/feed/{id}`, with the value of the first column being used as the value for `{id}`. The other columns in the row are assumed to summarize the resource and are included in the feed. A separate resource template for the full representation of the resource should also be defined. Result Format: JSON

p_source
The source implementation for the selected HTTP method.

p_items_per_page
The default pagination for a resource handler HTTP operation GET method, that is, the number of rows to return on each page of a JSON format result set based on a database query. Default: NULL (defers to the resource module setting).

p_mimes_allowed
Comma-separated list of MIME types that the handler will accept. Applies to PUT and POST only.

p_comments
Comment text.

Usage Notes

Only one handler for each HTTP method (source type) is permitted.

Examples

The following example defines a POST handler to the `/my/tickets/` resource to accept new tickets.

```
BEGIN
  ORDS_ADMIN.DEFINE_HANDLER(
    p_schema => 'tickets',
    p_module_name => 'my.tickets',
    p_pattern => '.',
    p_method  => 'POST',
    p_mimes_allowed => 'application/json',
    p_source_type => ords_admin.source_type_plsql,
    p_source => '
      declare
        l_owner varchar2(255);
        l_payload blob;
        l_id number;
      begin
        l_payload := :body;
        l_owner := :owner;
        if ( l_owner is null ) then
          l_owner := :current_user;
        end if;
        l_id := ticket_api.create_ticket(
          p_json_entity => l_payload,
          p_author => l_owner
        );
        :location := ''./'' || l_id;
        :status := 201;
      end;
      '
  );
END;
/
```

ORDS_ADMIN.DEFINE_MODULE

Format

```
ORDS_ADMIN.DEFINE_MODULE(
   p_schema          IN ords_schemas.parsing_schema%type,
   p_module_name     IN ords_modules.name%type,
   p_base_path       IN ords_modules.uri_prefix%type,
   p_items_per_page  IN ords_modules.items_per_page%type DEFAULT 25,
   p_status          IN ords_modules.status%type DEFAULT 'PUBLISHED',
   p_comments        IN ords_modules.comments%type DEFAULT NULL);
```

Description

DEFINE_MODULE defines a resource module. If the module already exists, then the module and any existing templates will be replaced by this definition; otherwise, a new module is created.

Parameters

p_schema
Name of the schema. This parameter is mandatory.

p_module_name
Name of the owning RESTful service module. Case sensitive.

p_base_path
The base of the URI that is used to access this RESTful service. Example: `hr/` means that all URIs starting with `hr/` will be serviced by this resource module.

p_items_per_page
The default pagination for a resource handler HTTP operation GET method, that is, the number of rows to return on each page of a JSON format result set based on a database query. Default: 25.

p_status
Publication status. Valid values: `PUBLISHED` (default) or `NOT_PUBLISHED`.

p_comments
Comment text.

Usage Notes

(None.)

Examples

The following example creates a simple module.

```
BEGIN
  ORDS_ADMIN.DEFINE_MODULE(
    p_schema => 'tickets',
    p_module_name => 'my.tickets',
    p_base_path => '/my/tickets/'
  );
END;
/
```

ORDS_ADMIN.DEFINE_PARAMETER

Format

```
ORDS_ADMIN.DEFINE_PARAMETER(
   p_schema             IN ords_schemas.parsing_schema%type,
   p_module_name        IN ords_modules.name%type,
   p_pattern            IN ords_templates.uri_template%type,
   p_method             IN ords_handlers.method%type,
   p_name               IN ords_parameters.name%type ,
   p_bind_variable_name IN ords_parameters.bind_variable_name%type
                           DEFAULT NULL,
   p_source_type        IN ords_parameters.source_type%type DEFAULT 'HEADER',
   p_param_type         IN ords_parameters.param_type%type DEFAULT 'STRING',
   p_access_method      IN ords_parameters.access_method%type DEFAULT 'IN',
   p_comments           IN ords_parameters.comments%type DEFAULT NULL);
```

Description

DEFINE_PARAMETER defines a module handler parameter. If the parameter already exists, then the parameter will be replaced by this definition; otherwise, a new parameter is created.

Parameters

p_schema
Name of the schema. This parameter is mandatory.

p_module_name
Name of the owning RESTful service module. Case sensitive.

p_pattern
Matching pattern for the owning resource template.

p_method
The owning handler HTTP Method. Valid values: `GET` (retrieves a representation of a resource), `POST` (creates a new resource or adds a resource to a collection), `PUT` (updates an existing resource), `DELETE` (deletes an existing resource).

p_name
The name of the parameter, as it is named in the URI Template or HTTP Header. Used to map names that are not valid SQL parameter names.

p_bind_variable_name
The name of the parameter, as it will be referred to in the SQL. If NULL is specified, then the parameter is unbound.

p_source_type
The type that is identified if the parameter originates in the URI Template or a HTTP Header. Valid values: `HEADER`, `RESPONSE`, `URI`.

p_param_type
The native type of the parameter. Valid values: `STRING`, `INT`, `DOUBLE`, `BOOLEAN`, `LONG`, `TIMESTAMP`.

p_access_method
The parameter access method. Indicates if the parameter is an input value, output value, or both. Valid values: `IN`, `OUT`, `INOUT`.

p_comments
Comment text.

Usage Notes

All parameters must have unique names and variable names for the same handler.

Examples

The following example defines an outbound parameter on the POST handler to store the location of the created ticket.

```
BEGIN
  ORDS_ADMIN.DEFINE_PARAMETER(
    p_schema => 'tickets',
```

```
    p_module_name => 'my.tickets',
    p_pattern => '.',
    p_method => 'POST',
    p_name => 'X-APEX-FORWARD',
    p_bind_variable_name => 'location',
    p_source_type => 'HEADER',
    p_access_method => 'OUT'
  );
END;
/
```

The following example defines an outbound parameter on the POST handler to store the HTTP status of the operation.

```
BEGIN
  ORDS_ADMIN.DEFINE_PARAMETER(
    p_schema => 'tickets',
    p_module_name => 'my.tickets',
    p_pattern => '.',
    p_method => 'POST',
    p_name => 'X-APEX-STATUS-CODE',
    p_bind_variable_name => 'status',
    p_source_type => 'HEADER',
    p_access_method => 'OUT'
  );
END;
/
```

ORDS_ADMIN.DEFINE_PRIVILEGE

Format

```
ORDS_ADMIN.DEFINE_PRIVILEGE(
   p_schema            IN ords_schemas.parsing_schema%type,
   p_privilege_name    IN sec_privileges.name%type,
   p_roles             IN owa.vc_arr,
   p_patterns          IN owa.vc_arr,
   p_modules           IN owa.vc_arr,
   p_label             IN sec_privileges.label%type DEFAULT NULL,
   p_description       IN sec_privileges.description%type DEFAULT NULL,
   p_comments          IN sec_privileges.comments%type DEFAULT NULL);
or
ORDS_ADMIN.DEFINE_PRIVILEGE(
   p_schema            IN ords_schemas.parsing_schema%type,
   p_privilege_name    IN sec_privileges.name%type,
   p_roles             IN owa.vc_arr,
   p_patterns          IN owa.vc_arr,
   p_label             IN sec_privileges.label%type DEFAULT NULL,
   p_description       IN sec_privileges.description%type DEFAULT NULL,
   p_comments          IN sec_privileges.comments%type DEFAULT NULL);
or
ORDS_ADMIN.DEFINE_PRIVILEGE(
   p_schema            IN ords_schemas.parsing_schema%type,
   p_privilege_name    IN sec_privileges.name%type,
   p_roles             IN owa.vc_arr,
   p_label             IN sec_privileges.label%type DEFAULT NULL,
   p_description       IN sec_privileges.description%type DEFAULT NULL,
   p_comments          IN sec_privileges.comments%type DEFAULT NULL);
```

Description

DEFINE_PRIVILEGE defines an Oracle REST Data Services privilege. If the privilege already exists, then the privilege and any existing patterns and any associations with modules and roles will be replaced by this definition; otherwise, a new privilege is created.

Parameters

p_schema
Name of the schema. This parameter is mandatory.

p_privilege_name
Name of the privilege. No spaces allowed.

p_roles
The names of the roles, at least one of which the privilege requires. May be empty, in which case the user must be authenticated but does not require any specific role; however, must not be null. Unauthenticated users will be denied access.

p_patterns
A list of patterns.

p_modules
A list of module names referencing modules created for the current schema.

p_label
Name of this security constraint as displayed to an end user. May be null.

p_description
A brief description of the purpose of the resources protected by this constraint.

p_comments
Comment text.

Usage Notes

`p_roles`, `p_patterns`, and `p_modules` do not accept null values. If no value is to be passed, then either choose the appropriate procedure specification or pass an empty `owa.vc_arr` value.

Examples

The following example creates a privilege connected to roles, patterns, and modules:

```
DECLARE
  l_priv_roles owa.vc_arr;
  l_priv_patterns owa.vc_arr;
  l_priv_modules owa.vc_arr;
BEGIN
  l_priv_roles(1) := 'Tickets User';
  l_priv_patterns(1) := '/my/*';
  l_priv_patterns(2) := '/comments/*';
  l_priv_patterns(3) := '/tickets_feed/*';
  l_priv_patterns(4) := '/tickets/*';
  l_priv_patterns(5) := '/categories/*';
  l_priv_patterns(6) := '/stats/*';
```

```
  l_priv_modules(1) := 'my.tickets';

  ords_admin.create_role(
    p_schema    => 'tickets',
    p_role_name => 'Tickets User'
  );

  ords_admin.define_privilege(
    p_schema             => 'tickets',
    p_privilege_name     => 'tickets.privilege',
    p_roles              => l_priv_roles,
    p_patterns           => l_priv_patterns,
    P_modules            => l_priv_modules,
    p_label              => 'Task Ticketing Access',
    p_description        => 'Provides the ability to create, ' ||
                            'update and delete tickets ' ||
                            'and post comments on tickets'
  );
END;
/
```

The following example creates a privilege connected to roles and patterns:

```
DECLARE
  l_priv_roles owa.vc_arr;
  l_priv_patterns owa.vc_arr;
BEGIN
  l_priv_roles(1) := 'Tickets User';
  l_priv_patterns(1) := '/my/*';
  l_priv_patterns(2) := '/comments/*';
  l_priv_patterns(3) := '/tickets_feed/*';
  l_priv_patterns(4) := '/tickets/*';
  l_priv_patterns(5) := '/categories/*';
  l_priv_patterns(6) := '/stats/*';

  ords_admin.create_role(
    p_schema    => 'tickets',
    p_role_name => 'Tickets User'
  );

  ords_admin.define_privilege(
    p_schema             => 'tickets',
    p_privilege_name     => 'tickets.privilege',
    p_roles              => l_priv_roles,
    p_patterns           => l_priv_patterns,
    p_label              => 'Task Ticketing Access',
    p_description        => 'Provides the ability to create, ' ||
                            'update and delete tickets ' ||
                            'and post comments on tickets'
  );
END;
/
```

The following example creates a privilege connected to roles:

```
DECLARE
  l_priv_roles owa.vc_arr;
BEGIN
  l_priv_roles(1) := 'Tickets User';

  ords_admin.create_role(
```

```
    p_schema    => 'tickets',
    p_role_name => 'Tickets User'
  };

  ords_admin.define_privilege(
    p_schema             => 'tickets',
    p_privilege_name     => 'tickets.privilege',
    p_roles              => l_priv_roles,
    p_label              => 'Task Ticketing Access',
    p_description        => 'Provides the ability to create, ' ||
                            'update and delete tickets ' ||
                            'and post comments on tickets'
  );
END;
/
```

ORDS_ADMIN.DEFINE_SERVICE

Format

```
ORDS_ADMIN.DEFINE_SERVICE(
   p_schema             IN ords_schemas.parsing_schema%type,
   p_module_name        IN ords_modules.name%type,
   p_base_path          IN ords_modules.uri_prefix%type,
   p_pattern            IN ords_templates.uri_template%type,
   p_method             IN ords_handlers.method%type DEFAULT 'GET',
   p_source_type        IN ords_handlers.source_type%type
                           DEFAULT ords_admin.source_type_collection_feed,
   p_source             IN ords_handlers.source%type,
   p_items_per_page     IN ords_modules.items_per_page%type DEFAULT 25,
   p_status             IN ords_modules.status%type DEFAULT 'PUBLISHED',
   p_etag_type          IN ords_templates.etag_type%type DEFAULT 'HASH',
   p_etag_query         IN ords_templates.etag_query%type DEFAULT NULL,
   p_mimes_allowed      IN ords_handlers.mimes_allowed%type DEFAULT NULL,
   p_module_comments    IN ords_modules.comments%type DEFAULT NULL,
   p_template_comments  IN ords_modules.comments%type DEFAULT NULL,
   p_handler_comments   IN ords_modules.comments%type DEFAULT NULL);
```

Description

DEFINE_SERVICE defines a resource module, template, and handler in one call. If the module already exists, then the module and any existing templates will be replaced by this definition; otherwise, a new module is created.

Parameters

p_schema
Name of the schema. This parameter is mandatory.

p_module_name
Name of the RESTful service module. Case sensitive. Must be unique.

p_base_path
The base of the URI that is used to access this RESTful service. Example: `hr/` means that all URIs starting with `hr/` will be serviced by this resource module.

p_pattern
A matching pattern for the resource template. For example, a pattern of `/objects/:object/:id?` will match `/objects/emp/101` (matches a request for the item in the `emp` resource with `id` of 101) and will also match `/objects/emp/`. (Matches a request for the `emp` resource, because the `:id` parameter is annotated with the `?` modifier, which indicates that the `id` parameter is optional.)

p_method
The HTTP Method to which this handler will respond. Valid values: `GET` (retrieves a representation of a resource), `POST` (creates a new resource or adds a resource to a collection), `PUT` (updates an existing resource), `DELETE` (deletes an existing resource).

p_source_type
The HTTP request method for this handler. Valid values:

- `source_type_collection_feed`. Executes a SQL query and transforms the result set into an Oracle REST Data Services Standard JSON representation. Available when the HTTP method is GET. Result Format: JSON
- `source_type_collection_item`. Executes a SQL query returning one row of data into a Oracle REST Data Services Standard JSON representation. Available when the HTTP method is GET. Result Format: JSON
- `source_type_media`. Executes a SQL query conforming to a specific format and turns the result set into a binary representation with an accompanying HTTP Content-Type header identifying the Internet media type of the representation. Result Format: Binary
- `source_type_plsql`. Executes an anonymous PL/SQL block and transforms any OUT or IN/OUT parameters into a JSON representation. Available only when the HTTP method is DELETE, PUT, or POST. Result Format: JSON
- `source_type_query || source_type_csv_query`. Executes a SQL query and transforms the result set into either an Oracle REST Data Services legacy JavaScript Object Notation (JSON) or CSV representation, depending on the format selected. Available when the HTTP method is GET. Result Format: JSON or CSV
- `source_type_query_one_row`. Executes a SQL query returning one row of data into an Oracle REST Data Services legacy JSON representation. Available when the HTTP method is GET. Result Format: JSON
- `source_type_feed`. Executes a SQL query and transforms the results into a JSON Feed representation. Each item in the feed contains a summary of a resource and a hyperlink to a full representation of the resource. The first column in each row in the result set must be a unique identifier for the row and is used to form a hyperlink of the form: `path/to/feed/{id}`, with the value of the first column being used as the value for `{id}`. The other columns in the row are assumed to summarize the resource and are included in the feed. A separate resource template for the full representation of the resource should also be defined. Result Format: JSON

p_source
The source implementation for the selected HTTP method.

p_items_per_page
The default pagination for a resource handler HTTP operation GET method, that is, the number of rows to return on each page of a JSON format result set based on a database query. Default: NULL (defers to the resource module setting).

p_status
Publication status. Valid values: `PUBLISHED` (default) or `NOT_PUBLISHED`.

p_etag_type
A type of entity tag to be used by the resource template. An entity tag is an HTTP Header that acts as a version identifier for a resource. Use entity tag headers to avoid retrieving previously retrieved resources and to perform optimistic locking when updating resources. Valid values are `HASH`, `QUERY`, `NONE`:

- `HASH` (known as Secure HASH): The contents of the returned resource representation are hashed using a secure digest function to provide a unique fingerprint for a given resource version.
- `QUERY`: Manually define a query that uniquely identifies a resource version. A manually defined query can often generate an entity tag more efficiently than hashing the entire resource representation.
- `NONE`: Do not generate an entity tag.

p_etag_query
Query that is used to generate the entity tag.

p_mimes_allowed
Comma-separated list of MIME types that the handler will accept. Applies to PUT and POST only.

p_module_comments
Comment text.

p_template_comments
Comment text.

p_handler_comments
Comment text.

Usage Notes

Creates a resource module, template, and handler in one call.

Examples

The following example defines a REST service that retrieves the current user's tickets.

```
BEGIN
  ORDS_ADMIN.DEFINE_SERVICE(
    p_schema => 'tickets',
    p_module_name => 'my.tickets',
    p_base_path => '/my/tickets/',
    p_pattern => '.',
    p_source => 'select t.id "$.id", t.id, t.title from tickets t' ||
                ' where t.owner = :current_user order by t.updated_on desc'
  );
END;
/
```

The following example defines a REST service that retrieves tickets filtered by category.

```
BEGIN
  ORDS_ADMIN.DEFINE_SERVICE(
    p_schema => 'tickets',
    p_module_name => 'by.category',
    p_base_path => '/by/category/',
    p_pattern => ':category_id',
    p_source => 'select ''../../my/tickets/'' ||
                  t.id "$.id", t.id, t.title' ||
                ' from tickets t, categories c, ticket_categories tc' ||
                ' where c.id = :category_id and c.id = tc.category_id and' ||
                ' tc.ticket_id = t.id order by t.updated_on desc'
  );
END;
/
```

ORDS_ADMIN.DEFINE_TEMPLATE

Format

```
ORDS_ADMIN.DEFINE_TEMPLATE(
   p_schema       IN ords_schemas.parsing_schema%type,
   p_module_name  IN ords_modules.name%type,
   p_pattern      IN ords_templates.uri_template%type,
   p_priority     IN ords_templates.priority%type DEFAULT 0,
   p_etag_type    IN ords_templates.etag_type%type DEFAULT 'HASH',
   p_etag_query   IN ords_templates.etag_query%type DEFAULT NULL,
   p_comments     IN ords_templates.comments%type DEFAULT NULL);
```

Description

DEFINE_TEMPLATE defines a resource template. If the template already exists, then the template and any existing handlers will be replaced by this definition; otherwise, a new template is created.

Parameters

p_schema
Name of the schema. This parameter is mandatory.

p_module_name
Name of the owning RESTful service module. Case sensitive.

p_pattern
A matching pattern for the resource template. For example, a pattern of `/objects/:object/:id?` will match `/objects/emp/101` (matches a request for the item in the `emp` resource with `id` of 101) and will also match `/objects/emp/`. (Matches a request for the `emp` resource, because the `:id` parameter is annotated with the `?` modifier, which indicates that the `id` parameter is optional.)

p_priority
The priority for the order of how the resource template should be evaluated: 0 (low priority. the default) through 9 (high priority).

p_etag_type
A type of entity tag to be used by the resource template. An entity tag is an HTTP Header that acts as a version identifier for a resource. Use entity tag headers to avoid retrieving previously retrieved resources and to perform optimistic locking when updating resources. Valid values are `HASH`, `QUERY`, `NONE`:

- `HASH` (known as Secure HASH): The contents of the returned resource representation are hashed using a secure digest function to provide a unique fingerprint for a given resource version.
- `QUERY`: Manually define a query that uniquely identifies a resource version. A manually defined query can often generate an entity tag more efficiently than hashing the entire resource representation.
- `NONE`: Do not generate an entity tag.

p_etag_query
Query that is used to generate the entity tag.

p_comments
Comment text.

Usage Notes

he resource template pattern must be unique with a resource module.

Examples

The following example defines a resource for displaying ticket items.

```
BEGIN
  ORDS_ADMIN.DEFINE_TEMPLATE(
    p_schema => 'tickets',
    p_module_name => 'my.tickets',
    p_pattern => '/:id'
  );
END;
/
```

ORDS_ADMIN.DELETE_MODULE

Format

```
ORDS_ADMIN.DELETE_MODULE(
   p_schema       IN ords_schemas.parsing_schema%type,
   p_module_name  IN ords_modules.name%type);
```

Description

DELETE_MODULE deletes a resource module.

Parameters

p_schema
Name of the schema. This parameter is mandatory.

p_module_name
Name of the owning RESTful service module. Case sensitive.

Usage Notes

If the module does not already exist or is accessible to the current user, then no exception is raised.

Examples

The following example deletes a resource module.

```
BEGIN
  ORDS_ADMIN.DELETE_MODULE(
    p_schema => 'tickets',
    p_module_name => 'my.tickets'
  );
END;
/
```

ORDS_ADMIN.DELETE_PRIVILEGE

Description

DELETE_PRIVILEGE deletes a privilege.

Parameters

p_schema
Name of the schema. This parameter is mandatory.

p_name
Name of the privilege.

Usage Notes

If the privilege does not already exist, then no exception is raised.

Examples

The following example deletes a privilege.

```
BEGIN
  ORDS_ADMIN.DELETE_PRIVILEGE(
    p_schema => 'tickets',
    p_name => 'tickets.privilege'
  );
END;
/
```

ORDS_ADMIN.DELETE_ROLE

Format

```
ORDS_ADMIN.DELETE_ROLE(
   p_schema    IN ords_schemas.parsing_schema%type,
   p_role_name IN sec_roles.name%type);
```

Description

DELETE_ROLE deletes the named role.

Parameters

p_name
Name of the role.

Usage Notes

This will also delete any association between the role and any privileges that reference the role.

No exception is produced if the role does not already exist.

Examples

The following example deletes a role.

```
BEGIN
  ORDS_ADMIN.DELETE_ROLE(
    p_schema => 'tickets',
    p_role_name => 'Tickets User'
  );
END;
/
```

ORDS_ADMIN.DROP_REST_FOR_SCHEMA

Format

```
ORDS_ADMIN.DROP_REST_FOR_SCHEMA(
   p_schema ords_schemas.parsing_schema%type);
```

Description

DROP_REST_FOR_SCHEMA deletes all auto-REST Oracle REST Data Services metadata for the associated schema.

Parameters

p_schema
Name of the schema.

Usage Notes

This procedure effectively "undoes" the actions performed by the `ORDS.Enable_Schema` procedure.

Examples

The following example deletes all auto-REST Oracle REST Data Services metadata for the TICKETS schema.

```
BEGIN
  ORDS_ADMIN.DROP_REST_FOR_SCHEMA(
    p_schema => 'tickets'
```

```
  );
END;
/
```

ORDS_ADMIN.ENABLE_OBJECT

Format

```
ORDS_ADMIN.ENABLE_OBJECT(
   p_enabled         IN boolean DEFAULT TRUE,
   p_schema          IN ords_schemas.parsing_schema%,
   p_object          IN ords_objects.parsing_object%type,
   p_object_type     IN ords_objects.type%type DEFAULT 'TABLE',
   p_object_alias    IN ords_objects.object_alias%type DEFAULT NULL,
   p_auto_rest_auth  IN boolean DEFAULT NULL);
```

Description

ENABLE_OBJECT enables Oracle REST Data Services access to a specified table or view in a schema.

Parameters

p_enabled
`TRUE` to enable access; `FALSE` to disable access.

p_schema
Name of the schema for the table or view. This parameter is mandatory.

p_object
Name of the table or view.

p_object_type
Type of the object: `TABLE` (default) or `VIEW`.

p_object_alias
Alias of the object.

p_auto_rest_auth
Controls whether Oracle REST Data Services should require user authorization before allowing access to the Oracle REST Data Services metadata for this object. If this value is `TRUE`, then the service is protected by the following roles:

- `oracle.dbtools.autorest.any.schema`
- `oracle.dbtools.role.autorest.<SCHEMANAME>.<OBJECTNAME>`

Usage Notes

None.

Examples

The following example enables a table named CATEGORIES.

```
BEGIN
  ORDS_ADMIN.ENABLE_OBJECT(
    p_schema => 'tickets',
    p_object=>'CATEGORIES'
```

```
  );
END;
/
```

The following example enables a view named TICKETS_FEED.

```
BEGIN
  ORDS_ADMIN.ENABLE_OBJECT(
    p_schema => 'tickets',
    p_object => 'TICKETS_FEED',
    p_object_type => 'VIEW'
  );
END;
/
```

ORDS_ADMIN.DROP_REST_FOR_OBJECT

Format

```
ORDS_ADMIN.DROP_REST_FOR_OBJECT(
   p_schema          IN ords_schemas.parsing_schema%,
   p_object          IN ords_objects.parsing_object%type);
```

Description

DROP_REST_FOR_OBJECT deletes all auto-REST Oracle REST Data Services metadata for the associated schema object.

Parameters

p_schema
Name of the schema.

p_object
Name of the table or view.

Usage Notes

This procedure effectively "undoes" the actions performed by the `ORDS_ADMIN.ENABLE_OBJECT` procedure.

Examples

The following example deletes all auto-REST Oracle REST Data Services metadata for the TICKETS schema CATEGORIES table.

```
BEGIN
  ORDS_ADMIN.DROP_REST_FOR_OBJECT(
    p_schema => 'tickets',
    p_object=>'CATEGORIES'
  );
END;
/
```

ORDS_ADMIN.ENABLE_SCHEMA

Format

```
ORDS_ADMIN.ENABLE_SCHEMA
   p_enabled             IN boolean DEFAULT TRUE,
   p_schema              IN ords_schemas.parsing_schema%type,
   p_url_mapping_type    IN ords_url_mappings.type%type DEFAULT 'BASE_PATH',
   p_url_mapping_pattern IN ords_url_mappings.pattern%type DEFAULT NULL,
   p_auto_rest_auth      IN boolean DEFAULT NULL);
```

Description

ENABLE_SCHEMA enables Oracle REST Data Services to access the named schema.

Parameters

p_enabled
`TRUE` to enable Oracle REST Data Services access; `FALSE` to disable Oracle REST Data Services access.

p_schema
Name of the schema. This parameter is mandatory.

p_url_mapping_type
URL Mapping type: `BASE_PATH` or `BASE_URL`.

p_url_mapping_pattern
URL mapping pattern.

p_auto_rest_auth
For a schema, controls whether Oracle REST Data Services should require user authorization before allowing access to the Oracle REST Data Services metadata catalog of this schema.

Usage Notes

None.

Examples

The following example enables the current schema.

```
BEGIN
  ORDS_ADMIN.ENABLE_SCHEMA(
    p_schema => 'tickets'
  );
END;
/
```

ORDS_ADMIN.PUBLISH_MODULE

Format

```
ORDS_ADMIN.PUBLISH_MODULE(
   p_schema       IN ords_schemas.parsing_schema%type,
   p_module_name  IN ords_modules.name%type,
   p_status       IN ords_modules.status%type DEFAULT 'PUBLISHED');
```

Description

PUBLISH_MODULE changes the publication status of an Oracle REST Data Services resource module.

Parameters

p_schema
Name of the schema. This parameter is mandatory.

p_module_name
Current name of the RESTful service module. Case sensitive.

p_status
Publication status. Valid values: `PUBLISHED` (default) or `NOT_PUBLISHED`.

Usage Notes

(None.)

Examples

The following example publishes a previously defined module named `my.tickets`.

```
BEGIN
  ORDS_ADMIN.PUBLISH_MODULE(
    p_schema => 'tickets',
    p_module_name => 'my.tickets'
  );
END;
/
```

ORDS_ADMIN.RENAME_MODULE

Format

```
ORDS_ADMIN.RENAME_MODULE(
   p_schema         IN ords_schemas.parsing_schema%type,
   p_module_name    IN ords_modules.name%type,
   p_new_name       IN ords_modules.name%type DEFAULT NULL,
   p_new_base_path  IN ords_modules.uri_prefix%type DEFAULT NULL);
```

Description

RENAME_MODULE lets you change the name or the base path, or both, of an Oracle REST Data Services resource module.

Parameters

p_schema
Name of the schema. This parameter is mandatory.

p_module_name
Current name of the RESTful service module. Case sensitive.

p_new_name
New name to be assigned to the RESTful service module. Case sensitive. If this parameter is null, the name is not changed.

p_new_base_path
The base of the URI to be used to access this RESTful service. Example: hr/ means that all URIs starting with hr/ will be serviced by this resource module. If this parameter is null, the base path is not changed.

Usage Notes

Both the new resource module name and the base path must be unique within the enabled schema.

Examples

The following example renames resource module `my.tickets` to `old.tickets`.

```
BEGIN
  ORDS_ADMIN.RENAME_MODULE(
    p_schema => 'tickets',
    p_module_name =>'my.tickets',
    p_new_name=>'old.tickets',
    p_new_base_path=>'/old/tickets/');
END;
/
```

ORDS_ADMIN.RENAME_PRIVILEGE

Format

```
ORDS_ADMIN.RENAME_PRIVILEGE(
   p_schema    IN ords_schemas.parsing_schema%type,
   p_name      IN sec_privileges.name%type,
   p_new_name  IN sec_privileges.name%type);
```

Description

RENAME_PRIVILEGE renames a privilege.

Parameters

p_schema
Name of the schema. This parameter is mandatory.

p_name
Current name of the privilege.

p_new_name
New name to be assigned to the privilege.

Usage Notes

(None.)

Examples

The following example renames the privilege `tickets.privilege` to `old.tickets.privilege`.

```
BEGIN
  ORDS_ADMIN.RENAME_PRIVILEGE(
    p_schema => 'tickets',
    p_name =>'tickets.privilege',
    p_new_name=>'old.tickets.privilege');
END;
/
```

ORDS_ADMIN.RENAME_ROLE

Format

```
ORDS_ADMIN.RENAME_ROLE(
   p_schema     IN ords_schemas.parsing_schema%type,
   p_role_name  IN sec_roles.name%type,
   p_new_name   IN sec_roles.name%type);
```

Description

RENAME_ROLE renames a role.

Parameters

p_schema
Name of the schema. This parameter is mandatory.

p_role_name
Current name of the role.

p_new_name
New name to be assigned to the role.

Usage Notes

`p_role_name` must exist.

Examples

The following example renames an existing role.

```
BEGIN
  ORDS_ADMIN.RENAME_ROLE(
    p_schema=>'tickets',
    p_role_name=>'Tickets User',
    p_new_name=>'Legacy Tickets User');
END;
/
```

ORDS_ADMIN.SET_MODULE_ORIGINS_ALLOWED

Format

```
ORDS_ADMIN.SET_MODULE_ORIGINS_ALLOWED
   p_schema          IN ords_schemas.parsing_schema%type,
   p_module_name     IN ords_modules.name%type,
   p_origins_allowed IN sec_origins_allowed_modules.origins_allowed%type);
```

Description

SET_MODULE_ORIGINS_ALLOWED configures the allowed origins for a resource module. Any existing allowed origins will be replaced.

Parameters

p_schema
Name of the schema. This parameter is mandatory.

p_module_name
Name of the resource module.

p_origins_allowed
A comma-separated list of URL prefixes. If the list is empty, any existing origins are removed.

Usage Notes

To indicate no allowed origins for a resource module (and remove any existing allowed origins), specify an empty `p_origins_allowed` value.

Examples

The following restricts the resource module `my.tickets` to two specified origins.

```
BEGIN
  ORDS_ADMIN.SET_MODULE_ORIGINS_ALLOWED(
    p_schema          => 'tickets',
    p_module_name     => 'my.tickets',
    p_origins_allowed => 'http://example.com,https://example.com');
END;
/
```

ORDS_ADMIN.SET_URL_MAPPING

Format

```
ORDS_ADMIN.SET_URL_MAPPING
   p_schema              IN ords_schemas.parsing_schema%,
   p_url_mapping_type    IN ords_url_mappings.type%type,
   p_url_mapping_pattern IN ords_url_mappings.pattern%type);
```

Description

SET_URL_MAPPING configures how the specified schema is mapped to request URLs.

Parameters

p_schema
Name of the schema to map. This parameter is mandatory.

p_url_mapping_type
URL Mapping type: `BASE_PATH` or `BASE_URL`.

p_url_mapping_pattern
URL mapping pattern.

Usage Notes

(None.)

Examples

The following example creates a `BASE_PATH` mapping for the tickets user.

```
BEGIN
  ORDS_ADMIN.SET_URL_MAPPING(
    p_schema              => 'tickets',
    p_url_mapping_type    => 'BASE_PATH',
    p_url_mapping_pattern => 'https://example.com/ords/ticketing'
  );
END;
/
```

ORDS_ADMIN.ENABLE_HOUSEKEEPING_JOB

Format

```
ORDS_ADMIN.ENABLE_HOUSEKEEPING_JOB(p_enabled IN boolean DEFAULT TRUE);
```

Description

`ENABLE_HOUSEKEEPING_JOB` creates and enables or disables the ORDS `DBMS_SCHEDULER` housekeeping job. The job name is `ORDS_HOUSEKEEPING_JOB` which replaces the deprecated job, `CLEAN_OLD_ORDS_SESSIONS`.

Parameters

p_enabled
`TRUE` to enable ORDS HOUSEKEEPING_JOB; `FALSE` to disable it. A NULL value will create and enable the job if it does not already exist otherwise its enablement state will remain changed.

Usage Notes

The job runs every hour and performs housekeeping actions on the ORDS metadata repository. No commit is required.

Examples

The following example enables the housekeeping job:

```
EXECUTE ORDS_ADMIN.ENABLE_HOUSEKEEPING_JOB;
```

ORDS_ADMIN.DROP_HOUSEKEEPING_JOB

Format

```
ORDS_ADMIN.DROP_HOUSEKEEPING_JOB;
```

Description

`DROP_HOUSEKEEPING_JOB` drops the ORDS `DBMS_SCHEDULER` housekeeping job. The job name is `ORDS_HOUSEKEEPING_JOB`.

Parameters

None.

Usage Notes

No commit is required.

Examples

The following example drops the housekeeping job:

```
EXECUTE ORDS_ADMIN.DROP_HOUSEKEEPING_JOB;
```

ORDS_ADMIN.PERFORM_HOUSEKEEPING

Format

```
ORDS_ADMIN.PERFORM_HOUSEKEEPING;
```

Description

`PERFORM_HOUSEKEEPING` performs ORDS housekeeping actions immediately. The following action is performed:

- Removes expired sessions that are older than one day.

Parameters

None.

Usage Notes

No commit is required.

Examples

The following example performs the housekeeping actions immediately against the ORDS metadata repository:

```
EXECUTE ORDS_ADMIN.PERFORM_HOUSEKEEPING;
```

ORDS_ADMIN.SET_SESSION_DEFAULTS

Format

```
ORDS_ADMIN.SET_SESSION_DEFAULTS(
       p_runtime_user IN varchar2);
```

Description

Sets the default values that apply for the duration of the database session.

Parameters

p_runtime_user
Sets a runtime user as the target while REST enabling or disabling the schemas. Otherwise all runtime users are targeted.

Usage Notes

`NULL` values have no effect. Use `RESET_SESSION_DEFAULTS` function to reset the values and start again.

Examples

The following example sets the `HR` user as the only grantee target for the “connect through” proxy privilege when a schema is REST enabled or disabled:

```
BEGIN
  ORDS_ADMIN.SET_SESSION_DEFAULTS(
    p_runtime_user => 'HR');
END;
/
```

ORDS_ADMIN.RESET_SESSION_DEFAULTS

Format

```
ORDS_ADMIN.RESET_SESSION_DEFAULTS
```

Description

Resets the session defaults back to the initial values.

Parameters

None.

Usage Notes

Use `SET_SESSION_DEFAULTS` function to set the default values that were reset using this function.

Examples

The following example resets all the session default values:

```
BEGIN
  ORDS_ADMIN.RESET_SESSION_DEFAULTS;
END;
/
```

ORDS_ADMIN.PROVISION_ADMIN_ROLE

Format

```
ORDS_ADMIN.PROVISION_ADMIN_ROLE(
       p_ user IN varchar2);
```

Description

Provision a database user with the ORDS Administrator role so that it can administer ORDS.

Parameters

p_user
The name of the user to be provisioned.

Usage Notes

User `ORDS_PUBLIC_USER` cannot be configured using this interface.

Examples

The following example provisions the ORDS administrator role to the `HR` user:

```
BEGIN
  ORDS_ADMIN.PROVISION_ADMIN_ROLE(
  p_user => 'HR'
  );
END;
/
```

ORDS_ADMIN.PROVISION_RUNTIME_ROLE

Format

```
ORDS_ADMIN.PROVISION_RUNTIME_ROLE(
       p_user                  IN  varchar2,
       p_proxy_enabled_schemas IN  boolean DEFAULT TRUE);
```

Description

Provision a database user so that it can act as an ORDS runtime user.

Parameters

p_user
The name of the user to be provisioned.

p_proxy_enabled_schemas
When the value is set to TRUE, “connect through” proxy grants are added for any enabled schemas.

Usage Notes

ORDS_PUBLIC_USER is an example of a runtime user. Additional changes to the ORDS configuration are required to use a user other than the ORDS_PUBLIC_USER.

Examples

The following example provisions the ORDS runtime role to the HR user and grants it the “connect through” proxy privilege for all the enabled schemas:

```
BEGIN
  ORDS_ADMIN.PROVISION_RUNTIME_ROLE(
   p_user => 'HR',
   p_proxy_enabled_schemas => TRUE
 );
END;
/
```

ORDS_ADMIN.UNPROVISION_ ROLES

Format

```
ORDS_ADMIN.UNPROVISION_ROLES(
       p_user               IN  varchar2,
       p_administrator_role IN  boolean DEFAULT NULL,
       p_runtime_role       IN  boolean DEFAULT NULL);
```

Description

Unprovision the ORDS database roles.

Parameters

p_user
The name of the user to be unprovisioned.

p_administrator_role
Unprovision as an admin user.

p_runtime_role
Unprovision as a runtime user.

Usage Notes

NULL boolean values are evaluated to TRUE unless any value is set to TRUE. In such case, NULL values are evaluated to FALSE. So, by default all the roles are unprovisioned unless an explicit choice is made.

Examples

The following example unprovisions the ORDS administrator role from the HR user:

```
BEGIN
  ORDS_ADMIN.UNPROVISION_ROLES (
    p_user => 'HR',
    p_administrator_role => TRUE);
END;
/
```

10

Implicit Parameters

This chapter describes the implicit parameters used in REST service handlers that are not explicitly declared. Oracle REST Data Services (ORDS) adds these parameters automatically to the resource handlers.

List of Implicit Parameters

The following table lists the implicit parameters:

> **Note:**
>
> Parameter names are case sensitive. For example, :CURRENT_USER is not a valid implicit parameter.

Table 10-1 List of Implicit Parameters

Name	Type	Access Mode	HTTP Header	Description	Introduced
`:body`	BLOB	IN	N/A	Specifies the body of the request as a temporary BLOB.	2.0
`:body_text`	CLOB	IN	N/A	Specifies the body of the request as a temporary CLOB.	18.3

Table 10-1 (Cont.) List of Implicit Parameters

Name	Type	Access Mode	HTTP Header	Description	Introduced
`:content_type`	VARCHAR	IN	`Content-Type`	Specifies the MIME type of the request body, as indicated by the Content-Type request header.	2.0
`:current_user`	VARCHAR	IN	N/A	Specifies the authenticated user for the request. If no user is authenticated, then the value is set to null.	2.0
`:forward_location`	VARCHAR	OUT	`X-ORDS-FORWARD-LOCATION`	Specifies the location where Oracle REST Data Services must forward a GET request to produce the response for this request.	18.3

Table 10-1 (Cont.) List of Implicit Parameters

Name	Type	Access Mode	HTTP Header	Description	Introduced
`:fetch_offset`	NUMBER	IN	N/A	Specifies the zero-based offset of the first row to be displayed on a page.	18.3
`:fetch_size`	NUMBER	IN	N/A	Specifies the maximum number of rows to be retrieved on a page.	18.3
`:page_offset`	NUMBER	IN	N/A	Specifies the zero based page offset in a paginated request. **Note:** The `:page_offset` parameter is deprecated. Use `:row_offset` parameter instead.	2.0

Table 10-1 (Cont.) List of Implicit Parameters

Name	Type	Access Mode	HTTP Header	Description	Introduced
`:page_size`	NUMBER	IN	N/A	Specifies the maximum number of rows to be retrieved on a page. The `:page_size` parameter is deprecated. Use `:fetch_size` parameter instead.	2.0
`:row_offset`	NUMBER	IN	N/A	Specifies the one-based index of the first row to be displayed in a paginated request.	3.0
`:row_count`	NUMBER	IN	N/A	Specifies the one-based index of the last row to be displayed in a paginated request.	3.0

Table 10-1 (Cont.) List of Implicit Parameters

Name	Type	Access Mode	HTTP Header	Description	Introduced
:status_code	NUMBER	OUT	X-ORDS-STATUS-CODE	Specifies the HTTP status code for the request.	18.3

About the :body parameter

The :body implicit parameter is used in the resource handlers to receive the contents of the request body as a temporary BLOB.

> **Note:**
>
> Only POST or PUT requests can have a request body. The HTTP specification does not permit request bodies on GET or DELETE requests.

Example 10-1 Example

The following example illustrates a PL/SQL block that stores the request body in a database table:

```
begin
 insert into tab (content) values (:body);
end;
```

> **Note:**
>
> The :body implicit parameter **must** be dereferenced exactly once in a PL/SQL block. If it is dereferenced more than once, then the second and subsequent dereferences will appear to be empty. This is because the client sends the request body only once.

The following example will **not** work as intended because it dereferences the :body parameter twice:

```
begin
 insert into tab1(content) values (:body); -- request body will be
inserted
 insert into tab2(content) values (:body); -- an empty blob will be
inserted
end;
```

To avoid this limitation, the `:body` parameter value must be assigned to a local PL/SQL variable before it is used. This enables the local variable to be dereferenced more than once:

```
declare
 l_content blob := :body;
begin
 insert into tabl(content) values(l_content);
 insert into tab2(content) values(l_content);
end;
```

About the :body_text Parameter

The `:body_text` implicit parameter is used in the resource handlers to receive the contents of the request body as a temporary CLOB. Typically, the content of the request body is textual (for example JSON or HTML content) and so, receiving the request body as a CLOB saves the resource handler author from the effort of converting the `:body` BLOB parameter to a CLOB instance. Similar to the `:body` parameter, the `:body_text` parameter **must** be dereferenced only once in a PL/SQL block.

It is recommended to assign the `:body_text` value to a local PL/SQL variable, and the PL/SQL variable is used throughout the PL/SQL block.

About the :content_type Parameter

The `:content_type` implicit parameter provides the value of the Content-Type request header supplied with the request. If no Content-Type header is present in the request, then a null value is returned.

About the :current_user Parameter

The `:current_user` implicit parameter provides the identity of the user authenticated for the request.

> **Note:**
>
> In a scenario, where the user is not authenticated, the value is set to null. For example, if the request is for a public resource, then the value will be set to null.

About the :status_code Parameter

The `:status_code` implicit parameter enables a resource handler to indicate the HTTP status code value to include in a response. The value must be one of the numeric values defined in the **HTTP Specification** document.

About the :forward_location Parameter

The `:forward_location` implicit parameter provides a mechanism for PL/SQL based resource handlers to produce a response for a request.

Consider a POST request that results in the creation of a new resource. Typically, the response of a POST request for REST APIs contains the location of the newly created resource (in the Location response header) along with the representation of the new resource. The presence of the Location header in the response indicates that there must be a GET resource handler that can produce a response for the specified location.

Instead of applying logic to the POST resource handler to render the representation of the new resource in the response, the resource handler can delegate that task to the existing GET Resource Handler.

