

Oracle® REST Data Services

Developer's Guide



Release 24.1
F91834-