The following resource handler defines a POST handler that delegates the generation of the response to a GET resource handler:

```
ords.define_handler(
  p_module_name => 'tickets.collection',
  p_pattern => '.',
  p_method  => 'POST',
  p_mimes_allowed => 'application/json',
  p_source_type => ords.source_type_plsql,
  p_source => '
   declare
    l_owner varchar2(255);
    l_payload clob;
    l_id number;
   begin
    l_payload := :body_text;
    l_owner := :current_user;
    l_id := ticket_api.create_ticket(
      p_json_entity => l_payload,
      p_author => l_owner
    );
    :forward_location := ''./'' || l_id;
    :status_code := 201;
   end;
  '
);
```

Where:

- The `ords.define_handler` API is used to add a POST handler to an existing resource module named `tickets.collection`.
- The `p_pattern` with value '.' indicates that the POST handler should be bound to the root resource of the resource module. If the base path of the `tickets.collection`' is `/tickets/`, then the POST handler is bound to the `/tickets/` URL path.

- The `p_mimes_allowed` value indicates that the POST request must have a Content-Type header value of `application/json`'.
- The `p_source_type` value indicates that the source of the POST handler is a PL/SQL block.
- The `p_source` value contains the source of the PL/SQL block:

  Where:

  > **Note:**
  >
  > The `:body_text` implicit parameter is assigned to a local variable, so that it can be dereferenced more than once.

  – The identity of the user, making the POST request, is determined from the `:current_user` implicit parameter.

  – The PL/SQL block, delegates the task of storing the request payload to a PL/SQL package level function. The PL/SQL block should only contain logic to bridge from the HTTP request to the PL/SQL package invocation.

  > **Note:**
  >
  > When all the data modification operations are wrapped in a PL/SQL API, the PL/SQL block can be independently unit tested. Long and complicated PL/SQL blocks are an anti-pattern indicative of code that is difficult to test and maintain.

  – The PL/SQL package level function returns the ID of the newly created resource.

  – The `:forward_location` implicit parameter is assigned the value of `'./' || l_id`. For example, if the value of `l_id` is 4256, then the value of `:forward_location` is `/tickets/4256` .

  When ORDS evaluates the preceding PL/SQL block and checks the value assigned to the `:forward_location` implicit parameter, it initiates a GET request against the specified location (for example, `/tickets/4256`) and return the response generated by the GET request as the response of the POST request. In addition, ORDS includes a location response header with the fully resolved URL of the `:forward_location` value.

  – The `:status_code` implicit parameter is assigned the HTTP response status code value. The 201 (Created) status code indicates that a new resource is created. This value will override the status code generated by the GET request.

About the Pagination Implicit Parameters

The following table lists the pagination implicit parameters:

> **Note:**
>
> Oracle REST Data Services reserves the use of the query parameters, `page`, `offset`, and `limit`. It is not permitted to define REST services that use named bind parameters with any of the preceding query parameter names. Alternatively, REST services must use the appropriate pagination implicit parameters defined in the following table:

Table 10-2 Pagination Implicit Parameters

Name	Description	Status
`:page_offset`	Specifies the zero based page offset in a pagination request.	Deprecated
`:page_size`	Specifies the maximum number of rows to be retrieved on a page.	Deprecated
`:row_offset`	Specifies the index of the first row to be displayed in a pagination request.	Not Recommended
`:row_count`	Specifies the index of the last row to displayed in a pagination request.	Not Recommended
`:fetch_offset`	Specifies the zero based index of the first row to be displayed on a page.	Recommended
`:fetch_size`	Specifies the maximum number of rows to be retrieved on a page.	Recommended

About the :page_offset Parameter

The `:page_offset` implicit parameter is provided for backward compatibility, so it is used only with `source_type_query` source type resource handlers.

> **Note:**
>
> - The `source_type_query` source type is deprecated, instead use the `source_type_collection` feed parameter.
> - The `:page_offset` implicit parameter is deprecated, instead use the `:row_offset` implicit parameter.

About the :page_size Parameter

The `:page_size` implicit parameter is used to indicate the maximum number of rows to be retrieved on a page. `:page_size` parameter is provided for backward compatibility. This parameter is deprecated, instead use `:fetch_size` implicit parameter.

About the :row_offset Parameter

The `:row_offset` implicit parameter indicates the number of the first row to be displayed on a page. The `:row_offset` implicit parameter is used when you are using both a wrapper pagination query and `row_number()` (used in Oracle 11g and earlier releases). Starting Oracle 12c or later releases, Oracle recommends using the `:fetch_offset` implicit parameter and a row limiting clause instead of the `:row_offset` parameter.

About the :row_count Parameter

The `:row_count` implicit parameter is used to indicate the number of rows to be displayed on a page. The `:row_count` value is the value of the sum of `:row_offset` and the pagination size. The `:row_count` implicit parameter is useful when implementing pagination using a wrapper pagination query and `row_number()`method that was used in Oracle database 11g and earlier releases. Starting Oracle Database release 12c or later, Oracle recommends that you use `:fetch_size` parameter and a row limiting clause instead.

About the :fetch_offset Parameter

The `:fetch_offset` implicit parameter is used to indicate the zero based offset of the first row to display in a given page. The `:fetch_offset` implicit parameter is used when you implement pagination using a row limiting clause, which is recommended for use with Oracle 12c and later releases.

About the :fetch_size Parameter

The `:fetch_size` implicit parameter is used to indicate the maximum number of rows to retrieve on a page. ORDS always sets the value of `:fetch_size` to the pagination size plus one. The presence or absence of the extra row helps ORDS in determining if there is a subsequent page in the results or not.

> **Note:**
>
> The extra row that is queried is never displayed on the page.

About Automatic Pagination

This section describes the automatic pagination process.

If a GET resource handler source type, `source_type_collection_feed` or `source_type_query` has a non zero pagination size (`p_items_per_page`) and the source of the GET resource handler does not dereference any of the implicit pagination parameters discussed in the preceding sections, then ORDS automatically wraps the query in a pagination clause to constrain the query results to include only the values from the requested page. With automatic pagination, the resource handler

author needs to specify only the pagination size, and ORDS automatically handles the remaining effort in paginating the resource.

> **Note:**
>
> All resource modules have a default pagination size (`p_items_per_page`) of 25. So, by default automatic pagination is enabled.

About Manual Pagination

This section describes the manual pagination process.

In some scenarios, a GET resource handler needs to perform pagination on its own rather than delegating the pagination process to ORDS. In such cases, the source of the GET resource handler will dereference one or more implicit pagination parameters discussed in the preceding sections.

> **Note:**
>
> The GET resource handler must specify the desired pagination size so that ORDS can correctly calculate the required values for the implicit pagination parameters.

Examples

Manual pagination example using row limiting clause

The following example defines a REST service that uses a row limiting clause to paginate the query result set. This is the recommended way to implement manual pagination:

```
begin
 ords.define_service(
   p_module_name => 'example.paging',
   p_base_path => '/example/',
   p_pattern => '/paged',
   p_items_per_page => 7,
   p_source => 'select * from emp e order by empno desc
offset :fetch_offset rows fetch next :fetch_size rows only'
 );
 commit;
end;
```

Manual pagination example using row_number() method

The following example defines a REST service that uses a wrapper query and `row_number()` method. This approach is not recommended.

```
begin
ords.define_service(
   p_module_name => 'example.paging',
```

```
   p_base_path => '/example/',
   p_pattern => '/paged',
   p_items_per_page => 7,
   p_source => 'select * from (select q_.* , row_number() over (order
by 1) rn__ from (select * from emp e order by empno desc) q_ )where
rn__ between :row_offset and :row_count'
 );
 commit;
end;
```

11

OAUTH PL/SQL Package Reference

The OAUTH PL/SQL package contains procedures for implementing OAuth authentication using Oracle REST Data Services.

Related Topics

- Using the Oracle REST Data Services PL/SQL API

OAUTH.CREATE_CLIENT

Format

```
OAUTH.CREATE_CLIENT(
   p_name            VARCHAR2 IN,
   p_grant_type      VARCHAR2 IN,
   p_owner           VARCHAR2 IN DEFAULT NULL,
   p_description     VARCHAR2 IN DEFAULT NULL,
   p_allowed_origins VARCHAR2 IN DEFAULT NULL,
   p_redirect_uri    VARCHAR2 IN DEFAULT NULL,
   p_support_email   VARCHAR2 IN DEFAULT NULL,
   p_support_uri     VARCHAR2 IN DEFAULT NULL,
   p_privilege_names VARCHAR2 IN)
```

Description

Creates an OAuth client registration.

Parameters

p_name
Name for the client, displayed to the end user during the approval phase of three-legged OAuth. Must be unique.

p_grant_type
Must be one of `authorization_code`, `implicit`, or `client_credentials`.

p_owner
Name of the party that owns the client application.

p_description
Description of the purpose of the client, displayed to the end user during the approval phase of three-legged OAuth. May be null if `p_grant_type` is `client_credentials`; otherwise, must not be null.

p_allowed_origins
A comma-separated list of URL prefixes. If the list is empty, any existing origins are removed.

p_redirect_uri
Client-controlled URI to which redirect containing an OAuth access token or error will be sent. May be null if `p_grant_type` is `client_credentials`; otherwise, must not be null.

p_support_email
The email where end users can contact the client for support.

p_support_uri
The URI where end users can contact the client for support. Example: `http://www.myclientdomain.com/support/`

p_privilege_names
List of comma-separated privileges that the client wants to access.

Usage Notes

To have the operation take effect, use the COMMIT statement after calling this procedure.

Examples

The following example creates an OAuth client registration.

```
BEGIN
  OAUTH.create_client(
    'CLIENT_TEST',
    'authorization_code',
    'test_user',
    'This is a test description.',
    '',
    'https://example.org/my_redirect/#/',
    'test@example.org',
    'https://example.org/help/#/',
     'MyPrivilege'
    );
    COMMIT;
END;
/
```

OAUTH.DELETE_CLIENT

Format

```
OAUTH.DELETE_CLIENT(
   p_name VARCHAR2 IN);
```

Description

Deletes an OAuth client registration.

Parameters

p_name
Name of the client registration to be deleted.

Usage Notes

To have the operation take effect, use the COMMIT statement after calling this procedure.

Examples

The following example deletes an OAuth client registration.

```
BEGIN
  OAUTH.delete_client(
    'CLIENT_TEST'
    );
  COMMIT;
END;
/
```

OAUTH.GRANT_CLIENT_ROLE

Format

```
OAUTH.GRANT_CLIENT_ROLE(
   p_client_name VARCHAR2 IN,
   p_role_name   VARCHAR2 IN);
```

Description

Grant an OAuth client the specified role, enabling clients performing two-legged OAuth to access privileges requiring the role.

Parameters

p_client_name
Name of the OAuth client.

p_role_name
Name of the role to be granted.

Usage Notes

To have the operation take effect, use the COMMIT statement after calling this procedure.

Examples

The following example creates a role and grants that role to an OAuth client.

```
BEGIN
  ORDS.create_role(p_role_name => 'CLIENT_TEST_ROLE');

  OAUTH.grant_client_role(
    'CLIENT_TEST',
    'CLIENT_TEST_ROLE'
    );
  COMMIT;
END;
/
```

OAUTH.RENAME_CLIENT

Format

```
OAUTH.RENAME_CLIENT(
   p_name     VARCHAR2 IN,
   p_new_name VARCHAR2 IN);
```

Description

Renames a client.

Parameters

p_name
Current name for the client.

p_new_name
New name for the client.

Usage Notes

The client name is displayed to the end user during the approval phase of three-legged OAuth.

To have the operation take effect, use the COMMIT statement after calling this procedure.

Examples

The following example renames a client.

```
BEGIN
  OAUTH.rename_client(
    'CLIENT_TEST',
    'CLIENT_TEST_RENAMED'
    );
  COMMIT;
END;
/
```

OAUTH.REVOKE_CLIENT_ROLE

Format

```
OAUTH.REVOKE_CLIENT_ROLE(
   p_client_name  VARCHAR2 IN,
   p_role_name    VARCHAR2 IN);
```

Description

Revokes the specified role from an OAuth client, preventing the client from accessing privileges requiring the role through two-legged OAuth.

Parameters

p_client_name
Name of the OAuth client.

p_role_name
Name of the role to be revoked

Usage Notes

To have the operation take effect, use the COMMIT statement after calling this procedure.

Examples

The following example revokes a specified role from an OAuth client.

```
BEGIN
  OAUTH.revoke_client_role(
    'CLIENT_TEST_RENAMED',
    'CLIENT_TEST_ROLE'
    );
  COMMIT;
END;
/
```

OAUTH.UPDATE_CLIENT

Format

```
OAUTH.UPDATE_CLIENT(
  p_name             VARCHAR2 IN,
  p_description      VARCHAR2 IN,
  p_origins_allowed  VARCHAR2 IN,
  p_redirect_uri     VARCHAR2 IN,
  p_support_email    VARCHAR2 IN,
  p_suppor_uri       VARCHAR2 IN,
  p_privilege_names  t_ords_vchar_tab IN);
```

Description

Updates the client information (except name). Any null values will not alter the existing client property.

Parameters

p_name
Name of the client that requires the owner, description, origins allowed, support e-mail, support URI, and/or privilege modification.

p_description
Description of the purpose of the client, displayed to the end user during the approval phase of three-legged OAuth.

p_redirect_uri
Client-controlled URI to which a redirect containing the OAuth access token/error will be sent. If this parameter is null, the existing `p_redirect_uri` value (if any) is not changed.

p_support_email
The email address where end users can contact the client for support.

p_support_uri
The URI where end users can contact the client for support. Example: `http://www.myclientdomain.com/support/`

p_privilege_names
List of names of the privileges that the client wishes to access.

Usage Notes

To have the operation take effect, use the COMMIT statement after calling this procedure.

If you want to rename the client, use the `OAUTH.RENAME_CLIENT` procedure.

Example to Updates the Description of the Specified Client

The following example updates the description of the client with the name matching the value for `p_name`.

```
BEGIN
  ORDS_METADATA.OAUTH.update_client(
    p_name => 'CLIENT_TEST_RENAMED',
    p_description => 'The description was altered',
    p_origins_allowed => null,
    p_redirect_uri => null,
    p_support_email => null,
    p_support_uri => null,
    p_privilege_names => null);
  COMMIT;
END;
/
```

Example 11-1 Example to Add Multiple Privileges

The following example adds a second privilege:

```
declare
 my_privs t_ords_vchar_tab  := t_ords_vchar_tab ();
begin
 my_privs.EXTEND (3);
 my_privs(1):='tst.privilege1';
 my_privs(2):='tst.privilege2';
.
 oauth.update_client(
    p_name => 'Test_Client',
    p_owner => 'scott',
    p_description => 'Description',
    p_grant_type => 'client_credentials',
    p_redirect_uri => '/abc/efg/',
    p_privilege_names => my_privs);
commit;
end;
```

Related Topics

- OAUTH.RENAME_CLIENT

12

Enabling ORDS Database API

This section describes how to enable the Oracle REST Data Services (ORDS) Database API.

ORDS database API is a database management and monitoring REST API embedded into Oracle REST Data Services. Depending on the database version and configuration, ORDS database API provides services such as manage pluggable databases, export data and review database performance. By default, the ORDS database API feature is disabled when you install ORDS for the first time.

Basic Setup to Enable ORDS Database API

This section explains the basic setup to enable the ORDS database API.

To enable the ORDS database API, set the `database.api.enabled` property to `true` and then restart ORDS:

```
java -jar ords.war set-property database.api.enabled true
```

To access the ORDS database API, you can use one of the following available authentication methods available:

- Database authentication using database username and password
- Through a mid-tier user with the SQL Administrator, or System Administrator role

> **Note:**
>
> There are certain endpoints that are accessible only by certain roles. The REST APIs for Oracle Database documentation provides information on which roles can access each endpoint.

To enable database authentication, you must set the `restEnabledSql.active` property to `true` as shown in the following code snippet and then restart ORDS:

```
java -jar ords.war set-property restEnabledSql.active true
```

For the database authentication, ensure that the administrator schema is ORDS enabled and is granted with the DBA role in an 11gR2 environment or the PDB_DBA role for 12c and higher versions of the database before the schema is used to execute the database API queries in the database. This is done for each non-CDB or pluggable database in which you want to use the database. For more information, refer to "REST-Enabling the Oracle Database Schema" and "ORDS_ADMIN.ENABLE_SCHEMA" sections.

> **Note:**
>
> In the following example, sqlplus command-line utility is used to connect to the SALESPDB database as the system user to configure the PDBADMIN user in that database. The mechanism to connect to the database and performing the steps will differ depending on your environment settings.

For example, to use PDBADMIN schema, in the SALESPDB database for ORDS Database API services, use the following commands in the database.

```
sqlplus system@SALESPDB
GRANT PDB_DBA TO PDBADMIN;
BEGIN
ORDS_ADMIN.ENABLE_SCHEMA(p_schema => 'PDBADMIN');
END;
/
```

The PDBADMIN user is now ready to use the ORDS database API services.

To list the tables in the database, send a `GET` request to `https://<server>/ords/salespdb/pdbadmin/_/db-api/stable/database/objects/tables/`

On request, you must provide the username and password. If you are using a browser, ORDS provides a link to login and authenticate the request. Once you are authenticated, your browser will have an access cookie, and you do not have to specify the user credentials until that cookie expires.

The same service can be accessed through command line utilities such as curl:

```
curl --user pdbadmin:password https://<server>/ords/salespdb/
pdbadmin/_/db-api/stable/database/objects/tables/
```

An OpenAPI V3 document that describes the available ORDS database API services can be accessed at `https://<server>/ords/<my database>/<my admin schema>/_/db-api/stable/metadata-catalog/openapi.json`. With the exception of `https://<server>/ords/<my database>/<my admin schema>/_/db-api/stable/databases/pdbs/`, all other ORDS database API services are made available.

Related Topics

- REST-Enabling the Oracle Database Schema
- ORDS_ADMIN.ENABLE_SCHEMA

Advanced Setup to Enable the ORDS Database API

This section describes the configuration options for using ORDS database API with various database topologies.

> **Note:**
>
> **Disabling management services**: When the value of `database.api.management.services.disabled` property is set to `true`, the following ORDS Database API services are disabled:
>
> - **DBCA Jobs**: DELETE, GET and POST
> - **DBCA Templates**: GET
> - **Oracle Home Environment**: GET
> - **PDB Lifecycle**: DELETE, GET, POST
> - **Open Service Broker**- DELETE, GET and PUT

Pluggable Database Lifecycle Management

This section describes how to enable the Pluggable Database (PDB) lifecycle management operations. Pluggable Database management is performed in the Container Database (CDB) and includes create, clone, plug, unplug and delete operations.

You cannot have an ORDS enabled schema in the container database. To perform the PDB lifecycle management operations, the default CDB administrator credentials, `db.cdb.adminUser` and `db.cdb.adminUser.password` must be defined in the connection pool. In this case, specifying an user schema in the URI is not required.

To define the default CDB administrator credentials, perform the following steps:

1. Create the CDB administrator user and grant the SYSDBA privilege. In this example, the user is called `C##DBAPI_CDB_ADMIN`. However, any suitable common user name can be used.

   ```
   CREATE USER C##DBAPI_CDB_ADMIN IDENTIFIED BY <PASSWORD>;
   GRANT SYSDBA TO C##DBAPI_CDB_ADMIN CONTAINER = ALL;
   ```

2. Set the `db.cdb.adminUser` and `db.cdb.adminUser.password` properties for the connection pool.

   ```
   echo db.cdb.adminUser=C##DBAPI_CDB_ADMIN as SYSDBA > cdbAdmin.properties
   echo db.cdb.adminUser.password=<PASSWORD> >> cdbAdmin.properties
   java -jar ords.war set-properties --conf apex_pu cdbAdmin.properties
   rm cdbAdmin.properties
   ```

The ORDS role, SQL Administrator must be used to access the `https://<server>/ords/_/db-api/stable/database/pdbs/` services.

Disabling PDB Lifecycle Management

This section describes how to disable the PDB lifecycle management services.

You can enable ORDS database API and disable the PDB related services at `https://<server>/ords/_/db-api/stable/databases/pdbs/`.

When the optional CDB administrator credentials are not set, a `HTTP 503 Service Unavailable` response is produced if a user attempts to access `https://<server>/ords/_/db-api/stable/databases/pdbs/`.

To clearly indicate that the PDB operations are disabled for the ORDS installation, set the `database.api.management.services.disabled` property to `true` as shown in the following code snippet and then restart ORDS:

```
java -jar ords.war set-property
database.api.management.services.disabled true
```

This will produce a response, `HTTP 503 Service Unavailable` with an explanatory reason.

Creating a Default Administrator

This section describes how to create and use the default administrator user for the non-CDB or PDB connections.

The ORDS database API service operations are not schema specific. By configuring the default administrator credentials, `db.adminUser` and `db.adminUser.password` in the connection pool, you can execute the corresponding SQL statements as the default administrator user. The ORDS database API endpoints can be executed using a specified ORDS enabled schema if the schema has the DBA role. However, it is not necessary to do so when the default administrator credentials are configured.

> **Note:**
>
> The user credentials must be the same across all the pluggable databases and therefore it is recommended to create the common user in the CDB.

To create the default administrator and grant the DBA role, perform the following steps:

1. Create the default administrator user and grant the DBA role. In this example, the user is called `C##_DBAPI_DEFAULT_ADMIN`. However, any suitable common user name can be used as shown in the following code snippet:

   ```
   CREATE USER C##_DBAPI_DEFAULT_ADMIN IDENTIFIED BY <PASSWORD> CONTAINER = ALL;

   GRANT DBA TO C##_DBAPI_DEFAULT_ADMIN CONTAINER = ALL;
   ```

2. Set the `db.adminUser` and `db.adminUser.password` properties for the connection pool as shown in the following code snipet:

   ```
   echo db.adminUser=C##_DBAPI_PDB_ADMIN > pdbAdmin.properties
       echo db.adminUser.password=<PASSWORD> >> pdbAdmin.properties
       java -jar ords.war set-properties --conf apex_pu pdbAdmin.properties
       rm pdbAdmin.properties
   ```

A schema is not required to be provided in the URI request.

For example, `https://<server>/ords/salespdb/_/db-api/stable/database/datapump/jobs/` lists all the data pump jobs in the `salespdb`, and queries in that database are executed as the `db.adminUser` user.

The ORDS role `SQL Administrator`, is required to use the database API services.

Configuration of Database API Environment Services

This section describes how to configure ORDS Database API environment services.

Starting with ORDS 19.2 release, on a system with ORDS installed, you can perform the set of environment services operations.

For example, the following endpoint lists all the databases discovered in the Oracle Home:

```
https://<server>/ords/_/db-api/stable/environment/databases/
```

You must have the ORDS System Administrator role to use the ORDS database API environment services. The environment services provide information about the database Oracle Home on the host machine and a RESTful interface to the Oracle Database Configuration Assistant to create or delete the databases.

Similar to pluggable database lifecycle management, the environment services can be disabled.

To disable the environment services, set the `database.api.management.services.disabled` property to `true` as follows and then restart ORDS:

```
java -jar ords.war set-property database.api.management.services.disabled
true
```

Configuration of Database API with Open Service Broker API Compatible Platforms

This section describes how to configure and use the ORDS database API with Open Service Broker API compatible platforms.

The ORDS database API provides a service broker for each registered connection pool. Service brokers compliant with the Open Service Broker API specification, allow platforms to provision a new instance of a service. With ORDS as an Open Service Broker to an Oracle database, customers can provision pluggable databases and database users. The nature of the database dictates the service offering that the ORDS database API provides.

Table 12-1 Open Service Broker Service Catalog

Database Type	Service	Plans	Prerequisites
Container Database	**`create-pluggable-database`**. Create a new pluggable database in the Oracle multitenant container database.	**`clone-database`** Create a new pluggable database in the container database by cloning another local pluggable database. Any ORDS REST enabled schemas in the source database is REST enabled in the new database. **`create-database`** Create a new pluggable database from PDB$SEED. The pluggable database administrator account is automatically rest enabled.	Pluggable database lifecycle management must be configured.
Non-Container or Pluggable Database	**`create-oracle-database-user`** Create and configure an Oracle database user with an account through which the user can log in to the database.	**`create-standard-database-user`** Create an Oracle database user with the specified roles and privileges. The objects of the user are stored in the default database tablespace. The temporary segments of the user are stored in the default temporary database tablespace. **`create-ords-enabled-database-user`** Create an Oracle database user with an ORDS enabled schema. The objects of the user are stored in the default database tablespace. The temporary segments of the user are stored in the default temporary database tablespace.	None

To register the service broker URL with your Open Service Broker compliant platform, it depends on how the pool is registered with ORDS and the database type. Oracle

recommends that you use HTTPS with Open Service Broker endpoints. The process of registering a service broker differs depending on the platform.
The Service Broker URL for ORDS follows the following pattern:

- **`create-oracle-database-user`**
  To register the non-CDB or PDB service catalog, you must use the service broker URL for the non-CDB or PDB pool. The format is as follows:

  ```
  https://<server>/ords/<my database>/<my admin schema>/_/db-api/stable/
  openservicebroker/
  ```

  Using the SALESPDB example with PDBADMIN as an ORDS enabled schema, the URL is as follows:

  ```
  https://<server>/ords/salespdb/pdbadmin/_/db-api/stable/
  openservicebroker/
  ```

  > **Note:**
  >
  > `<my database>` can be the default database connection.

  This configuration is common when customers are using ORDS directly with a single database. With this configuration, the example URL is `https://<server>/ords/pdbadmin_/db-api/stable/openservicebroker/`.

- **Supported Open Service Broker Operations**
  ORDS database API supports the synchronous provisioning operation. Other Open Service Broker operations such as deprovisioning and service binding are not supported.

- **Disabling the Service Broker for a Specific Pool**
  To disable the Open Service Broker services available for a specific pool, set the feature. `openservicebroker.exclude` property to `true` by specifying the pool name as follows:

  ```
  java -jar ords.war set-property --conf <pool name>
  feature.openservicebroker.exclude true
  ```
  And then restart ORDS.

  When you use ORDS directly with a container database and pluggable database mapping at runtime, disabling the Open Service Broker for the container disables the broker for all pluggable databases in the container. In such case, the configuration is defined in the container database pool configuration file.

A

Oracle REST Data Services Database Type Mappings

This appendix describes the REST Data Services database type mappings along with the structural database types.

Oracle Built-in Types

Data Type	JSON Data Type	REST Version	Value Example	Description
NUMBER	number	v1	`"big" : 1234567890` `"bigger" : 1.2345678901e10`	Represented with all significant digits. An exponent is used when the number exceeds 10 digits.
RAW	string	Custom	`"code" : "SEVMTE8gV09STEQh"`	Base64 bit encoding is used
DATE	string	v1.2	`"start" : "1995-06-02T04:29:11Z"`	Represented using ISO 8601 format in UTC time zone
TIMESTAMP	string	v1.2	`when : "1995-06-02T04:29:11.002Z"`	Represented using ISO 8601 format in UTC time zone
TIMESTAMP WITH LOCAL TIME ZONE	string	v1.2	"at" : "1995-06-02T04:29:11.002Z"	Represented using ISO 8601 format. The local time zone is converted to UTC time zone as the local time zone specification does not apply for a transfer encoding.
CHAR	string	v1	`"message" : "Hello World! "`	Represented with trailing spaces. This may be required as padding for PUT or POST methods. For example, `"abc "`.
ROWID	string	Custom	`"id" : "AAAGq9AAEAAAA0bAAA"`	Output as the native Oracle textual representation. For example, equivalent to the following conversion: `SELECT ROWIDTOCHAR(id) id FROM DUAL`.
UROWID	string	Custom	`"uid" : "AAAGq9AAEAAAA0bAAA"`	Output as the native Oracle textual representation. For example, equivalent to the following conversion: `SELECT CAST(uid as VARCHAR(4000)) id FROM DUAL`.

Data Type	JSON Data Type	REST Version	Value Example	Description
FLOAT	number	v1	`*as NUMBER`	
NCHAR	string	v1	`"message" : "Hello World!"`	Represented using unicode character where the character is not supported by the body character set.
NVARCHAR2	string	v1	`"message" : "Hello World!"`	Represented using unicode character where the character is not supported by the body character set.
VARCHAR2	string	v1	`"message" : "Hello World!"`	
BINARY_FLOAT	number	v1	`*as NUMBER`	
BINARY_DOUBLE	number	v1	`*as NUMBER`	
TIMESTAMP WITH TIME ZONE	object	v1.2	`"event" : "1995-06-02T 04:29:11.002 Z" "when" : "1995-06-02T 04:29:11.002 Z"`	Represented using ISO 8601 format in UTC time zone. The value represents the same point in time but the original time zone is lost.
INTERVAL YEAR TO MONTH	object	Custom	`"until" : "P-123Y3M" "until" : "P3M"`	Represented using ISO 8601 "Duration" format. Zero duration components are considered optional.
INTERVAL DAY TO SECOND	object	Custom	`"until" : "P-5DT3H55M" "until" : "PT3H55M"`	Represented using ISO 8601 "Duration" format. Zero duration components are considered optional
LONG	string	v1	`*as VARCHAR`	
LONG RAW	string	Custom	`"long_code" : { "SEVMTE8gV09 STEQh"`	

Data Type	JSON Data Type	REST Version	Value Example	Description
BLOB	string	Custom	`"bin" : { "base64_value" : "bGVhc3VyZS4=" }`	
CLOB	string	Custom	`"text" : { "value" : "Hello World! " }`	
BFILE	Object	Custom	`"file" : { "locator" : "TARGET_DIR" , "filename" : "myfile" }`	
BOOLEAN	true|false	v1	`"right" : true "wrong" : false`	

Handling Structural Database Types

This section explains how structural database types are handled.

Object Types

An exception to this is where ORDS has adopted an accepted encoding for an Industry Standard type such as GeoJSON.

Following is a sample code snippet:

```
"address" : {

"number" : 42,
```

```
"street" : "Wallaby Way",

"city" : "Sydney"

}
```

Inheritance

Object type inheritance is not supported. For marshalling purposes, all object types are treated as if they are left concrete types.

PL/SQL Records

PL/SQL Records are not supported.

VARRAYS

VARRAYS are mapped directly to the JSON array type.

Following is a sample code snippet:

```
"addresses" : [

{

"__db_type" : "MY_SCHEMA.AUS_ADDRESS",

"number" : 42,

"street" : "Wallaby Way",

"city" : "Sydney"

},

{

"__db_type" : "MY_SCHEMA.UK_ADDRESS"

"number : 1,

"street" : "Oracle Parkway"

"city" : "Reading"

"postcode" : "RG6 1RA"

}

]
```

Element Inheritance

If the type of a VARRAY element instance is a sub-type of the defined type, then it becomes mandatory to add the `__db_type` named value, as explained in the object types section.

Associative Arrays

Associative arrays (formally known as PL/SQL table or index-by table) fall into following two categories:

- **Indexed by an integer value**: A sparsely populated indexed array. This type of array may not yield a value for a given index. When this type of array is converted to and from JSON, the index is ignored, removing the indexable value gaps. This will have the side-effect that a sparsely populated indexed array that is passed as an IN/OUT parameter through a PL/SQL procedure without any changes, could still appear to have been changed, as the indexable value gaps would have been removed.

  Following is a sample code snippet:

  ```
  "avg_values" : [

  34,

  57,

  86,

  3235

  ]

  :
  ```

- **Not indexed by an integer value**: For example, VARCHAR. This category is rarely used and not supported by the Oracle JDBC API.

Oracle Geospacial Encoding

Oracle Geospacial types comprises of more than the predefined Oracle Object types. However, recognized JSON encoding call, GeoJSON is used to encode the instance data.

Related Topics

- GeoJSON standard documentation

Enabling Database Mapping Support

This section shows how to enable the extended database mapping support.

To enable the extended database mapping support, the following code snippet must be added to the Oracle REST Data Services `defaults.xml` file, which is located in the Oracle REST Data Services configuration `ords` directory:

```
<entry key="misc.datatypes.enable">true</entry>
```

B

About the Oracle REST Data Services Configuration Files

The section describes the Oracle REST Data Services configuration files.

Topics:

- Locating Configuration Files
- Setting the Location of the Configuration Files
- Understanding the Configuration Folder Structure
- Understanding the Configuration File Format
- Understanding Configurable Parameters

Locating Configuration Files

Use the `configdir` command to display the current location of the configuration files:

```
java -jar ords.war configdir
```

If the configuration folder has not yet been configured, the message: `The config.dir setting is not set`, is displayed. If it has been configured, the current value of the setting is displayed.

Setting the Location of the Configuration Files

To change the location of the configuration folder use the `configdir` command:

```
java -jar ords.war configdir </path/to/config>
```

Where:

- `</path/to/config>` is the location where the configuration files are stored.

Understanding the Configuration Folder Structure

The configuration folder has the following structure:

```
./
|
+-defaults.xml
+-apex.properties*
+-url-mapping.xml
|
+conf/
  |
  +-apex.xml
  +-apex_al.xml
```

```
+-apex_rt.xml
+-apex_pu.xml
|
...
+-(db-name).xml
+-(db-name)_al.xml
+-(db-name)_rt.xml
+-(db-name)_pu.xml
```

Global settings that apply to all database connections are stored in `defaults.xml`.

Settings specific to a particular database connection (for example, the default `apex` connection) are stored in `conf/<db-name>.xml`, where *<db-name>* is the name of the database connection.

If the database connection uses Oracle Application Express RESTful Services, the files with names including `_al.xml`, `_rt.xml`, and `_pu.xml` store the configuration for the `APEX_LISTENER`, `APEX_REST_PUBLIC_USER`, and `ORDS_PUBLIC_USER` database users, respectively.

If the database connection uses Oracle REST Data Services RESTful Services, the file *<db-name>*`_pu.xml` stores the configuration for the `ORDS_PUBLIC_USER` database user.

Understanding the Configuration File Format

Configuration files use the standard Java XML properties file format, where each configuration setting contains a key and a corresponding value. The following is an example of a `defaults.xml` file:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>

<entry key="db.connectionType">basic</entry>
<entry key="db.hostname">localhost</entry>
<entry key="db.port">1521</entry>
<entry key="db.sid">orcl</entry>

<entry key="jdbc.DriverType">thin</entry>
<entry key="jdbc.InitialLimit">3</entry>
<entry key="jdbc.MinLimit">1</entry>
<entry key="jdbc.MaxLimit">10</entry>
<entry key="jdbc.MaxStatementsLimit">10</entry>
<entry key="jdbc.InactivityTimeout">1800</entry>
<entry key="jdbc.statementTimeout">900</entry>
<entry key="jdbc.MaxConnectionReuseCount">1000</entry>

</properties>
```

Understanding the url-mapping.xml File Format

The `url-mapping.xml` file stores the rules that route requests to the appropriate database when more than one database is configured. The following is an example of a `url-mapping.xml` file:

```
<pool-config xmlns="http://xmlns.oracle.com/apex/pool-config">
 <pool name="sales_db"
```

```
   base-path="/sales"
   workspace-id="sales_rest"/>
</pool-config>
```

Understanding Configurable Parameters

Table B-1 lists editable parameters for the `defaults.xml` and `(db-name).xml` configuration files.

> **Note:**
>
> Oracle recommends users to use the Oracle REST Data Services command-line interface and Oracle SQL Developer Oracle REST Data Services Administration to edit the configuration files.

Table B-1 Oracle REST Data Services Configuration Files Parameters

Key	Type	Description	Example	Setting Type
`apex.docTable`	string	This parameter is deprecated, instead use `owa.docTable` parameter.	`MYDOCTABLE`	Pool specific
`database.api.enabled`	string	Specifies whether the Database API is enabled. Supported values: • true • false (default)		
`db.connectionType`	string	The type of connection. Supported values: • basic • tns • customurl	`basic`	Pool specific
`db.customURL`	string	Specifies the JDBC URL connection to connect to the database.	`jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=myhost)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=ora111.example.com)))`	Pool specific
`db.hostname`	string	Specifies the host system for the Oracle database.	`myhostname`	Pool specific
`db.password`	string	Specifies the password of the specified database user. Include an exclamation at the beginning of the password so that it can be stored encrypted.	`!password4user`	Pool specific

Table B-1 (Cont.) Oracle REST Data Services Configuration Files Parameters

Key	Type	Description	Example	Setting Type
db.port	numeric	Specifies the database listener port.	1521	Pool specific
db.servicename	string	Specifies the network service name of the database.	ora111.example.com	Pool specific
db.serviceNameSuffix	string	Specifies that the pool points to a CDB, and that the PDBs connected to that CDB should be made addressable by Oracle REST Data Services (see Making All PDBs Addressable by Oracle REST Data Services (Pluggable Mapping)).	apex_pu	Pool specific
db.sid	string	Specifies the name of the database.	ora111	Pool specific
db.tnsAliasName	string	Specifies the TNS alias name that matches the name in the tnsnames.ora file.	MY_TNSALIAS	Pool specific
db.tnsDirectory	string	The directory location of your tnsnames.ora file.	C:\ORACLE\NETWORK\ADMIN	Pool specific
db.username	string	Specifies the name of the database user for the connection.	APEX_PUBLIC_USER	Pool specific
debug.printDebugToScreen	boolean	Specifies whether to display error messages on the browser. Supported values: • true • false (default)	false	Global
error.keepErrorMessages	boolean	Specifies whether to retain the error messages. Supported values: • true • false (default)	true	Global
error.maxEntries	numeric	Specifies the total number of error messages to retain. Defaults to 50.	50	Global
error.externalPath	string	Specifies the path to a folder that contains the custom error page.	/path/to/error/pages/folder/	Global
icap.port	numeric	Specifies the Internet Content Adaptation Protocol (ICAP) Port to virus scan files. The icap.port is required to have a value.	5555	Global
icap.server	string	Specifies the Internet Content Adaptation Protocol (ICAP) Server name to virus scan files. The icap.server is required to have a value.	servername	Global

Table B-1 (Cont.) Oracle REST Data Services Configuration Files Parameters

Key	Type	Description	Example	Setting Type
`jdbc.DriverType`	string	Specifies the JDBC driver type. Supported values: • thin • oci8	`thin`	Pool specific
`jdbc.InactivityTimeout`	numeric	Specifies how long an available connection can remain idle before it is closed. The inactivity connection timeout is in seconds. Defaults to 1800.	`1800`	Pool specific
`jdbc.InitialLimit`	numeric	Specifies the initial size for the number of connections that will be created. Defaults to 3. (The default is low, and should probably be set higher in most production environments.)	`10`	Pool specific
`jdbc.MaxConnectionReuseCount`	numeric	Specifies the maximum number of times to reuse a connection before it is discarded and replaced with a new connection. Defaults to 1000.	`1000`	Pool specific
`jdbc.MaxLimit`	numeric	Specifies the maximum number of connections. Defaults to 10. (Might be too low for some production environments.)	`20`	Pool specific
`jdbc.auth.enabled`	boolean	Specifies if the PL/SQL Gateway calls can be authenticated using database users. If the value is `true` then this feature is enabled. If the value is `false`, then this feature is disabled. The default value is `false`. Oracle recommends not to use this feature. This feature used only to facilitate customers migrating from mod_plsql.	`false`	Pool specific
`jdbc.MaxStatementsLimit`	numeric	Specifies the maximum number of statements to cache for each connection. Defaults to 10.	`10`	Pool specific
`jdbc.MinLimit`	numeric	Specifies the minimum number of connections. Defaults to 1.	`1`	Pool specific

Table B-1 (Cont.) Oracle REST Data Services Configuration Files Parameters

Key	Type	Description	Example	Setting Type
`jdbc.statementTimeout`	numeric	Specifies how long a borrowed (in use) connection can remain unused before it is considered as abandoned and reclaimed. The abandoned connection timeout is in seconds. Defaults to 900.	`900`	Pool specific
`log.logging`	boolean	Specifies whether to retain the log messages. Supported values: • true • false (default)	`true`	Global
`log.maxEntries`	numeric	Specifies the total number of log messages to retain. Defaults to 50.	`50`	Global
`log.procedure`	boolean	Specifies whether procedures are to be logged. Supported values: • true • false (default)	`false`	Global
`misc.defaultPage`	string	Specifies the default page to display. The Oracle REST Data Services home page, **apex**, is commonly used.	`apex`	Pool specific
`misc.pagination.maxRows`	numeric	Specifies the maximum number of rows that will be returned from a query when processing a RESTful service and that will be returned from a nested cursor in a result set. Affects all RESTful services generated through a SQL query, regardless of whether the resource is paginated. Defaults to 10000.	`300`	Pool specific

Table B-1 (Cont.) Oracle REST Data Services Configuration Files Parameters

Key	Type	Description	Example	Setting Type
`owa.docTable`	string	Specifies the name of the document table used by the file upload. Defaults to `FLOWS_FILES.WWV_FLOW_FILE_OBJECTS$` value. **Note:** For APEX 4.x and above this parameter should not be used.	`MYDOCTABLE`	Pool specific
`procedure.postProcess`	string	Specifies the procedure name(s) to execute after executing the procedure specified on the URL. Multiple procedure names must be separated by commas.	`SCHEMA1.SUBMIT.REQUEST,FINISHTASK`	Pool specific
`procedure.preProcess`	string	Specifies the procedure name(s) to execute prior to executing the procedure specified on the URL. Multiple procedure names must be separated by commas.	`SCOTT.PREPROC1, INITIALIZE, PKG1.PROC`	Pool specific

Table B-1 (Cont.) Oracle REST Data Services Configuration Files Parameters

Key	Type	Description	Example	Setting Type
`procedure.rest.preHook`	string	Specifies the function to be invoked prior to dispatching each Oracle REST Data Services based REST Service. The function can perform configuration of the database session, perform additional validation or authorization of the request. If the function returns `true`, then processing of the request continues. If the function returns `false`, then processing of the request is aborted and an HTTP 403 Forbidden status is returned.	`MYAPP.VALIDATE_REST_CALL`	Pool specific
`security.disableDefaultExclusionList`	boolean	If this value is set to `true`, then the Oracle REST Data Services internal exclusion list is not enforced. **Note**: The Oracle REST Data Services internal exclusion list blocks the users from accessing the following: • sys.* • dbms_* • utl_* • owa_* • owa.* • htp.* • htf.* • wpg_docload.* Supported values: • `true` • `false (default)` Oracle recommends that you do not set this value to `true`. That is, do not disable the default internal exclusion list. The only possible exception is temporarily disabling the internal exclusion list for debugging purposes.	`false`	Global

Table B-1 (Cont.) Oracle REST Data Services Configuration Files Parameters

Key	Type	Description	Example	Setting Type
`security.exclusionList`	string	Specifies a pattern for procedures, packages, or schema names which are forbidden to be directly executed from a browser. Procedure names can contain the wildcard characters asterisk (*) or question mark (?). Use an asterisk (*) to substitute zero or more characters and a question mark (?) to substitute for any one character. **Note**: Separate multiple patterns using commas.	`customer_account,bank*, employe?`	Global
`security.inclusionList`	string	Specifies a pattern for procedures, packages, or schema names which are allowed to be directly executed from a browser. Procedure names can contain the wildcard characters asterisk (*) or question mark (?). Use an asterisk (*) to substitute zero or more characters and a question mark (?) to substitute for any one character. **Note**: Separate multiple patterns using commas.	`apex, p, v, f, wwv_*, y*, c*`	Global
`security.maxEntries`	numeric	Specifies the maximum number of cached procedure validations. Defaults to 2000. Set this value to 0 to force the validation procedure to be invoked on each request.	`2000`	Global
`security.requestAuthenticationFunction`	string	Specifies an authentication function to determine if the requested procedure in the URL should be allowed or disallowed for processing. The function should return `true` if the procedure is allowed; otherwise, it should return false. If it returns false, Oracle REST Data Services will return `WWW-Authenticate` in the response header.	`owa_custom.authorize`	Pool specific

Table B-1 (Cont.) Oracle REST Data Services Configuration Files Parameters

Key	Type	Description	Example	Setting Type
`security.requestValidationFunction`	string	Specifies a validation function to determine if the requested procedure in the URL should be allowed or disallowed for processing. The function should return true if the procedure is allowed; otherwise, return false.	`CHECK_VALID_PROCEDURE`	Pool specific
`security.verifySSL`	boolean	Specifies whether HTTPS is available in your environment. Supported values: • true (default) • false If you change the value to false, see Using OAuth2 in Non-HTTPS Environments.	`true`	Global
`soda.defaultLimit`	string	When using the SODA REST API, specifies the default number of documents returned for a GET request on a collection when a limit is not specified in the URL. Must be a positive integer, or "unlimited" for no limit. Defaults to 100.	`75`	Pool specific
`soda.maxLimit`	string	When using the SODA REST API, specifies the maximum number of documents that will be returned for a GET request on a collection URL, regardless of any limit specified in the URL. Must be a positive integer, or "unlimited" for no limit. Defaults to 1000.	`700`	Pool specific
`restEnabledSql.active`	boolean	Specifies whether the REST-Enabled SQL service is active. Supported values: • true • false (default)	`true`	Pool specific

See Also:

For more information, see Configuring and Installing Oracle REST Data Services and "Oracle REST Data Services Administration" in *Oracle SQL Developer User's Guide*.

C

Troubleshooting Oracle REST Data Services

This appendix contains information on troubleshooting Oracle REST Data Services.

Topics:

- Enabling Detailed Request Error Messages
- Configuring Application Express Static Resources with Oracle REST Data Services

Enabling Detailed Request Error Messages

To enable detailed request error messages, add the following setting to the Oracle REST Data Services configuration file named: `defaults.xml`:

```
<entry key="debug.printDebugToScreen">true</entry>
```

When this setting is present in `defaults.xml`, any request that produces an error response includes a detailed message, including a stack trace. This setting must not be enabled on productions systems due to the risk of sensitive information being revealed to an attacker.

ORDS User Defined Service

The following table lists the ORDS user defined services:

Table C-1 List of ORDS user defined service

Service	Response
--HTTP 200 BEGIN ORDS.define_service(p_module_name => 'test1', p_base_path => 'test1/', p_pattern => 'ok/', p_method => 'GET', p_source_type => ORDS.source_type_collection_feed, p_source => 'SELECT * FROM dual', p_items_per_page => 0);	>curl -i -X GET --user DEMO:demo -k http://localhost:8082/ords/demo/test1/ok/ HTTP/1.1 200 OK Date: Thu, 19 Mar 2020 17:18:05 GMT Content-Type: application/json ETag: "BLNTmyd/ZM889Q0GlgJ1t7lkSYo2kpAVIv4CY5dvtp9NI/Em1DJRzpmE5Bg/4GiKifewtzuJA6i+YCgdxETWWQ==" Transfer-Encoding: chunked

Table C-1 (Cont.) List of ORDS user defined service

Service	Response
COMMIT; END; /	

Table C-1 (Cont.) List of ORDS user defined service

Service	Response
--HTTP 200 , p_source _type => ORDS.sou rce_type _collect ion_feed , BEGIN ORDS.def ine_serv ice(p_module _name => 'test2', p_base_p ath => 'test2/' , p_patter n => 'norows/ ', p_method => 'GET', p_source _type => ORDS.sou rce_type _collect ion_feed , p_source => 'SELECT	>curl --head -i -X GET -- user DEMO:dem o -k http:// localhos t:8082/ ords/ demo/ test2/ norows/ HTTP/1.1 200 OK Date: Thu, 19 Mar 2020 17:18:28 GMT Content- Type: applicat ion/json ETag: "aZVsHTw ewrbbkl6 wHNcTa3R FFdEsbdt DRBTSlR9 3r/ vBmDvVsg ud2rFqLD I65UKxzS ElnAAMQd lBj/ sB9ywWqQ ==" Transfer - Encoding : chunked

Table C-1 (Cont.) List of ORDS user defined service

Service	Response
`* FROM dual where 1 = 2', p_items_per_page => 0);  COMMIT; END; /`	

Table C-1 (Cont.) List of ORDS user defined service

Service	Response
create table no_rows (col1 int); --HTTP 200 , p_source_type => ORDS.source_type_collection_feed , BEGIN ORDS.define_service(p_module_name => 'test2b' , p_base_path => 'test2b/' , p_pattern => 'norows/' , p_method => 'GET', p_source_type => ORDS.source_type_collect	>curl --head -i -X GET --user DEMO:demo -k http://localhost:8082/ords/demo/test2b/norows/ HTTP/1.1 200 OK Date: Thu, 19 Mar 2020 17:18:34 GMT Content-Type: application/json ETag: "Ns/ghFxVWYPHUyZT53HNl6EMVlQUXD5wmz3eoO15dlY6nSVkk2FX3sNw3Yvq87SdLYAlCLeuqb4N4DQrcy+0Q==" Transfer-Encoding: chunked

Table C-1 (Cont.) List of ORDS user defined service

Service	Response
ion_feed , p_source => 'SELECT * FROM no_rows' , p_items_ per_page => 0); COMMIT; END; /	

Table C-1 (Cont.) List of ORDS user defined service

Service	Response
--HTTP 404 , p_source _type => ORDS.sou rce_type _collect ion_item , BEGIN ORDS.def ine_serv ice(p_module _name => 'test2c' , p_base_p ath => 'test2c/ ', p_patter n => 'norows/ ', p_method => 'GET', p_source _type => ORDS.sou rce_type _collect ion_item , p_source => 'SELECT	>curl --head -i -X GET -- user DEMO:dem o -k http:// localhos t:8082/ ords/ demo/ test2c/ norows/ HTTP/1.1 404 Not Found Content- Type: text/ html Content- Length: 16127

Table C-1 (Cont.) List of ORDS user defined service

Service	Response
`* FROM dual where 1 = 2', p_items_per_page => 0); COMMIT; END; /`	

Table C-1 (Cont.) List of ORDS user defined service

Service	Response
`--HTTP 404 BEGIN ORDS.define_service( p_module_name => 'test3', p_base_path => 'test3/', p_pattern => 'doesnotexist/', p_method => 'GET', p_source_type => ORDS.source_type_collection_feed, p_source => 'SELECT 10 as A FROM doesnotexist', p_items_per_page`	`>curl --head -i -X GET --user DEMO:demo -k http://localhost:8082/ords/demo/test3/doesnotexist/ HTTP/1.1 403 Forbidden Content-Type: text/html Error-Reason: error="missing.object"; error_description*=UTF-8' 'The request could not be processed because a table or view referenced 0by the SQL statement being evaluate`

Table C-1 (Cont.) List of ORDS user defined service

Service	Response
=> 0); COMMIT; END; /	d is not accessible or does not exist Content-Length: 16327

Table C-1 (Cont.) List of ORDS user defined service

Service	Response
--HTTP 555 BEGIN ORDS.define_service(p_module_name => 'test4', p_base_path => 'test4/', p_pattern => 'badsyntax/', p_method => 'GET', p_source_type => ORDS.source_type_collection_feed, p_source => 'SELECT 10', p_items_per_page => 0); COMMIT;	>curl --head -i -X GET --user DEMO:demo -k http://localhost:8082/ords/demo/test4/badsyntax/ HTTP/1.1 500 Server Error Content-Type: text/html Error-Reason: error="resource.generator.evaluation"; error_description*=UTF-8' 'The request could not be processed because an error occurred whilst attempting to

Table C-1 (Cont.) List of ORDS user defined service

Service	Response
END; /	evaluate the SQL statement associated with this resource. Please check the SQL statement is correctly formed and executes without error. SQL Error Code ORA-00923 FROM keyword not found where expected Error Message. Content-Length: 16514

Configuring Application Express Static Resources with Oracle REST Data Services

When using Oracle REST Data Services, a blank page might be displayed when attempting to access an Oracle Application Express page, for example, when attempting to display `https://example/ords/`. This problem is caused by an improper

configuration of Application Express static resources, which causes the JavaScript and CSS resources required by Application Express not to be found and the Application Express page not to render correctly.

The specific cause can be any of the following:

- Forgetting to ensure that the Application Express static images are located on the same server as the Oracle REST Data Services instance
- Forgetting to deploy `i.war` on WebLogic Server
- Specifying an incorrect path when using the `java -jar ords.war static` command to generate `i.war`
- Configuring Application Express to use a nondefault context path for static resources (`/i`) and not specifying the same context path (using the `--context-path` option) when using `java -jar ords.war static`
- Moving, renaming, or deleting the folder pointed to by `i.war` after deploying `i.war`
- When running in Standalone mode, entering an incorrect path (or not specifying a path) when prompted on the first run of Standalone mode
- When running in Standalone mode, entering an incorrect path with the `--static-images` option
- Upgrading to a new version of Application Express and forgetting to reconfigure and redeploy `i.war` to point to the static resources for the new Application Express version, or in Standalone mode forgetting to update the location by using the `--apex-images` option

To help in diagnosing the problem, you can try to access the `apex_version.txt` file. For example, if your Application Express deployment is located at `https://example.com/ords/` and your static resources have been deployed at `https://example.com/i/`, use a browser to access the following URL:

```
https://example.com/i/apex_version.txt
```

If you get a `404 Not Found` error, then check the preceding list of possible specific causes, including `i.war` not being deployed or not pointing to a folder containing Application Express static resources.

If a plain text file is displayed, it should contain text like the following:

```
Application Express Version:  4.2.1
```

Check that the version number matches the version of Application Express that is deployed on the database. If the numbers do not match, check if you have made an error mentioned in the last item in the preceding list of possible specific causes, because Oracle REST Data Services is not configured to use the correct version of the Application Express static resources to match the Application Express version in the database.

If you need help in solving the problem, check the information in this book about creating and deploying `i.war` for your environment, such as WebLogic Server.

You can also get detailed help on the static listener command by entering the following at a command prompt:

```
java -jar ords.war help static
```

D

Creating an Image Gallery

This tutorial explains an extended example that builds an image gallery service for storing and retrieving images. This tutorial uses Oracle Application Express.

Topics:

- Before You Begin
- Creating the Gallery Database Table
- Creating the Gallery RESTful Service Module
- Trying Out the Gallery RESTful Service
- Creating the Gallery Application
- Trying Out the Gallery Application
- Securing the Gallery RESTful Services
- Accessing the RESTful Services from a Third Party Application

> **See Also:**
>
> To do this tutorial, you must be familiar with the concepts and techniques covered in Developing Oracle REST Data Services Applications.

Before You Begin

This section describes some common conventions used in this example as well as best practices regarding API entry points.

Topics:

- About URIs
- About Browser Support
- Creating an Application Express Workspace

About URIs

Throughout this example, URIs and URI Templates are referenced using an abbreviated form that omits the host name, context root and workspace path prefix. Consider the following example:

```
gallery/images/
```

To access this URI in your Web browser, you would use a URI in the following format:

```
https://<host>:<port>/ords/<workspace>/gallery/images/
```

where

- <host> is the host on which Oracle REST Data Services is running.
- <port> is the port on which Oracle REST Data Services is listening.
- /ords is the context root where Oracle REST Data Services is deployed.
- /<workspace>/ is the workspace path prefix of the Oracle Application Express workspace where the RESTful Service is defined.

About Browser Support

This example uses many modern features defined in HTML5 and related specifications. It has only been tested in Mozilla Firefox and Google Chrome. It has not been tested in Microsoft Internet Explorer or on smart-phone/tablet web browsers. Please use recent versions of either Mozilla Firefox or Google Chrome for this example.

Creating an Application Express Workspace

To follow the instructions for creation the Gallery example application and related objects, first, create a new Oracle Application Express Workspace (in Full Development mode). See the Oracle Application Express Documentation for details on how to do this.

Call the workspace resteasy and call the administrator user of the workspace resteasy_admin. Ensure the resteasy_admin user is a a member of the RESTful Services user group.

Creating the Gallery Database Table

To create the Gallery database table, follow these steps:

1. Log into the resteasy workspace.
2. Navigate to **SQL Workshop** and then **SQL Commands**.
3. Enter or copy and paste in the following SQL:

```
CREATE SEQUENCE GALLERY_SEQ
/
CREATE TABLE GALLERY (
  ID NUMBER NOT NULL ENABLE,
  TITLE VARCHAR2(1000) NOT NULL ENABLE,
  CONTENT_TYPE VARCHAR2(1000) NOT NULL ENABLE,
  IMAGE BLOB NOT NULL ENABLE,
  CONSTRAINT GALLERY_PK PRIMARY KEY (ID) ENABLE
)
/
CREATE OR REPLACE TRIGGER BI_GALLERY
 before insert on GALLERY for each row
 begin
  if :NEW.ID is null then
   select GALLERY_SEQ.nextval into :NEW.ID from sys.dual;
  end if;
 end;
 /
```

```
ALTER TRIGGER BI_GALLERY ENABLE
/
```

Creating the Gallery RESTful Service Module

To create the Gallery RESTful services module, follow these steps:

1. Navigate to **SQL Workshop** and then **RESTful Services**.
2. Click **Create** on the right side, and enter the following information:
   - **Name**: `gallery.example`
   - **URI Prefix**: `gallery/`
   - **URI Template**: `images/`
   - **Method**: `POST`
   - **Source**: Enter or copy and paste in the following:

     ```
     declare
      image_id integer;
     begin
      insert into gallery (title,content_type,image)
                  values  (:title,:content_type,:body)
                  returning id into image_id;
      :status := 201;
      :location := image_id;
     end;
     ```
3. Click **Create Module**.
4. Click the `POST` handler under `images/`
5. For **Requires Secure Access**, select `No`.
6. Click **Create Parameter**, and enter the following:
   - **Name**: `Slug`
   - **Bind Variable Name**: `title`
7. Click **Create**.
8. Click **Create Parameter** on the bottom right, and enter the following information:
   - **Name**: `X-APEX-FORWARD`
   - **Bind Variable Name**: `location`
   - **Access Method**: `OUT`
9. Click **Create**.
10. Click **Create Parameter** on the bottom right, and enter the following information:
    - **Name**: `X-APEX-STATUS-CODE`
    - **Bind Variable Name**: `status`
    - **Access Method**: `OUT`
    - **Parameter Type**: `Integer`
11. Click **Create**.

At this point you have created the module with a single service that can store new images. Next, add a service to display the list of stored images:

1. Navigate to **SQL Workshop** and then **RESTful Services**.
2. Click the module named `gallery.example`.
3. Click **Create Handler** under `images/`, and enter the following information:
   - **Method**: `GET`
   - **Source Type**: `Feed`
   - **Requires Secure Access**: `No`
   - **Source**: Enter or copy and paste in the following:

     ```
     select id,title,content_type from gallery order by id desc
     ```
4. Click **Create**.

At this point you have created the service to store and list images. Next, add a service to display individual images:

1. Navigate to **SQL Workshop** and then **RESTful Services**.
2. Click the module named `gallery.example`.
3. Click **Create Template** under gallery.example, and enter the following information:
   - **URI Template**: `images/{id}`
4. Click **Create**.
5. Click **Create Handler** under images/{id}, and enter the following information:
   - **Method**: `GET`
   - **Source Type**: `Media Resource`
   - **Requires Secure Access**: `No`
   - **Source**: Enter or copy and paste in the following:

     ```
     select content_type, image from gallery where id = :id
     ```
6. Click **Create**.

Trying Out the Gallery RESTful Service

To try out the Gallery RESTful Service, follow these steps:

1. Navigate to **SQL Workshop** and then **RESTful Services**.
2. Click the module named `gallery.example`.
3. Click the `GET` handler located under `images/`.
4. Click **Test**.

   The following URI should be displayed in the browser:

   ```
   https://<host>:<port>/ords/resteasy/gallery/images/
   ```

   Content similar to the following should be displayed:

   ```
   {"next":
    {"$ref":
   ```

```
  "http://localhost:8080/ords/resteasy/gallery/images/?page=1"
 },
 "items":[]
}
```

- The content is a JSON document that lists the location of each image in the gallery, but since you have not yet added any images, the list (the `items[]` element) is empty.
- The JSON has no extra white space to minimize its size, this can make it difficult to decipher, it is recommended to add a JSON viewing plugin to your browser to make viewing the JSON easier.

To create an Oracle Application Express application to enable users to add and view images in the gallery, see Creating the Gallery Application.

Creating the Gallery Application

To create an Oracle Application Express application that uses the gallery RESTful Services, follow these steps:

1. Navigate to **Application Builder**.
2. Click **Create**.
3. Choose `Database`, then click **Next**.
4. Enter `Image Gallery` in the **Name** field, then click **Next**.
5. Click **Create Application**, and then **Create Application** again to confirm creation of the application.
6. Click `page 1, Home`.
7. Under **Regions** click the + (plus sign) icon to create a new region.
8. For **Region Type**, choose `HTML` and click **Next**, then click **Next** on the next page.
9. For **Region Template**, choose `No Template`.
10. For **Title**, enter `Tasks`, and click **Next**.
11. For **Enter HTML Text Region Source**, specify:

    ```
    <a class="button" id="upload-btn">Upload Image</a>
    ```

12. Click **Create Region**.
13. Under **Regions**, click the + (plus sign) icon to create a new region.
14. For **Region Type**, choose `HTML`, and click Next, then Next again.
15. For **Region Template**, choose `DIV Region with ID`.
16. For **Title**, enter `Images`.
17. Click **Create Region**.
18. Click the `Images` region and click the **Attributes** tab.
19. For **Static ID**, enter `images`, and click **Apply Changes**.
20. Under **Page**, click the Edit icon, then click the **JavaScript** tab.
21. For **Function and Global Variable Declaration**, enter or copy and paste in the following:

```
var workspace_path_prefix = 'resteasy';
 var gallery_url = './' + workspace_path_prefix + '/gallery/images/';
 function uploadFiles(url, fileOrBlob, onload) {
  var name = 'unspecified';
  if ( fileOrBlob['name'] ) {
   name = fileOrBlob.name;
  }
  var xhr = new XMLHttpRequest();
  xhr.open('POST', url, true);
  xhr.setRequestHeader('Slug',name);
  xhr.onload = onload;
  xhr.send(fileOrBlob);
 }

 function createUploader() {
  var $upload = $('<div id="uploader" title="Image Upload"\
   style="display:none">\
   <form>\
    <fieldset>\
     <label for="file">File</label>\
     <input type="file" name="file" id="file"\
      class="text ui-widget-content ui-corner-all"/>\
    </fieldset>\
   </form>\
  </div>');
  $(document.body).append($upload);
  $upload.dialog({
   autoOpen:false,
   modal: true,
   buttons: {
    "Upload": function() {
     var file = document.querySelector('input[type="file"]');
     uploadFiles(gallery_url,file.files[0],function() {
      $('#uploader').dialog("close");
      getImages();
     });
    },
    "Cancel": function() {
     $('#uploader').dialog("close");
    }
   }
  });
  $('#upload-btn').click(function() {
   $('#uploader').dialog("open");
  });
 }

 function getImages() {
  var xhr = new XMLHttpRequest();
  xhr.open('GET', gallery_url);
  xhr.onload = function(e) {
   var data = JSON.parse(this.response);
   $('#image-list').remove();
   var $images = $('<ol id="image-list"></ol>');
   for ( i in data.items ) {
    var item = data.items[i];
    var uri = item.uri['$ref'];
    var $image = $('<li></li>')
                 .append('<a href="' + uri + '" + title="' +
                         item.title + '"><img src="graphics/'+ uri +
                         '"></a>');
```

```
   $images.append($image);
  }
  $('#images').append($images);
 }
 xhr.send();
}
```

22. For Execute when Page Loads, enter or copy and paste in the following:

```
createUploader();
getImages();
```

23. Click **Apply Changes**.
24. Under **Page**, click the Edit icon, then click the **CSS** tab.
25. For **Inline**, enter or copy and paste in the following:

```
a img { border:none; }
#images ol { margin: 1em auto; width: 100%; }
#images li { display: inline; }
#images a { background: #fff; display: inline; float: left;
            margin: 0 0 27px 30px; width: auto; padding: 10px 10px 15px;
            textalign: center; text-decoration: none; color: #333;
            font-size: 18px; -webkit-box-shadow: 0 3px 6px rgba(0,0,0,.25);
            -moz-boxshadow: 0 3px 6px rgba(0,0,0,.25); }
#images img { display: block; width: 190px; margin-bottom: 12px; }
label {font-weight: bold; text-align: right;float: left;
       width: 120px; margin-right: 0.625em; }
label :after {content(":")}
input, textarea { width: 250px; margin-bottom: 5px;textalign: left}
textarea {height: 150px;}
br { clear: left; }
#images a:after { content: attr(title); }
.button {
  border-top: 1px solid #96d1f8;
  background: #65a9d7;
  background:
   -webkit-gradient(linear,left top,left bottom,
                    from(#3e779d),to(#65a9d7));
  background:
   -webkit-linear-gradient(top, #3e779d, #65a9d7);
  background:
   -moz-linear-gradient(top, #3e779d, #65a9d7);
  background: -ms-linear-gradient(top, #3e779d, #65a9d7);
  background: -o-linear-gradient(top, #3e779d, #65a9d7);
  padding: 5px 10px;
  -webkit-border-radius: 8px;
  -moz-border-radius: 8px;
  border-radius: 8px;
  -webkit-box-shadow: rgba(0,0,0,1) 0 1px 0;
  -moz-box-shadow: rgba(0,0,0,1) 0 1px 0;
  box-shadow: rgba(0,0,0,1) 0 1px 0;
  text-shadow: rgba(0,0,0,.4) 0 1px 0;
  color: white;
  font-size: 14px;
  text-decoration: none;
  vertical-align: middle;
  }

 .button:hover {
  border-top-color: #28597a;
```

```
 background: #28597a;
 color: #ccc;
 cursor: pointer;
}

.button:active {
 border-top-color: #1b435e;
 background: #1b435e;
}
```

26. Click **Apply Changes**.

Trying Out the Gallery Application

To try out the Gallery application, follow these steps:

1. Navigate to Application Builder.
2. Click Run beside the Image Gallery application.
3. Log in as the `resteasy_admin` user.
4. Click Upload Image.
5. Choose an image file (a `JPEG` or `PNG` file) and click Upload.

   The application displays the uploaded image.

Securing the Gallery RESTful Services

It is not wise to allow public access to the image uploading service, and it is probably not ideal to allow public access to the images in the gallery either. Therefore, you should protect access to the RESTful services.

RESTful Services support two kinds of authentication:

- First Party Authentication. This is authentication intended to be used by the party who created the RESTful service, enabling an Application Express application to easily consume a protected RESTful service. The application must be located with the RESTful service, that is, it must be located in the same Oracle Application Express workspace. The application must use the standard Oracle Application Express authentication.
- Third Party Authentication. This is authentication intended to be used by third party applications not related to the party who created the RESTful service. Third party authentication relies on the OAuth 2.0 protocol.

Topics:

- Protecting the RESTful Services
- Modifying the Application to Use First Party Authentication

Protecting the RESTful Services

To protect the RESTful services, follow these steps:

1. Navigate to **SQL Workshop** and then **RESTful Services**.

2. Click **RESTful Service Privileges** in the section labeled **Tasks**.
3. Click **Create**, and enter the following:
   - **Name**: `example.gallery`
   - **Label**: `Gallery Access`
   - **Assigned Groups**: `RESTful Services`
   - **Description**: `View and Post images in the Gallery`
   - **Protected Modules**: `gallery.example`
4. Click **Create**.

To check that access to the RESTful Service is now restricted, follow these steps:

1. Navigate to **SQL Workshop** and then **RESTful Services**.
2. Click the module named `gallery.example`.
3. Click the `GET` handler located under `images/`.
4. Click **Test**.

   The URI in the following format should be displayed in the browser:

   ```
   https://<host>:<port>/ords/resteasy/gallery/images
   ```

   An error page should be displayed with the error message:

   ```
   401 Unauthorized.
   ```

> **See Also:**
>
> This is the expected result, because a protected RESTful Service cannot be accessed unless proper credentials are provided. To add the required credentials to the request, see Modifying the Application to Use First Party Authentication

Modifying the Application to Use First Party Authentication

First Party Authentication relies on the cookie and user session established by the Application Express application, but Oracle REST Data Services needs additional information to enable it to verify the cookie. It needs to know the application ID and the current session ID. This information is always known to the Application Express application, and must be included with the request made to the RESTful service by adding the custom `Apex-Session` HTTP header to each request sent to a RESTful Service. The application ID and session ID are sent as the value of the header, separated from each other by a comma delimiter. For example:

```
GET /ords/resteasy/gallery/images/
Host: server.example.com
Apex-Session: 102,6028968452563
```

Sometimes it is not possible to include a custom header in the HTTP request. For example, when displaying an image in an HTML page using the `<img>` tag, an alternative mechanism is used for these scenarios. The application ID and session ID are included in a query parameter named `_apex_session`, which is added to

the Request URI, which contains the application ID and session ID separated by a comma. For example:

```
<img src="graphics/101?_apex_session=102,6028968452563">
```

Note that this approach must only be used when it is not possible to use a custom header. Otherwise, this approach is discouraged because of the increased risk of the session ID being inadvertently stored or disclosed due to its inclusion in the URI.

To modify the application to add the first party authentication information to each request, follow these steps:

1. Navigate to **Application Builder**.
2. Click the Edit button beside the Image Gallery application.
3. Click the first page, named `Home`.
4. Under **Page** click the Edit icon, and click the **JavaScript** tab.
5. Add the following at the start of the **Function and Global Variable Declaration** field:

```
function setApexSession(pathOrXhr) {
 var appId = $v('pFlowId');
 var sessionId = $v('pInstance');
 var apexSession = appId + ',' + sessionId;
 if ( typeof pathOrXhr === 'string' ) {
  var path = pathOrXhr;
  if ( path.indexOf('?') == -1 ) {
   path = path + '?_apex_session=' + apexSession;
  } else {
   path = path + '&_apex_session=' + apexSession;
  }
  return path;
 } else {
  var xhr = pathOrXhr;
  xhr.setRequestHeader('Apex-Session',apexSession);
  return xhr;
 }
}
```

6. This defines a JavaScript function named `setApexSession()` which will add the first party authentication information to an `XMLHttpRequest` object or a string containing a path.

   Now you must modify the existing JavaScript code to add call this function when appropriate.

7. After the line reading `xhr.open('POST',url,true);`, add the following line:

```
setApexSession(xhr);
```

8. After the line reading `xhr.open('GET', gallery_url);`, add the following line:

```
setApexSession(xhr);
```

9. Change the line reading `var uri = item.uri['$ref'];` to:

```
var uri = setApexSession(item.uri['$ref']);
```

10. Click **Apply Changes**.
11. Try running the application as before. It should work, because it is now providing the RESTful Services with the required authentication information.

Accessing the RESTful Services from a Third Party Application

If third parties want to consume and use the Gallery RESTful services, they must register the third party application in order to gain OAuth 2.0 credentials, which can then be used to initiate an interactive process by which users can authorize the third party application to access the RESTful Services on their behalf.

Once an application is registered, it can then acquire an access token. The access token must be provided with each request to a protected RESTful Service. Oracle REST Data Services verifies the access token before allowing access to the RESTful service.

OAuth 2.0 defines a number of different protocol flows that can be used by applications to acquire an access token. Oracle REST Data Services supports two of these protocol flows:

- **Authorization Code**. This flow is used when the third party application is able to keep its client credentials secure, for example, a third party website that is properly secured.
- **Implicit Grant**. This flow is used when the third party application cannot assure that its credentials would remain secret, for example, a JavaScript-based browser application or a native smartphone application.

The first step is to register the third party application. To demonstrate this, you will create a user representing the third party developer, and then use that user to register an application.

The steps in the related topics create a user in the `RESTEASY` workspace user repository and perform related actions.

> **Note:**
>
> In addition to authenticating users defined in workspace user repositories, Oracle REST Data Services can also authenticate against any user repository accessible from WebLogic Server. For information, see Authenticating Against WebLogic Server.

Topics:

- Creating the Third Party Developer User
- Registering the Third Party Application
- Acquiring an Access Token
- Using an Access Token
- About Browser Origins
- Configuring a RESTful Service for Cross Origin Resource Sharing
- Acquiring a Token Using the Authorization Code Protocol Flow
- About Securing the Access Token

Creating the Third Party Developer User

To create the third party developer user (the user account for the third party developer who wants to register an application to access the RESTful services), follow these steps:

1. Navigate to **Administration**.
2. Click **Manage Users and Groups**.
3. Click **Create User**, and enter the following information:
   - **Username**: `3rdparty_dev`
   - **Email Address**: Email address for this developer user
   - **Password**: Password for this user
   - **User Groups**: `OAuth 2.0 Client Developer`
4. Click **Create User**.

Registering the Third Party Application

To register the third party application to use the `Implicit Grant` OAuth 2.0 protocol flow, follow these steps:

1. Go to the following URI in your browser:

   `https://`*`server:port`*`/ords/resteasy/ui/oauth2/clients/`
2. Enter the credentials of the `3rdparty_dev` user created above, and click Sign In.
3. Click `Register Client`, and enter the following information:
   - **Name**: `3rd Party Gallery`
   - **Description**: `Demonstrates consuming the Gallery RESTful Service`
   - **Response Type**: `Token`
   - **Redirect URI**: `https://example.org/`
   - **Support Emai**l: Desired email address
   - **Required Scopes**: `Gallery Access`
4. Click **Register**.
5. Click **3rd Party Gallery** in the list that appears on the next page.
6. Note the values of the Client Identifier and the Authorization URI fields.

Acquiring an Access Token

To acquire an access token, a user must be prompted to approve access. To initiate the approval process, direct the user to the approval page using the following URI:

```
https://server:port/ords/resteasy/oauth2/auth?response_type=token&\
                                              client_id=CLIENT_IDENTIFIER&\
                                              state=STATE
```

where:

- `CLIENT_IDENTIFIER` is the Client Identifier assigned to the application when it was registered.
- `STATE` is a unique value generated by the application used to prevent Cross Site Request Forgery (CSRF) attacks.

Note the following about the Oracle REST Data Services OAuth 2.0 implementation:

- The OAuth 2.0 specification allows two optional parameters to be supplied in the above request:
  - `redirect_uri`: Identifies the location where the authorization server will redirect back to after the user has approved/denied access.
  - `scope`: Identifies the RESTful Service Privileges that the client wishes to access.

  Oracle REST Data Services does not support either of these parameters: both of these values are specified when the client is registered, so it would be redundant to repeat them here. Any values supplied for these parameters will be ignored.
- The OAuth 2.0 specification *recommends* the use of the `state` parameter, but Oracle REST Data Services ***requires*** the use of the parameter because of its importance in helping to prevent CSRF attacks.
- The response type is also specified when the application is registered, and thus the `response_type` parameter is also redundant; however, the OAuth 2.0 specification states the parameter is always required, so it must be included. It is an error if the `response_type` value differs from the registered response type.

When the preceding URI is accessed in a browser, the user is prompted to sign on, and then prompted to review the application's request for access and choose whether to approve or deny access.

If the user approves the request, then the browser will be redirected back to the registered redirect URI, and the access token will be encoded in the fragment portion of the URI:

```
https://example.org/#token_type=bearer&\
                                        access_token=ACCESS_TOKEN&\
                                        expires_in=TOKEN_LIFETIME&\
                                        state=STATE
```

where:

- `example.org` is used for illustrative purposes only. In a real application `example.org` will be replaced with the URL of the third party application that is requesting access.
- `ACCESS_TOKEN` is the unique, unguessable access token assigned to the current user session, and which must be provided with subsequent requests to the RESTful service.
- `TOKEN_LIFETIME` is the number of seconds for which the access token is valid.
- `STATE` is the unique value supplied by the application at the start of the authorization flow. If the returned `state` value does not match the initial `state` value, then there is an error condition, and the access token must not be used, because it is possible an attacker is attempting to subvert the authorization process through a CSRF attack.

> **Note:**
>
> You can modify the default OAuth access token duration (or lifetime) for all the generated access tokens. To achieve this, add the `security.oauth.tokenLifetime` entry to the `defaults.xml` configuration file in the following way, with the OAuth access token duration specified in seconds:
> `<entry key="security.oauth.tokenLifetime”>600</entry>`

If the user denies the request, or the user is not authorized to access the RESTful Service, the browser will be redirected back to the registered redirect URI, and an error message will be encoded in the fragment portion of the URI:

```
https://example.org/#error=access_denied&state=STATE
```

where:

- `error=access_denied` informs the client that the user is not authorized to access the RESTful Service, or chose not to approve access to the application.
- `STATE` is the unique value supplied by the application at the start of the authorization flow. If the returned `state` value does not match the initial `state` value, then there is an error condition, the client should ignore this response. It is possible an attacker is attempting to subvert the authorization process via a CSRF attack.

Using an Access Token

After the application has acquired an access token, the access token must be included with each request made to the protected RESTful service. To do this, an `Authorization` header is added to the HTTP request, with the following syntax:

```
Authorization: Bearer ACCESS_TOKEN
```

where:

- `ACCESS_TOKEN` is the access token value.

For example, a JavaScript-based browser application might invoke the Gallery service as follows:

```
var accessToken = ... /* initialize with the value of the access token */
var xhr = new XMLHttpRequest();
xhr.open('GET', 'https://server:port/ords/resteasy/gallery/images/',true);
/* Add the Access Token to the request */
xhr.setRequestHeader('Authorization', 'Bearer ' + accessToken);
xhr.onload = function(e) {
 /* logic to process the returned JSON document */
 ...
};
xhr.send();
```

The preceding example uses the `XMLHttpRequest.setRequestHeader(name,value)` function to add the `Authorization` header to the HTTP request. If the access token

is valid, then the server will respond with a JSON document listing the images in the gallery.

About Browser Origins

One of the key security concepts of web browsers is the **Same Origin Policy**, which permits scripts running on pages originating from the same web site (an Origin) to access each other's data with no restrictions, but prevents access to data originating from other web sites.

An origin is defined by the protocol, host name and port of a web-site. For example `https://example.com` is one origin and `https://another.example.com` is a different origin, because the host name differs. Similarly, `http://example.com` is a different origin than `https://example.com` because the protocol differs. Finally, `http://example.com` is a different origin from `http://example.com:8080` because the port differs.

For example, if a third party client of the Gallery RESTful service is located at:

```
https://thirdparty.com/gallery.html
```

and the Gallery RESTful service is located at:

```
https://example.com/ords/resteasy/gallery/images/
```

then the Same Origin Policy will prevent `gallery.html` making an `XMLHttpRequest` to `https://example.com/ords/resteasy/gallery/images/`, because scripts in the `https://thirdparty.com` origin can only access data from that same origin, and `https://example.com` is clearly a different origin.

This is proper if the authors of `https://example.com` do not trust the authors of `https://thirdparty.com`. However, if the authors do have reason to trust each other, then the Same Origin Policy is too restrictive. Fortunately, a protocol called **Cross Origin Resource Sharing (CORS)**, provides a means for `https://example.com` to inform the web browser that it trusts `https://thirdparty.com` and thus to instruct the browser to permit `gallery.html` to make an `XMLHttpRequest` to `https://example.com/ords/resteasy/gallery/images/`.

Configuring a RESTful Service for Cross Origin Resource Sharing

To configure a RESTful service for Cross Origin Resource Sharing, follow these steps:

1. Navigate to **SQL Workshop** and then **RESTful Services**.
2. Click the module named `gallery.example`.
3. For **Origins Allowed**, enter the origins that are permitted to access the RESTful service (origins are separated by a comma).
4. Press **Apply Changes**

Acquiring a Token Using the Authorization Code Protocol Flow

Other sections have explained acquiring an access token using the OAuth 2.0 Implicit protocol flow. This section explains how to do the same using the Authorization Code protocol flow. The process is slightly more involved than for the Implicit protocol flow, because it requires exchanging an authorization code for an access token.

This section will mimic this exchange process using cURL.

Topics:

- Registering the Client Application
- Acquiring an Authorization Code
- Exchanging an Authorization Code for an Access Token
- Extending OAuth 2.0 Session Duration

Registering the Client Application

[To register the client, follow these steps:

1. Go to the following URI in your browser:

   `https://server:port/ords/resteasy/ui/oauth2/clients/`

2. Enter the credentials of the `3rdparty_dev` user, and click **Sign In**.
3. Click **Register Client**, and enter the following information:
   - **Name**: `Another Gallery`
   - **Description**: `Demonstrates using the Authorization Code OAuth 2.0 Protocol Flow`
   - **Response Type**: `Code`
   - **Redirect URI** : `https://gallery.example.demo`
   - **Support EMail**: any desired email address
   - **Required Scopes**: `Gallery Access`
4. Click **Register**.
5. Click **3rd Party Gallery** in the list that appears on the next page.
6. Note the values of the Client Identifier, Client Secret, and the Authorization URI fields.

Acquiring an Authorization Code

The first step in the Authorization Code protocol flow is to acquire an authorization code. An authorization code is a short lived token that when presented along with the application's client identifier and secret can be exchanged for an access token.

To acquire an access token, the user must be prompted to approve access. To initiate the approval process, direct the user to the approval page using a URI in the following format:

```
https://server:port/ords/resteasy/oauth2/auth?response_type=code&\
                                              client_id=CLIENT_IDENTIFIER&\
                                              state=STATE
```

where:

- `CLIENT_IDENTIFIER` is the Client Identifier assigned to the application when it was registered.

- `STATE` is a unique value generated by the application used to prevent Cross Site Request Forgery (CSRF) attacks.

If the user approves the request, then the browser will be redirected back to the registered redirect URI, and the access token will be encoded in the query string portion of the URI:

```
https://gallery.example.demo?code=AUTHORIZATION_CODE&state=STATE
```

where:

- `AUTHORIZATION_CODE` is the authorization code value.
- `STATE` is the unique value supplied by the application at the start of the authorization flow. If the returned `state` value does not match the initial `state` value, then there is an error condition, the authorization code must not be used. It is possible an attacker is attempting to subvert the authorization process via a CSRF attack.

  Because the registered `https://gallery.example.demo` redirect URI does not exist, the browser will report a server not found error, but for the purposes of this example, this does not matter, because you can still see the authorization code value encoded in the URI. Note the value of the `code` parameter, because it will be used while Exchanging an Authorization Code for an Access Token.

Exchanging an Authorization Code for an Access Token

In this section you will use cURL to exchange the authorization code for an access token. To exchange an authorization code the application must make an HTTP request to the Oracle REST Data Services OAuth 2.0 token endpoint, providing the authorization code and its client identifier and secret. If the credentials are correct, Oracle REST Data Services responds with a JSON document containing the access token. Note that the application makes the HTTP request from its server side (where the client identifier and secret are securely stored) directly to Oracle REST Data Services; the web-browser is not involved at all in this step of the protocol flow.

Use a cURL command in the following format to exchange the authorization code for an access token:

```
curl -i -d "grant_type=authorization_code&code=AUTHORIZATION_CODE" \
     --user CLIENT_IDENTIFER:CLIENT_SECRET \
     https://server:port/ords/resteasy/oauth2/token
```

where:

- `AUTHORIZATION_CODE` is the authorization code value (which was encoded in the `code` parameter of the query string in the redirect URI in the previous section).
- `CLIENT_IDENTIFER` is the client identifier value.
- `CLIENT_SECRET` is the client secret value.

cURL translates the above commands into an HTTP request like the following:

```
POST /ords/resteasy/oauth2/token HTTP/1.1
Authorization: Basic Q0xJRU5UX0lERU5USUZJRVI6Q0xJRU5UX1NFQ1JFVA==
Host: server:port
Accept: */*
Content-Length: 59
Content-Type: application/x-www-form-urlencoded
```

```
grant_type=authorization_code&code=AUTHORIZATION_CODE
```

where:

- The request is an HTTP `POST` to the `oauth2/token` OAuth 2.0 token endpoint.
- The `Authorization` header uses the HTTP `BASIC` authentication protocol to encode the client identifier and secret to assert the application's identity.
- The `Content-Type` of the request is form data (`application/x-www-form-urlencoded`) and the content of the request is the form data asserting the OAuth 2.0 token grant type and the OAuth 2.0 authorization code value.

The preceding HTTP request will produce a response like the following:

```
HTTP/1.1 200 OK
ETag: "..."
Content-Type: application/json

{
 "access_token":"04tss-gM35uOeQzR_2ve4Q..",
 "token_type":"bearer",
 "expires_in":3600,
 "refresh_token":"UX4FVHhPFJl6GokvTXYw0A.."
}
```

The response is a JSON document containing the access token along with a refresh token. After the application has acquired an access token, the access token must be included with each request made to the protected RESTful Service. To do this an `Authorization` header is added to the HTTP request, with the following syntax:

```
Authorization: Bearer ACCESS_TOKEN
```

Related Topics

- Extending OAuth 2.0 Session Duration

Extending OAuth 2.0 Session Duration

To extend the lifetime of an OAuth 2.0 session, a refresh token can be exchanged for a new access token with a new expiration time. Note that refresh tokens are only issued for the Authorization Code protocol flow.

The application makes a similar request to that used to exchange an authorization code for an access token. Use a cURL command in the following format to exchange the refresh token for an access token:

```
curl -i -d "grant_type=refresh_token&refresh_token=REFRESH_TOKEN" \
     --user CLIENT_IDENTIFER:CLIENT_SECRET \
     https://server:port/ords/resteasy/oauth2/token
```

where:

- `REFRESH_TOKEN` is the refresh token value returned when the access token was initially issued.
- `CLIENT_IDENTIFER` is the client identifier value.
- `CLIENT_SECRET` is the client secret value.

cURL translates the above commands into an HTTP request like the following:

```
POST /ords/resteasy/oauth2/token HTTP/1.1
Authorization: Basic Q0xJRU5UX0lERU5USUZJRVI6Q0xJRU5UX1NFQ1JFVA==
Host: server:port
Accept: */*
Content-Length: 53
Content-Type: application/x-www-form-urlencoded

grant_type=refresh_token&refresh_token=REFRESH_TOKEN
```

where:

- The request is an HTTP `POST` to the `oauth2/token` OAuth 2.0 token endpoint.
- The `Authorization` header uses the HTTP `BASIC` authentication protocol to encode the client identifier and secret to assert the application's identity.
- The `Content-Type` of the request is form data (`application/x-www-form-urlencoded`) and the content of the request is the form data asserting the OAuth 2.0 token grant type and the refresh token value.

The preceding HTTP request will produce a response like the following:

```
HTTP/1.1 200 OK
ETag: "..."
Content-Type: application/json

{
 "access_token":"hECH_Fc7os2KtXT4pDfkzw..",
 "token_type":"bearer",
 "expires_in":3600,
 "refresh_token":"-7OBQKc_gUQG93ZHCi08Hg.."
}
```

The response is a JSON document containing the new access token along with a new refresh token. The existing access token and refresh token are invalidated, and any attempt to access a service using the old access token will fail.

About Securing the Access Token

In OAuth 2.0 the access token is the sole credential required to provide access to a protected service. It is, therefore, essential to keep the access token secure. Follow these guidelines to help keep the token secure:

- It is strongly recommended to use HTTPS for all protected RESTful Services. This prevents snooping attacks where an attacker may be able to steal access tokens by eavesdropping on insecure channels. It also prevents attackers from viewing the sensitive data that may be present in the payload of the requests.
- Ensure that the client application is not located in a browser origin with other applications or scripts that cannot be trusted. For example assume that user Alice has a client application hosted at the following location:

  `https://sharedhosting.com/alice/application`

  If another user (such as Fred) is also able to host his application in the same origin, for example, at:

  `https://sharedhosting.com/fred/trouble`

  then it will be easy for `/fred/trouble` to steal any access token acquired by `/alice/application`, because they share the same origin `https://`

`sharedhost.com`, and thus the browser will not prevent either application from accessing the other's data.

To protect against this scenario, Alice's application must be deployed in its own origin, for example:

```
https://alice.sharedhosting.com/application
```

or:

```
https://application.alice.sharedhosting.com
```

or:

```
https://aliceapp.com
```

E

Third-Party License Information

Oracle REST Data Services contains third-party code. See the Oracle Database Licensing Information 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.

jackson-annotations 2.10.3

```
Jackson Annotations
Copyright (c) 2019 Tatu Saloranta

LICENSE: Apache 2.0

                                 Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

      "License" shall mean the terms and conditions for use,
reproduction,
      and distribution as defined by Sections 1 through 9 of this
document.

      "Licensor" shall mean the copyright owner or entity authorized by
      the copyright owner that is granting the License.

      "Legal Entity" shall mean the union of the acting entity and all
      other entities that control, are controlled by, or are under
common
      control with that entity. For the purposes of this definition,
      "control" means (i) the power, direct or indirect, to cause the
      direction or management of such entity, whether by contract or
      otherwise, or (ii) ownership of fifty percent (50%) or more of the
      outstanding shares, or (iii) beneficial ownership of such entity.

      "You" (or "Your") shall mean an individual or Legal Entity
      exercising permissions granted by this License.

      "Source" form shall mean the preferred form for making
modifications,
      including but not limited to software source code, documentation
      source, and configuration files.
```

```
      "Object" form shall mean any form resulting from mechanical
      transformation or translation of a Source form, including but
      not limited to compiled object code, generated documentation,
      and conversions to other media types.

      "Work" shall mean the work of authorship, whether in Source or
      Object form, made available under the License, as indicated by a
      copyright notice that is included in or attached to the work
      (an example is provided in the Appendix below).

      "Derivative Works" shall mean any work, whether in Source or
Object
      form, that is based on (or derived from) the Work and for which
the
      editorial revisions, annotations, elaborations, or other
modifications
      represent, as a whole, an original work of authorship. For the
purposes
      of this License, Derivative Works shall not include works that
remain
      separable from, or merely link (or bind by name) to the
interfaces of,
      the Work and Derivative Works thereof.

      "Contribution" shall mean any work of authorship, including
      the original version of the Work and any modifications or
additions
      to that Work or Derivative Works thereof, that is intentionally
      submitted to Licensor for inclusion in the Work by the copyright
owner
      or by an individual or Legal Entity authorized to submit on
behalf of
      the copyright owner. For the purposes of this definition,
"submitted"
      means any form of electronic, verbal, or written communication
sent
      to the Licensor or its representatives, including but not limited
to
      communication on electronic mailing lists, source code control
systems,
      and issue tracking systems that are managed by, or on behalf of,
the
      Licensor for the purpose of discussing and improving the Work, but
      excluding communication that is conspicuously marked or otherwise
      designated in writing by the copyright owner as "Not a
Contribution."

      "Contributor" shall mean Licensor and any individual or Legal
Entity
      on behalf of whom a Contribution has been received by Licensor and
      subsequently incorporated within the Work.

   2. Grant of Copyright License. Subject to the terms and conditions of
      this License, each Contributor hereby grants to You a perpetual,
```

worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have
made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute
must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works;
or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The
contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside

```
            or as an addendum to the NOTICE text from the Work, provided
            that such additional attribution notices cannot be construed
            as modifying the License.

         You may add Your own copyright statement to Your modifications and
         may provide additional or different license terms and conditions
         for use, reproduction, or distribution of Your modifications, or
         for any such Derivative Works as a whole, provided Your use,
         reproduction, and distribution of the Work otherwise complies with
         the conditions stated in this License.

      5. Submission of Contributions. Unless You explicitly state
otherwise,
         any Contribution intentionally submitted for inclusion in the Work
         by You to the Licensor shall be under the terms and conditions of
         this License, without any additional terms or conditions.
         Notwithstanding the above, nothing herein shall supersede or
modify
         the terms of any separate license agreement you may have executed
         with Licensor regarding such Contributions.

      6. Trademarks. This License does not grant permission to use the
trade
         names, trademarks, service marks, or product names of the
Licensor,
         except as required for reasonable and customary use in describing
the
         origin of the Work and reproducing the content of the NOTICE file.

      7. Disclaimer of Warranty. Unless required by applicable law or
         agreed to in writing, Licensor provides the Work (and each
         Contributor provides its Contributions) on an "AS IS" BASIS,
         WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
         implied, including, without limitation, any warranties or
conditions
         of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
         PARTICULAR PURPOSE. You are solely responsible for determining the
         appropriateness of using or redistributing the Work and assume any
         risks associated with Your exercise of permissions under this
License.

      8. Limitation of Liability. In no event and under no legal theory,
         whether in tort (including negligence), contract, or otherwise,
         unless required by applicable law (such as deliberate and grossly
         negligent acts) or agreed to in writing, shall any Contributor be
         liable to You for damages, including any direct, indirect,
special,
         incidental, or consequential damages of any character arising as a
         result of this License or out of the use or inability to use the
         Work (including but not limited to damages for loss of goodwill,
         work stoppage, computer failure or malfunction, or any and all
         other commercial damages or losses), even if such Contributor
         has been advised of the possibility of such damages.

      9. Accepting Warranty or Additional Liability. While redistributing
```

```
      the Work or Derivative Works thereof, You may choose to offer,
      and charge a fee for, acceptance of support, warranty, indemnity,
      or other liability obligations and/or rights consistent with this
      License. However, in accepting such obligations, You may act only
      on Your own behalf and on Your sole responsibility, not on behalf
      of any other Contributor, and only if You agree to indemnify,
      defend, and hold each Contributor harmless for any liability
      incurred by, or claims asserted against, such Contributor by
reason
      of your accepting any such warranty or additional liability.

   END OF TERMS AND CONDITIONS

   APPENDIX: How to apply the Apache License to your work.

      To apply the Apache License to your work, attach the following
      boilerplate notice, with the fields enclosed by brackets "[]"
      replaced with your own identifying information. (Don't include
      the brackets!)  The text should be enclosed in the appropriate
      comment syntax for the file format. We also recommend that a
      file or class name and description of purpose be included on the
      same "printed page" as the copyright notice for easier
      identification within third-party archives.

   Copyright [yyyy] [name of copyright owner]

   Licensed under the Apache License, Version 2.0 (the "License");
   you may not use this file except in compliance with the License.
   You may obtain a copy of the License at

       http://www.apache.org/licenses/LICENSE-2.0

   Unless required by applicable law or agreed to in writing, software
   distributed under the License is distributed on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
   See the License for the specific language governing permissions and
   limitations under the License.
```

jackson-databind 2.10.3

```
Jackson Databind
Copyright (c) 2019 Tatu Saloranta

LICENSE: Apache 2.0

                                 Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.
```

```
      "License" shall mean the terms and conditions for use,
reproduction,
      and distribution as defined by Sections 1 through 9 of this
document.

      "Licensor" shall mean the copyright owner or entity authorized by
      the copyright owner that is granting the License.

      "Legal Entity" shall mean the union of the acting entity and all
      other entities that control, are controlled by, or are under
common
      control with that entity. For the purposes of this definition,
      "control" means (i) the power, direct or indirect, to cause the
      direction or management of such entity, whether by contract or
      otherwise, or (ii) ownership of fifty percent (50%) or more of the
      outstanding shares, or (iii) beneficial ownership of such entity.

      "You" (or "Your") shall mean an individual or Legal Entity
      exercising permissions granted by this License.

      "Source" form shall mean the preferred form for making
modifications,
      including but not limited to software source code, documentation
      source, and configuration files.

      "Object" form shall mean any form resulting from mechanical
      transformation or translation of a Source form, including but
      not limited to compiled object code, generated documentation,
      and conversions to other media types.

      "Work" shall mean the work of authorship, whether in Source or
      Object form, made available under the License, as indicated by a
      copyright notice that is included in or attached to the work
      (an example is provided in the Appendix below).

      "Derivative Works" shall mean any work, whether in Source or
Object
      form, that is based on (or derived from) the Work and for which
the
      editorial revisions, annotations, elaborations, or other
modifications
      represent, as a whole, an original work of authorship. For the
purposes
      of this License, Derivative Works shall not include works that
remain
      separable from, or merely link (or bind by name) to the
interfaces of,
      the Work and Derivative Works thereof.

      "Contribution" shall mean any work of authorship, including
      the original version of the Work and any modifications or
additions
      to that Work or Derivative Works thereof, that is intentionally
      submitted to Licensor for inclusion in the Work by the copyright
```

```
owner
      or by an individual or Legal Entity authorized to submit on
behalf of
      the copyright owner. For the purposes of this definition,
"submitted"
      means any form of electronic, verbal, or written communication
sent
      to the Licensor or its representatives, including but not limited
to
      communication on electronic mailing lists, source code control
systems,
      and issue tracking systems that are managed by, or on behalf of,
the
      Licensor for the purpose of discussing and improving the Work, but
      excluding communication that is conspicuously marked or otherwise
      designated in writing by the copyright owner as "Not a
Contribution."

      "Contributor" shall mean Licensor and any individual or Legal
Entity
      on behalf of whom a Contribution has been received by Licensor and
      subsequently incorporated within the Work.

   2. Grant of Copyright License. Subject to the terms and conditions of
      this License, each Contributor hereby grants to You a perpetual,
      worldwide, non-exclusive, no-charge, royalty-free, irrevocable
      copyright license to reproduce, prepare Derivative Works of,
      publicly display, publicly perform, sublicense, and distribute the
      Work and such Derivative Works in Source or Object form.

   3. Grant of Patent License. Subject to the terms and conditions of
      this License, each Contributor hereby grants to You a perpetual,
      worldwide, non-exclusive, no-charge, royalty-free, irrevocable
      (except as stated in this section) patent license to make, have
made,
      use, offer to sell, sell, import, and otherwise transfer the Work,
      where such license applies only to those patent claims licensable
      by such Contributor that are necessarily infringed by their
      Contribution(s) alone or by combination of their Contribution(s)
      with the Work to which such Contribution(s) was submitted. If You
      institute patent litigation against any entity (including a
      cross-claim or counterclaim in a lawsuit) alleging that the Work
      or a Contribution incorporated within the Work constitutes direct
      or contributory patent infringement, then any patent licenses
      granted to You under this License for that Work shall terminate
      as of the date such litigation is filed.

   4. Redistribution. You may reproduce and distribute copies of the
      Work or Derivative Works thereof in any medium, with or without
      modifications, and in Source or Object form, provided that You
      meet the following conditions:

      (a) You must give any other recipients of the Work or
          Derivative Works a copy of this License; and
```

```
      (b) You must cause any modified files to carry prominent notices
          stating that You changed the files; and

      (c) You must retain, in the Source form of any Derivative Works
          that You distribute, all copyright, patent, trademark, and
          attribution notices from the Source form of the Work,
          excluding those notices that do not pertain to any part of
          the Derivative Works; and

      (d) If the Work includes a "NOTICE" text file as part of its
          distribution, then any Derivative Works that You distribute
must
          include a readable copy of the attribution notices contained
          within such NOTICE file, excluding those notices that do not
          pertain to any part of the Derivative Works, in at least one
          of the following places: within a NOTICE text file distributed
          as part of the Derivative Works; within the Source form or
          documentation, if provided along with the Derivative Works;
or,
          within a display generated by the Derivative Works, if and
          wherever such third-party notices normally appear. The
contents
          of the NOTICE file are for informational purposes only and
          do not modify the License. You may add Your own attribution
          notices within Derivative Works that You distribute, alongside
          or as an addendum to the NOTICE text from the Work, provided
          that such additional attribution notices cannot be construed
          as modifying the License.

      You may add Your own copyright statement to Your modifications and
      may provide additional or different license terms and conditions
      for use, reproduction, or distribution of Your modifications, or
      for any such Derivative Works as a whole, provided Your use,
      reproduction, and distribution of the Work otherwise complies with
      the conditions stated in this License.

   5. Submission of Contributions. Unless You explicitly state
otherwise,
      any Contribution intentionally submitted for inclusion in the Work
      by You to the Licensor shall be under the terms and conditions of
      this License, without any additional terms or conditions.
      Notwithstanding the above, nothing herein shall supersede or
modify
      the terms of any separate license agreement you may have executed
      with Licensor regarding such Contributions.

   6. Trademarks. This License does not grant permission to use the
trade
      names, trademarks, service marks, or product names of the
Licensor,
      except as required for reasonable and customary use in describing
the
      origin of the Work and reproducing the content of the NOTICE file.

   7. Disclaimer of Warranty. Unless required by applicable law or
```

```
      agreed to in writing, Licensor provides the Work (and each
      Contributor provides its Contributions) on an "AS IS" BASIS,
      WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
      implied, including, without limitation, any warranties or
conditions
      of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
      PARTICULAR PURPOSE. You are solely responsible for determining the
      appropriateness of using or redistributing the Work and assume any
      risks associated with Your exercise of permissions under this
License.

   8. Limitation of Liability. In no event and under no legal theory,
      whether in tort (including negligence), contract, or otherwise,
      unless required by applicable law (such as deliberate and grossly
      negligent acts) or agreed to in writing, shall any Contributor be
      liable to You for damages, including any direct, indirect,
special,
      incidental, or consequential damages of any character arising as a
      result of this License or out of the use or inability to use the
      Work (including but not limited to damages for loss of goodwill,
      work stoppage, computer failure or malfunction, or any and all
      other commercial damages or losses), even if such Contributor
      has been advised of the possibility of such damages.

   9. Accepting Warranty or Additional Liability. While redistributing
      the Work or Derivative Works thereof, You may choose to offer,
      and charge a fee for, acceptance of support, warranty, indemnity,
      or other liability obligations and/or rights consistent with this
      License. However, in accepting such obligations, You may act only
      on Your own behalf and on Your sole responsibility, not on behalf
      of any other Contributor, and only if You agree to indemnify,
      defend, and hold each Contributor harmless for any liability
      incurred by, or claims asserted against, such Contributor by
reason
      of your accepting any such warranty or additional liability.

   END OF TERMS AND CONDITIONS

   APPENDIX: How to apply the Apache License to your work.

      To apply the Apache License to your work, attach the following
      boilerplate notice, with the fields enclosed by brackets "[]"
      replaced with your own identifying information. (Don't include
      the brackets!)  The text should be enclosed in the appropriate
      comment syntax for the file format. We also recommend that a
      file or class name and description of purpose be included on the
      same "printed page" as the copyright notice for easier
      identification within third-party archives.

   Copyright [yyyy] [name of copyright owner]

   Licensed under the Apache License, Version 2.0 (the "License");
   you may not use this file except in compliance with the License.
   You may obtain a copy of the License at
```

```
       http://www.apache.org/licenses/LICENSE-2.0

   Unless required by applicable law or agreed to in writing, software
   distributed under the License is distributed on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
   See the License for the specific language governing permissions and
   limitations under the License.

COPYRIGHT NOTICE
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi),
and has
been in development since 2007.
It is currently developed by a community of developers, as well as
supported
commercially by FasterXML.com.

-----------------jackson-core 2.10.2 ----------------------
COPYRIGHT: Copyright (c) 2007- Tatu Saloranta, tatu.saloranta@iki.fi
LICENSE: Apache 2.0

## Credits

A list of contributors may be found from CREDITS file, which is included
in some artifacts (usually source distributions); but is always
available
from the source code management (SCM) system project uses.
-----------------jackson-annotations 2.10.2 ----------------------
COPYRIGHT: Copyright (c) 2007- Tatu Saloranta, tatu.saloranta@iki.fi
LICENSE: Apache 2.0
```

jackson-dataformat-xml 2.10.3

```
Top Level Component Name: Jackson Data Format XML
Top Level Component License:

                                 Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

      "License" shall mean the terms and conditions for use,
reproduction,
      and distribution as defined by Sections 1 through 9 of this
document.

      "Licensor" shall mean the copyright owner or entity authorized by
```

```
      the copyright owner that is granting the License.

      "Legal Entity" shall mean the union of the acting entity and all
      other entities that control, are controlled by, or are under
common
      control with that entity. For the purposes of this definition,
      "control" means (i) the power, direct or indirect, to cause the
      direction or management of such entity, whether by contract or
      otherwise, or (ii) ownership of fifty percent (50%) or more of the
      outstanding shares, or (iii) beneficial ownership of such entity.

      "You" (or "Your") shall mean an individual or Legal Entity
      exercising permissions granted by this License.

      "Source" form shall mean the preferred form for making
modifications,
      including but not limited to software source code, documentation
      source, and configuration files.

      "Object" form shall mean any form resulting from mechanical
      transformation or translation of a Source form, including but
      not limited to compiled object code, generated documentation,
      and conversions to other media types.

      "Work" shall mean the work of authorship, whether in Source or
      Object form, made available under the License, as indicated by a
      copyright notice that is included in or attached to the work
      (an example is provided in the Appendix below).

      "Derivative Works" shall mean any work, whether in Source or
Object
      form, that is based on (or derived from) the Work and for which
the
      editorial revisions, annotations, elaborations, or other
modifications
      represent, as a whole, an original work of authorship. For the
purposes
      of this License, Derivative Works shall not include works that
remain
      separable from, or merely link (or bind by name) to the
interfaces of,
      the Work and Derivative Works thereof.

      "Contribution" shall mean any work of authorship, including
      the original version of the Work and any modifications or
additions
      to that Work or Derivative Works thereof, that is intentionally
      submitted to Licensor for inclusion in the Work by the copyright
owner
      or by an individual or Legal Entity authorized to submit on
behalf of
      the copyright owner. For the purposes of this definition,
"submitted"
      means any form of electronic, verbal, or written communication
sent
```

```
      to the Licensor or its representatives, including but not limited
to
      communication on electronic mailing lists, source code control
systems,
      and issue tracking systems that are managed by, or on behalf of,
the
      Licensor for the purpose of discussing and improving the Work, but
      excluding communication that is conspicuously marked or otherwise
      designated in writing by the copyright owner as "Not a
Contribution."

      "Contributor" shall mean Licensor and any individual or Legal
Entity
      on behalf of whom a Contribution has been received by Licensor and
      subsequently incorporated within the Work.

   2. Grant of Copyright License. Subject to the terms and conditions of
      this License, each Contributor hereby grants to You a perpetual,
      worldwide, non-exclusive, no-charge, royalty-free, irrevocable
      copyright license to reproduce, prepare Derivative Works of,
      publicly display, publicly perform, sublicense, and distribute the
      Work and such Derivative Works in Source or Object form.

   3. Grant of Patent License. Subject to the terms and conditions of
      this License, each Contributor hereby grants to You a perpetual,
      worldwide, non-exclusive, no-charge, royalty-free, irrevocable
      (except as stated in this section) patent license to make, have
made,
      use, offer to sell, sell, import, and otherwise transfer the Work,
      where such license applies only to those patent claims licensable
      by such Contributor that are necessarily infringed by their
      Contribution(s) alone or by combination of their Contribution(s)
      with the Work to which such Contribution(s) was submitted. If You
      institute patent litigation against any entity (including a
      cross-claim or counterclaim in a lawsuit) alleging that the Work
      or a Contribution incorporated within the Work constitutes direct
      or contributory patent infringement, then any patent licenses
      granted to You under this License for that Work shall terminate
      as of the date such litigation is filed.

   4. Redistribution. You may reproduce and distribute copies of the
      Work or Derivative Works thereof in any medium, with or without
      modifications, and in Source or Object form, provided that You
      meet the following conditions:

         (a) You must give any other recipients of the Work or
             Derivative Works a copy of this License; and

         (b) You must cause any modified files to carry prominent notices
             stating that You changed the files; and

         (c) You must retain, in the Source form of any Derivative Works
             that You distribute, all copyright, patent, trademark, and
             attribution notices from the Source form of the Work,
             excluding those notices that do not pertain to any part of
```

```
          the Derivative Works; and

      (d) If the Work includes a "NOTICE" text file as part of its
          distribution, then any Derivative Works that You distribute
must
          include a readable copy of the attribution notices contained
          within such NOTICE file, excluding those notices that do not
          pertain to any part of the Derivative Works, in at least one
          of the following places: within a NOTICE text file distributed
          as part of the Derivative Works; within the Source form or
          documentation, if provided along with the Derivative Works;
or,
          within a display generated by the Derivative Works, if and
          wherever such third-party notices normally appear. The
contents
          of the NOTICE file are for informational purposes only and
          do not modify the License. You may add Your own attribution
          notices within Derivative Works that You distribute, alongside
          or as an addendum to the NOTICE text from the Work, provided
          that such additional attribution notices cannot be construed
          as modifying the License.

      You may add Your own copyright statement to Your modifications and
      may provide additional or different license terms and conditions
      for use, reproduction, or distribution of Your modifications, or
      for any such Derivative Works as a whole, provided Your use,
      reproduction, and distribution of the Work otherwise complies with
      the conditions stated in this License.

   5. Submission of Contributions. Unless You explicitly state
otherwise,
      any Contribution intentionally submitted for inclusion in the Work
      by You to the Licensor shall be under the terms and conditions of
      this License, without any additional terms or conditions.
      Notwithstanding the above, nothing herein shall supersede or
modify
      the terms of any separate license agreement you may have executed
      with Licensor regarding such Contributions.

   6. Trademarks. This License does not grant permission to use the
trade
      names, trademarks, service marks, or product names of the
Licensor,
      except as required for reasonable and customary use in describing
the
      origin of the Work and reproducing the content of the NOTICE file.

   7. Disclaimer of Warranty. Unless required by applicable law or
      agreed to in writing, Licensor provides the Work (and each
      Contributor provides its Contributions) on an "AS IS" BASIS,
      WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
      implied, including, without limitation, any warranties or
conditions
      of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
      PARTICULAR PURPOSE. You are solely responsible for determining the
```

```
      appropriateness of using or redistributing the Work and assume any
      risks associated with Your exercise of permissions under this
License.

   8. Limitation of Liability. In no event and under no legal theory,
      whether in tort (including negligence), contract, or otherwise,
      unless required by applicable law (such as deliberate and grossly
      negligent acts) or agreed to in writing, shall any Contributor be
      liable to You for damages, including any direct, indirect,
special,
      incidental, or consequential damages of any character arising as a
      result of this License or out of the use or inability to use the
      Work (including but not limited to damages for loss of goodwill,
      work stoppage, computer failure or malfunction, or any and all
      other commercial damages or losses), even if such Contributor
      has been advised of the possibility of such damages.

   9. Accepting Warranty or Additional Liability. While redistributing
      the Work or Derivative Works thereof, You may choose to offer,
      and charge a fee for, acceptance of support, warranty, indemnity,
      or other liability obligations and/or rights consistent with this
      License. However, in accepting such obligations, You may act only
      on Your own behalf and on Your sole responsibility, not on behalf
      of any other Contributor, and only if You agree to indemnify,
      defend, and hold each Contributor harmless for any liability
      incurred by, or claims asserted against, such Contributor by
reason
      of your accepting any such warranty or additional liability.

   END OF TERMS AND CONDITIONS

============================================================
Top Level Component Copyright:

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi),
and has
been in development since 2007.
It is currently developed by a community of developers, as well as
supported
commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different
licenses.
To find the details that apply to this artifact see the accompanying
LICENSE file.
For more information, including possible other licensing options,
contact
FasterXML.com (http://fasterxml.com).

## Credits
```

```
A list of contributors may be found from CREDITS file, which is included
in some artifacts (usually source distributions); but is always
available
from the source code management (SCM) system project uses.

=============================================================

jackson-dataformat-xml has these fourth-party dependencies that are
also licensed under the
Apache 2.0 license:

  * jackson-core
  * jackson-annotations
  * jackson-databind
  * jackson-module-jaxb-annotations
  * com.fasterxml.woodstox/woodstox-core
  * stax2-api

=============================================================

woodstox-core depends on org.codehous.woodstox/stax2-api 3.1.4, which
is licensed under
the BSD 2-clause license.

Contents of stax2-api/src/main/resources/META-INF/LICENSE:

    This copy of Stax2 API is licensed under the
    Simplified BSF License (also known as "2-clause BSD", or "FreeBSD
License")
    See the License for details about distribution rights, and the
    specific rights regarding derivate works.

    You may obtain a copy of the License at:

    http://www.opensource.org/licenses/bsd-license.php

    with details of:

     = FasterXML.com
     = 2010-

Contents of https://opensource.org/licenses/bsd-license.php:

    Copyright

    Redistribution and use in source and binary forms, with or without
modification, are
    permitted provided that the following conditions are met:

    1. Redistributions of source code must retain the above copyright
notice, this list of
    conditions and the following disclaimer.

    2. Redistributions in binary form must reproduce the above
```

```
copyright notice, this list
    of conditions and the following disclaimer in the documentation
and/or other materials
    provided with the distribution.

    THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS
    OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY
    AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
SHALL THE COPYRIGHT HOLDER OR
    CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
    DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF
    USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY,
    WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN
    ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

=============================================================

Stax2 API  License :
Note: This license has also been called the "Simplified BSD License"
and the "FreeBSD License". See also the 3-clause BSD License.

Copyright

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE
USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Stax2 API Copyright:
```

```
This copy of Stax2 API is licensed under the
Simplified BSD License (also known as "2-clause BSD", or "FreeBSD
License")
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.opensource.org/licenses/bsd-license.php

with details of:

 = FasterXML.com
 = 2010-


============================================================

Jackson Annotations
Copyright (c) 2019 Tatu Saloranta

LICENSE: Apache 2.0

============================================================

Jackson Core
Copyright © 2008–2019 FasterXML. All rights reserved.

This copy of Jackson JSON processor streaming parser/generator is
licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

NOTICE FILE:
===============
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi),
and has
been in development since 2007.
It is currently developed by a community of developers, as well as
supported
commercially by FasterXML.com.

## Licensing

Jackson core and extension components may licensed under different
licenses.
To find the details that apply to this artifact see the accompanying
```

```
LICENSE file.
For more information, including possible other licensing options,
contact
FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included
in some artifacts (usually source distributions); but is always
available
from the source code management (SCM) system project uses.
===============

Jackson Databind
Copyright (c) 2019 Tatu Saloranta

License: Apache 2.0

===============

Jackson Databind
Copyright (c) 2019 Tatu Saloranta

License: Apache 2.0
```

jackson-core 2.10.3

```
Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

      "License" shall mean the terms and conditions for use,
reproduction,
      and distribution as defined by Sections 1 through 9 of this
document.

      "Licensor" shall mean the copyright owner or entity authorized by
      the copyright owner that is granting the License.

      "Legal Entity" shall mean the union of the acting entity and all
      other entities that control, are controlled by, or are under
common
      control with that entity. For the purposes of this definition,
      "control" means (i) the power, direct or indirect, to cause the
      direction or management of such entity, whether by contract or
      otherwise, or (ii) ownership of fifty percent (50%) or more of the
      outstanding shares, or (iii) beneficial ownership of such entity.

      "You" (or "Your") shall mean an individual or Legal Entity
```

```
      exercising permissions granted by this License.

      "Source" form shall mean the preferred form for making
modifications,
      including but not limited to software source code, documentation
      source, and configuration files.

      "Object" form shall mean any form resulting from mechanical
      transformation or translation of a Source form, including but
      not limited to compiled object code, generated documentation,
      and conversions to other media types.

      "Work" shall mean the work of authorship, whether in Source or
      Object form, made available under the License, as indicated by a
      copyright notice that is included in or attached to the work
      (an example is provided in the Appendix below).

      "Derivative Works" shall mean any work, whether in Source or
Object
      form, that is based on (or derived from) the Work and for which
the
      editorial revisions, annotations, elaborations, or other
modifications
      represent, as a whole, an original work of authorship. For the
purposes
      of this License, Derivative Works shall not include works that
remain
      separable from, or merely link (or bind by name) to the
interfaces of,
      the Work and Derivative Works thereof.

      "Contribution" shall mean any work of authorship, including
      the original version of the Work and any modifications or
additions
      to that Work or Derivative Works thereof, that is intentionally
      submitted to Licensor for inclusion in the Work by the copyright
owner
      or by an individual or Legal Entity authorized to submit on
behalf of
      the copyright owner. For the purposes of this definition,
"submitted"
      means any form of electronic, verbal, or written communication
sent
      to the Licensor or its representatives, including but not limited
to
      communication on electronic mailing lists, source code control
systems,
      and issue tracking systems that are managed by, or on behalf of,
the
      Licensor for the purpose of discussing and improving the Work, but
      excluding communication that is conspicuously marked or otherwise
      designated in writing by the copyright owner as "Not a
Contribution."

      "Contributor" shall mean Licensor and any individual or Legal
```

```
Entity
      on behalf of whom a Contribution has been received by Licensor and
      subsequently incorporated within the Work.

   2. Grant of Copyright License. Subject to the terms and conditions of
      this License, each Contributor hereby grants to You a perpetual,
      worldwide, non-exclusive, no-charge, royalty-free, irrevocable
      copyright license to reproduce, prepare Derivative Works of,
      publicly display, publicly perform, sublicense, and distribute the
      Work and such Derivative Works in Source or Object form.

   3. Grant of Patent License. Subject to the terms and conditions of
      this License, each Contributor hereby grants to You a perpetual,
      worldwide, non-exclusive, no-charge, royalty-free, irrevocable
      (except as stated in this section) patent license to make, have
made,
      use, offer to sell, sell, import, and otherwise transfer the Work,
      where such license applies only to those patent claims licensable
      by such Contributor that are necessarily infringed by their
      Contribution(s) alone or by combination of their Contribution(s)
      with the Work to which such Contribution(s) was submitted. If You
      institute patent litigation against any entity (including a
      cross-claim or counterclaim in a lawsuit) alleging that the Work
      or a Contribution incorporated within the Work constitutes direct
      or contributory patent infringement, then any patent licenses
      granted to You under this License for that Work shall terminate
      as of the date such litigation is filed.

   4. Redistribution. You may reproduce and distribute copies of the
      Work or Derivative Works thereof in any medium, with or without
      modifications, and in Source or Object form, provided that You
      meet the following conditions:

      (a) You must give any other recipients of the Work or
          Derivative Works a copy of this License; and

      (b) You must cause any modified files to carry prominent notices
          stating that You changed the files; and

      (c) You must retain, in the Source form of any Derivative Works
          that You distribute, all copyright, patent, trademark, and
          attribution notices from the Source form of the Work,
          excluding those notices that do not pertain to any part of
          the Derivative Works; and

      (d) If the Work includes a "NOTICE" text file as part of its
          distribution, then any Derivative Works that You distribute
must
          include a readable copy of the attribution notices contained
          within such NOTICE file, excluding those notices that do not
          pertain to any part of the Derivative Works, in at least one
          of the following places: within a NOTICE text file distributed
          as part of the Derivative Works; within the Source form or
          documentation, if provided along with the Derivative Works;
or,
```

```
          within a display generated by the Derivative Works, if and
          wherever such third-party notices normally appear. The
contents
          of the NOTICE file are for informational purposes only and
          do not modify the License. You may add Your own attribution
          notices within Derivative Works that You distribute, alongside
          or as an addendum to the NOTICE text from the Work, provided
          that such additional attribution notices cannot be construed
          as modifying the License.

      You may add Your own copyright statement to Your modifications and
      may provide additional or different license terms and conditions
      for use, reproduction, or distribution of Your modifications, or
      for any such Derivative Works as a whole, provided Your use,
      reproduction, and distribution of the Work otherwise complies with
      the conditions stated in this License.

   5. Submission of Contributions. Unless You explicitly state
otherwise,
      any Contribution intentionally submitted for inclusion in the Work
      by You to the Licensor shall be under the terms and conditions of
      this License, without any additional terms or conditions.
      Notwithstanding the above, nothing herein shall supersede or
modify
      the terms of any separate license agreement you may have executed
      with Licensor regarding such Contributions.

   6. Trademarks. This License does not grant permission to use the
trade
      names, trademarks, service marks, or product names of the
Licensor,
      except as required for reasonable and customary use in describing
the
      origin of the Work and reproducing the content of the NOTICE file.

   7. Disclaimer of Warranty. Unless required by applicable law or
      agreed to in writing, Licensor provides the Work (and each
      Contributor provides its Contributions) on an "AS IS" BASIS,
      WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
      implied, including, without limitation, any warranties or
conditions
      of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
      PARTICULAR PURPOSE. You are solely responsible for determining the
      appropriateness of using or redistributing the Work and assume any
      risks associated with Your exercise of permissions under this
License.

   8. Limitation of Liability. In no event and under no legal theory,
      whether in tort (including negligence), contract, or otherwise,
      unless required by applicable law (such as deliberate and grossly
      negligent acts) or agreed to in writing, shall any Contributor be
      liable to You for damages, including any direct, indirect,
special,
      incidental, or consequential damages of any character arising as a
      result of this License or out of the use or inability to use the
```

```
      Work (including but not limited to damages for loss of goodwill,
      work stoppage, computer failure or malfunction, or any and all
      other commercial damages or losses), even if such Contributor
      has been advised of the possibility of such damages.

   9. Accepting Warranty or Additional Liability. While redistributing
      the Work or Derivative Works thereof, You may choose to offer,
      and charge a fee for, acceptance of support, warranty, indemnity,
      or other liability obligations and/or rights consistent with this
      License. However, in accepting such obligations, You may act only
      on Your own behalf and on Your sole responsibility, not on behalf
      of any other Contributor, and only if You agree to indemnify,
      defend, and hold each Contributor harmless for any liability
      incurred by, or claims asserted against, such Contributor by
reason
      of your accepting any such warranty or additional liability.

   END OF TERMS AND CONDITIONS

   APPENDIX: How to apply the Apache License to your work.

      To apply the Apache License to your work, attach the following
      boilerplate notice, with the fields enclosed by brackets "[]"
      replaced with your own identifying information. (Don't include
      the brackets!)  The text should be enclosed in the appropriate
      comment syntax for the file format. We also recommend that a
      file or class name and description of purpose be included on the
      same "printed page" as the copyright notice for easier
      identification within third-party archives.

   Copyright [yyyy] [name of copyright owner]

   Licensed under the Apache License, Version 2.0 (the "License");
   you may not use this file except in compliance with the License.
   You may obtain a copy of the License at

       http://www.apache.org/licenses/LICENSE-2.0

   Unless required by applicable law or agreed to in writing, software
   distributed under the License is distributed on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
   See the License for the specific language governing permissions and
   limitations under the License.
```

Google Guava 29.0

```
COPYRIGHT: Copyright (C) 2008 The Guava Authors

LICENSE: Apache 2.0

                                 Apache License
```

```
                                 Version 2.0, January 2004
                              http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

      "License" shall mean the terms and conditions for use,
reproduction,
      and distribution as defined by Sections 1 through 9 of this
document.

      "Licensor" shall mean the copyright owner or entity authorized by
      the copyright owner that is granting the License.

      "Legal Entity" shall mean the union of the acting entity and all
      other entities that control, are controlled by, or are under
common
      control with that entity. For the purposes of this definition,
      "control" means (i) the power, direct or indirect, to cause the
      direction or management of such entity, whether by contract or
      otherwise, or (ii) ownership of fifty percent (50%) or more of the
      outstanding shares, or (iii) beneficial ownership of such entity.

      "You" (or "Your") shall mean an individual or Legal Entity
      exercising permissions granted by this License.

      "Source" form shall mean the preferred form for making
modifications,
      including but not limited to software source code, documentation
      source, and configuration files.

      "Object" form shall mean any form resulting from mechanical
      transformation or translation of a Source form, including but
      not limited to compiled object code, generated documentation,
      and conversions to other media types.

      "Work" shall mean the work of authorship, whether in Source or
      Object form, made available under the License, as indicated by a
      copyright notice that is included in or attached to the work
      (an example is provided in the Appendix below).

      "Derivative Works" shall mean any work, whether in Source or
Object
      form, that is based on (or derived from) the Work and for which
the
      editorial revisions, annotations, elaborations, or other
modifications
      represent, as a whole, an original work of authorship. For the
purposes
      of this License, Derivative Works shall not include works that
remain
      separable from, or merely link (or bind by name) to the
interfaces of,
      the Work and Derivative Works thereof.
```

```
      "Contribution" shall mean any work of authorship, including
      the original version of the Work and any modifications or
additions
      to that Work or Derivative Works thereof, that is intentionally
      submitted to Licensor for inclusion in the Work by the copyright
owner
      or by an individual or Legal Entity authorized to submit on
behalf of
      the copyright owner. For the purposes of this definition,
"submitted"
      means any form of electronic, verbal, or written communication
sent
      to the Licensor or its representatives, including but not limited
to
      communication on electronic mailing lists, source code control
systems,
      and issue tracking systems that are managed by, or on behalf of,
the
      Licensor for the purpose of discussing and improving the Work, but
      excluding communication that is conspicuously marked or otherwise
      designated in writing by the copyright owner as "Not a
Contribution."

      "Contributor" shall mean Licensor and any individual or Legal
Entity
      on behalf of whom a Contribution has been received by Licensor and
      subsequently incorporated within the Work.

   2. Grant of Copyright License. Subject to the terms and conditions of
      this License, each Contributor hereby grants to You a perpetual,
      worldwide, non-exclusive, no-charge, royalty-free, irrevocable
      copyright license to reproduce, prepare Derivative Works of,
      publicly display, publicly perform, sublicense, and distribute the
      Work and such Derivative Works in Source or Object form.

   3. Grant of Patent License. Subject to the terms and conditions of
      this License, each Contributor hereby grants to You a perpetual,
      worldwide, non-exclusive, no-charge, royalty-free, irrevocable
      (except as stated in this section) patent license to make, have
made,
      use, offer to sell, sell, import, and otherwise transfer the Work,
      where such license applies only to those patent claims licensable
      by such Contributor that are necessarily infringed by their
      Contribution(s) alone or by combination of their Contribution(s)
      with the Work to which such Contribution(s) was submitted. If You
      institute patent litigation against any entity (including a
      cross-claim or counterclaim in a lawsuit) alleging that the Work
      or a Contribution incorporated within the Work constitutes direct
      or contributory patent infringement, then any patent licenses
      granted to You under this License for that Work shall terminate
      as of the date such litigation is filed.

   4. Redistribution. You may reproduce and distribute copies of the
      Work or Derivative Works thereof in any medium, with or without
```

```
      modifications, and in Source or Object form, provided that You
      meet the following conditions:

      (a) You must give any other recipients of the Work or
          Derivative Works a copy of this License; and

      (b) You must cause any modified files to carry prominent notices
          stating that You changed the files; and

      (c) You must retain, in the Source form of any Derivative Works
          that You distribute, all copyright, patent, trademark, and
          attribution notices from the Source form of the Work,
          excluding those notices that do not pertain to any part of
          the Derivative Works; and

      (d) If the Work includes a "NOTICE" text file as part of its
          distribution, then any Derivative Works that You distribute
must
          include a readable copy of the attribution notices contained
          within such NOTICE file, excluding those notices that do not
          pertain to any part of the Derivative Works, in at least one
          of the following places: within a NOTICE text file distributed
          as part of the Derivative Works; within the Source form or
          documentation, if provided along with the Derivative Works;
or,
          within a display generated by the Derivative Works, if and
          wherever such third-party notices normally appear. The
contents
          of the NOTICE file are for informational purposes only and
          do not modify the License. You may add Your own attribution
          notices within Derivative Works that You distribute, alongside
          or as an addendum to the NOTICE text from the Work, provided
          that such additional attribution notices cannot be construed
          as modifying the License.

      You may add Your own copyright statement to Your modifications and
      may provide additional or different license terms and conditions
      for use, reproduction, or distribution of Your modifications, or
      for any such Derivative Works as a whole, provided Your use,
      reproduction, and distribution of the Work otherwise complies with
      the conditions stated in this License.

   5. Submission of Contributions. Unless You explicitly state
otherwise,
      any Contribution intentionally submitted for inclusion in the Work
      by You to the Licensor shall be under the terms and conditions of
      this License, without any additional terms or conditions.
      Notwithstanding the above, nothing herein shall supersede or
modify
      the terms of any separate license agreement you may have executed
      with Licensor regarding such Contributions.

   6. Trademarks. This License does not grant permission to use the
trade
      names, trademarks, service marks, or product names of the
```

```
Licensor,
      except as required for reasonable and customary use in describing
the
      origin of the Work and reproducing the content of the NOTICE file.

   7. Disclaimer of Warranty. Unless required by applicable law or
      agreed to in writing, Licensor provides the Work (and each
      Contributor provides its Contributions) on an "AS IS" BASIS,
      WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
      implied, including, without limitation, any warranties or
conditions
      of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
      PARTICULAR PURPOSE. You are solely responsible for determining the
      appropriateness of using or redistributing the Work and assume any
      risks associated with Your exercise of permissions under this
License.

   8. Limitation of Liability. In no event and under no legal theory,
      whether in tort (including negligence), contract, or otherwise,
      unless required by applicable law (such as deliberate and grossly
      negligent acts) or agreed to in writing, shall any Contributor be
      liable to You for damages, including any direct, indirect,
special,
      incidental, or consequential damages of any character arising as a
      result of this License or out of the use or inability to use the
      Work (including but not limited to damages for loss of goodwill,
      work stoppage, computer failure or malfunction, or any and all
      other commercial damages or losses), even if such Contributor
      has been advised of the possibility of such damages.

   9. Accepting Warranty or Additional Liability. While redistributing
      the Work or Derivative Works thereof, You may choose to offer,
      and charge a fee for, acceptance of support, warranty, indemnity,
      or other liability obligations and/or rights consistent with this
      License. However, in accepting such obligations, You may act only
      on Your own behalf and on Your sole responsibility, not on behalf
      of any other Contributor, and only if You agree to indemnify,
      defend, and hold each Contributor harmless for any liability
      incurred by, or claims asserted against, such Contributor by
reason
      of your accepting any such warranty or additional liability.

   END OF TERMS AND CONDITIONS



=========================
Modules which can optionally be built:
guava-bom
guava-gwt
guava-testlib
guava-tests
guava
refactorings
```

```
util


========================
Compile time Dependencies:

com.google.code.findbugs » jsr305 3.0.2
COPYRIGHT: (C) University of Maryland

LICENSE:
The JSR-305 reference implementation (lib/jsr305.jar) is
distributed under the terms of the New BSD license:

  http://www.opensource.org/licenses/bsd-license.php

See the JSR-305 home page for more information:

http://code.google.com/p/jsr-305/

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---
com.google.errorprone » error_prone_annotations 2.3.4
COPYRIGHT: Copyright 2017 The Error Prone Authors.
LICENSE: Apache 2.0 https://github.com/google/error-prone/blob/v2.2.0/
COPYING

---
com.google.guava » failureaccess 1.0.1
(part of google guava 28.1-jre. same license and copyright)
```

```
---
com.google.guava » listenablefuture  9999.0-empty-...
(part of google guava 28.1-jre. same license and copyright)

---
com.google.j2objc » j2objc-annotations 1.3
COPYRIGHT: (C) Daniel Connelly and Google, Inc.
LICENSE: Apache 2.0 https://github.com/google/j2objc/blob/1.1/LICENSE

---
org.checkerframework » checker-qual 2.10 MIT license
COPYRIGHT: Copyright 2004-present by the Checker Framework developers

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
================================================================================
===
```

Jetty 9.4.28

```
This program and the accompanying materials are made available under the
terms of the Eclipse Public License 1.0 which is available at
https://www.eclipse.org/org/documents/epl-1.0/EPL-1.0.txt
or the Apache Software License 2.0 which is available at
https://www.apache.org/licenses/LICENSE-2.0

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE
PUBLIC
LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE
PROGRAM
CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS
```

```
"Contribution" means:

a) in the case of the initial Contributor, the initial code and
documentation
   distributed under this Agreement, and
b) in the case of each subsequent Contributor:
    i) changes to the Program, and
   ii) additions to the Program;

   where such changes and/or additions to the Program originate from
and are
   distributed by that particular Contributor. A Contribution
'originates'
   from a Contributor if it was added to the Program by such Contributor
   itself or anyone acting on such Contributor's behalf. Contributions
do not
   include additions to the Program which: (i) are separate modules of
   software distributed in conjunction with the Program under their own
   license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which
are
necessarily infringed by the use or sale of its Contribution alone or
when
combined with the Program.

"Program" means the Contributions distributed in accordance with this
Agreement.

"Recipient" means anyone who receives the Program under this Agreement,
including all Contributors.

2. GRANT OF RIGHTS
  a) Subject to the terms of this Agreement, each Contributor hereby
grants
     Recipient a non-exclusive, worldwide, royalty-free copyright
license to
     reproduce, prepare derivative works of, publicly display, publicly
     perform, distribute and sublicense the Contribution of such
Contributor,
     if any, and such derivative works, in source code and object code
form.
  b) Subject to the terms of this Agreement, each Contributor hereby
grants
     Recipient a non-exclusive, worldwide, royalty-free patent license
under
     Licensed Patents to make, use, sell, offer to sell, import and
otherwise
     transfer the Contribution of such Contributor, if any, in source
code and
     object code form. This patent license shall apply to the
combination of
     the Contribution and the Program if, at the time the Contribution
```

```
is
     added by the Contributor, such addition of the Contribution causes
such
     combination to be covered by the Licensed Patents. The patent
license
     shall not apply to any other combinations which include the
Contribution.
     No hardware per se is licensed hereunder.
  c) Recipient understands that although each Contributor grants the
licenses
     to its Contributions set forth herein, no assurances are provided
by any
     Contributor that the Program does not infringe the patent or other
     intellectual property rights of any other entity. Each Contributor
     disclaims any liability to Recipient for claims brought by any
other
     entity based on infringement of intellectual property rights or
     otherwise. As a condition to exercising the rights and licenses
granted
     hereunder, each Recipient hereby assumes sole responsibility to
secure
     any other intellectual property rights needed, if any. For
example, if a
     third party patent license is required to allow Recipient to
distribute
     the Program, it is Recipient's responsibility to acquire that
license
     before distributing the Program.
  d) Each Contributor represents that to its knowledge it has sufficient
     copyright rights in its Contribution, if any, to grant the
copyright
     license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form
under
its own license agreement, provided that:

  a) it complies with the terms and conditions of this Agreement; and
  b) its license agreement:
      i) effectively disclaims on behalf of all Contributors all
warranties
         and conditions, express and implied, including warranties or
         conditions of title and non-infringement, and implied
warranties or
         conditions of merchantability and fitness for a particular
purpose;
     ii) effectively excludes on behalf of all Contributors all
liability for
         damages, including direct, indirect, special, incidental and
         consequential damages, such as lost profits;
    iii) states that any provisions which differ from this Agreement are
         offered by that Contributor alone and not by any other party;
and
```

```
     iv) states that source code for the Program is available from such
         Contributor, and informs licensees how to obtain it in a
reasonable
         manner on or through a medium customarily used for software
exchange.

When the Program is made available in source code form:

  a) it must be made available under this Agreement; and
  b) a copy of this Agreement must be included with each copy of the
Program.
     Contributors may not remove or alter any copyright notices
contained
     within the Program.

Each Contributor must identify itself as the originator of its
Contribution,
if
any, in a manner that reasonably allows subsequent Recipients to
identify the
originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities
with
respect to end users, business partners and the like. While this
license is
intended to facilitate the commercial use of the Program, the
Contributor who
includes the Program in a commercial product offering should do so in a
manner
which does not create potential liability for other Contributors.
Therefore,
if a Contributor includes the Program in a commercial product offering,
such
Contributor ("Commercial Contributor") hereby agrees to defend and
indemnify
every other Contributor ("Indemnified Contributor") against any losses,
damages and costs (collectively "Losses") arising from claims, lawsuits
and
other legal actions brought by a third party against the Indemnified
Contributor to the extent caused by the acts or omissions of such
Commercial
Contributor in connection with its distribution of the Program in a
commercial
product offering. The obligations in this section do not apply to any
claims
or Losses relating to any actual or alleged intellectual property
infringement. In order to qualify, an Indemnified Contributor must:
a) promptly notify the Commercial Contributor in writing of such claim,
and
b) allow the Commercial Contributor to control, and cooperate with the
Commercial Contributor in, the defense and any related settlement
negotiations. The Indemnified Contributor may participate in any such
```

```
claim at
its own expense.

For example, a Contributor might include the Program in a commercial
product
offering, Product X. That Contributor is then a Commercial Contributor.
If
that Commercial Contributor then makes performance claims, or offers
warranties related to Product X, those performance claims and
warranties are
such Commercial Contributor's responsibility alone. Under this section,
the
Commercial Contributor would have to defend claims against the other
Contributors related to those performance claims and warranties, and if
a
court requires any other Contributor to pay any damages as a result, the
Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS
PROVIDED ON AN
"AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER
EXPRESS OR
IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF
TITLE,
NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
Each
Recipient is solely responsible for determining the appropriateness of
using
and distributing the Program and assumes all risks associated with its
exercise of rights under this Agreement , including but not limited to
the
risks and costs of program errors, compliance with applicable laws,
damage to
or loss of data, programs or equipment, and unavailability or
interruption of
operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR
ANY
CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT,
INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT
LIMITATION
LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE
EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE
POSSIBILITY
OF SUCH DAMAGES.

7. GENERAL
```

```
If any provision of this Agreement is invalid or unenforceable under
applicable law, it shall not affect the validity or enforceability of
the
remainder of the terms of this Agreement, and without further action by
the
parties hereto, such provision shall be reformed to the minimum extent
necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including
a
cross-claim or counterclaim in a lawsuit) alleging that the Program
itself
(excluding combinations of the Program with other software or hardware)
infringes such Recipient's patent(s), then such Recipient's rights
granted
under Section 2(b) shall terminate as of the date such litigation is
filed.

All Recipient's rights under this Agreement shall terminate if it fails
to
comply with any of the material terms or conditions of this Agreement
and does
not cure such failure in a reasonable period of time after becoming
aware of
such noncompliance. If all Recipient's rights under this Agreement
terminate,
Recipient agrees to cease use and distribution of the Program as soon as
reasonably practicable. However, Recipient's obligations under this
Agreement
and any licenses granted by Recipient relating to the Program shall
continue
and survive.

Everyone is permitted to copy and distribute copies of this Agreement,
but in
order to avoid inconsistency the Agreement is copyrighted and may only
be
modified in the following manner. The Agreement Steward reserves the
right to
publish new versions (including revisions) of this Agreement from time
to
time. No one other than the Agreement Steward has the right to modify
this
Agreement. The Eclipse Foundation is the initial Agreement Steward. The
Eclipse Foundation may assign the responsibility to serve as the
Agreement
Steward to a suitable separate entity. Each new version of the
Agreement will
be given a distinguishing version number. The Program (including
Contributions) may always be distributed subject to the version of the
Agreement under which it was received. In addition, after a new version
of the
Agreement is published, Contributor may elect to distribute the Program
(including its Contributions) under the new version. Except as expressly
```

stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement,
whether expressly, by implication, estoppel or otherwise. All rights in the
Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this
Agreement will bring a legal action under this Agreement more than one year
after the cause of action arose. Each party waives its rights to a jury trial in
any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

```
      "Work" shall mean the work of authorship, whether in Source or
      Object form, made available under the License, as indicated by a
      copyright notice that is included in or attached to the work
      (an example is provided in the Appendix below).

      "Derivative Works" shall mean any work, whether in Source or
Object
      form, that is based on (or derived from) the Work and for which
the
      editorial revisions, annotations, elaborations, or other
modifications
      represent, as a whole, an original work of authorship. For the
purposes
      of this License, Derivative Works shall not include works that
remain
      separable from, or merely link (or bind by name) to the
interfaces of,
      the Work and Derivative Works thereof.

      "Contribution" shall mean any work of authorship, including
      the original version of the Work and any modifications or
additions
      to that Work or Derivative Works thereof, that is intentionally
      submitted to Licensor for inclusion in the Work by the copyright
owner
      or by an individual or Legal Entity authorized to submit on
behalf of
      the copyright owner. For the purposes of this definition,
"submitted"
      means any form of electronic, verbal, or written communication
sent
      to the Licensor or its representatives, including but not limited
to
      communication on electronic mailing lists, source code control
systems,
      and issue tracking systems that are managed by, or on behalf of,
the
      Licensor for the purpose of discussing and improving the Work, but
      excluding communication that is conspicuously marked or otherwise
      designated in writing by the copyright owner as "Not a
Contribution."

      "Contributor" shall mean Licensor and any individual or Legal
Entity
      on behalf of whom a Contribution has been received by Licensor and
      subsequently incorporated within the Work.

   2. Grant of Copyright License. Subject to the terms and conditions of
      this License, each Contributor hereby grants to You a perpetual,
      worldwide, non-exclusive, no-charge, royalty-free, irrevocable
      copyright license to reproduce, prepare Derivative Works of,
      publicly display, publicly perform, sublicense, and distribute the
      Work and such Derivative Works in Source or Object form.
```

```
   3. Grant of Patent License. Subject to the terms and conditions of
      this License, each Contributor hereby grants to You a perpetual,
      worldwide, non-exclusive, no-charge, royalty-free, irrevocable
      (except as stated in this section) patent license to make, have
made,
      use, offer to sell, sell, import, and otherwise transfer the Work,
      where such license applies only to those patent claims licensable
      by such Contributor that are necessarily infringed by their
      Contribution(s) alone or by combination of their Contribution(s)
      with the Work to which such Contribution(s) was submitted. If You
      institute patent litigation against any entity (including a
      cross-claim or counterclaim in a lawsuit) alleging that the Work
      or a Contribution incorporated within the Work constitutes direct
      or contributory patent infringement, then any patent licenses
      granted to You under this License for that Work shall terminate
      as of the date such litigation is filed.

   4. Redistribution. You may reproduce and distribute copies of the
      Work or Derivative Works thereof in any medium, with or without
      modifications, and in Source or Object form, provided that You
      meet the following conditions:

      (a) You must give any other recipients of the Work or
          Derivative Works a copy of this License; and

      (b) You must cause any modified files to carry prominent notices
          stating that You changed the files; and

      (c) You must retain, in the Source form of any Derivative Works
          that You distribute, all copyright, patent, trademark, and
          attribution notices from the Source form of the Work,
          excluding those notices that do not pertain to any part of
          the Derivative Works; and

      (d) If the Work includes a "NOTICE" text file as part of its
          distribution, then any Derivative Works that You distribute
must
          include a readable copy of the attribution notices contained
          within such NOTICE file, excluding those notices that do not
          pertain to any part of the Derivative Works, in at least one
          of the following places: within a NOTICE text file distributed
          as part of the Derivative Works; within the Source form or
          documentation, if provided along with the Derivative Works;
or,
          within a display generated by the Derivative Works, if and
          wherever such third-party notices normally appear. The
contents
          of the NOTICE file are for informational purposes only and
          do not modify the License. You may add Your own attribution
          notices within Derivative Works that You distribute, alongside
          or as an addendum to the NOTICE text from the Work, provided
          that such additional attribution notices cannot be construed
          as modifying the License.

      You may add Your own copyright statement to Your modifications and
```

```
      may provide additional or different license terms and conditions
      for use, reproduction, or distribution of Your modifications, or
      for any such Derivative Works as a whole, provided Your use,
      reproduction, and distribution of the Work otherwise complies with
      the conditions stated in this License.

   5. Submission of Contributions. Unless You explicitly state
otherwise,
      any Contribution intentionally submitted for inclusion in the Work
      by You to the Licensor shall be under the terms and conditions of
      this License, without any additional terms or conditions.
      Notwithstanding the above, nothing herein shall supersede or
modify
      the terms of any separate license agreement you may have executed
      with Licensor regarding such Contributions.

   6. Trademarks. This License does not grant permission to use the
trade
      names, trademarks, service marks, or product names of the
Licensor,
      except as required for reasonable and customary use in describing
the
      origin of the Work and reproducing the content of the NOTICE file.

   7. Disclaimer of Warranty. Unless required by applicable law or
      agreed to in writing, Licensor provides the Work (and each
      Contributor provides its Contributions) on an "AS IS" BASIS,
      WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
      implied, including, without limitation, any warranties or
conditions
      of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
      PARTICULAR PURPOSE. You are solely responsible for determining the
      appropriateness of using or redistributing the Work and assume any
      risks associated with Your exercise of permissions under this
License.

   8. Limitation of Liability. In no event and under no legal theory,
      whether in tort (including negligence), contract, or otherwise,
      unless required by applicable law (such as deliberate and grossly
      negligent acts) or agreed to in writing, shall any Contributor be
      liable to You for damages, including any direct, indirect,
special,
      incidental, or consequential damages of any character arising as a
      result of this License or out of the use or inability to use the
      Work (including but not limited to damages for loss of goodwill,
      work stoppage, computer failure or malfunction, or any and all
      other commercial damages or losses), even if such Contributor
      has been advised of the possibility of such damages.

   9. Accepting Warranty or Additional Liability. While redistributing
      the Work or Derivative Works thereof, You may choose to offer,
      and charge a fee for, acceptance of support, warranty, indemnity,
      or other liability obligations and/or rights consistent with this
      License. However, in accepting such obligations, You may act only
      on Your own behalf and on Your sole responsibility, not on behalf
```

```
      of any other Contributor, and only if You agree to indemnify,
      defend, and hold each Contributor harmless for any liability
      incurred by, or claims asserted against, such Contributor by
reason
      of your accepting any such warranty or additional liability.

   END OF TERMS AND CONDITIONS

   APPENDIX: How to apply the Apache License to your work.

      To apply the Apache License to your work, attach the following
      boilerplate notice, with the fields enclosed by brackets "[]"
      replaced with your own identifying information. (Don't include
      the brackets!)  The text should be enclosed in the appropriate
      comment syntax for the file format. We also recommend that a
      file or class name and description of purpose be included on the
      same "printed page" as the copyright notice for easier
      identification within third-party archives.

   Copyright [yyyy] [name of copyright owner]

   Licensed under the Apache License, Version 2.0 (the "License");
   you may not use this file except in compliance with the License.
   You may obtain a copy of the License at

       http://www.apache.org/licenses/LICENSE-2.0

   Unless required by applicable law or agreed to in writing, software
   distributed under the License is distributed on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
   See the License for the specific language governing permissions and
limitations under the License.

Notice:
=============================================================
 Jetty Web Container
 Copyright 1995-2019 Mort Bay Consulting Pty Ltd.
=============================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
unless otherwise noted.

Jetty is dual licensed under both

  * The Apache 2.0 License
    http://www.apache.org/licenses/LICENSE-2.0.html

      and

  * The Eclipse Public 1.0 License
    http://www.eclipse.org/legal/epl-v10.html

Jetty may be distributed under either license.
```

```
------
Eclipse

The following artifacts are EPL.
 * org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
 * org.eclipse.jetty.orbit:javax.security.auth.message


The following artifacts are EPL and CDDL 1.0.
 * org.eclipse.jetty.orbit:javax.mail.glassfish


------
OW2

The following artifacts are licensed by the OW2 Foundation according to
the
terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm


------
Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl


------
MortBay

The following artifacts are ASL2 licensed.  Based on selected classes
from
following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
  org.apache.tomcat:tomcat-jasper
  org.apache.tomcat:tomcat-juli
  org.apache.tomcat:tomcat-jsp-api
  org.apache.tomcat:tomcat-el-api
  org.apache.tomcat:tomcat-jasper-el
  org.apache.tomcat:tomcat-api
  org.apache.tomcat:tomcat-util-scan
  org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
  org.apache.tomcat:tomcat-jasper-el
  org.apache.tomcat:tomcat-el-api
```

```
------
Assorted

The UnixCrypt.java code implements the one way cryptography used by
Unix systems for simple password protection.  Copyright 1996 Aki
Yoshida,
modified April 2001  by Iris Van den Broeke, Daniel Deville.
Permission to use, copy, modify and distribute UnixCrypt
for non-commercial or commercial purposes and without fee is
granted provided that the copyright notice appears in all copies.


**************
FROM POM FILE:


org.ow2.asm asm / asm-common / asm-tree / asm-analysis 7.0
COPYRIGHT and LICENSE:
ASM: a very small and fast Java bytecode manipulation framework
Copyright (c) 2000-2011 INRIA, France Telecom
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
  documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its
  contributors may be used to endorse or promote products derived from
  this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.


---
org.eclipse.jetty.toolchain jetty-schema 3.1.2

COPYRIGHT: Copyright 1995-2018 Mort Bay Consulting Pty Ltd.
```

```
LICENSE: Apache Software License - Version 2.0 + Eclipse Public License
- Version 1.0


---
org.eclipse.jetty.orbit javax.security.auth.message 1.0.0.v201108011116
org.eclipse.jetty.orbit javax.activation 1.1.0.v201105071233
COPYRIGHT: Mort Bay Consulting
LICENSE: Apache Software License - Version 2.0 + Eclipse Public License
- Version 1.0


---
org.mortbay.jasper apache-jsp 8.5.40
COPYRIGHT: Mort Bay Consulting
LICENSE: Apache 2.0 https://github.com/jetty-project/jasper-jsp/blob/
jasper-jsp-8.5.40/LICENSE


---
org.apache.taglibs taglibs-standard-imp 1.2.5
org.apache.taglibs taglibs-standard-spec 1.2.5
COPYRIGHT: Copyright © 2001-2015 The Apache Software Foundation. All
Rights Reserved.
LICENSE: Apache 2.0 https://www.apache.org/licenses/LICENSE-2.0


---
org.eclipse.jetty.toolchain jetty-perf-helper 1.0.5
COPYRIGHT and LICENSE:

//  Copyright (c) 1995-2014 Mort Bay Consulting Pty. Ltd.
//
------------------------------------------------------------------------
//  All rights reserved. This program and the accompanying materials
//  are made available under the terms of the Eclipse Public License
v1.0
//  and Apache License v2.0 which accompanies this distribution.
//
//      The Eclipse Public License is available at
//      http://www.eclipse.org/legal/epl-v10.html
//
//      The Apache License v2.0 is available at
//      http://www.opensource.org/licenses/apache2.0.php
//
//  You may elect to redistribute this code under either of these
licenses.



---
org.slf4j jcl104-over-slf4j 1.7.25
org.slf4j log4j-over-slf4j 1.7.25
org.slf4j slf4j-api 1.7.25
```

```
org.slf4j slf4j-simple 1.7.25
COPYRIGHT and LICENSE:
Copyright (c) 2004-2017 QOS.ch
All rights reserved.

Permission is hereby granted, free  of charge, to any person obtaining
a  copy  of this  software  and  associated  documentation files  (the
"Software"), to  deal in  the Software without  restriction, including
without limitation  the rights to  use, copy, modify,  merge, publish,
distribute,  sublicense, and/or sell  copies of  the Software,  and to
permit persons to whom the Software  is furnished to do so, subject to
the following conditions:

The  above  copyright  notice  and  this permission  notice  shall  be
included in all copies or substantial portions of the Software.

THE  SOFTWARE IS  PROVIDED  "AS  IS", WITHOUT  WARRANTY  OF ANY  KIND,
EXPRESS OR  IMPLIED, INCLUDING  BUT NOT LIMITED  TO THE  WARRANTIES OF
MERCHANTABILITY,    FITNESS    FOR    A   PARTICULAR    PURPOSE    AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE,  ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



---
org.jboss.logging 3.3.2.Final
COPYRIGHT: Copyright 2010 Red Hat, Inc.
LICENSE: Apache 2.0 https://github.com/jboss-logging/jboss-logging/blob/
3.3.2.Final/src/main/resources/META-INF/LICENSE.txt


---
com.github.jnr jnr-unixsocket 0.22

COPYRIGHT:
Copyright (C) 2016 Fritz Elfert
Copyright (C) 2009 Wayne Meissner
Copyright (C) 2016 Marcus Linke
Copyright (C) 2014 Greg Vanore

LICENSE: Apache 2.0

  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
  See the License for the specific language governing permissions and
```

```
limitations under the License.



---
org.eclipse.jetty.alpn » alpn-api 1.1.3.v20160715
COPYRIGHT and LICENSE:
//  Copyright (c) 1995-2015 Mort Bay Consulting Pty. Ltd.
//
------------------------------------------------------------------------
//  All rights reserved. This program and the accompanying materials
//  are made available under the terms of the Eclipse Public License
v1.0
//  and Apache License v2.0 which accompanies this distribution.
//
//      The Eclipse Public License is available at
//      http://www.eclipse.org/legal/epl-v10.html
//
//      The Apache License v2.0 is available at
//      http://www.opensource.org/licenses/apache2.0.php
//
//  You may elect to redistribute this code under either of these
licenses.



---
org.infinispan » infinispan-core  9.4.8.Final
org.infinispan » infinispan-query 9.4.8.Final
org.infinispan » infinispan-client-hotrod 9.4.8.Final
org.infinispan » infinispan-remote-query-client 9.4.8.Final
Copyright The Infinispan Project Team
LICENSE: Apache 2.0 https://github.com/infinispan/infinispan/blob/
9.4.8.Final/LICENSE.md


---
org.conscrypt » conscrypt-openjdk-uber v2.1.0
COPYRIGHT and LICENSE:
Copyright 2016 The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

     http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```

```
---
org.mongodb » mongo-java-driver 2.13.2
COPYRIGHT: Copyright 2008-present MongoDB, Inc.
LICENSE: Apache 2.0 https://github.com/mongodb/mongo-java-driver/blob/
r2.13.2/LICENSE.txt

NOTICE FILE:
The MongoDB Java Driver uses third-party libraries or other resources
that may
be distributed under licenses different than the MongoDB Java Driver
software.

In the event that we accidentally failed to list a required notice,
please bring it to our attention through any of the ways detailed here:

           mongodb-dev@googlegroups.com

The attached notices are provided for information only.

For any licenses that require disclosure of source, sources are
available at
https://github.com/mongodb/mongo-java-driver.


1) The following files: Immutable.java, NotThreadSafe.java,
ThreadSafe.java

   Copyright (c) 2005 Brian Goetz and Tim Peierls
   Released under the Creative Commons Attribution License (http://
creativecommons.org/licenses/by/2.5)
   Official home: http://www.jcip.net

   Any republication or derived work distributed in source code form
   must include this copyright and license notice.

2) The following files: Assertions.java, AbstractCopyOnWriteMap.java,
CopyOnWriteMap.java

    Copyright (c) 2008-2014 Atlassian Pty Ltd

    Licensed under the Apache License, Version 2.0 (the "License");
    you may not use this file except in compliance with the License.
    You may obtain a copy of the License at

        http://www.apache.org/licenses/LICENSE-2.0

    Unless required by applicable law or agreed to in writing, software
    distributed under the License is distributed on an "AS IS" BASIS,
    WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
    See the License for the specific language governing permissions and
    limitations under the License.

3) The following files: Beta.java, UnsignedLongs.java,
UnsignedLongsTest.java
```

```
    Copyright 2010 The Guava Authors
    Copyright 2011 The Guava Authors

    Licensed under the Apache License, Version 2.0 (the "License");
    you may not use this file except in compliance with the License.
    You may obtain a copy of the License at

        http://www.apache.org/licenses/LICENSE-2.0

    Unless required by applicable law or agreed to in writing, software
    distributed under the License is distributed on an "AS IS" BASIS,
    WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
    See the License for the specific language governing permissions and
    limitations under the License.

4) The following files: ReadTimeoutHandler.java

    Copyright 2008-present MongoDB, Inc.
    Copyright 2012 The Netty Project

    Licensed under the Apache License, Version 2.0 (the "License");
    you may not use this file except in compliance with the License.
    You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

    Unless required by applicable law or agreed to in writing, software
    distributed under the License is distributed on an "AS IS" BASIS,
    WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
    See the License for the specific language governing permissions and
    limitations under the License.

5) The following files: InstantCodec.java, Jsr310CodecProvider.java,
LocalDateCodec.java, LocalDateTimeCodec.java, LocalTimeCodec.java

    Copyright 2008-present MongoDB, Inc.
    Copyright 2018 Cezary Bartosiak

    Licensed under the Apache License, Version 2.0 (the "License");
    you may not use this file except in compliance with the License.
    You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

    Unless required by applicable law or agreed to in writing, software
    distributed under the License is distributed on an "AS IS" BASIS,
    WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
    See the License for the specific language governing permissions and
    limitations under the License.

6) The following files: SaslPrep.java
```

```
     Copyright 2008-present MongoDB, Inc.
     Copyright 2017 Tom Bentley

     Licensed under the Apache License, Version 2.0 (the "License");
     you may not use this file except in compliance with the License.
     You may obtain a copy of the License at

       http://www.apache.org/licenses/LICENSE-2.0

     Unless required by applicable law or agreed to in writing, software
     distributed under the License is distributed on an "AS IS" BASIS,
     WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
     See the License for the specific language governing permissions and
     limitations under the License.

7) The following files (originally from https://github.com/
marianobarrios/tls-channel):

       AsynchronousTlsChannel.java
       AsynchronousTlsChannelGroup.java
       ExtendedAsynchronousByteChannel.java
       BufferHolder.java
       ByteBufferSet.java
       ByteBufferUtil.java
       TlsChannelImpl.java
       TlsChannelCallbackException.java
       Util.java
       BufferAllocator.java
       ClientTlsChannel.java
       NeedsReadException.java
       NeedsTaskException.java
       NeedsWriteException.java
       TlsChannel.java
       TlsChannelBuilder.java
       TlsChannelFlowControlException.java
       TrackingAllocator.java
       WouldBlockException.java

     Copyright (c) [2015-2018] all contributors

     MIT License

     Permission is hereby granted, free of charge, to any person
obtaining a copy
     of this software and associated documentation files (the
"Software"), to deal
     in the Software without restriction, including without limitation
the rights
     to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell
     copies of the Software, and to permit persons to whom the Software
is
     furnished to do so, subject to the following conditions:
```

```
     The above copyright notice and this permission notice shall be
included in all
     copies or substantial portions of the Software.

     THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR
     IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY,
     FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT
SHALL THE
     AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER
     LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM,
     OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE
SOFTWARE.


---
org.springframework » spring-beans 3.2.8.RELEASE
COPYRIGHT: Copyright (c) 2002-2014 Pivotal, Inc.
LICENSE: Apache 2.0 The Spring Framework is released under version 2.0
of the Apache License.

---
org.jboss.logging » jboss-logging 3.3.2.Final
COPYRIGHT: Copyright 2010 Red Hat, Inc.
LICENSE: Apache 2.0 https://github.com/jboss-logging/jboss-logging/blob/
3.3.2.Final/src/main/resources/META-INF/LICENSE.txt


---
com.hazelcast » hazelcast 3.9.4
com.hazelcast » hazelcast-client 3.9.4
LICENSE: Apache 2.0 https://github.com/hazelcast/hazelcast/blob/v3.9.4/
LICENSE
NOTICE FILE:
The packages:

com.hazelcast.util.collection
com.hazelcast.internal.util.concurrent

and the classes:

com.hazelcast.util.QuickMath
com.hazelcast.client.impl.protocol.util.UnsafeBuffer
com.hazelcast.client.impl.protocol.util.BufferBuilder

contain code originating from the Agrona project
(https://github.com/real-logic/Agrona).

The class com.hazelcast.util.HashUtil contains code originating
from the Koloboke project (https://github.com/OpenHFT/Koloboke).
```

```
The class classloading.ThreadLocalLeakTestUtils contains code
originating
from the Tomcat project (https://github.com/apache/tomcat).

com.hazelcast.internal.cluster.fd.PhiAccrualFailureDetector contains
code originating
from the Akka project (https://github.com/akka/akka/).


---
ant » ant 1.6.5
ant » ant-launcher 1.6.5
COPYRIGHT:  Copyright 1999-2012 The Apache Software Foundation

LICENSE: Aapche 2.0 https://github.com/apache/ant/blob/rel/1.6.5/LICENSE

NOTICE FILE:
   This product includes software developed by
   The Apache Software Foundation (http://www.apache.org/).

   This product includes also software developed by :
     - the W3C consortium (http://www.w3c.org) ,
     - the SAX project (http://www.saxproject.org)

   Please read the different LICENSE files present in the root
directory of
this distribution.



---
org.apache.ant » ant 1.8.4
COPYRIGHT:  Copyright 1999-2012 The Apache Software Foundation
LICENSE: Apache 2.0 https://github.com/apache/ant/blob/rel/1.8.4/LICENSE

===
The  task is based on code Copyright (c) 2002, Landmark
Graphics Corp that has been kindly donated to the Apache Software
Foundation.


===
W3C® SOFTWARE NOTICE AND LICENSE
http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231

This work (and included software, documentation such as READMEs, or
other
related items) is being provided by the copyright holders under the
following
license. By obtaining, using and/or copying this work, you (the
licensee) agree
that you have read, understood, and will comply with the following
terms and
conditions.
```

```
Permission to copy, modify, and distribute this software and its
documentation,
with or without modification, for any purpose and without fee or
royalty is
hereby granted, provided that you include the following on ALL copies
of the
software and documentation or portions thereof, including modifications:

  1. The full text of this NOTICE in a location viewable to users of the
     redistributed or derivative work.
  2. Any pre-existing intellectual property disclaimers, notices, or
terms
     and conditions. If none exist, the W3C Software Short Notice
should be
     included (hypertext is preferred, text is permitted) within the
body
     of any redistributed or derivative code.
  3. Notice of any changes or modifications to the files, including the
date
     changes were made. (We recommend you provide URIs to the location
from
     which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT
HOLDERS MAKE
NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT
LIMITED
TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE
OR THAT
THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD
PARTY
PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL
OR
CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR
DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in
advertising or
publicity pertaining to the software without specific, written prior
permission.
Title to copyright in this software and any associated documentation
will at
all times remain with copyright holders.

_____

This formulation of W3C's notice and license became active on December
31 2002.
This version removes the copyright ownership notice such that this
license can
be used with materials other than those owned by the W3C, reflects that
ERCIM
```

```
is now a host of the W3C, includes references to this specific dated
version of
the license, and removes the ambiguous grant of "use". Otherwise, this
version
is the same as the previous version and is written so as to preserve
the Free
Software Foundation's assessment of GPL compatibility and OSI's
certification
under the Open Source Definition. Please see our Copyright FAQ for
common
questions about using materials from our site, including specific terms
and
conditions for packages like libwww, Amaya, and Jigsaw. Other questions
about
this notice can be directed to site-policy@w3.org.

Joseph Reagle

This license came from: http://www.megginson.com/SAX/copying.html
  However please note future versions of SAX may be covered
  under http://saxproject.org/?selected=pd

SAX2 is Free!

I hereby abandon any property rights to SAX 2.0 (the Simple API for
XML), and release all of the SAX 2.0 source code, compiled code, and
documentation contained in this distribution into the Public Domain.
SAX comes with NO WARRANTY or guarantee of fitness for any
purpose.

David Megginson, david@megginson.com
2000-05-05

--------------------------------------------------------------------------
-------------
LICENSES per Deps:
==========================================================================
=============
    BSD 3-Clause "New" or "Revised" License (BSD-3-Clause):

        * API Common (com.google.api:api-common:1.0.0 - https://
github.com/googleapis)
        * GAX (Google Api eXtensions) (com.google.api:gax:1.0.0 -
https://github.com/googleapis)
        * ParaNamer Core (com.thoughtworks.paranamer:paranamer:2.3 -
http://paranamer.codehaus.org/paranamer)
        * io.grpc:grpc-all (io.grpc:grpc-all:1.0.1 - https://github.com/
grpc/grpc-java)
        * io.grpc:grpc-auth (io.grpc:grpc-auth:1.0.1 - https://
github.com/grpc/grpc-java)
        * io.grpc:grpc-context (io.grpc:grpc-context:1.0.1 - https://
github.com/grpc/grpc-java)
        * io.grpc:grpc-core (io.grpc:grpc-core:1.0.1 - https://
github.com/grpc/grpc-java)
        * io.grpc:grpc-netty (io.grpc:grpc-netty:1.0.1 - https://
```

```
github.com/grpc/grpc-java)
        * io.grpc:grpc-okhttp (io.grpc:grpc-okhttp:1.0.1 - https://
github.com/grpc/grpc-java)
        * io.grpc:grpc-protobuf (io.grpc:grpc-protobuf:1.0.1 - https://
github.com/grpc/grpc-java)
        * io.grpc:grpc-protobuf-lite (io.grpc:grpc-protobuf-lite:1.0.1
- https://github.com/grpc/grpc-java)
        * io.grpc:grpc-protobuf-nano (io.grpc:grpc-protobuf-nano:1.0.1
- https://github.com/grpc/grpc-java)
        * io.grpc:grpc-stub (io.grpc:grpc-stub:1.0.1 - https://
github.com/grpc/grpc-java)
        * ANTLR 3 Runtime (org.antlr:antlr-runtime:3.5.2 - http://
www.antlr.org)
        * Hamcrest (org.hamcrest:hamcrest:2.1 - http://hamcrest.org/
JavaHamcrest/)
        * asm (org.ow2.asm:asm:7.1 - http://asm.ow2.org/)
        * asm-analysis (org.ow2.asm:asm-analysis:7.1 - http://
asm.ow2.org/)
        * asm-commons (org.ow2.asm:asm-commons:7.1 - http://
asm.ow2.org/)
        * asm-tree (org.ow2.asm:asm-tree:7.1 - http://asm.ow2.org/)
        * ASM Util (org.ow2.asm:asm-util:5.0.3 - http://
asm.objectweb.org/asm-util/)
        * ThreeTen backport (org.threeten:threetenbp:1.3.3 - https://
www.threeten.org/threetenbp)
        * Google Auth Library for Java - Credentials
(com.google.auth:google-auth-library-credentials:0.6.1 - https://
github.com/google/google-auth-library-java/google-auth-library-
credentials)
        * Google Auth Library for Java - OAuth2 HTTP
(com.google.auth:google-auth-library-oauth2-http:0.6.1 - https://
github.com/google/google-auth-library-java/google-auth-library-oauth2-
http)
        * Protocol Buffers [Core] (com.google.protobuf:protobuf-
java:3.0.2 - https://developers.google.com/protocol-buffers/protobuf-
java/)
        * Protocol Buffers [Core] (com.google.protobuf:protobuf-
java:3.2.0 - https://developers.google.com/protocol-buffers/protobuf-
java/)
        * Protocol Buffers [Util] (com.google.protobuf:protobuf-java-
util:3.2.0 - https://developers.google.com/protocol-buffers/protobuf-
java-util/)
        * Protocol Buffers [Lite] (com.google.protobuf:protobuf-
lite:3.0.1 - https://developers.google.com/protocol-buffers/protobuf-
lite/)
        * Protocol Buffer JavaNano API
(com.google.protobuf.nano:protobuf-javanano:3.0.0-alpha-5 - https://
developers.google.com/protocol-buffers/)

Note: This license has also been called the "New BSD License" or
"Modified BSD License". See also the 2-clause BSD License.

Copyright

Redistribution and use in source and binary forms, with or without
```

```
modification, are permitted provided that the following conditions are
met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

================================================================================
======

    Eclipse Public License - Version 1.0:

        * Http Services Servlet (org.eclipse.equinox.http:servlet:1.0.0-
v20070606 - no url defined)
        * Jetty :: Apache JSP Implementation (org.eclipse.jetty:apache-
jsp:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Apache :: JSTL module (org.eclipse.jetty:apache-
jstl:9.4.21.v20190916 - http://tomcat.apache.org/taglibs/standard/)
        * Jetty :: Infinispan Session Manager Common
(org.eclipse.jetty:infinispan-common:9.4.21.v20190916 - http://
www.eclipse.org/jetty)
        * Jetty :: Infinispan Session Manager Remote
(org.eclipse.jetty:infinispan-remote-query:9.4.21.v20190916 - http://
www.eclipse.org/jetty)
        * Jetty :: ALPN :: Client (org.eclipse.jetty:jetty-alpn-
client:9.4.21.v20190916 - https://eclipse.org/jetty/jetty-alpn-parent/
jetty-alpn-client)
        * Jetty :: ALPN :: Conscrypt Server Implementation
(org.eclipse.jetty:jetty-alpn-conscrypt-server:9.4.21.v20190916 -
https://eclipse.org/jetty/jetty-alpn-parent/jetty-alpn-conscrypt-server)
        * Jetty :: ALPN :: OpenJDK8 Client Implementation
(org.eclipse.jetty:jetty-alpn-openjdk8-client:9.4.21.v20190916 -
https://eclipse.org/jetty/jetty-alpn-parent/jetty-alpn-openjdk8-client)
        * Jetty :: ALPN :: OpenJDK8 Server Implementation
(org.eclipse.jetty:jetty-alpn-openjdk8-server:9.4.21.v20190916 -
https://eclipse.org/jetty/jetty-alpn-parent/jetty-alpn-openjdk8-server)
```

```
        * Jetty :: ALPN :: Server (org.eclipse.jetty:jetty-alpn-
server:9.4.21.v20190916 - https://eclipse.org/jetty/jetty-alpn-parent/
jetty-alpn-server)
        * Jetty :: Servlet Annotations (org.eclipse.jetty:jetty-
annotations:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: Asynchronous HTTP Client (org.eclipse.jetty:jetty-
client:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: Continuation (org.eclipse.jetty:jetty-
continuation:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: Deployers (org.eclipse.jetty:jetty-
deploy:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: Distribution Assemblies (org.eclipse.jetty:jetty-
distribution:9.4.21.v20190916 - https://eclipse.org/jetty/jetty-
distribution)
        * Jetty :: Hazelcast Session Manager (org.eclipse.jetty:jetty-
hazelcast:9.4.21.v20190916 - https://eclipse.org/jetty/jetty-hazelcast)
        * Jetty :: Http Utility (org.eclipse.jetty:jetty-
http:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: IO Utility (org.eclipse.jetty:jetty-
io:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: JAAS (org.eclipse.jetty:jetty-jaas:9.4.21.v20190916
- https://eclipse.org/jetty/jetty-jaas)
        * Jetty :: JMX Management (org.eclipse.jetty:jetty-
jmx:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: JNDI Naming (org.eclipse.jetty:jetty-
jndi:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: NoSQL Session Managers (org.eclipse.jetty:jetty-
nosql:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: Plus (org.eclipse.jetty:jetty-plus:9.4.21.v20190916
- http://www.eclipse.org/jetty)
        * Jetty :: Proxy (org.eclipse.jetty:jetty-
proxy:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: Quick Start (org.eclipse.jetty:jetty-
quickstart:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: Rewrite Handler (org.eclipse.jetty:jetty-
rewrite:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: Security (org.eclipse.jetty:jetty-
security:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: Server Core (org.eclipse.jetty:jetty-
server:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: Servlet Handling (org.eclipse.jetty:jetty-
servlet:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: Utility Servlets and Filters
(org.eclipse.jetty:jetty-servlets:9.4.21.v20190916 - http://
www.eclipse.org/jetty)
        * Jetty :: Utilities (org.eclipse.jetty:jetty-util:9.3.0.RC0 -
http://www.eclipse.org/jetty)
        * Jetty :: Utilities (org.eclipse.jetty:jetty-
util:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: Utilities :: Ajax(JSON) (org.eclipse.jetty:jetty-
util-ajax:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: Webapp Application Support (org.eclipse.jetty:jetty-
webapp:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: XML utilities (org.eclipse.jetty:jetty-
xml:9.4.21.v20190916 - http://www.eclipse.org/jetty)
```

```
        * Test :: Jetty Test Webapp (org.eclipse.jetty:test-jetty-
webapp:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: ALPN :: API (org.eclipse.jetty.alpn:alpn-
api:1.1.3.v20160715 - http://www.eclipse.org/jetty/alpn-api)
        * Example Async Rest :: Jar (org.eclipse.jetty.example-async-
rest:example-async-rest-jar:9.4.21.v20190916 - https://eclipse.org/
jetty/examples-parent/example-async-rest/example-async-rest-jar)
        * Jetty :: FastCGI :: Client (org.eclipse.jetty.fcgi:fcgi-
client:9.4.21.v20190916 - https://eclipse.org/jetty/fcgi-parent/fcgi-
client)
        * Jetty :: GCloud :: Session Manager
(org.eclipse.jetty.gcloud:jetty-gcloud-session-manager:9.4.21.v20190916
- https://eclipse.org/jetty/gcloud-parent/jetty-gcloud-session-manager)
        * Jetty :: HTTP2 :: Client (org.eclipse.jetty.http2:http2-
client:9.4.21.v20190916 - https://eclipse.org/jetty/http2-parent/http2-
client)
        * Jetty :: HTTP2 :: Common (org.eclipse.jetty.http2:http2-
common:9.4.21.v20190916 - https://eclipse.org/jetty/http2-parent/http2-
common)
        * Jetty :: HTTP2 :: HPACK (org.eclipse.jetty.http2:http2-
hpack:9.4.21.v20190916 - https://eclipse.org/jetty/http2-parent/http2-
hpack)
        * Jetty :: HTTP2 :: Server (org.eclipse.jetty.http2:http2-
server:9.4.21.v20190916 - https://eclipse.org/jetty/http2-parent/http2-
server)
        * Jetty :: Memcached :: Sessions
(org.eclipse.jetty.memcached:jetty-memcached-sessions:9.4.21.v20190916
- https://eclipse.org/jetty/memcached-parent/jetty-memcached-sessions)
        * Jetty Orbit :: Activation
(org.eclipse.jetty.orbit:javax.activation:1.1.0.v201105071233 - http://
www.eclipse.org/jetty/jetty-orbit/javax.activation)
        * Jetty Orbit :: Glassfish Mail
(org.eclipse.jetty.orbit:javax.mail.glassfish:1.4.1.v201005082020 -
http://www.eclipse.org/jetty/jetty-orbit/javax.mail.glassfish)
        * Jetty Orbit :: JASPI API
(org.eclipse.jetty.orbit:javax.security.auth.message:1.0.0.v201108011116
 - http://www.eclipse.org/jetty/jetty-orbit/javax.security.auth.message)
        * Jetty Orbit :: JSP JSTL
(org.eclipse.jetty.orbit:javax.servlet.jsp.jstl:1.2.0.v201105211821 -
http://www.eclipse.org/jetty/jetty-orbit/javax.servlet.jsp.jstl)
        * Jetty :: OSGi :: Boot (org.eclipse.jetty.osgi:jetty-osgi-
boot:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty Tests :: JMX :: WebApp (org.eclipse.jetty.tests:jmx-
webapp:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Test :: CDI :: Common Demo Webapp
(org.eclipse.jetty.tests:test-cdi-common-webapp:9.4.21.v20190916 -
http://www.eclipse.org/jetty/test-cdi-common-webapp)
        * Jetty Tests :: WebApp :: Servlet Spec ::
ServletContainerInitializer Test Jar (org.eclipse.jetty.tests:test-
container-initializer:9.4.21.v20190916 - http://www.eclipse.org/jetty/
test-servlet-spec-parent/test-container-initializer)
        * Jetty Tests :: WebApp :: JNDI (org.eclipse.jetty.tests:test-
jndi-webapp:9.4.21.v20190916 - http://www.eclipse.org/jetty/test-jndi-
webapp)
        * Jetty Tests :: WebApp :: Mock Resources
```

```
(org.eclipse.jetty.tests:test-mock-resources:9.4.21.v20190916 - http://
www.eclipse.org/jetty/test-mock-resources)
        * Jetty Tests :: Sessions :: Common
(org.eclipse.jetty.tests:test-sessions-common:9.4.21.v20190916 - http://
www.eclipse.org/jetty)
        * Jetty Tests :: Webapps :: Spec Webapp
(org.eclipse.jetty.tests:test-spec-webapp:9.4.21.v20190916 - http://
www.eclipse.org/jetty/test-servlet-spec-parent/test-spec-webapp)
        * Jetty Tests :: WebApp :: Servlet Spec :: Fragment Jar
(org.eclipse.jetty.tests:test-web-fragment:9.4.21.v20190916 - http://
www.eclipse.org/jetty/test-servlet-spec-parent/test-web-fragment)
        * Jetty :: Servlet API and Schemas for OSGi
(org.eclipse.jetty.toolchain:jetty-osgi-servlet-api:3.1.0.M3 - http://
www.eclipse.org/jetty/jetty-toolchain/jetty-osgi-servlet-api)
        * Jetty :: Schemas (org.eclipse.jetty.toolchain:jetty-
schemas:3.1.2 - https://eclipse.org/jetty/jetty-schemas)
        * Jetty Toolchain :: Test Helper
(org.eclipse.jetty.toolchain:jetty-test-helper:5.3 - https://
eclipse.org/jetty/jetty-test-helper)
        * Jetty :: Websocket :: javax.websocket :: Client
Implementation (org.eclipse.jetty.websocket:javax-websocket-client-
impl:9.4.21.v20190916 - http://www.eclipse.org/jetty/javax-websocket-
client-impl)
        * Jetty :: Websocket :: javax.websocket.server :: Server
Implementation (org.eclipse.jetty.websocket:javax-websocket-server-
impl:9.4.21.v20190916 - http://www.eclipse.org/jetty/javax-websocket-
server-impl)
        * Jetty :: Websocket :: API
(org.eclipse.jetty.websocket:websocket-api:9.4.21.v20190916 - http://
www.eclipse.org/jetty/websocket-api)
        * Jetty :: Websocket :: Client
(org.eclipse.jetty.websocket:websocket-client:9.4.21.v20190916 - http://
www.eclipse.org/jetty/websocket-client)
        * Jetty :: Websocket :: Common
(org.eclipse.jetty.websocket:websocket-common:9.4.21.v20190916 - http://
www.eclipse.org/jetty/websocket-common)
        * Jetty :: Websocket :: Server
(org.eclipse.jetty.websocket:websocket-server:9.4.21.v20190916 - http://
www.eclipse.org/jetty/websocket-server)
        * Jetty :: Websocket :: Servlet Interface
(org.eclipse.jetty.websocket:websocket-servlet:9.4.21.v20190916 -
http://www.eclipse.org/jetty/websocket-servlet)
        * org.eclipse.osgi
(org.eclipse.osgi:org.eclipse.osgi:3.6.0.v20100517 - http://
www.eclipse.org/org.eclipse.osgi)
        * org.eclipse.osgi.services
(org.eclipse.osgi:org.eclipse.osgi.services:3.2.100.v20100503 - http://
www.eclipse.org/org.eclipse.osgi.services)
        * org.eclipse.sisu.inject
(org.eclipse.sisu:org.eclipse.sisu.inject:0.3.3 - http://
www.eclipse.org/sisu/org.eclipse.sisu.inject/)
        * org.eclipse.sisu.plexus
(org.eclipse.sisu:org.eclipse.sisu.plexus:0.3.3 - http://
www.eclipse.org/sisu/org.eclipse.sisu.plexus/)
```

```
See above for Eclipse Public License - v 1.0

================================================================================
=============

    Eclipse Public License - v 2.0:

        * jnr-posix (com.github.jnr:jnr-posix:3.0.47 - http://
nexus.sonatype.org/oss-repository-hosting.html/jnr-posix)
        * Eclipse Compiler for Java(TM) (org.eclipse.jdt:ecj:3.17.0 -
http://www.eclipse.org/jdt)

Eclipse Public License - v 2.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE
PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF
THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

    1. DEFINITIONS

    "Contribution" means:
        a) in the case of the initial Contributor, the initial content
Distributed under this Agreement, and
        b) in the case of each subsequent Contributor:
            i) changes to the Program, and
            ii) additions to the Program;

        where such changes and/or additions to the Program originate
from and are Distributed by that particular Contributor. A Contribution
"originates" from a Contributor if it was added to the Program by such
Contributor itself or anyone acting on such Contributor's behalf.
Contributions do not include changes or additions to the Program that
are not Modified Works.

    "Contributor" means any person or entity that Distributes the
Program.

    "Licensed Patents" mean patent claims licensable by a Contributor
which are necessarily infringed by the use or sale of its Contribution
alone or when combined with the Program.

    "Program" means the Contributions Distributed in accordance with
this Agreement.

    "Recipient" means anyone who receives the Program under this
Agreement or any Secondary License (as applicable), including
Contributors.

    "Derivative Works" shall mean any work, whether in Source Code or
other form, that is based on (or derived from) the Program and for
which the editorial revisions, annotations, elaborations, or other
modifications represent, as a whole, an original work of authorship.

    "Modified Works" shall mean any work in Source Code or other form
that results from an addition to, deletion from, or modification of the
```

contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Secondary License" means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability ("notices") contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement.

In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the

```
date such litigation is filed.

    All Recipient's rights under this Agreement shall terminate if it
fails to comply with any of the material terms or conditions of this
Agreement and does not cure such failure in a reasonable period of time
after becoming aware of such noncompliance. If all Recipient's rights
under this Agreement terminate, Recipient agrees to cease use and
distribution of the Program as soon as reasonably practicable. However,
Recipient's obligations under this Agreement and any licenses granted
by Recipient relating to the Program shall continue and survive.

    Everyone is permitted to copy and distribute copies of this
Agreement, but in order to avoid inconsistency the Agreement is
copyrighted and may only be modified in the following manner. The
Agreement Steward reserves the right to publish new versions (including
revisions) of this Agreement from time to time. No one other than the
Agreement Steward has the right to modify this Agreement. The Eclipse
Foundation is the initial Agreement Steward. The Eclipse Foundation may
assign the responsibility to serve as the Agreement Steward to a
suitable separate entity. Each new version of the Agreement will be
given a distinguishing version number. The Program (including
Contributions) may always be Distributed subject to the version of the
Agreement under which it was received. In addition, after a new version
of the Agreement is published, Contributor may elect to Distribute the
Program (including its Contributions) under the new version.

    Except as expressly stated in Sections 2(a) and 2(b) above,
Recipient receives no rights or licenses to the intellectual property
of any Contributor under this Agreement, whether expressly, by
implication, estoppel or otherwise. All rights in the Program not
expressly granted under this Agreement are reserved. Nothing in this
Agreement is intended to be enforceable by any entity that is not a
Contributor or Recipient. No third-party beneficiary rights are created
under this Agreement.
    Exhibit A - Form of Secondary Licenses Notice

    "This Source Code may also be made available under the following
Secondary Licenses when the conditions for such availability set forth
in the Eclipse Public License, v. 2.0 are satisfied: {name license(s),
version(s), and exceptions or additional permissions here}."

    Simply including a copy of this Agreement, including this Exhibit A
is not sufficient to license the Source Code under Secondary Licenses.

    If it is not possible or desirable to put the notice in a
particular file, then You may include the notice in a location (such as
a LICENSE file in a relevant directory) where a recipient would be
likely to look for such a notice.

    You may add additional accurate notices of copyright ownership.


================================================================================
=============
```

GNU General Public License Version 2:

* jnr-posix (com.github.jnr:jnr-posix:3.0.47 - http://nexus.sonatype.org/oss-repository-hosting.html/jnr-posix)
* Checker Qual (org.checkerframework:checker-compat-qual:2.0.0 - http://checkerframework.org)

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original

authors' reputations.

  Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

  The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

  0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

  1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

  2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

    a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

    b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

```
    c) If the modified program normally reads commands interactively
    when run, you must cause it, when started running for such
    interactive use in the most ordinary way, to print or display an
    announcement including an appropriate copyright notice and a
    notice that there is no warranty (or else, saying that you provide
    a warranty) and that users may redistribute the program under
    these conditions, and telling the user how to view a copy of this
    License.  (Exception: if the Program itself is interactive but
    does not normally print such an announcement, your work based on
    the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole.  If
identifiable sections of that work are not derived from the Program,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works.  But when you
distribute the same sections as part of a whole which is a work based
on the Program, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote
it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program
with the Program (or with a work based on the Program) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

  3. You may copy and distribute the Program (or a work based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

    a) Accompany it with the complete corresponding machine-readable
    source code, which must be distributed under the terms of Sections
    1 and 2 above on a medium customarily used for software
interchange; or,

    b) Accompany it with a written offer, valid for at least three
    years, to give any third party, for a charge no more than your
    cost of physically performing source distribution, a complete
    machine-readable copy of the corresponding source code, to be
    distributed under the terms of Sections 1 and 2 above on a medium
    customarily used for software interchange; or,

    c) Accompany it with the information you received as to the offer
    to distribute corresponding source code.  (This alternative is
    allowed only for noncommercial distribution and only if you
    received the program in object code or executable form with such
    an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for
```

making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

  4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

  5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

  6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

  7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under

any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

  8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

  9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of
this License, you may choose any version ever published by the Free Software
Foundation.

  10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

Copyright (C)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

```
    You should have received a copy of the GNU General Public License
along
    with this program; if not, write to the Free Software Foundation,
Inc.,
    51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

    Gnomovision version 69, Copyright (C) year name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
`show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the
appropriate
parts of the General Public License.  Of course, the commands you use
may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary.  Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  `Gnomovision' (which makes passes at compilers) written by James
Hacker.

  , 1 April 1989
  Ty Coon, President of Vice

This General Public License does not permit incorporating your program
into
proprietary programs.  If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with
the
library.  If this is what you want to do, use the GNU Lesser General
Public License instead of this License.

================================================================================
=============

    GNU Lesser General Public License (LGPL), Version 2.1:

        * jnr-posix (com.github.jnr:jnr-posix:3.0.47 - http://
nexus.sonatype.org/oss-repository-hosting.html/jnr-posix)
        * Hibernate Search Engine (org.hibernate:hibernate-search-
engine:5.10.3.Final - http://search.hibernate.org/hibernate-search-
engine)
        * Hibernate Search Avro Serialization (org.hibernate:hibernate-
search-serialization-avro:5.10.3.Final - http://search.hibernate.org/
```

hibernate-search-serialization-avro)
        * Hibernate Commons Annotations (org.hibernate.common:hibernate-commons-annotations:5.0.4.Final - http://hibernate.org)

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that
there is no warranty for the free library. Also, if the library is
modified by someone else and passed on, the recipients should know
that what they have is not the original version, so that the original
author's reputation will not be affected by problems that might be
introduced by others.
\u000C
Finally, software patents pose a constant threat to the existence of
any free program. We wish to make sure that a company cannot
effectively restrict the users of a free program by obtaining a
restrictive license from a patent holder. Therefore, we insist that
any patent license obtained for a version of the library must be
consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the
ordinary GNU General Public License. This license, the GNU Lesser
General Public License, applies to certain designated libraries, and
is quite different from the ordinary General Public License. We use
this license for certain libraries in order to permit linking those
libraries into non-free programs.

When a program is linked with a library, whether statically or using
a shared library, the combination of the two is legally speaking a
combined work, a derivative of the original library. The ordinary
General Public License therefore permits such linking only if the
entire combination fits its criteria of freedom. The Lesser General
Public License permits more lax criteria for linking other code with
the library.

We call this license the "Lesser" General Public License because it
does Less to protect the user's freedom than the ordinary General
Public License. It also provides other free software developers Less
of an advantage over competing non-free programs. These disadvantages
are the reason we use the ordinary General Public License for many
libraries. However, the Lesser license provides advantages in certain
special circumstances.

For example, on rare occasions, there may be a special need to
encourage the widest possible use of a certain library, so that it
becomes
a de-facto standard. To achieve this, non-free programs must be
allowed to use the library. A more frequent case is that a free
library does the same job as widely used non-free libraries. In this
case, there is little to gain by limiting the free library to free
software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free
programs enables a greater number of people to use a large body of
free software. For example, permission to use the GNU C Library in
non-free programs enables many more people to use the whole GNU
operating system, as well as its variant, the GNU/Linux operating
system.

Although the Lesser General Public License is Less protective of the
users' freedom, it does ensure that the user of a program that is

linked with the Library has the freedom and the wherewithal to run
that program using a modified version of the Library.

  The precise terms and conditions for copying, distribution and
modification follow. Pay close attention to the difference between a
"work based on the library" and a "work that uses the library". The
former contains code derived from the library, whereas the latter must
be combined with the library in order to run.
\u000C
                  GNU LESSER GENERAL PUBLIC LICENSE
   TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

  0. This License Agreement applies to any software library or other
program which contains a notice placed by the copyright holder or
other authorized party saying it may be distributed under the terms of
this Lesser General Public License (also called "this License").
Each licensee is addressed as "you".

  A "library" means a collection of software functions and/or data
prepared so as to be conveniently linked with application programs
(which use some of those functions and data) to form executables.

  The "Library", below, refers to any such software library or work
which has been distributed under these terms. A "work based on the
Library" means either the Library or any derivative work under
copyright law: that is to say, a work containing the Library or a
portion of it, either verbatim or with modifications and/or translated
straightforwardly into another language. (Hereinafter, translation is
included without limitation in the term "modification".)

  "Source code" for a work means the preferred form of the work for
making modifications to it. For a library, complete source code means
all the source code for all modules it contains, plus any associated
interface definition files, plus the scripts used to control compilation
and installation of the library.

  Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running a program using the Library is not restricted, and output from
such a program is covered only if its contents constitute a work based
on the Library (independent of the use of the Library in a tool for
writing it). Whether that is true depends on what the Library does
and what the program that uses the Library does.

  1. You may copy and distribute verbatim copies of the Library's
complete source code as you receive it, in any medium, provided that
you conspicuously and appropriately publish on each copy an
appropriate copyright notice and disclaimer of warranty; keep intact
all the notices that refer to this License and to the absence of any
warranty; and distribute a copy of this License along with the
Library.

  You may charge a fee for the physical act of transferring a copy,
and you may at your option offer warranty protection in exchange for a
fee.

\u000C
  2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

    a) The modified work must itself be a software library.

    b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

    c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

    d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

    (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

  3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the

ordinary GNU General Public License has appeared, then you can specify
that version instead if you wish.) Do not make any other change in
these notices.
\u000C
  Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all
subsequent copies and derivative works made from that copy.

  This option is useful when you wish to copy part of the code of
the Library into a program that is not a library.

  4. You may copy and distribute the Library (or a portion or
derivative of it, under Section 2) in object code or executable form
under the terms of Sections 1 and 2 above provided that you accompany
it with the complete corresponding machine-readable source code, which
must be distributed under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

  If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not
compelled to copy the source along with the object code.

  5. A program that contains no derivative of any portion of the
Library, but is designed to work with the Library by being compiled or
linked with it, is called a "work that uses the Library". Such a
work, in isolation, is not a derivative work of the Library, and
therefore falls outside the scope of this License.

  However, linking a "work that uses the Library" with the Library
creates an executable that is a derivative of the Library (because it
contains portions of the Library), rather than a "work that uses the
library". The executable is therefore covered by this License.
Section 6 states terms for distribution of such executables.

  When a "work that uses the Library" uses material from a header file
that is part of the Library, the object code for the work may be a
derivative work of the Library even though the source code is not.
Whether this is true is especially significant if the work can be
linked without the Library, or if the work is itself a library. The
threshold for this to be true is not precisely defined by law.

  If such an object file uses only numerical parameters, data
structure layouts and accessors, and small macros and small inline
functions (ten lines or less in length), then the use of the object
file is unrestricted, regardless of whether it is legally a derivative
work. (Executables containing this object code plus portions of the
Library will still fall under Section 6.)

  Otherwise, if the work is a derivative of the Library, you may
distribute the object code for the work under the terms of Section 6.
Any executables containing that work also fall under Section 6,
whether or not they are linked directly with the Library itself.
\u000C

```
  6. As an exception to the Sections above, you may also combine or
link a "work that uses the Library" with the Library to produce a
work containing portions of the Library, and distribute that work
under terms of your choice, provided that the terms permit
modification of the work for the customer's own use and reverse
engineering for debugging such modifications.

  You must give prominent notice with each copy of the work that the
Library is used in it and that the Library and its use are covered by
this License.  You must supply a copy of this License.  If the work
during execution displays copyright notices, you must include the
copyright notice for the Library among them, as well as a reference
directing the user to the copy of this License.  Also, you must do one
of these things:

    a) Accompany the work with the complete corresponding
    machine-readable source code for the Library including whatever
    changes were used in the work (which must be distributed under
    Sections 1 and 2 above); and, if the work is an executable linked
    with the Library, with the complete machine-readable "work that
    uses the Library", as object code and/or source code, so that the
    user can modify the Library and then relink to produce a modified
    executable containing the modified Library.  (It is understood
    that the user who changes the contents of definitions files in the
    Library will not necessarily be able to recompile the application
    to use the modified definitions.)

    b) Use a suitable shared library mechanism for linking with the
    Library.  A suitable mechanism is one that (1) uses at run time a
    copy of the library already present on the user's computer system,
    rather than copying library functions into the executable, and (2)
    will operate properly with a modified version of the library, if
    the user installs one, as long as the modified version is
    interface-compatible with the version that the work was made with.

    c) Accompany the work with a written offer, valid for at
    least three years, to give the same user the materials
    specified in Subsection 6a, above, for a charge no more
    than the cost of performing this distribution.

    d) If distribution of the work is made by offering access to copy
    from a designated place, offer equivalent access to copy the above
    specified materials from the same place.

    e) Verify that the user has already received a copy of these
    materials or that you have already sent this user a copy.

  For an executable, the required form of the "work that uses the
Library" must include any data and utility programs needed for
reproducing the executable from it.  However, as a special exception,
the materials to be distributed need not include anything that is
normally distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies
the executable.
```

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.
\u000C
7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.
\u000C
11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you

may not distribute the Library at all. For example, if a patent
license would not permit royalty-free redistribution of the Library by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under
any
particular circumstance, the balance of the section is intended to
apply,
and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

  12. If the distribution and/or use of the Library is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Library under this License may
add
an explicit geographical distribution limitation excluding those
countries,
so that distribution is permitted only in or among countries not thus
excluded. In such case, this License incorporates the limitation as if
written in the body of this License.

  13. The Free Software Foundation may publish revised and/or new
versions of the Lesser General Public License from time to time.
Such new versions will be similar in spirit to the present version,
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library
specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation. If the Library does not specify a
license version number, you may choose any version ever published by
the Free Software Foundation.
\u000C
  14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission. For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this. Our
decision will be guided by the two goals of preserving the free status

of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

END OF TERMS AND CONDITIONS
\u000C
How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of
the
ordinary General Public License).

To apply these terms, attach the following notices to the library.
It is
safest to attach them to the start of each source file to most
effectively
convey the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

Copyright (C)

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU

```
    Lesser General Public License for more details.

    You should have received a copy of the GNU Lesser General Public
    License along with this library; if not, write to the Free Software
    Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301  USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary.  Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the
  library `Frob' (a library for tweaking knobs) written by James Random
Hacker.

  , 1 April 1990
  Ty Coon, President of Vice

That's all there is to it!

================================================================================
=============

    MIT License:

        * jnr-x86asm (com.github.jnr:jnr-x86asm:1.0.2 - http://
github.com/jnr/jnr-x86asm)
        * JOpt Simple (net.sf.jopt-simple:jopt-simple:4.6 - http://
pholser.github.com/jopt-simple)
        * Checker Qual (org.checkerframework:checker-compat-qual:2.0.0
- http://checkerframework.org)
        * Animal Sniffer Annotations (org.codehaus.mojo:animal-sniffer-
annotations:1.14 - http://mojo.codehaus.org/animal-sniffer/animal-
sniffer-annotations)
        * Animal Sniffer Annotations (org.codehaus.mojo:animal-sniffer-
annotations:1.9 - http://mojo.codehaus.org/animal-sniffer/animal-
sniffer-annotations)
        * JCL 1.2 implemented over SLF4J (org.slf4j:jcl-over-
slf4j:1.7.25 - http://www.slf4j.org)
        * SLF4J API Module (org.slf4j:slf4j-api:1.7.25 - http://
www.slf4j.org)
        * SLF4J Simple Binding (org.slf4j:slf4j-simple:1.7.25 - http://
www.slf4j.org)

Copyright

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:
```

```
The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

================================================================================
=============

    Public Domain:

        * AOP alliance (aopalliance:aopalliance:1.0 - http://
aopalliance.sourceforge.net)
        * Backport of JSR 166 (backport-util-concurrent:backport-util-
concurrent:3.1 - http://backport-jsr166.sourceforge.net/)
        * reactive-streams (org.reactivestreams:reactive-streams:1.0.2
- http://www.reactive-streams.org/)
     CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT
PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS DOCUMENT DOES NOT CREATE
AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS
INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES
REGARDING THE USE OF THIS DOCUMENT OR THE INFORMATION OR WORKS PROVIDED
HEREUNDER, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM THE USE
OF THIS DOCUMENT OR THE INFORMATION OR WORKS PROVIDED HEREUNDER.

Statement of Purpose

The laws of most jurisdictions throughout the world automatically
confer exclusive Copyright and Related Rights (defined below) upon the
creator and subsequent owner(s) (each and all, an "owner") of an
original work of authorship and/or a database (each, a "Work").

Certain owners wish to permanently relinquish those rights to a Work
for the purpose of contributing to a commons of creative, cultural and
scientific works ("Commons") that the public can reliably and without
fear of later claims of infringement build upon, modify, incorporate in
other works, reuse and redistribute as freely as possible in any form
whatsoever and for any purposes, including without limitation
commercial purposes. These owners may contribute to the Commons to
promote the ideal of a free culture and the further production of
creative, cultural and scientific works, or to gain reputation or
greater distribution for their Work in part through the use and efforts
of others.

For these and/or other purposes and motivations, and without any
expectation of additional consideration or compensation, the person
associating CC0 with a Work (the "Affirmer"), to the extent that he or
she is an owner of Copyright and Related Rights in the Work,
voluntarily elects to apply CC0 to the Work and publicly distribute the
```

Work under its terms, with knowledge of his or her Copyright and Related Rights in the Work and the meaning and intended legal effect of CC0 on those rights.

1. Copyright and Related Rights. A Work made available under CC0 may be protected by copyright and related or neighboring rights ("Copyright and Related Rights"). Copyright and Related Rights include, but are not limited to, the following:

the right to reproduce, adapt, distribute, perform, display, communicate, and translate a Work;
moral rights retained by the original author(s) and/or performer(s);
publicity and privacy rights pertaining to a person's image or likeness depicted in a Work;
rights protecting against unfair competition in regards to a Work, subject to the limitations in paragraph 4(a), below;
rights protecting the extraction, dissemination, use and reuse of data in a Work;
database rights (such as those arising under Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, and under any national implementation thereof, including any amended or successor version of such directive); and
other similar, equivalent or corresponding rights throughout the world based on applicable law or treaty, and any national implementations thereof.

2. Waiver. To the greatest extent permitted by, but not in contravention of, applicable law, Affirmer hereby overtly, fully, permanently, irrevocably and unconditionally waives, abandons, and surrenders all of Affirmer's Copyright and Related Rights and associated claims and causes of action, whether now known or unknown (including existing as well as future claims and causes of action), in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "Waiver"). Affirmer makes the Waiver for the benefit of each member of the public at large and to the detriment of Affirmer's heirs and successors, fully intending that such Waiver shall not be subject to revocation, rescission, cancellation, termination, or any other legal or equitable action to disrupt the quiet enjoyment of the Work by the public as contemplated by Affirmer's express Statement of Purpose.

3. Public License Fallback. Should any part of the Waiver for any reason be judged legally invalid or ineffective under applicable law, then the Waiver shall be preserved to the maximum extent permitted taking into account Affirmer's express Statement of Purpose. In addition, to the extent the Waiver is so judged Affirmer hereby grants to each affected person a royalty-free, non transferable, non sublicensable, non exclusive, irrevocable and unconditional license to exercise Affirmer's Copyright and Related Rights in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in

```
any current or future medium and for any number of copies, and (iv) for
any purpose whatsoever, including without limitation commercial,
advertising or promotional purposes (the "License"). The License shall
be deemed effective as of the date CC0 was applied by Affirmer to the
Work. Should any part of the License for any reason be judged legally
invalid or ineffective under applicable law, such partial invalidity or
ineffectiveness shall not invalidate the remainder of the License, and
in such case Affirmer hereby affirms that he or she will not (i)
exercise any of his or her remaining Copyright and Related Rights in
the Work or (ii) assert any associated claims and causes of action with
respect to the Work, in either case contrary to Affirmer's express
Statement of Purpose.

4. Limitations and Disclaimers.

    No trademark or patent rights held by Affirmer are waived,
abandoned, surrendered, licensed or otherwise affected by this document.
    Affirmer offers the Work as-is and makes no representations or
warranties of any kind concerning the Work, express, implied, statutory
or otherwise, including without limitation warranties of title,
merchantability, fitness for a particular purpose, non infringement, or
the absence of latent or other defects, accuracy, or the present or
absence of errors, whether or not discoverable, all to the greatest
extent permissible under applicable law.
    Affirmer disclaims responsibility for clearing rights of other
persons that may apply to the Work or any use thereof, including
without limitation any person's Copyright and Related Rights in the
Work. Further, Affirmer disclaims responsibility for obtaining any
necessary consents, permissions or other rights required for any use of
the Work.
    Affirmer understands and acknowledges that Creative Commons is not
a party to this document and has no duty or obligation with respect to
this CC0 or use of the Work.

================================================================================
=============

    The Apache Software License, Version 2.0:

        * ant (ant:ant:1.6.5 - http://www.apache.org/ant/)
        * A Swiss Army Knife for OSGi (biz.aQute.bnd:bndlib:2.4.0 -
http://www.aQute.biz/Code/Bnd/bndlib)
        * Jackson-core (com.fasterxml.jackson.core:jackson-core:2.1.3 -
http://wiki.fasterxml.com/JacksonHome)
        * Caffeine cache (com.github.ben-manes.caffeine:caffeine:2.6.2
- https://github.com/ben-manes/caffeine)
        * jffi (com.github.jnr:jffi:1.2.16 - http://github.com/jnr/jffi)
        * jffi (com.github.jnr:jffi:1.2.17 - http://github.com/jnr/jffi)
        * jnr-a64asm (com.github.jnr:jnr-a64asm:1.0.0 - http://
nexus.sonatype.org/oss-repository-hosting.html/jnr-a64asm)
        * jnr-constants (com.github.jnr:jnr-constants:0.9.11 - http://
github.com/jnr/jnr-constants)
        * jnr-enxio (com.github.jnr:jnr-enxio:0.20 - http://
github.com/jnr/jnr-enxio)
        * jnr-ffi (com.github.jnr:jnr-ffi:2.1.9 - http://github.com/jnr/
```

```
jnr-ffi)
        * jnr-unixsocket (com.github.jnr:jnr-unixsocket:0.22 - http://
github.com/jnr/jnr-unixsocket)
        * Google APIs Client Library for Java (com.google.api-
client:google-api-client:1.21.0 - https://github.com/google/google-api-
java-client/google-api-client)
        * com.google.api.grpc:grpc-google-common-protos
(com.google.api.grpc:grpc-google-common-protos:0.1.0 - https://
github.com/googleapis/googleapis)
        * com.google.api.grpc:proto-google-common-protos
(com.google.api.grpc:proto-google-common-protos:0.1.9 - https://
github.com/googleapis/googleapis)
        * com.google.api.grpc:proto-google-iam-v1
(com.google.api.grpc:proto-google-iam-v1:0.1.9 - https://github.com/
googleapis/googleapis)
        * AutoValue (com.google.auto.value:auto-value:1.2 - https://
github.com/google/auto/auto-value)
        * com.google.cloud.datastore:datastore-v1-proto-client
(com.google.cloud.datastore:datastore-v1-proto-client:1.3.0 - https://
cloud.google.com/datastore/datastore-v1-proto-client/)
        * com.google.cloud.datastore:datastore-v1-protos
(com.google.cloud.datastore:datastore-v1-protos:1.3.0 - https://
cloud.google.com/datastore/)
        * FindBugs-jsr305 (com.google.code.findbugs:jsr305:3.0.0 -
http://findbugs.sourceforge.net/)
        * Gson (com.google.code.gson:gson:2.7 - https://github.com/
google/gson/gson)
        * Gson (com.google.code.gson:gson:2.8.1 - https://github.com/
google/gson/gson)
        * error-prone annotations
(com.google.errorprone:error_prone_annotations:2.1.3 - http://
nexus.sonatype.org/oss-repository-hosting.html/error_prone_parent/
error_prone_annotations)
        * Guava: Google Core Libraries for Java
(com.google.guava:guava:19.0 - https://github.com/google/guava/guava)
        * Guava: Google Core Libraries for Java
(com.google.guava:guava:25.1-android - https://github.com/google/guava/
guava)
        * Google HTTP Client Library for Java (com.google.http-
client:google-http-client:1.21.0 - https://github.com/google/google-
http-java-client/google-http-client)
        * Google App Engine extensions to the Google HTTP Client
Library for Java. (com.google.http-client:google-http-client-
appengine:1.21.0 - https://github.com/google/google-http-java-client/
google-http-client-appengine)
        * Jackson extensions to the Google HTTP Client Library for
Java. (com.google.http-client:google-http-client-jackson:1.21.0 -
https://github.com/google/google-http-java-client/google-http-client-
jackson)
        * Jackson 2 extensions to the Google HTTP Client Library for
Java. (com.google.http-client:google-http-client-jackson2:1.21.0 -
https://github.com/google/google-http-java-client/google-http-client-
jackson2)
        * Protocol Buffer extensions to the Google HTTP Client Library
for Java. (com.google.http-client:google-http-client-protobuf:1.20.0 -
```

```
https://github.com/google/google-http-java-client/google-http-client-
protobuf)
        * Google Guice - Core Library (com.google.inject:guice:4.2.1 -
https://github.com/google/guice/guice)
        * J2ObjC Annotations (com.google.j2objc:j2objc-annotations:1.1
- https://github.com/google/j2objc/)
        * Google OAuth Client Library for Java (com.google.oauth-
client:google-oauth-client:1.21.0 - https://github.com/google/google-
oauth-java-client/google-oauth-client)
        * xmemcached (com.googlecode.xmemcached:xmemcached:2.4.5 -
https://github.com/killme2008/xmemcached)
        * hazelcast (com.hazelcast:hazelcast:3.9.4 - http://
www.hazelcast.com/hazelcast/)
        * hazelcast-client (com.hazelcast:hazelcast-client:3.9.4 -
http://www.hazelcast.com/hazelcast-client/)
        * ProtoParser (com.squareup:protoparser:4.0.3 - http://
github.com/square/protoparser/)
        * OkHttp (com.squareup.okhttp:okhttp:2.5.0 - https://github.com/
square/okhttp/okhttp)
        * Okio (com.squareup.okio:okio:1.6.0 - https://github.com/
square/okio/okio)
        * Apache Commons Codec (commons-codec:commons-codec:1.10 -
http://commons.apache.org/proper/commons-codec/)
        * Codec (commons-codec:commons-codec:1.3 - http://
jakarta.apache.org/commons/codec/)
        * Commons Codec (commons-codec:commons-codec:1.6 - http://
commons.apache.org/codec/)
        * Apache Commons IO (commons-io:commons-io:2.5 - http://
commons.apache.org/proper/commons-io/)
        * Commons Logging (commons-logging:commons-logging:1.1.1 -
http://commons.apache.org/logging)
        * Commons Logging (commons-logging:commons-logging:1.1.3 -
http://commons.apache.org/proper/commons-logging/)
        * Netty/Buffer (io.netty:netty-buffer:4.1.28.Final - http://
netty.io/netty-buffer/)
        * Netty/Buffer (io.netty:netty-buffer:4.1.3.Final - http://
netty.io/netty-buffer/)
        * Netty/Codec (io.netty:netty-codec:4.1.28.Final - http://
netty.io/netty-codec/)
        * Netty/Codec (io.netty:netty-codec:4.1.3.Final - http://
netty.io/netty-codec/)
        * Netty/Codec/HTTP (io.netty:netty-codec-http:4.1.3.Final -
http://netty.io/netty-codec-http/)
        * Netty/Codec/HTTP2 (io.netty:netty-codec-http2:4.1.3.Final -
http://netty.io/netty-codec-http2/)
        * Netty/Common (io.netty:netty-common:4.1.28.Final - http://
netty.io/netty-common/)
        * Netty/Common (io.netty:netty-common:4.1.3.Final - http://
netty.io/netty-common/)
        * Netty/Handler (io.netty:netty-handler:4.1.28.Final - http://
netty.io/netty-handler/)
        * Netty/Handler (io.netty:netty-handler:4.1.3.Final - http://
netty.io/netty-handler/)
        * Netty/Resolver (io.netty:netty-resolver:4.1.28.Final - http://
netty.io/netty-resolver/)
```

```
        * Netty/Resolver (io.netty:netty-resolver:4.1.3.Final - http://
netty.io/netty-resolver/)
        * Netty/Transport (io.netty:netty-transport:4.1.28.Final -
http://netty.io/netty-transport/)
        * Netty/Transport (io.netty:netty-transport:4.1.3.Final -
http://netty.io/netty-transport/)
        * Netty/Transport/Native/Epoll (io.netty:netty-transport-native-
epoll:4.1.28.Final - http://netty.io/netty-transport-native-epoll/)
        * Netty/Transport/Native/Unix/Common (io.netty:netty-transport-
native-unix-common:4.1.28.Final - http://netty.io/netty-transport-
native-unix-common/)
        * RxJava (io.reactivex.rxjava2:rxjava:2.2.4 - https://
github.com/ReactiveX/RxJava)
        * Joda-Time (joda-time:joda-time:2.9.2 - http://www.joda.org/
joda-time/)
        * Apache Ant Core (org.apache.ant:ant:1.8.4 - http://
ant.apache.org/ant/)
        * Apache Ant Launcher (org.apache.ant:ant-launcher:1.8.4 -
http://ant.apache.org/ant-launcher/)
        * Apache Avro (org.apache.avro:avro:1.7.6 - http://
avro.apache.org)
        * Commons Compress (org.apache.commons:commons-compress:1.4.1 -
http://commons.apache.org/compress/)
        * Apache Commons Lang (org.apache.commons:commons-lang3:3.8.1 -
http://commons.apache.org/proper/commons-lang/)
        * Commons Math (org.apache.commons:commons-math3:3.2 - http://
commons.apache.org/proper/commons-math/)
        * Apache Felix Framework
(org.apache.felix:org.apache.felix.framework:6.0.3 - http://
felix.apache.org/org.apache.felix.framework/)
        * HttpClient (org.apache.httpcomponents:httpclient:4.0.1 -
http://hc.apache.org/httpcomponents-client)
        * Apache HttpClient (org.apache.httpcomponents:httpclient:4.5.6
- http://hc.apache.org/httpcomponents-client)
        * HttpCore (org.apache.httpcomponents:httpcore:4.0.1 - http://
hc.apache.org/httpcomponents-core/)
        * Apache HttpCore (org.apache.httpcomponents:httpcore:4.4.10 -
http://hc.apache.org/httpcomponents-core-ga)
        * Lucene Common Analyzers (org.apache.lucene:lucene-analyzers-
common:5.5.5 - http://lucene.apache.org/lucene-parent/lucene-analyzers-
common)
        * Lucene Core (org.apache.lucene:lucene-core:5.5.5 - http://
lucene.apache.org/lucene-parent/lucene-core)
        * Lucene Facets (org.apache.lucene:lucene-facet:5.5.5 - http://
lucene.apache.org/lucene-parent/lucene-facet)
        * Lucene Miscellaneous (org.apache.lucene:lucene-misc:5.5.5 -
http://lucene.apache.org/lucene-parent/lucene-misc)
        * Lucene Queries (org.apache.lucene:lucene-queries:5.5.5 -
http://lucene.apache.org/lucene-parent/lucene-queries)
        * Lucene QueryParsers (org.apache.lucene:lucene-
queryparser:5.5.5 - http://lucene.apache.org/lucene-parent/lucene-
queryparser)
        * EL 2.2 plugin (org.apache.openwebbeans:openwebbeans-
el22:2.0.11 - openwebbeans.apache.org/openwebbeans-el22)
        * OpenWebBeans Core (org.apache.openwebbeans:openwebbeans-
```

```
impl:2.0.11 - openwebbeans.apache.org/openwebbeans-impl)
        * Jetty 9 plugin (org.apache.openwebbeans:openwebbeans-
jetty9:2.0.11 - openwebbeans.apache.org/openwebbeans-jetty9)
        * SPI definition (org.apache.openwebbeans:openwebbeans-
spi:2.0.11 - openwebbeans.apache.org/openwebbeans-spi)
        * Web plugin (org.apache.openwebbeans:openwebbeans-web:2.0.11 -
openwebbeans.apache.org/openwebbeans-web)
        * Apache Standard Taglib Implementation
(org.apache.taglibs:taglibs-standard-impl:1.2.5 - http://
tomcat.apache.org/taglibs/standard-1.2.5/taglibs-standard-impl)
        * Apache Standard Taglib Specification API
(org.apache.taglibs:taglibs-standard-spec:1.2.5 - http://
tomcat.apache.org/taglibs/standard-1.2.5/taglibs-standard-spec)
        * org.apiguardian:apiguardian-api (org.apiguardian:apiguardian-
api:1.1.0 - https://github.com/apiguardian-team/apiguardian)
        * Jackson (org.codehaus.jackson:jackson-core-asl:1.9.11 -
http://jackson.codehaus.org)
        * Jackson (org.codehaus.jackson:jackson-core-asl:1.9.13 -
http://jackson.codehaus.org)
        * Data Mapper for Jackson (org.codehaus.jackson:jackson-mapper-
asl:1.9.13 - http://jackson.codehaus.org)
        * Plexus Classworlds (org.codehaus.plexus:plexus-
classworlds:2.5.2 - http://plexus.codehaus.org/plexus-classworlds/)
        * Plexus :: Component Annotations (org.codehaus.plexus:plexus-
component-annotations:1.5.5 - http://plexus.codehaus.org/plexus-
containers/plexus-component-annotations/)
        * Plexus :: Component Annotations (org.codehaus.plexus:plexus-
component-annotations:1.7.1 - http://codehaus-plexus.github.io/plexus-
containers/plexus-component-annotations/)
        * Plexus Interpolation API (org.codehaus.plexus:plexus-
interpolation:1.11 - http://plexus.codehaus.org/plexus-components/
plexus-interpolation)
        * Plexus Interpolation API (org.codehaus.plexus:plexus-
interpolation:1.25 - http://codehaus-plexus.github.io/plexus-
interpolation/)
        * Plexus Common Utilities (org.codehaus.plexus:plexus-
utils:3.1.0 - http://codehaus-plexus.github.io/plexus-utils/)
        * org.conscrypt:conscrypt-openjdk-uber (org.conscrypt:conscrypt-
openjdk-uber:2.1.0 - https://conscrypt.org/)
        * Jetty :: Apache JSP Implementation (org.eclipse.jetty:apache-
jsp:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Apache :: JSTL module (org.eclipse.jetty:apache-
jstl:9.4.21.v20190916 - http://tomcat.apache.org/taglibs/standard/)
        * Jetty :: Infinispan Session Manager Common
(org.eclipse.jetty:infinispan-common:9.4.21.v20190916 - http://
www.eclipse.org/jetty)
        * Jetty :: Infinispan Session Manager Remote
(org.eclipse.jetty:infinispan-remote-query:9.4.21.v20190916 - http://
www.eclipse.org/jetty)
        * Jetty :: ALPN :: Client (org.eclipse.jetty:jetty-alpn-
client:9.4.21.v20190916 - https://eclipse.org/jetty/jetty-alpn-parent/
jetty-alpn-client)
        * Jetty :: ALPN :: Conscrypt Server Implementation
(org.eclipse.jetty:jetty-alpn-conscrypt-server:9.4.21.v20190916 -
https://eclipse.org/jetty/jetty-alpn-parent/jetty-alpn-conscrypt-server)
```

```
        * Jetty :: ALPN :: OpenJDK8 Client Implementation
(org.eclipse.jetty:jetty-alpn-openjdk8-client:9.4.21.v20190916 -
https://eclipse.org/jetty/jetty-alpn-parent/jetty-alpn-openjdk8-client)
        * Jetty :: ALPN :: OpenJDK8 Server Implementation
(org.eclipse.jetty:jetty-alpn-openjdk8-server:9.4.21.v20190916 -
https://eclipse.org/jetty/jetty-alpn-parent/jetty-alpn-openjdk8-server)
        * Jetty :: ALPN :: Server (org.eclipse.jetty:jetty-alpn-
server:9.4.21.v20190916 - https://eclipse.org/jetty/jetty-alpn-parent/
jetty-alpn-server)
        * Jetty :: Servlet Annotations (org.eclipse.jetty:jetty-
annotations:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: Asynchronous HTTP Client (org.eclipse.jetty:jetty-
client:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: Continuation (org.eclipse.jetty:jetty-
continuation:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: Deployers (org.eclipse.jetty:jetty-
deploy:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: Distribution Assemblies (org.eclipse.jetty:jetty-
distribution:9.4.21.v20190916 - https://eclipse.org/jetty/jetty-
distribution)
        * Jetty :: Hazelcast Session Manager (org.eclipse.jetty:jetty-
hazelcast:9.4.21.v20190916 - https://eclipse.org/jetty/jetty-hazelcast)
        * Jetty :: Http Utility (org.eclipse.jetty:jetty-
http:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: IO Utility (org.eclipse.jetty:jetty-
io:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: JAAS (org.eclipse.jetty:jetty-jaas:9.4.21.v20190916
- https://eclipse.org/jetty/jetty-jaas)
        * Jetty :: JMX Management (org.eclipse.jetty:jetty-
jmx:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: JNDI Naming (org.eclipse.jetty:jetty-
jndi:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: NoSQL Session Managers (org.eclipse.jetty:jetty-
nosql:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: Plus (org.eclipse.jetty:jetty-plus:9.4.21.v20190916
- http://www.eclipse.org/jetty)
        * Jetty :: Proxy (org.eclipse.jetty:jetty-
proxy:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: Quick Start (org.eclipse.jetty:jetty-
quickstart:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: Rewrite Handler (org.eclipse.jetty:jetty-
rewrite:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: Security (org.eclipse.jetty:jetty-
security:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: Server Core (org.eclipse.jetty:jetty-
server:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: Servlet Handling (org.eclipse.jetty:jetty-
servlet:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: Utility Servlets and Filters
(org.eclipse.jetty:jetty-servlets:9.4.21.v20190916 - http://
www.eclipse.org/jetty)
        * Jetty :: Utilities (org.eclipse.jetty:jetty-util:9.3.0.RC0 -
http://www.eclipse.org/jetty)
        * Jetty :: Utilities (org.eclipse.jetty:jetty-
util:9.4.21.v20190916 - http://www.eclipse.org/jetty)
```

```
        * Jetty :: Utilities :: Ajax(JSON) (org.eclipse.jetty:jetty-
util-ajax:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: Webapp Application Support (org.eclipse.jetty:jetty-
webapp:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: XML utilities (org.eclipse.jetty:jetty-
xml:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Test :: Jetty Test Webapp (org.eclipse.jetty:test-jetty-
webapp:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty :: ALPN :: API (org.eclipse.jetty.alpn:alpn-
api:1.1.3.v20160715 - http://www.eclipse.org/jetty/alpn-api)
        * Example Async Rest :: Jar (org.eclipse.jetty.example-async-
rest:example-async-rest-jar:9.4.21.v20190916 - https://eclipse.org/
jetty/examples-parent/example-async-rest/example-async-rest-jar)
        * Jetty :: FastCGI :: Client (org.eclipse.jetty.fcgi:fcgi-
client:9.4.21.v20190916 - https://eclipse.org/jetty/fcgi-parent/fcgi-
client)
        * Jetty :: GCloud :: Session Manager
(org.eclipse.jetty.gcloud:jetty-gcloud-session-manager:9.4.21.v20190916
- https://eclipse.org/jetty/gcloud-parent/jetty-gcloud-session-manager)
        * Jetty :: HTTP2 :: Client (org.eclipse.jetty.http2:http2-
client:9.4.21.v20190916 - https://eclipse.org/jetty/http2-parent/http2-
client)
        * Jetty :: HTTP2 :: Common (org.eclipse.jetty.http2:http2-
common:9.4.21.v20190916 - https://eclipse.org/jetty/http2-parent/http2-
common)
        * Jetty :: HTTP2 :: HPACK (org.eclipse.jetty.http2:http2-
hpack:9.4.21.v20190916 - https://eclipse.org/jetty/http2-parent/http2-
hpack)
        * Jetty :: HTTP2 :: Server (org.eclipse.jetty.http2:http2-
server:9.4.21.v20190916 - https://eclipse.org/jetty/http2-parent/http2-
server)
        * Jetty :: Memcached :: Sessions
(org.eclipse.jetty.memcached:jetty-memcached-sessions:9.4.21.v20190916
- https://eclipse.org/jetty/memcached-parent/jetty-memcached-sessions)
        * Jetty Orbit :: Activation
(org.eclipse.jetty.orbit:javax.activation:1.1.0.v201105071233 - http://
www.eclipse.org/jetty/jetty-orbit/javax.activation)
        * Jetty Orbit :: Glassfish Mail
(org.eclipse.jetty.orbit:javax.mail.glassfish:1.4.1.v201005082020 -
http://www.eclipse.org/jetty/jetty-orbit/javax.mail.glassfish)
        * Jetty Orbit :: JASPI API
(org.eclipse.jetty.orbit:javax.security.auth.message:1.0.0.v201108011116
 - http://www.eclipse.org/jetty/jetty-orbit/javax.security.auth.message)
        * Jetty Orbit :: JSP JSTL
(org.eclipse.jetty.orbit:javax.servlet.jsp.jstl:1.2.0.v201105211821 -
http://www.eclipse.org/jetty/jetty-orbit/javax.servlet.jsp.jstl)
        * Jetty :: OSGi :: Boot (org.eclipse.jetty.osgi:jetty-osgi-
boot:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Jetty Tests :: JMX :: WebApp (org.eclipse.jetty.tests:jmx-
webapp:9.4.21.v20190916 - http://www.eclipse.org/jetty)
        * Test :: CDI :: Common Demo Webapp
(org.eclipse.jetty.tests:test-cdi-common-webapp:9.4.21.v20190916 -
http://www.eclipse.org/jetty/test-cdi-common-webapp)
        * Jetty Tests :: WebApp :: Servlet Spec ::
ServletContainerInitializer Test Jar (org.eclipse.jetty.tests:test-
```

```
container-initializer:9.4.21.v20190916 - http://www.eclipse.org/jetty/
test-servlet-spec-parent/test-container-initializer)
        * Jetty Tests :: WebApp :: JNDI (org.eclipse.jetty.tests:test-
jndi-webapp:9.4.21.v20190916 - http://www.eclipse.org/jetty/test-jndi-
webapp)
        * Jetty Tests :: WebApp :: Mock Resources
(org.eclipse.jetty.tests:test-mock-resources:9.4.21.v20190916 - http://
www.eclipse.org/jetty/test-mock-resources)
        * Jetty Tests :: Sessions :: Common
(org.eclipse.jetty.tests:test-sessions-common:9.4.21.v20190916 - http://
www.eclipse.org/jetty)
        * Jetty Tests :: Webapps :: Spec Webapp
(org.eclipse.jetty.tests:test-spec-webapp:9.4.21.v20190916 - http://
www.eclipse.org/jetty/test-servlet-spec-parent/test-spec-webapp)
        * Jetty Tests :: WebApp :: Servlet Spec :: Fragment Jar
(org.eclipse.jetty.tests:test-web-fragment:9.4.21.v20190916 - http://
www.eclipse.org/jetty/test-servlet-spec-parent/test-web-fragment)
        * Jetty :: Servlet API and Schemas for OSGi
(org.eclipse.jetty.toolchain:jetty-osgi-servlet-api:3.1.0.M3 - http://
www.eclipse.org/jetty/jetty-toolchain/jetty-osgi-servlet-api)
        * Jetty :: Schemas (org.eclipse.jetty.toolchain:jetty-
schemas:3.1.2 - https://eclipse.org/jetty/jetty-schemas)
        * Jetty Toolchain :: Test Helper
(org.eclipse.jetty.toolchain:jetty-test-helper:5.3 - https://
eclipse.org/jetty/jetty-test-helper)
        * Jetty :: Websocket :: javax.websocket :: Client
Implementation (org.eclipse.jetty.websocket:javax-websocket-client-
impl:9.4.21.v20190916 - http://www.eclipse.org/jetty/javax-websocket-
client-impl)
        * Jetty :: Websocket :: javax.websocket.server :: Server
Implementation (org.eclipse.jetty.websocket:javax-websocket-server-
impl:9.4.21.v20190916 - http://www.eclipse.org/jetty/javax-websocket-
server-impl)
        * Jetty :: Websocket :: API
(org.eclipse.jetty.websocket:websocket-api:9.4.21.v20190916 - http://
www.eclipse.org/jetty/websocket-api)
        * Jetty :: Websocket :: Client
(org.eclipse.jetty.websocket:websocket-client:9.4.21.v20190916 - http://
www.eclipse.org/jetty/websocket-client)
        * Jetty :: Websocket :: Common
(org.eclipse.jetty.websocket:websocket-common:9.4.21.v20190916 - http://
www.eclipse.org/jetty/websocket-common)
        * Jetty :: Websocket :: Server
(org.eclipse.jetty.websocket:websocket-server:9.4.21.v20190916 - http://
www.eclipse.org/jetty/websocket-server)
        * Jetty :: Websocket :: Servlet Interface
(org.eclipse.jetty.websocket:websocket-servlet:9.4.21.v20190916 -
http://www.eclipse.org/jetty/websocket-servlet)
        * Infinispan Hot Rod Client (org.infinispan:infinispan-client-
hotrod:9.4.8.Final - http://www.infinispan.org/infinispan-client-hotrod)
        * Infinispan Commons (org.infinispan:infinispan-
commons:9.4.8.Final - http://www.infinispan.org/infinispan-commons)
        * Infinispan Core (org.infinispan:infinispan-core:9.4.8.Final -
http://www.infinispan.org/infinispan-core)
        * Infinispan Directory Provider for Hibernate Search
```

```
(org.infinispan:infinispan-directory-provider:9.4.8.Final - http://
www.infinispan.org/infinispan-lucene-parent/infinispan-directory-
provider)
        * Infinispan Lucene Directory Implementation
(org.infinispan:infinispan-lucene-directory:9.4.8.Final - http://
www.infinispan.org/infinispan-lucene-parent/infinispan-lucene-directory)
        * Infinispan Object Querying and Filtering API
(org.infinispan:infinispan-objectfilter:9.4.8.Final - http://
www.infinispan.org/infinispan-objectfilter)
        * Infinispan Query (org.infinispan:infinispan-query:9.4.8.Final
- http://www.infinispan.org/infinispan-query)
        * Infinispan Query DSL API (org.infinispan:infinispan-query-
dsl:9.4.8.Final - http://www.infinispan.org/infinispan-query-dsl)
        * Infinispan Remote Query Client (org.infinispan:infinispan-
remote-query-client:9.4.8.Final - http://www.infinispan.org/infinispan-
remote-query-client)
        * ProtoStream - core
(org.infinispan.protostream:protostream:4.2.2.Final - http://
infinispan.org/protostream)
        * classfilewriter (org.jboss.classfilewriter:jboss-
classfilewriter:1.2.4.Final - https://github.com/jbossas/jboss-
classfilewriter)
        * JBoss Logging 3 (org.jboss.logging:jboss-logging:3.3.2.Final
- http://www.jboss.org)
        * JBoss Marshalling OSGi Bundle (org.jboss.marshalling:jboss-
marshalling-osgi:2.0.6.Final - http://www.jboss.org/jboss-marshalling-
parent/jboss-marshalling-osgi)
        * Weld APIs (org.jboss.weld:weld-api:3.1.SP1 - http://weld.cdi-
spec.org)
        * Weld Implementation (Core) (org.jboss.weld:weld-core-
impl:3.1.2.Final - http://weld.cdi-spec.org)
        * Weld SPIs for container integration (org.jboss.weld:weld-
spi:3.1.SP1 - http://weld.cdi-spec.org)
        * Weld Environment Common (org.jboss.weld.environment:weld-
environment-common:3.1.2.Final - http://weld.cdi-spec.org)
        * Weld JSF (org.jboss.weld.module:weld-jsf:3.1.2.Final - http://
weld.cdi-spec.org)
        * Weld Web (org.jboss.weld.module:weld-web:3.1.2.Final - http://
weld.cdi-spec.org)
        * Weld Probe Core (org.jboss.weld.probe:weld-probe-
core:3.1.2.Final - http://weld.cdi-spec.org/weld-core-parent/weld-probe-
parent/weld-probe-core)
        * Weld Servlet Core (org.jboss.weld.servlet:weld-servlet-
core:3.1.2.Final - http://weld.cdi-spec.org)
        * JGroups (org.jgroups:jgroups:4.0.18.Final - http://
www.jgroups.org)
        * MongoDB Java Driver (org.mongodb:mongo-java-driver:2.13.2 -
http://www.mongodb.org)
        * MortBay :: Apache EL :: API and Implementation
(org.mortbay.jasper:apache-el:8.5.40 - https://github.com/jetty-project/
jasper-jsp/apache-el)
        * MortBay :: Apache Jasper :: JSP Implementation
(org.mortbay.jasper:apache-jsp:8.5.40 - https://github.com/jetty-
project/jasper-jsp/apache-jsp)
        * org.opentest4j:opentest4j (org.opentest4j:opentest4j:1.2.0 -
```

```
https://github.com/ota4j-team/opentest4j)
        * Plexus Cipher: encryption/decryption Component
(org.sonatype.plexus:plexus-cipher:1.4 - http://spice.sonatype.org/
plexus-cipher)
        * Plexus Security Dispatcher Component
(org.sonatype.plexus:plexus-sec-dispatcher:1.4 - http://
spice.sonatype.org/plexus-sec-dispatcher)
        * Sisu - Guice (org.sonatype.sisu:sisu-guice:2.1.7 - http://
forge.sonatype.com/sisu-guice/)
        * Sisu - Inject (JSR330 bean support) (org.sonatype.sisu:sisu-
inject-bean:1.4.2 - http://sisu.sonatype.org/sisu-inject/guice-bean/
sisu-inject-bean/)
        * Sisu - Inject (Plexus bean support) (org.sonatype.sisu:sisu-
inject-plexus:1.4.2 - http://sisu.sonatype.org/sisu-inject/guice-bean/
guice-plexus/sisu-inject-plexus/)
        * Spring Beans (org.springframework:spring-beans:3.2.8.RELEASE
- https://github.com/SpringSource/spring-framework)
        * Spring Core (org.springframework:spring-core:3.2.8.RELEASE -
https://github.com/SpringSource/spring-framework)
        * Google Cloud Core (com.google.cloud:google-cloud-core:1.0.0 -
https://github.com/GoogleCloudPlatform/google-cloud-java/tree/master/
google-cloud-core)
        * Google Cloud Core HTTP (com.google.cloud:google-cloud-core-
http:1.0.0 - https://github.com/GoogleCloudPlatform/google-cloud-java/
tree/master/google-cloud-core-http)
        * Google Cloud Datastore (com.google.cloud:google-cloud-
datastore:1.0.0 - https://github.com/GoogleCloudPlatform/google-cloud-
java/tree/master/google-cloud-datastore)

See above for The Apache Software License, Version 2.0

================================================================================
=============

    The JSON License:

        * JSON in Java (org.json:json:20160810 - https://github.com/
douglascrockford/JSON-java)

Copyright (c) 2002 JSON.org

Permission is hereby granted, free of charge, to any person obtaining a
copy
of this software and associated documentation files (the "Software"),
to deal
in the Software without restriction, including without limitation the
rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included
in all
copies or substantial portions of the Software.
```

```
The Software shall be used for Good, not Evil.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
IN THE
SOFTWARE.
```

Javassist 3.27

```
Copyright (C) 1999-2019 by Shigeru Chiba, All rights reserved.
Contributors Bill Burke, Jason Green.

Javassist (JAVA programming ASSISTant) makes Java bytecode manipulation
simple. It is a class library for editing bytecodes in Java; it enables
Java programs to define a new class at runtime and to modify a class
file when the JVM loads it. Unlike other similar bytecode editors,
Javassist provides two levels of API: source level and bytecode level.
If the users use the source- level API, they can edit a class file
without knowledge of the specifications of the Java bytecode. The whole
API is designed with only the vocabulary of the Java language. You can
even specify inserted bytecode in the form of source text; Javassist
compiles it on the fly. On the other hand, the bytecode-level API
allows the users to directly edit a class file as other editors.

This software may distributed under the Mozilla Public License Version
1.1, the GNU Lesser General Public License Version 2.1 or later, or the
Apache License Version 2.0.Apache License. Oracle elects Apache 2.0

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the
copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other
entities that control, are controlled by, or are under common control
```

with that entity. For the purposes of this definition, "control" means
(i) the power, direct or indirect, to cause the direction or management
of such entity, whether by contract or otherwise, or (ii) ownership
of fifty percent (50%) or more of the outstanding shares, or (iii)
beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising
permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation, and
conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object
form, made available under the License, as indicated by a copyright
notice that is included in or attached to the work (an example is
provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the
original version of the Work and any modifications or additions to
that Work or Derivative Works thereof, that is intentionally submitted
to Licensor for inclusion in the Work by the copyright owner or by
an individual or Legal Entity authorized to submit on behalf of the
copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of,
the Licensor for the purpose of discussing and improving the Work,
but excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and
conditions of this License, each Contributor hereby grants to
You a perpetual, worldwide, non-exclusive, no-charge, royalty-free,
irrevocable copyright license to reproduce, prepare Derivative Works
of, publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

    You must give any other recipients of the Work or Derivative Works a copy of this License; and
    You must cause any modified files to carry prominent notices stating that You changed the files; and
    You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
    If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

    You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding

the above, nothing herein shall supersede or modify the terms of
any separate license agreement you may have executed with Licensor
regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each Contributor
provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR
CONDITIONS OF ANY KIND, either express or implied, including, without
limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT,
MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are
solely responsible for determining the appropriateness of using or
redistributing the Work and assume any risks associated with Your
exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise, unless
required by applicable law (such as deliberate and grossly negligent
acts) or agreed to in writing, shall any Contributor be liable to You
for damages, including any direct, indirect, special, incidental, or
consequential damages of any character arising as a result of this
License or out of the use or inability to use the Work (including but
not limited to damages for loss of goodwill, work stoppage, computer
failure or malfunction, or any and all other commercial damages or
losses), even if such Contributor has been advised of the possibility
of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the
Work or Derivative Works thereof, You may choose to offer, and charge a
fee for, acceptance of support, warranty, indemnity, or other liability
obligations and/or rights consistent with this License. However, in
accepting such obligations, You may act only on Your own behalf and
on Your sole responsibility, not on behalf of any other Contributor,
and only if You agree to indemnify, defend, and hold each Contributor
harmless for any liability incurred by, or claims asserted against,
such Contributor by reason of your accepting any such warranty or
additional liability.

END OF TERMS AND CONDITIONS

avsc 5.4.21

Copyright (c) 2015-2017, Matthieu Monsch.

Permission is hereby granted, free of charge, to any person obtaining a
copy of
this software and associated documentation files (the "Software"), to
deal in
the Software without restriction, including without limitation the

```
rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies
of the Software, and to permit persons to whom the Software is
furnished to do
so, subject to the following conditions:

The above copyright notice and this permission notice shall be included
in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
IN THE
SOFTWARE.
```

babel-polyfill 7.8.7

```
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.


4th party Dependencies
============================================================
Dependencies
```

```
================================================================
MIT
====
Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.

 @babel/cli
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
cli
publisher: Sebastian McKenzie
email: sebmck@gmail.com
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/code-frame
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
code-frame
publisher: Sebastian McKenzie
email: sebmck@gmail.com
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/core
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
core
publisher: Sebastian McKenzie
email: sebmck@gmail.com
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/generator
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
generator
publisher: Sebastian McKenzie
email: sebmck@gmail.com
Copyright (c) 2014-present Sebastian McKenzie and other contributors
```

```
 @babel/helper-annotate-as-pure
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
helper-annotate-as-pure
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/helper-builder-binary-assignment-operator-visitor
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
helper-builder-binary-assignment-operator-visitor
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/helper-call-delegate
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
helper-call-delegate
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/helper-create-class-features-plugin
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
helper-create-class-features-plugin
publisher: The Babel Team
url: https://babeljs.io/team
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/helper-define-map
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
helper-define-map
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/helper-explode-assignable-expression
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
helper-explode-assignable-expression
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/helper-function-name
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
helper-function-name
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/helper-get-function-arity
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
helper-get-function-arity
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/helper-hoist-variables
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
helper-hoist-variables
```

```
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/helper-member-expression-to-functions
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
helper-member-expression-to-functions
publisher: Justin Ridgewell
email: justin@ridgewell.name
Copyright (c) 2014-present Sebastian McKenzie and other contributors

licenseFile: d:\elearning\node_modules\@babel\helper-member-expression-
to-functions\LICENSE
 @babel/helper-module-imports
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
helper-module-imports
publisher: Logan Smyth
email: loganfsmyth@gmail.com
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/helper-module-transforms
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
helper-module-transforms
publisher: Logan Smyth
email: loganfsmyth@gmail.com
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/helper-optimise-call-expression
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
helper-optimise-call-expression
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/helper-plugin-utils
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
helper-plugin-utils
publisher: Logan Smyth
email: loganfsmyth@gmail.com
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/helper-regex
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
helper-regex
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/helper-remap-async-to-generator
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
helper-remap-async-to-generator
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/helper-replace-supers
```

```
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
helper-replace-supers
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/helper-simple-access
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
helper-simple-access
publisher: Logan Smyth
email: loganfsmyth@gmail.com
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/helper-split-export-declaration
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
helper-split-export-declaration
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/helper-wrap-function
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
helper-wrap-function
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/helpers
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
helpers
publisher: Sebastian McKenzie
email: sebmck@gmail.com
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/highlight
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
highlight
publisher: suchipi
email: me@suchipi.com
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/parser
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
parser
publisher: Sebastian McKenzie
email: sebmck@gmail.com
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-proposal-async-generator-functions
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-proposal-async-generator-functions
Copyright (c) 2014-present Sebastian McKenzie and other contributors
```

```
 @babel/plugin-proposal-class-properties
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-proposal-class-properties
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-proposal-json-strings
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-proposal-json-strings
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-proposal-object-rest-spread
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-proposal-object-rest-spread
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-proposal-optional-catch-binding
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-proposal-optional-catch-binding
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-proposal-unicode-property-regex
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-proposal-unicode-property-regex
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-syntax-async-generators
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-syntax-async-generators
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-syntax-json-strings
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-syntax-json-strings
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-syntax-object-rest-spread
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-syntax-object-rest-spread
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-syntax-optional-catch-binding
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-syntax-optional-catch-binding
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-syntax-typescript
```

```
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-syntax-typescript
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-arrow-functions
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-arrow-functions
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-async-to-generator
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-async-to-generator
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-block-scoped-functions
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-block-scoped-functions
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-block-scoping
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-block-scoping
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-classes
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-classes
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-computed-properties
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-computed-properties
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-destructuring
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-destructuring
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-dotall-regex
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-dotall-regex
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-duplicate-keys
licenses: MIT
```

```
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-duplicate-keys
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-exponentiation-operator
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-exponentiation-operator
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-for-of
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-for-of
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-function-name
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-function-name
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-literals
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-literals
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-member-expression-literals
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-member-expression-literals
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-modules-amd
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-modules-amd
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-modules-commonjs
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-modules-commonjs
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-modules-systemjs
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-modules-systemjs
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-modules-umd
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
```

```
plugin-transform-modules-umd
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-named-capturing-groups-regex
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-named-capturing-groups-regex
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-new-target
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-new-target
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-object-super
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-object-super
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-parameters
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-parameters
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-property-literals
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-property-literals
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-regenerator
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-regenerator
publisher: Ben Newman
email: bn@cs.stanford.edu
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-reserved-words
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-reserved-words
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-shorthand-properties
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-shorthand-properties
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-spread
licenses: MIT
```

```
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-spread
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-sticky-regex
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-sticky-regex
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-template-literals
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-template-literals
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-typeof-symbol
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-typeof-symbol
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-typescript
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-typescript
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/plugin-transform-unicode-regex
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
plugin-transform-unicode-regex
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/preset-env
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
preset-env
publisher: Henry Zhu
email: hi@henryzoo.com
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/preset-typescript
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
preset-typescript
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/template
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
template
publisher: Sebastian McKenzie
email: sebmck@gmail.com
Copyright (c) 2014-present Sebastian McKenzie and other contributors
```

```
 @babel/traverse
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
traverse
publisher: Sebastian McKenzie
email: sebmck@gmail.com
Copyright (c) 2014-present Sebastian McKenzie and other contributors

 @babel/types
licenses: MIT
repository: https://github.com/babel/babel/tree/master/packages/babel-
types
publisher: Sebastian McKenzie
email: sebmck@gmail.com
Copyright (c) 2014-present Sebastian McKenzie and other contributors

ansi-styles@3.2.1
licenses: MIT
repository: https://github.com/chalk/ansi-styles
publisher: Sindre Sorhus
email: sindresorhus@gmail.com
url: sindresorhus.com
Copyright (c) Sindre Sorhus  (sindresorhus.com)

 arr-diff@4.0.0
licenses: MIT
repository: https://github.com/jonschlinkert/arr-diff
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2017, Jon Schlinkert

 arr-flatten@1.1.0
licenses: MIT
repository: https://github.com/jonschlinkert/arr-flatten
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2017, Jon Schlinkert.

 arr-union@3.1.0
licenses: MIT
repository: https://github.com/jonschlinkert/arr-union
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2016, Jon Schlinkert.

 array-unique@0.3.2
licenses: MIT
repository: https://github.com/jonschlinkert/array-unique
publisher: Jon Schlinkert
Copyright (c) 2014-2016, Jon Schlinkert
Copyright (c) 2014-2016, Jon Schlinkert

 assign-symbols@1.0.0
licenses: MIT
```

```
repository: https://github.com/jonschlinkert/assign-symbols
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015-present, Jon Schlinkert.

 async-each@1.0.3
licenses: MIT
repository: https://github.com/paulmillr/async-each
publisher: Paul Miller
url: https://paulmillr.com/
Copyright (c) 2016 Paul Miller (paulmillr.com)

 atob@2.1.2
licenses: (MIT OR Apache-2.0)
repository: git://git.coolaj86.com/coolaj86/atob.js
publisher: AJ ONeal
email: coolaj86@gmail.com
url: https://coolaj86.com
Copyright (c) 2015 AJ ONeal

 balanced-match@1.0.0
licenses: MIT
repository: https://github.com/juliangruber/balanced-match
publisher: Julian Gruber
email: mail@juliangruber.com
url: http://juliangruber.com
Copyright (c) 2013 Julian Gruber

 base@0.11.2
licenses: MIT
repository: https://github.com/node-base/base
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015-2017, Jon Schlinkert.

 binary-extensions@1.13.1
licenses: MIT
repository: https://github.com/sindresorhus/binary-extensions
publisher: Sindre Sorhus
email: sindresorhus@gmail.com
url: sindresorhus.com
Copyright (c) 2019 Sindre Sorhus  (https://sindresorhus.com), Paul
Miller (https://paulmillr.com)

 brace-expansion@1.1.11
licenses: MIT
repository: https://github.com/juliangruber/brace-expansion
publisher: Julian Gruber
email: mail@juliangruber.com
url: http://juliangruber.com
Copyright (c) 2013 Julian Gruber

 braces@2.3.2
licenses: MIT
repository: https://github.com/micromatch/braces
```

```
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2018, Jon Schlinkert.

 browserslist@4.5.6
licenses: MIT
repository: https://github.com/browserslist/browserslist
publisher: Andrey Sitnik
email: andrey@sitnik.ru
Copyright 2014 Andrey Sitnik

 cache-base@1.0.1
licenses: MIT
repository: https://github.com/jonschlinkert/cache-base
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2018, Jon Schlinkert.

 chalk@2.4.2
licenses: MIT
repository: https://github.com/chalk/chalk
Copyright (c) Sindre Sorhus  (sindresorhus.com)

 chokidar@2.1.5
licenses: MIT
repository: https://github.com/paulmillr/chokidar
publisher: Paul Miller
url: https://paulmillr.com
Copyright (c) 2012-2019 Paul Miller (https://paulmillr.com), Elan
Shanker

 class-utils@0.3.6
licenses: MIT
repository: https://github.com/jonschlinkert/class-utils
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015, 2017-2018, Jon Schlinkert.

 collection-visit@1.0.0
licenses: MIT
repository: https://github.com/jonschlinkert/collection-visit
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015-2016, Jon Schlinkert

 color-convert@1.9.3
licenses: MIT
repository: https://github.com/Qix-/color-convert
publisher: Heather Arthur
email: fayearthur@gmail.com
Copyright (c) 2011-2016 Heather Arthur

 color-name@1.1.3
licenses: MIT
repository: https://github.com/dfcreative/color-name
```

```
publisher: DY
email: dfcreative@gmail.com
Copyright (c) 2015 Dmitry Ivanov

 commander@2.20.0
licenses: MIT
repository: https://github.com/tj/commander.js
publisher: TJ Holowaychuk
email: tj@vision-media.ca
Copyright (c) 2011 TJ Holowaychuk

 component-emitter@1.3.0
licenses: MIT
repository: https://github.com/component/emitter
Copyright (c) 2014 Component contributors

 concat-map@0.0.1
licenses: MIT
repository: https://github.com/substack/node-concat-map
publisher: James Halliday
email: mail@substack.net
url: http://substack.net
No Copyright

 convert-source-map@1.6.0
licenses: MIT
repository: https://github.com/thlorenz/convert-source-map
publisher: Thorsten Lorenz
email: thlorenz@gmx.de
url: http://thlorenz.com
Copyright 2013 Thorsten Lorenz.

 copy-descriptor@0.1.1
licenses: MIT
repository: https://github.com/jonschlinkert/copy-descriptor
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015-2016, Jon Schlinkert

 core-js-compat@3.0.1
licenses: MIT
repository: https://github.com/zloirock/core-js
Copyright (c) 2014-2019 Denis Pushkarev

 core-js-pure@3.0.1
licenses: MIT
repository: https://github.com/zloirock/core-js
Copyright (c) 2014-2019 Denis Pushkarev

 core-js@3.0.1
licenses: MIT
repository: https://github.com/zloirock/core-js
Copyright (c) 2014-2019 Denis Pushkarev

 core-util-is@1.0.2
```

```
licenses: MIT
repository: https://github.com/isaacs/core-util-is
publisher: Isaac Z. Schlueter
email: i@izs.me
url: http://blog.izs.me/
Copyright Node.js contributors. All rights reserved

 debug@2.6.9
licenses: MIT
repository: https://github.com/visionmedia/debug
publisher: TJ Holowaychuk
email: tj@vision-media.ca
Copyright (c) 2014 TJ Holowaychuk

 debug@4.1.1
licenses: MIT
repository: https://github.com/visionmedia/debug
publisher: TJ Holowaychuk
email: tj@vision-media.ca
Copyright (c) 2014 TJ Holowaychuk

 decode-uri-component@0.2.0
licenses: MIT
repository: https://github.com/SamVerschueren/decode-uri-component
publisher: Sam Verschueren
email: sam.verschueren@gmail.com
url: github.com/SamVerschueren
Copyright (c) Sam Verschueren  (github.com/SamVerschueren)

 define-property@0.2.5
licenses: MIT
repository: https://github.com/jonschlinkert/define-property
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015-2018, Jon Schlinkert.

 define-property@1.0.0
licenses: MIT
repository: https://github.com/jonschlinkert/define-property
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015-2018, Jon Schlinkert.

 define-property@2.0.2
licenses: MIT
repository: https://github.com/jonschlinkert/define-property
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015-2018, Jon Schlinkert.

 escape-string-regexp@1.0.5
licenses: MIT
repository: https://github.com/sindresorhus/escape-string-regexp
publisher: Sindre Sorhus
email: sindresorhus@gmail.com
```

```
url: sindresorhus.com
Copyright (c) Sindre Sorhus  (sindresorhus.com)

 expand-brackets@2.1.4
licenses: MIT
repository: https://github.com/jonschlinkert/expand-brackets
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015-2018, Jon Schlinkert

 extend-shallow@2.0.1
licenses: MIT
repository: https://github.com/jonschlinkert/extend-shallow
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2015, 2017, Jon Schlinkert.

 extend-shallow@3.0.2
licenses: MIT
repository: https://github.com/jonschlinkert/extend-shallow
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2015, 2017, Jon Schlinkert.

 extglob@2.0.4
licenses: MIT
repository: https://github.com/micromatch/extglob
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015-2018, Jon Schlinkert.

 fill-range@4.0.0
licenses: MIT
repository: https://github.com/jonschlinkert/fill-range
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-present, Jon Schlinkert.

 for-in@1.0.2
licenses: MIT
repository: https://github.com/jonschlinkert/for-in
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2017, Jon Schlinkert

 fragment-cache@0.2.1
licenses: MIT
repository: https://github.com/jonschlinkert/fragment-cache
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2016-2017, Jon Schlinkert

 fs-readdir-recursive@1.1.0
licenses: MIT
repository: https://github.com/fs-utils/fs-readdir-recursive
```

```
publisher: Jonathan Ong
email: me@jongleberry.com
url: http://jongleberry.com
Copyright (c) 2014 Jonathan Ong me@jongleberry.com

 get-value@2.0.6
licenses: MIT
repository: https://github.com/jonschlinkert/get-value
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2018, Jon Schlinkert.

 globals@11.12.0
licenses: MIT
repository: https://github.com/sindresorhus/globals
publisher: Sindre Sorhus
email: sindresorhus@gmail.com
url: sindresorhus.com
Copyright (c) Sindre Sorhus  (sindresorhus.com)

 has-flag@3.0.0
licenses: MIT
repository: https://github.com/sindresorhus/has-flag
publisher: Sindre Sorhus
email: sindresorhus@gmail.com
url: sindresorhus.com
Copyright (c) Sindre Sorhus  (sindresorhus.com)

 has-value@0.3.1
licenses: MIT
repository: https://github.com/jonschlinkert/has-value
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2018, Jon Schlinkert.

 has-value@1.0.0
licenses: MIT
repository: https://github.com/jonschlinkert/has-value
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2018, Jon Schlinkert.

 has-values@0.1.4
licenses: MIT
repository: https://github.com/jonschlinkert/has-values
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2018, Jon Schlinkert.

 has-values@1.0.0
licenses: MIT
repository: https://github.com/jonschlinkert/has-values
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2018, Jon Schlinkert.
```

```
 invariant@2.2.4
licenses: MIT
repository: https://github.com/zertosh/invariant
publisher: Andres Suarez
email: zertosh@gmail.com
Copyright (c) 2013-present, Facebook, Inc.

 is-accessor-descriptor@0.1.6
licenses: MIT
repository: https://github.com/jonschlinkert/is-accessor-descriptor
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015-present, Jon Schlinkert.

 is-accessor-descriptor@1.0.0
licenses: MIT
repository: https://github.com/jonschlinkert/is-accessor-descriptor
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015-present, Jon Schlinkert.

 is-binary-path@1.0.1
licenses: MIT
repository: https://github.com/sindresorhus/is-binary-path
publisher: Sindre Sorhus
email: sindresorhus@gmail.com
url: sindresorhus.com
Copyright (c) 2019 Sindre Sorhus  (https://sindresorhus.com), Paul
Miller (https://paulmillr.com)

 is-buffer@1.1.6
licenses: MIT
repository: https://github.com/feross/is-buffer
publisher: Feross Aboukhadijeh
email: feross@feross.org
url: http://feross.org/
Copyright (c) Feross Aboukhadijeh

 is-data-descriptor@0.1.4
licenses: MIT
repository: https://github.com/jonschlinkert/is-data-descriptor
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015-present, Jon Schlinkert.

 is-data-descriptor@1.0.0
licenses: MIT
repository: https://github.com/jonschlinkert/is-data-descriptor
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015-present, Jon Schlinkert.

 is-descriptor@0.1.6
licenses: MIT
```

```
repository: https://github.com/jonschlinkert/is-descriptor
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015-present, Jon Schlinkert.

 is-descriptor@1.0.2
licenses: MIT
repository: https://github.com/jonschlinkert/is-descriptor
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015-present, Jon Schlinkert.

 is-extendable@0.1.1
licenses: MIT
repository: https://github.com/jonschlinkert/is-extendable
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015-2017, Jon Schlinkert.

 is-extendable@1.0.1
licenses: MIT
repository: https://github.com/jonschlinkert/is-extendable
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015-2017, Jon Schlinkert.

 is-extglob@2.1.1
licenses: MIT
repository: https://github.com/jonschlinkert/is-extglob
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2016, Jon Schlinkert

 is-glob@3.1.0
licenses: MIT
repository: https://github.com/jonschlinkert/is-glob
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2017, Jon Schlinkert.

 is-glob@4.0.1
licenses: MIT
repository: https://github.com/micromatch/is-glob
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2017, Jon Schlinkert.

 is-number@3.0.0
licenses: MIT
repository: https://github.com/jonschlinkert/is-number
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-present, Jon Schlinkert.

 is-plain-obj@1.1.0
```

```
licenses: MIT
repository: https://github.com/sindresorhus/is-plain-obj
publisher: Sindre Sorhus
email: sindresorhus@gmail.com
url: sindresorhus.com
Copyright (c) Sindre Sorhus  (sindresorhus.com)

 is-plain-object@2.0.4
licenses: MIT
repository: https://github.com/jonschlinkert/is-plain-object
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2017, Jon Schlinkert.

 is-windows@1.0.2
licenses: MIT
repository: https://github.com/jonschlinkert/is-windows
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015-2018, Jon Schlinkert.

 isarray@1.0.0
licenses: MIT
repository: https://github.com/juliangruber/isarray
publisher: Julian Gruber
email: mail@juliangruber.com
url: http://juliangruber.com
Copyright (c) 2013 Julian Gruber

 isobject@2.1.0
licenses: MIT
repository: https://github.com/jonschlinkert/isobject
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2017, Jon Schlinkert.

 isobject@3.0.1
licenses: MIT
repository: https://github.com/jonschlinkert/isobject
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2017, Jon Schlinkert.

 js-levenshtein@1.1.6
licenses: MIT
repository: https://github.com/gustf/js-levenshtein
publisher: Gustaf Andersson
email: gustaf@me.com
Copyright (c) 2017 Gustaf Andersson

 js-tokens@4.0.0
licenses: MIT
repository: https://github.com/lydell/js-tokens
publisher: Simon Lydell
Copyright (c) 2014, 2015, 2016, 2017, 2018 Simon Lydell
```

```
 jsesc@0.5.0
licenses: MIT
repository: https://github.com/mathiasbynens/jsesc
publisher: Mathias Bynens
url: http://mathiasbynens.be/
Copyright Mathias Bynens

 jsesc@2.5.2
licenses: MIT
repository: https://github.com/mathiasbynens/jsesc
publisher: Mathias Bynens
url: https://mathiasbynens.be/
Copyright Mathias Bynens

 json5@2.1.0
licenses: MIT
repository: https://github.com/json5/json5
publisher: Aseem Kishore
email: aseem.kishore@gmail.com
Copyright (c) 2012-2018 Aseem Kishore, and others.

 kind-of@3.2.2
licenses: MIT
repository: https://github.com/jonschlinkert/kind-of
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2017, Jon Schlinkert.

 kind-of@4.0.0
licenses: MIT
repository: https://github.com/jonschlinkert/kind-of
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2017, Jon Schlinkert.

 kind-of@5.1.0
licenses: MIT
repository: https://github.com/jonschlinkert/kind-of
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2017, Jon Schlinkert.

 kind-of@6.0.2
licenses: MIT
repository: https://github.com/jonschlinkert/kind-of
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2017, Jon Schlinkert.

 lodash@4.17.11
licenses: MIT
repository: https://github.com/lodash/lodash
publisher: John-David Dalton
email: john.david.dalton@gmail.com
```

```
url: http://allyoucanleet.com/
Copyright JS Foundation and other contributors

 loose-envify@1.4.0
licenses: MIT
repository: https://github.com/zertosh/loose-envify
publisher: Andres Suarez
email: zertosh@gmail.com
Copyright (c) 2015 Andres Suarez

 map-cache@0.2.2
licenses: MIT
repository: https://github.com/jonschlinkert/map-cache
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015-2016, Jon Schlinkert.

 map-visit@1.0.0
licenses: MIT
repository: https://github.com/jonschlinkert/map-visit
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015-2016, Jon Schlinkert

 micromatch@3.1.10
licenses: MIT
repository: https://github.com/micromatch/micromatch
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-present, Jon Schlinkert.

 minimist@0.0.8
licenses: MIT
repository: https://github.com/substack/minimist
publisher: James Halliday
email: mail@substack.net
url: http://substack.net
No Copyright info

 minimist@1.2.0
licenses: MIT
repository: https://github.com/substack/minimist
publisher: James Halliday
email: mail@substack.net
url: http://substack.net
No Copyright info

 mixin-deep@1.3.1
licenses: MIT
repository: https://github.com/jonschlinkert/mixin-deep
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-present, Jon Schlinkert.

 mkdirp@0.5.1
```

```
licenses: MIT
repository: https://github.com/substack/node-mkdirp
publisher: James Halliday
email: mail@substack.net
url: http://substack.net
Copyright 2010 James Halliday (mail@substack.net)

 ms@2.0.0
licenses: MIT
repository: https://github.com/zeit/ms
Copyright (c) 2016 Zeit, Inc.

 ms@2.1.1
licenses: MIT
repository: https://github.com/zeit/ms
Copyright (c) 2016 Zeit, Inc.

 nanomatch@1.2.13
licenses: MIT
repository: https://github.com/micromatch/nanomatch
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2016-2018, Jon Schlinkert.

 node-releases@1.1.17
licenses: MIT
repository: https://github.com/chicoxyzzy/node-releases
publisher: Sergey Rubanov
email: chi187@gmail.com
Copyright (c) 2017 Sergey Rubanov (https://github.com/chicoxyzzy)

 normalize-path@2.1.1
licenses: MIT
repository: https://github.com/jonschlinkert/normalize-path
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2018, Jon Schlinkert.

 normalize-path@3.0.0
licenses: MIT
repository: https://github.com/jonschlinkert/normalize-path
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2018, Jon Schlinkert.

 object-copy@0.1.0
licenses: MIT
repository: https://github.com/jonschlinkert/object-copy
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2016-2017, Jon Schlinkert

 object-visit@1.0.1
licenses: MIT
repository: https://github.com/jonschlinkert/object-visit
```

```
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015, 2017, Jon Schlinkert

 object.pick@1.3.0
licenses: MIT
repository: https://github.com/jonschlinkert/object.pick
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2016, Jon Schlinkert.

 pascalcase@0.1.1
licenses: MIT
repository: https://github.com/jonschlinkert/pascalcase
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015, Jon Schlinkert.

 path-dirname@1.0.2
licenses: MIT
repository: https://github.com/es128/path-dirname
publisher: Elan Shanker
Copyright (c) Elan Shanker and Node.js contributors. All rights
reserved.

 path-is-absolute@1.0.1
licenses: MIT
repository: https://github.com/sindresorhus/path-is-absolute
publisher: Sindre Sorhus
email: sindresorhus@gmail.com
url: sindresorhus.com
Copyright (c) Sindre Sorhus  (sindresorhus.com)

 path-parse@1.0.6
licenses: MIT
repository: https://github.com/jbgutierrez/path-parse
publisher: Javier Blanco
email: http://jbgutierrez.info
Copyright (c) 2015 Javier Blanco

 posix-character-classes@0.1.1
licenses: MIT
repository: https://github.com/jonschlinkert/posix-character-classes
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2016-2017, Jon Schlinkert

 private@0.1.8
licenses: MIT
repository: https://github.com/benjamn/private
publisher: Ben Newman
email: bn@cs.stanford.edu
Copyright (c) 2014 Ben Newman

 process-nextick-args@2.0.0
```

```
licenses: MIT
repository: https://github.com/calvinmetcalf/process-nextick-args
Copyright (c) 2015 Calvin Metcalf

 readable-stream@2.3.6
licenses: MIT
repository: https://github.com/nodejs/readable-stream
Copyright Node.js contributors. All rights reserved.

 readdirp@2.2.1
licenses: MIT
repository: https://github.com/paulmillr/readdirp
publisher: Thorsten Lorenz
email: thlorenz@gmx.de
url: thlorenz.com
Copyright (c) 2012-2019 Thorsten Lorenz, Paul Miller (https://
paulmillr.com)

 regenerate-unicode-properties@8.0.2
licenses: MIT
repository: https://github.com/mathiasbynens/regenerate-unicode-
properties
publisher: Mathias Bynens
url: https://mathiasbynens.be/
Copyright Mathias Bynens

 regenerate@1.4.0
licenses: MIT
repository: https://github.com/mathiasbynens/regenerate
publisher: Mathias Bynens
url: https://mathiasbynens.be/
Copyright Mathias Bynens

 regenerator-transform@0.13.4
licenses: MIT
repository: https://github.com/facebook/regenerator/tree/master/
packages/regenerator-transform
publisher: Ben Newman
email: bn@cs.stanford.edu
Copyright (c) 2014-present, Facebook, Inc.

 regex-not@1.0.2
licenses: MIT
repository: https://github.com/jonschlinkert/regex-not
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2016, 2018, Jon Schlinkert.

 regexp-tree@0.1.6
licenses: MIT
repository: https://github.com/DmitrySoshnikov/regexp-tree
publisher: Dmitry Soshnikov
Copyright (c) 2017 Dmitry Soshnikov

 regexpu-core@4.5.4
```

```
licenses: MIT
repository: https://github.com/mathiasbynens/regexpu-core
publisher: Mathias Bynens
url: https://mathiasbynens.be/
Copyright Mathias Bynens

 regjsgen@0.5.0
licenses: MIT
repository: https://github.com/bnjmnt4n/regjsgen
publisher: Benjamin Tan
url: https://bnjmnt4n.now.sh/
Copyright 2014-2018 Benjamin Tan

 repeat-element@1.1.3
licenses: MIT
repository: https://github.com/jonschlinkert/repeat-element
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015-present, Jon Schlinkert.

 repeat-string@1.6.1
licenses: MIT
repository: https://github.com/jonschlinkert/repeat-string
publisher: Jon Schlinkert
url: http://github.com/jonschlinkert
Copyright (c) 2014-2015, Jon Schlinkert.

 resolve-url@0.2.1
licenses: MIT
repository: https://github.com/lydell/resolve-url
publisher: Simon Lydell
Copyright (c) 2014 Simon Lydell

 resolve@1.10.1
licenses: MIT
repository: https://github.com/browserify/resolve
publisher: James Halliday
email: mail@substack.net
url: http://substack.net
No Copyright Info

 ret@0.1.15
licenses: MIT
repository: https://github.com/fent/ret.js
publisher: Roly Fentanes
url: https://github.com/fent
https://github.com/fent/ret.js

 safe-buffer@5.1.2
licenses: MIT
repository: https://github.com/feross/safe-buffer
publisher: Feross Aboukhadijeh
email: feross@feross.org
url: http://feross.org
Copyright (c) Feross Aboukhadijeh
```

```
 safe-regex@1.1.0
licenses: MIT
repository: https://github.com/substack/safe-regex
publisher: James Halliday
email: mail@substack.net
url: http://substack.net
https://github.com/substack/safe-regex

 set-value@0.4.3
licenses: MIT
repository: https://github.com/jonschlinkert/set-value
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2018, Jon Schlinkert.

 set-value@2.0.0
licenses: MIT
repository: https://github.com/jonschlinkert/set-value
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2014-2018, Jon Schlinkert.

 slash@2.0.0
licenses: MIT
repository: https://github.com/sindresorhus/slash
publisher: Sindre Sorhus
email: sindresorhus@gmail.com
url: sindresorhus.com
Copyright (c) Sindre Sorhus  (sindresorhus.com)

 snapdragon-node@2.1.1
licenses: MIT
repository: https://github.com/jonschlinkert/snapdragon-node
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2017-present, Jon Schlinkert.

 snapdragon-util@3.0.1
licenses: MIT
repository: https://github.com/jonschlinkert/snapdragon-util
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2017-2018, Jon Schlinkert.

 snapdragon@0.8.2
licenses: MIT
repository: https://github.com/jonschlinkert/snapdragon
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2017-2018, Jon Schlinkert.

 source-map-resolve@0.5.2
licenses: MIT
repository: https://github.com/lydell/source-map-resolve
```

```
publisher: Simon Lydell
Copyright (c) 2014, 2015, 2016, 2017 Simon Lydell

 source-map-url@0.4.0
licenses: MIT
repository: https://github.com/lydell/source-map-url
publisher: Simon Lydell
Copyright (c) 2014 Simon Lydell

 split-string@3.1.0
licenses: MIT
repository: https://github.com/jonschlinkert/split-string
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015-present, Jon Schlinkert.

 static-extend@0.1.2
licenses: MIT
repository: https://github.com/jonschlinkert/static-extend
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2016, Jon Schlinkert.

 string_decoder@1.1.1
licenses: MIT
repository: https://github.com/nodejs/string_decoder
Copyright Node.js contributors. All rights reserved.

 supports-color@5.5.0
licenses: MIT
repository: https://github.com/chalk/supports-color
publisher: Sindre Sorhus
email: sindresorhus@gmail.com
url: sindresorhus.com
Copyright (c) Sindre Sorhus  (sindresorhus.com)

 to-fast-properties@2.0.0
licenses: MIT
repository: https://github.com/sindresorhus/to-fast-properties
publisher: Sindre Sorhus
email: sindresorhus@gmail.com
url: sindresorhus.com
Copyright (c) 2014 Petka Antonov
2015 Sindre Sorhus

 to-object-path@0.3.0
licenses: MIT
repository: https://github.com/jonschlinkert/to-object-path
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015-2016, Jon Schlinkert.

 to-regex-range@2.1.1
licenses: MIT
repository: https://github.com/micromatch/to-regex-range
```

```
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015-present, Jon Schlinkert.

 to-regex@3.0.2
licenses: MIT
repository: https://github.com/jonschlinkert/to-regex
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2016-2018, Jon Schlinkert.

 trim-right@1.0.1
licenses: MIT
repository: https://github.com/sindresorhus/trim-right
publisher: Sindre Sorhus
email: sindresorhus@gmail.com
url: sindresorhus.com
Copyright (c) Sindre Sorhus  (sindresorhus.com)

 unicode-canonical-property-names-ecmascript@1.0.4
licenses: MIT
repository: https://github.com/mathiasbynens/unicode-canonical-property-
names-ecmascript
publisher: Mathias Bynens
url: https://mathiasbynens.be/
Copyright Mathias Bynens

 unicode-match-property-ecmascript@1.0.4
licenses: MIT
repository: https://github.com/mathiasbynens/unicode-match-property-
ecmascript
publisher: Mathias Bynens
url: https://mathiasbynens.be/
Copyright Mathias Bynens

 unicode-match-property-value-ecmascript@1.1.0
licenses: MIT
repository: https://github.com/mathiasbynens/unicode-match-property-
value-ecmascript
publisher: Mathias Bynens
url: https://mathiasbynens.be/
Copyright Mathias Bynens

 unicode-property-aliases-ecmascript@1.0.5
licenses: MIT
repository: https://github.com/mathiasbynens/unicode-property-aliases-
ecmascript
publisher: Mathias Bynens
url: https://mathiasbynens.be/
Copyright Mathias Bynens

 union-value@1.0.0
licenses: MIT
repository: https://github.com/jonschlinkert/union-value
publisher: Jon Schlinkert
```

```
url: https://github.com/jonschlinkert
Copyright (c) 2015-present, Jon Schlinkert.

 unset-value@1.0.0
licenses: MIT
repository: https://github.com/jonschlinkert/unset-value
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015, 2017, Jon Schlinkert

 upath@1.1.2
licenses: MIT
repository: https://github.com/anodynos/upath
publisher: Angelos Pikoulas
email: agelos.pikoulas@gmail.com
Copyright(c) 2014-2017 Angelos Pikoulas (agelos.pikoulas@gmail.com)

 urix@0.1.0
licenses: MIT
repository: https://github.com/lydell/urix
publisher: Simon Lydell
Copyright (c) 2013 Simon Lydell

 use@3.1.1
licenses: MIT
repository: https://github.com/jonschlinkert/use
publisher: Jon Schlinkert
url: https://github.com/jonschlinkert
Copyright (c) 2015-present, Jon Schlinkert.

 util-deprecate@1.0.2
licenses: MIT
repository: https://github.com/TooTallNate/util-deprecate
publisher: Nathan Rajlich
email: nathan@tootallnate.net
url: http://n8.io/
Copyright (c) 2014 Nathan Rajlich

ISC
======
The ISC License

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR
IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
```

```
 anymatch@2.0.0
licenses: ISC
repository: https://github.com/micromatch/anymatch
publisher: Elan Shanker
url: http://github.com/es128
Copyright (c) 2014 Elan Shanker

 electron-to-chromium@1.3.131
licenses: ISC
repository: https://github.com/kilian/electron-to-chromium
publisher: Kilian Valkhof
Copyright 2018 Kilian Valkhof

 fs.realpath@1.0.0
licenses: ISC
repository: https://github.com/isaacs/fs.realpath
publisher: Isaac Z. Schlueter
email: i@izs.me
url: http://blog.izs.me/
Copyright (c) Isaac Z. Schlueter and Contributors

 glob-parent@3.1.0
licenses: ISC
repository: https://github.com/es128/glob-parent
publisher: Elan Shanker
url: https://github.com/es128
Copyright (c) 2015, 2019 Elan Shanker

 glob@7.1.3
licenses: ISC
repository: https://github.com/isaacs/node-glob
publisher: Isaac Z. Schlueter
email: i@izs.me
url: http://blog.izs.me/
Copyright (c) Isaac Z. Schlueter and Contributors

 graceful-fs@4.1.15
licenses: ISC
repository: https://github.com/isaacs/node-graceful-fs
Copyright (c) Isaac Z. Schlueter, Ben Noordhuis, and Contributors

 inflight@1.0.6
licenses: ISC
repository: https://github.com/npm/inflight
publisher: Isaac Z. Schlueter
email: i@izs.me
url: http://blog.izs.me/
Copyright (c) Isaac Z. Schlueter

 inherits@2.0.3
licenses: ISC
repository: https://github.com/isaacs/inherits
Copyright (c) Isaac Z. Schlueter

 minimatch@3.0.4
```

```
licenses: ISC
repository: https://github.com/isaacs/minimatch
publisher: Isaac Z. Schlueter
email: i@izs.me
url: http://blog.izs.me
Copyright (c) Isaac Z. Schlueter and Contributors

 once@1.4.0
licenses: ISC
repository: https://github.com/isaacs/once
publisher: Isaac Z. Schlueter
email: i@izs.me
url: http://blog.izs.me/
Copyright (c) Isaac Z. Schlueter and Contributors

 output-file-sync@2.0.1
licenses: ISC
repository: https://github.com/shinnn/output-file-sync
publisher: Shinnosuke Watanabe
url: https://github.com/shinnn
Copyright 2017 - 2018 Shinnosuke Watanabe

 remove-trailing-separator@1.1.0
licenses: ISC
repository: https://github.com/darsain/remove-trailing-separator
publisher: darsain
Copyright (c) 2017 Tomas Sardyha

 semver@5.7.0
licenses: ISC
repository: https://github.com/npm/node-semver
https://github.com/npm/node-semver

 semver@6.0.0
licenses: ISC
repository: https://github.com/npm/node-semver
Copyright (c) Isaac Z. Schlueter and Contributors

wrappy@1.0.2
licenses: ISC
repository: https://github.com/npm/wrappy
publisher: Isaac Z. Schlueter
email: i@izs.me
url: http://blog.izs.me/
Copyright (c) Isaac Z. Schlueter and Contributors

BSD
=========
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

  * Redistributions of source code must retain the above copyright
    notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above copyright
```

```
    notice, this list of conditions and the following disclaimer in the
    documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL  BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.


 esutils@2.0.2
licenses: BSD
repository: https://github.com/estools/esutils
Copyright (C) 2013 Yusuke Suzuki (twitter: @Constellation) and other
contributors.

 regjsparser@0.6.0
licenses: BSD-2-Clause
repository: https://github.com/jviereck/regjsparser
publisher: 'Julian Viereck'
email: julian.viereck@gmail.com
Copyright (c) Julian Viereck and Contributors, All Rights Reserved.

 source-map@0.5.7
licenses: BSD-3-Clause
repository: https://github.com/mozilla/source-map
publisher: Nick Fitzgerald
email: nfitzgerald@mozilla.com
Copyright (c) 2009-2011, Mozilla Foundation and contributors


CC-BY-4.0
=============
Attribution 4.0 International

=======================================================================

Creative Commons Corporation ("Creative Commons") is not a law firm and
does not provide legal services or legal advice. Distribution of
Creative Commons public licenses does not create a lawyer-client or
other relationship. Creative Commons makes its licenses and related
information available on an "as-is" basis. Creative Commons gives no
warranties regarding its licenses, any material licensed under their
```

terms and conditions, or any related information. Creative Commons disclaims all liability for damages resulting from their use to the fullest extent possible.

Using Creative Commons Public Licenses

Creative Commons public licenses provide a standard set of terms and conditions that creators and other rights holders may use to share original works of authorship and other material subject to copyright and certain other rights specified in the public license below. The following considerations are for informational purposes only, are not exhaustive, and do not form part of our licenses.

Considerations for licensors: Our public licenses are intended for use by those authorized to give the public permission to use material in ways otherwise restricted by copyright and certain other rights. Our licenses are irrevocable. Licensors should read and understand the terms and conditions of the license they choose before applying it. Licensors should also secure all rights necessary before applying our licenses so that the public can reuse the material as expected. Licensors should clearly mark any material not subject to the license. This includes other CC-licensed material, or material used under an exception or limitation to copyright. More considerations for licensors: wiki.creativecommons.org/Considerations_for_licensors

Considerations for the public: By using one of our public licenses, a licensor grants the public permission to use the licensed material under specified terms and conditions. If the licensor's permission is not necessary for any reason--for example, because of any applicable exception or limitation to copyright--then that use is not regulated by the license. Our licenses grant only permissions under copyright and certain other rights that a licensor has authority to grant. Use of the licensed material may still be restricted for other reasons, including because others have copyright or other rights in the material. A licensor may make special requests, such as asking that all changes be marked or described. Although not required by our licenses, you are encouraged to respect those requests where reasonable. More_considerations for the public: wiki.creativecommons.org/Considerations_for_licensees

==========

Creative Commons Attribution 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the

Licensed Material available under these terms and conditions.

Section 1 -- Definitions.

a. Adapted Material means material subject to Copyright and Similar Rights that is derived from or based upon the Licensed Material and in which the Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image.

b. Adapter's License means the license You apply to Your Copyright and Similar Rights in Your contributions to Adapted Material in accordance with the terms and conditions of this Public License.

c. Copyright and Similar Rights means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section 2(b)(1)-(2) are not Copyright and Similar Rights.

d. Effective Technological Measures means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.

e. Exceptions and Limitations means fair use, fair dealing, and/or any other exception or limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.

f. Licensed Material means the artistic or literary work, database, or other material to which the Licensor applied this Public License.

g. Licensed Rights means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.

h. Licensor means the individual(s) or entity(ies) granting rights under this Public License.

i. Share means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time

individually chosen by them.

j. Sui Generis Database Rights means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.

k. You means the individual or entity exercising the Licensed Rights under this Public License. Your has a corresponding meaning.

Section 2 -- Scope.

a. License grant.

1. Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights in the Licensed Material to:

a. reproduce and Share the Licensed Material, in whole or in part; and

b. produce, reproduce, and Share Adapted Material.

2. Exceptions and Limitations. For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions.

3. Term. The term of this Public License is specified in Section 6(a).

4. Media and formats; technical modifications allowed. The Licensor authorizes You to exercise the Licensed Rights in all media and formats whether now known or hereafter created, and to make technical modifications necessary to do so. The Licensor waives and/or agrees not to assert any right or authority to forbid You from making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures. For purposes of this Public License, simply making modifications authorized by this Section 2(a) (4) never produces Adapted Material.

5. Downstream recipients.

a. Offer from the Licensor -- Licensed Material. Every recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License.

b. No downstream restrictions. You may not offer or impose

any additional or different terms or conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any recipient of the Licensed Material.

6. No endorsement. Nothing in this Public License constitutes or may be construed as permission to assert or imply that You are, or that Your use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as provided in Section 3(a)(1)(A)(i).

b. Other rights.

1. Moral rights, such as the right of integrity, are not licensed under this Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise.

2. Patent and trademark rights are not licensed under this Public License.

3. To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties.

Section 3 -- License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

a. Attribution.

1. If You Share the Licensed Material (including in modified form), You must:

a. retain the following if it is supplied by the Licensor with the Licensed Material:

i. identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor (including by pseudonym if designated);

ii. a copyright notice;

iii. a notice that refers to this Public License;

iv. a notice that refers to the disclaimer of warranties;

v. a URI or hyperlink to the Licensed Material to the extent reasonably practicable;

b. indicate if You modified the Licensed Material and retain an indication of any previous modifications; and

c. indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License.

2. You may satisfy the conditions in Section 3(a)(1) in any reasonable manner based on the medium, means, and context in which You Share the Licensed Material. For example, it may be reasonable to satisfy the conditions by providing a URI or hyperlink to a resource that includes the required information.

3. If requested by the Licensor, You must remove any of the information required by Section 3(a)(1)(A) to the extent reasonably practicable.

4. If You Share Adapted Material You produce, the Adapter's License You apply must not prevent recipients of the Adapted Material from complying with this Public License.

Section 4 -- Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

a. for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database;

b. if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui Generis Database Rights (but not its individual contents) is Adapted Material; and

c. You must comply with the conditions in Section 3(a) if You Share all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights.

Section 5 -- Disclaimer of Warranties and Limitation of Liability.

a. UNLESS OTHERWISE SEPARATELY UNDERTAKEN BY THE LICENSOR, TO THE EXTENT POSSIBLE, THE LICENSOR OFFERS THE LICENSED MATERIAL AS-IS AND AS-AVAILABLE, AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE LICENSED MATERIAL, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHER. THIS INCLUDES, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OR ABSENCE OF ERRORS, WHETHER OR NOT KNOWN OR DISCOVERABLE. WHERE DISCLAIMERS OF WARRANTIES ARE NOT ALLOWED IN FULL OR IN PART, THIS DISCLAIMER MAY NOT APPLY TO YOU.

b. TO THE EXTENT POSSIBLE, IN NO EVENT WILL THE LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY (INCLUDING, WITHOUT LIMITATION, NEGLIGENCE) OR OTHERWISE FOR ANY DIRECT, SPECIAL, INDIRECT, INCIDENTAL, CONSEQUENTIAL, PUNITIVE, EXEMPLARY, OR OTHER LOSSES, COSTS, EXPENSES, OR DAMAGES ARISING OUT OF THIS PUBLIC LICENSE OR USE OF THE LICENSED MATERIAL, EVEN IF THE LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSSES, COSTS, EXPENSES, OR DAMAGES. WHERE A LIMITATION OF LIABILITY IS NOT ALLOWED IN FULL OR IN PART, THIS LIMITATION MAY NOT APPLY TO YOU.

c. The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

Section 6 -- Term and Termination.

a. This Public License applies for the term of the Copyright and Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate automatically.

b. Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates:

    1. automatically as of the date the violation is cured, provided it is cured within 30 days of Your discovery of the violation; or

    2. upon express reinstatement by the Licensor.

    For the avoidance of doubt, this Section 6(b) does not affect any right the Licensor may have to seek remedies for Your violations of this Public License.

c. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License.

d. Sections 1, 5, 6, 7, and 8 survive termination of this Public License.

Section 7 -- Other Terms and Conditions.

a. The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed.

b. Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License.

Section 8 -- Interpretation.

a. For the avoidance of doubt, this Public License does not, and shall not be interpreted to, reduce, limit, restrict, or impose conditions on any use of the Licensed Material that could lawfully be made without permission under this Public License.

b. To the extent possible, if any provision of this Public License is deemed unenforceable, it shall be automatically reformed to the minimum extent necessary to make it enforceable. If the provision cannot be reformed, it shall be severed from this Public License without affecting the enforceability of the remaining terms and conditions.

c. No term or condition of this Public License will be waived and no failure to comply consented to unless expressly agreed to by the Licensor.

d. Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver of, any privileges and immunities that apply to the Licensor or You, including from the legal processes of any jurisdiction or authority.

==========

Creative Commons is not a party to its public licenses. Notwithstanding, Creative Commons may elect to apply one of its public licenses to material it publishes and in those instances will be considered the “Licensor.” The text of the Creative Commons public licenses is dedicated to the public domain under the CC0 Public Domain Dedication. Except for the limited purpose of indicating that material is shared under a Creative Commons public license or as otherwise permitted by the Creative Commons policies published at creativecommons.org/policies, Creative Commons does not authorize the use of the trademark "Creative Commons" or any other trademark or logo of Creative Commons without its prior written consent including, without limitation, in connection with any unauthorized modifications to any of its public licenses or any other arrangements, understandings, or agreements concerning use of licensed material. For the avoidance of doubt, this paragraph does not form part of the public licenses.

Creative Commons may be contacted at creativecommons.org.

```
caniuse-lite@1.0.30000966
licenses: CC-BY-4.0
repository: https://github.com/ben-eb/caniuse-lite
publisher: Ben Briggs
email: beneb.info@gmail.com
The data in this repo is available for use under a CC BY 4.0 license
(http://creativecommons.org/licenses/by/4.0/).
For attribution just mention somewhere that the source is caniuse.com
```

CodeMirror 5.53.2

```
MIT License

Copyright (C) 2017 by Marijn Haverbeke  and others

Permission is hereby granted, free of charge, to any person obtaining a
copy
of this software and associated documentation files (the "Software"),
to deal
in the Software without restriction, including without limitation the
rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included
in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
IN
THE SOFTWARE.
```

Dexie.js 3.0.1

```
Dexie.js

Copyright (c) 2014-2017 David Fahlander
```

```
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

      "License" shall mean the terms and conditions for use,
reproduction,
      and distribution as defined by Sections 1 through 9 of this
document.

      "Licensor" shall mean the copyright owner or entity authorized by
      the copyright owner that is granting the License.

      "Legal Entity" shall mean the union of the acting entity and all
      other entities that control, are controlled by, or are under
common
      control with that entity. For the purposes of this definition,
      "control" means (i) the power, direct or indirect, to cause the
      direction or management of such entity, whether by contract or
      otherwise, or (ii) ownership of fifty percent (50%) or more of the
      outstanding shares, or (iii) beneficial ownership of such entity.

      "You" (or "Your") shall mean an individual or Legal Entity
      exercising permissions granted by this License.

      "Source" form shall mean the preferred form for making
modifications,
      including but not limited to software source code, documentation
      source, and configuration files.

      "Object" form shall mean any form resulting from mechanical
      transformation or translation of a Source form, including but
      not limited to compiled object code, generated documentation,
      and conversions to other media types.

      "Work" shall mean the work of authorship, whether in Source or
      Object form, made available under the License, as indicated by a
      copyright notice that is included in or attached to the work
      (an example is provided in the Appendix below).

      "Derivative Works" shall mean any work, whether in Source or
```

```
Object
      form, that is based on (or derived from) the Work and for which
the
      editorial revisions, annotations, elaborations, or other
modifications
      represent, as a whole, an original work of authorship. For the
purposes
      of this License, Derivative Works shall not include works that
remain
      separable from, or merely link (or bind by name) to the
interfaces of,
      the Work and Derivative Works thereof.

      "Contribution" shall mean any work of authorship, including
      the original version of the Work and any modifications or
additions
      to that Work or Derivative Works thereof, that is intentionally
      submitted to Licensor for inclusion in the Work by the copyright
owner
      or by an individual or Legal Entity authorized to submit on
behalf of
      the copyright owner. For the purposes of this definition,
"submitted"
      means any form of electronic, verbal, or written communication
sent
      to the Licensor or its representatives, including but not limited
to
      communication on electronic mailing lists, source code control
systems,
      and issue tracking systems that are managed by, or on behalf of,
the
      Licensor for the purpose of discussing and improving the Work, but
      excluding communication that is conspicuously marked or otherwise
      designated in writing by the copyright owner as "Not a
Contribution."

      "Contributor" shall mean Licensor and any individual or Legal
Entity
      on behalf of whom a Contribution has been received by Licensor and
      subsequently incorporated within the Work.

   2. Grant of Copyright License. Subject to the terms and conditions of
      this License, each Contributor hereby grants to You a perpetual,
      worldwide, non-exclusive, no-charge, royalty-free, irrevocable
      copyright license to reproduce, prepare Derivative Works of,
      publicly display, publicly perform, sublicense, and distribute the
      Work and such Derivative Works in Source or Object form.

   3. Grant of Patent License. Subject to the terms and conditions of
      this License, each Contributor hereby grants to You a perpetual,
      worldwide, non-exclusive, no-charge, royalty-free, irrevocable
      (except as stated in this section) patent license to make, have
made,
      use, offer to sell, sell, import, and otherwise transfer the Work,
      where such license applies only to those patent claims licensable
```

```
      by such Contributor that are necessarily infringed by their
      Contribution(s) alone or by combination of their Contribution(s)
      with the Work to which such Contribution(s) was submitted. If You
      institute patent litigation against any entity (including a
      cross-claim or counterclaim in a lawsuit) alleging that the Work
      or a Contribution incorporated within the Work constitutes direct
      or contributory patent infringement, then any patent licenses
      granted to You under this License for that Work shall terminate
      as of the date such litigation is filed.

   4. Redistribution. You may reproduce and distribute copies of the
      Work or Derivative Works thereof in any medium, with or without
      modifications, and in Source or Object form, provided that You
      meet the following conditions:

      (a) You must give any other recipients of the Work or
          Derivative Works a copy of this License; and

      (b) You must cause any modified files to carry prominent notices
          stating that You changed the files; and

      (c) You must retain, in the Source form of any Derivative Works
          that You distribute, all copyright, patent, trademark, and
          attribution notices from the Source form of the Work,
          excluding those notices that do not pertain to any part of
          the Derivative Works; and

      (d) If the Work includes a "NOTICE" text file as part of its
          distribution, then any Derivative Works that You distribute
must
          include a readable copy of the attribution notices contained
          within such NOTICE file, excluding those notices that do not
          pertain to any part of the Derivative Works, in at least one
          of the following places: within a NOTICE text file distributed
          as part of the Derivative Works; within the Source form or
          documentation, if provided along with the Derivative Works;
or,
          within a display generated by the Derivative Works, if and
          wherever such third-party notices normally appear. The
contents
          of the NOTICE file are for informational purposes only and
          do not modify the License. You may add Your own attribution
          notices within Derivative Works that You distribute, alongside
          or as an addendum to the NOTICE text from the Work, provided
          that such additional attribution notices cannot be construed
          as modifying the License.

      You may add Your own copyright statement to Your modifications and
      may provide additional or different license terms and conditions
      for use, reproduction, or distribution of Your modifications, or
      for any such Derivative Works as a whole, provided Your use,
      reproduction, and distribution of the Work otherwise complies with
      the conditions stated in this License.

   5. Submission of Contributions. Unless You explicitly state
```

```
otherwise,
      any Contribution intentionally submitted for inclusion in the Work
      by You to the Licensor shall be under the terms and conditions of
      this License, without any additional terms or conditions.
      Notwithstanding the above, nothing herein shall supersede or
modify
      the terms of any separate license agreement you may have executed
      with Licensor regarding such Contributions.

   6. Trademarks. This License does not grant permission to use the
trade
      names, trademarks, service marks, or product names of the
Licensor,
      except as required for reasonable and customary use in describing
the
      origin of the Work and reproducing the content of the NOTICE file.

   7. Disclaimer of Warranty. Unless required by applicable law or
      agreed to in writing, Licensor provides the Work (and each
      Contributor provides its Contributions) on an "AS IS" BASIS,
      WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
      implied, including, without limitation, any warranties or
conditions
      of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
      PARTICULAR PURPOSE. You are solely responsible for determining the
      appropriateness of using or redistributing the Work and assume any
      risks associated with Your exercise of permissions under this
License.

   8. Limitation of Liability. In no event and under no legal theory,
      whether in tort (including negligence), contract, or otherwise,
      unless required by applicable law (such as deliberate and grossly
      negligent acts) or agreed to in writing, shall any Contributor be
      liable to You for damages, including any direct, indirect,
special,
      incidental, or consequential damages of any character arising as a
      result of this License or out of the use or inability to use the
      Work (including but not limited to damages for loss of goodwill,
      work stoppage, computer failure or malfunction, or any and all
      other commercial damages or losses), even if such Contributor
      has been advised of the possibility of such damages.

   9. Accepting Warranty or Additional Liability. While redistributing
      the Work or Derivative Works thereof, You may choose to offer,
      and charge a fee for, acceptance of support, warranty, indemnity,
      or other liability obligations and/or rights consistent with this
      License. However, in accepting such obligations, You may act only
      on Your own behalf and on Your sole responsibility, not on behalf
      of any other Contributor, and only if You agree to indemnify,
      defend, and hold each Contributor harmless for any liability
      incurred by, or claims asserted against, such Contributor by
reason
      of your accepting any such warranty or additional liability.

   END OF TERMS AND CONDITIONS
```

```
   APPENDIX: How to apply the Apache License to your work.

      To apply the Apache License to your work, attach the following
      boilerplate notice, with the fields enclosed by brackets "{}"
      replaced with your own identifying information. (Don't include
      the brackets!)  The text should be enclosed in the appropriate
      comment syntax for the file format. We also recommend that a
      file or class name and description of purpose be included on the
      same "printed page" as the copyright notice for easier
      identification within third-party archives.

   Copyright {yyyy} {name of copyright owner}

   Licensed under the Apache License, Version 2.0 (the "License");
   you may not use this file except in compliance with the License.
   You may obtain a copy of the License at

       http://www.apache.org/licenses/LICENSE-2.0

   Unless required by applicable law or agreed to in writing, software
   distributed under the License is distributed on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
   See the License for the specific language governing permissions and
   limitations under the License.
```

SheetJS 0.15.5

```
SheetJS is licensed under Apache 2.0. License Text below.
=========
SheetJS bundles 4th party JSZip which is dependent on Pako. The
copyright and license text is here.
/*
JSZip - A Javascript class for generating and reading zip files

(c) 2009-2014 Stuart Knightley
Dual licenced under the MIT license or GPLv3. See https://
raw.github.com/Stuk/jszip/master/LICENSE.markdown.
JSZip uses the library pako released under the MIT license :
https://github.com/nodeca/pako/blob/master/LICENSE
Note: since JSZip 3 removed critical functionality, this version
assigns to the
`JSZipSync` variable.  Another JSZip version can be loaded in parallel.
*/
==============================================

Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
```

```
   1. Definitions.

      "License" shall mean the terms and conditions for use,
reproduction,
      and distribution as defined by Sections 1 through 9 of this
document.

      "Licensor" shall mean the copyright owner or entity authorized by
      the copyright owner that is granting the License.

      "Legal Entity" shall mean the union of the acting entity and all
      other entities that control, are controlled by, or are under
common
      control with that entity. For the purposes of this definition,
      "control" means (i) the power, direct or indirect, to cause the
      direction or management of such entity, whether by contract or
      otherwise, or (ii) ownership of fifty percent (50%) or more of the
      outstanding shares, or (iii) beneficial ownership of such entity.

      "You" (or "Your") shall mean an individual or Legal Entity
      exercising permissions granted by this License.

      "Source" form shall mean the preferred form for making
modifications,
      including but not limited to software source code, documentation
      source, and configuration files.

      "Object" form shall mean any form resulting from mechanical
      transformation or translation of a Source form, including but
      not limited to compiled object code, generated documentation,
      and conversions to other media types.

      "Work" shall mean the work of authorship, whether in Source or
      Object form, made available under the License, as indicated by a
      copyright notice that is included in or attached to the work
      (an example is provided in the Appendix below).

      "Derivative Works" shall mean any work, whether in Source or
Object
      form, that is based on (or derived from) the Work and for which
the
      editorial revisions, annotations, elaborations, or other
modifications
      represent, as a whole, an original work of authorship. For the
purposes
      of this License, Derivative Works shall not include works that
remain
      separable from, or merely link (or bind by name) to the
interfaces of,
      the Work and Derivative Works thereof.

      "Contribution" shall mean any work of authorship, including
      the original version of the Work and any modifications or
additions
      to that Work or Derivative Works thereof, that is intentionally
```

```
      submitted to Licensor for inclusion in the Work by the copyright
owner
      or by an individual or Legal Entity authorized to submit on
behalf of
      the copyright owner. For the purposes of this definition,
"submitted"
      means any form of electronic, verbal, or written communication
sent
      to the Licensor or its representatives, including but not limited
to
      communication on electronic mailing lists, source code control
systems,
      and issue tracking systems that are managed by, or on behalf of,
the
      Licensor for the purpose of discussing and improving the Work, but
      excluding communication that is conspicuously marked or otherwise
      designated in writing by the copyright owner as "Not a
Contribution."

      "Contributor" shall mean Licensor and any individual or Legal
Entity
      on behalf of whom a Contribution has been received by Licensor and
      subsequently incorporated within the Work.

   2. Grant of Copyright License. Subject to the terms and conditions of
      this License, each Contributor hereby grants to You a perpetual,
      worldwide, non-exclusive, no-charge, royalty-free, irrevocable
      copyright license to reproduce, prepare Derivative Works of,
      publicly display, publicly perform, sublicense, and distribute the
      Work and such Derivative Works in Source or Object form.

   3. Grant of Patent License. Subject to the terms and conditions of
      this License, each Contributor hereby grants to You a perpetual,
      worldwide, non-exclusive, no-charge, royalty-free, irrevocable
      (except as stated in this section) patent license to make, have
made,
      use, offer to sell, sell, import, and otherwise transfer the Work,
      where such license applies only to those patent claims licensable
      by such Contributor that are necessarily infringed by their
      Contribution(s) alone or by combination of their Contribution(s)
      with the Work to which such Contribution(s) was submitted. If You
      institute patent litigation against any entity (including a
      cross-claim or counterclaim in a lawsuit) alleging that the Work
      or a Contribution incorporated within the Work constitutes direct
      or contributory patent infringement, then any patent licenses
      granted to You under this License for that Work shall terminate
      as of the date such litigation is filed.

   4. Redistribution. You may reproduce and distribute copies of the
      Work or Derivative Works thereof in any medium, with or without
      modifications, and in Source or Object form, provided that You
      meet the following conditions:

      (a) You must give any other recipients of the Work or
          Derivative Works a copy of this License; and
```

```
      (b) You must cause any modified files to carry prominent notices
          stating that You changed the files; and

      (c) You must retain, in the Source form of any Derivative Works
          that You distribute, all copyright, patent, trademark, and
          attribution notices from the Source form of the Work,
          excluding those notices that do not pertain to any part of
          the Derivative Works; and

      (d) If the Work includes a "NOTICE" text file as part of its
          distribution, then any Derivative Works that You distribute
must
          include a readable copy of the attribution notices contained
          within such NOTICE file, excluding those notices that do not
          pertain to any part of the Derivative Works, in at least one
          of the following places: within a NOTICE text file distributed
          as part of the Derivative Works; within the Source form or
          documentation, if provided along with the Derivative Works;
or,
          within a display generated by the Derivative Works, if and
          wherever such third-party notices normally appear. The
contents
          of the NOTICE file are for informational purposes only and
          do not modify the License. You may add Your own attribution
          notices within Derivative Works that You distribute, alongside
          or as an addendum to the NOTICE text from the Work, provided
          that such additional attribution notices cannot be construed
          as modifying the License.

      You may add Your own copyright statement to Your modifications and
      may provide additional or different license terms and conditions
      for use, reproduction, or distribution of Your modifications, or
      for any such Derivative Works as a whole, provided Your use,
      reproduction, and distribution of the Work otherwise complies with
      the conditions stated in this License.

   5. Submission of Contributions. Unless You explicitly state
otherwise,
      any Contribution intentionally submitted for inclusion in the Work
      by You to the Licensor shall be under the terms and conditions of
      this License, without any additional terms or conditions.
      Notwithstanding the above, nothing herein shall supersede or
modify
      the terms of any separate license agreement you may have executed
      with Licensor regarding such Contributions.

   6. Trademarks. This License does not grant permission to use the
trade
      names, trademarks, service marks, or product names of the
Licensor,
      except as required for reasonable and customary use in describing
the
      origin of the Work and reproducing the content of the NOTICE file.
```

```
   7. Disclaimer of Warranty. Unless required by applicable law or
      agreed to in writing, Licensor provides the Work (and each
      Contributor provides its Contributions) on an "AS IS" BASIS,
      WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
      implied, including, without limitation, any warranties or
conditions
      of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
      PARTICULAR PURPOSE. You are solely responsible for determining the
      appropriateness of using or redistributing the Work and assume any
      risks associated with Your exercise of permissions under this
License.

   8. Limitation of Liability. In no event and under no legal theory,
      whether in tort (including negligence), contract, or otherwise,
      unless required by applicable law (such as deliberate and grossly
      negligent acts) or agreed to in writing, shall any Contributor be
      liable to You for damages, including any direct, indirect,
special,
      incidental, or consequential damages of any character arising as a
      result of this License or out of the use or inability to use the
      Work (including but not limited to damages for loss of goodwill,
      work stoppage, computer failure or malfunction, or any and all
      other commercial damages or losses), even if such Contributor
      has been advised of the possibility of such damages.

   9. Accepting Warranty or Additional Liability. While redistributing
      the Work or Derivative Works thereof, You may choose to offer,
      and charge a fee for, acceptance of support, warranty, indemnity,
      or other liability obligations and/or rights consistent with this
      License. However, in accepting such obligations, You may act only
      on Your own behalf and on Your sole responsibility, not on behalf
      of any other Contributor, and only if You agree to indemnify,
      defend, and hold each Contributor harmless for any liability
      incurred by, or claims asserted against, such Contributor by
reason
      of your accepting any such warranty or additional liability.

   END OF TERMS AND CONDITIONS

   APPENDIX: How to apply the Apache License to your work.

      To apply the Apache License to your work, attach the following
      boilerplate notice, with the fields enclosed by brackets "{}"
      replaced with your own identifying information. (Don't include
      the brackets!)  The text should be enclosed in the appropriate
      comment syntax for the file format. We also recommend that a
      file or class name and description of purpose be included on the
      same "printed page" as the copyright notice for easier
      identification within third-party archives.

   Copyright (C) 2012-present   SheetJS LLC

   Licensed under the Apache License, Version 2.0 (the "License");
   you may not use this file except in compliance with the License.
   You may obtain a copy of the License at
```

```
       http://www.apache.org/licenses/LICENSE-2.0

   Unless required by applicable law or agreed to in writing, software
   distributed under the License is distributed on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
   See the License for the specific language governing permissions and
   limitations under the License.
```

jQuery 3.5.1

```
Copyright JS Foundation and other contributors, https://js.foundation/

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

----------------------------------------------------------------------
--------

external/sizzle

Copyright JS Foundation and other contributors, https://js.foundation/

This software consists of voluntary contributions made by many
individuals. For exact contribution history, see the revision history
available at https://github.com/jquery/sizzle

The following license applies to all parts of this software except as
documented below:

====

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
```

```
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

All files located in the node_modules and external directories are
externally maintained libraries used by this software which have their
own licenses; we recommend you read them, as their terms may differ from
the terms above.
```

Index

A

access token
  acquiring, *D-12*
  securing, *D-19*
  using, *D-14*
Apache Tomcat, *1-2*
  about, *1-41*
  configuring Oracle REST Data Services images, *1-42*
  deploying to, *1-41*
  downloading, *1-41*
authentication
  against WebLogic user repositories, *5-77*

B

bequeath connection, *1-44*
browser origins, *D-15*

C

command-line interface, *1-4*
configdir command, *B-1*
  locating configuration files, *B-1*
  locating configuration folder, *B-1*
configuration file editable parameters
  jdbc.MaxConnectionReuseCount, *B-5*
  jdbc.MinLimit, *B-5*
  security.requestValidationFunction, *B-10*
  security.verifySSL, *B-10*
  soda.defaultLimit, *B-10*
  soda.maxLimit, *B-10*
configuration files, *B-1*
  format of, *B-2*
  locating using configdir command, *B-1*
configuration folder
  setting location, *B-1*
  structure of, *B-1*
CORS (Cross Origin Resource Sharing), *D-15*
CREATE_CLIENT procedure, *11-1*
CREATE_ROLE procedure, *8-1*, *9-1*
CREATE_SERVICE procedure (deprecated), *8-1*
Cross Origin Resource Sharing (CORS), *D-15*
Cross Site Request Forgery (CSRF) attacks, *D-13*
CSRF (Cross Site Request Forgery) attacks, *D-13*
cURL, *5-5*

D

database users, *1-4*
defaults.xml
  enabling detailed request error messages, *C-1*
defaults.xmll, file format, *B-2*
DEFINE_HANDLER procedure, *8-4*, *9-2*
DEFINE_MODULE procedure, *8-6*, *9-4*
DEFINE_PARAMETER procedure, *8-7*, *9-5*
DEFINE_PRIVILEGE procedure, *8-9*, *9-7*
DEFINE_SERVICE procedure, *8-11*, *9-10*
DEFINE_TEMPLATE procedure, *8-14*, *9-13*
DELETE_CLIENT procedure, *11-2*
DELETE_MODULE procedure, *8-16*, *9-14*
DELETE_PRIVILEGE procedure, *8-16*, *9-15*
DELETE_ROLE procedure, *8-17*, *9-15*
deploy options
  Apache Tomcat, *1-41*
deployment options
  Oracle WebLogic Server, *1-36*
DER
  converting private key to DER, *1-35*
downloading
  Apache Tomcat, *1-41*
  Oracle WebLogic Server, *1-37*
DROP_REST_FOR_SCHEMA procedure, *8-17*, *9-16*

E

ENABLE_OBJECT procedure, *8-18*, *9-17*, *9-24*, *9-25*
ENABLE_SCHEMA procedure, *8-19*, *9-19*

G

GlassFish Server
installing the deployment, 1-42
GRANT_CLIENT_ROLE procedure, 11-3
graphical user interface administration, 1-32

I

image gallery example, D-1
installation options
standalone mode, 1-33
installation overview, 1-3

J

Java EE application servers
about supported, 1-2
JSON
using to pass parameters, 5-32

M

multiple database configuration, 2-1
about the request URL, 2-1
configuring additional databases, 2-2
routing request rules, 2-3
routing requests based on URL prefix, 2-4

O

OAUTH package
CREATE_CLIENT, 11-1
DELETE_CLIENT, 11-2
GRANT_CLIENT_ROLE, 11-3
RENAME_CLIENT, 11-4
REVOKE_CLIENT_ROLE, 11-4
UPDATE_CLIENT, 11-5
OAuth2, default behavior, 1-33
Oracle REST Data Services
about, 1-1
about upgrading, 1-43
administering with graphical user interface, 1-32
bequeath connection, 1-44
caching, 2-5
configuration files, B-1
configuring, 1-3, 1-7, 2-1
configuring multiple databases, 2-1
configuring with command-line interface, 1-4, 1-7
database users, 1-4
developing RESTful services, 2-7
downloading, 1-3
Oracle REST Data Services *(continued)*
environment, 2-5
Excel settings, 2-5
installation overview, 1-3
installing, 1-3
PL/SQL API
PL/SQL API for Oracle REST Data Services, 5-82
pre- and post- processing, 2-5
running in standalone mode, 1-33
security, 2-5
system requirements, 1-2
Oracle REST Data Services configuration file
enabling detailed request error messages, C-1
Oracle REST Data Services package
CREATE_SERVICE (deprecated), 8-1
Oracle WebLogic Server, 1-2
about, 1-37
deploy to, 1-36
downloading, 1-37
installing, 1-36
installing the deployment, 1-38
ORDS package
CREATE_ROLE, 8-1, 9-1
DEFINE_HANDLER, 8-4, 9-2
DEFINE_MODULE, 8-6, 9-4
DEFINE_PARAMETER, 8-7, 9-5
DEFINE_PRIVILEGE, 8-9, 9-7
DEFINE_SERVICE, 8-11, 9-10
DEFINE_TEMPLATE, 8-14, 9-13
DELETE_MODULE, 8-16, 9-14
DELETE_PRIVILEGE, 8-16, 9-15
DELETE_ROLE, 8-17, 9-15
DROP_REST_FOR_SCHEMA, 8-17, 9-16
ENABLE_OBJECT, 8-18, 9-17, 9-24, 9-25
ENABLE_SCHEMA, 8-19, 9-19
PUBLISH_MODULE, 8-20, 9-20
RENAME_MODULE, 8-21, 9-20
RENAME_PRIVILEGE, 8-22, 9-21
RENAME_ROLE, 8-22, 9-22
SET_MODULE_ORIGINS_ALLOWED, 8-23, 9-23
SET_URL_MAPPING, 8-24, 9-23

P

passing parameters
using JSON, 5-32
using query strings, 5-40
using route patterns, 5-36
private key
converting to DER, 1-35
PUBLISH_MODULE procedure, 8-20, 9-20

R

RENAME_CLIENT procedure, *11-4*
RENAME_MODULE procedure, *8-21*, *9-20*
RENAME_PRIVILEGE procedure, *8-22*, *9-21*
RENAME_ROLE procedure, *8-22*, *9-22*
resource handler, *5-3*
resource module, *5-3*
resource template, *5-3*
RESTful services
  about, *5-2*
  accessing from third-party application, *D-11*
  configuring for cross-origin resource sharing, *D-15*
  developing, *2-7*
  getting started with, *5-2*
  image gallery example, *D-1*
  integrating with existing group/role models, *5-79*
  sample services, *5-55*
  securing, *D-8*
  terminology, *5-3*
  user roles, *5-74*
  using cURL, *5-5*
REVOKE_CLIENT_ROLE procedure, *11-4*
role-mapping.xml file, *5-79*
route pattern, *5-3*

S

Same Origin Policy, *D-15*
SET_MODULE_ORIGINS_ALLOWED procedure, *8-23*, *9-23*
SET_URL_MAPPING procedure, *8-24*, *9-23*
SQL Developer Oracle REST Data Services Administration, *1-32*
standalone mode
  starting, *1-34*
  stopping the server, *1-35*
standalone mode, running in, *1-33*
structure of configuration folder, *B-1*
supported Java EE application servers, *1-2*
system requirements, *1-2*

T

troubleshooting, *C-1*
  enabling detailed request error messages, *C-1*

U

UPDATE_CLIENT procedure, *11-5*
upsert operation, *5-15*
URI pattern, *5-3*
URI template, *5-3*
url-mapping.xml
  file format, *B-2*
  request rules routing, *B-2*
user roles for RESTful services, *5-74*
using query strings
  to pass optional parameters, *5-40*
using route patterns
  for passing required parameters, *5-36*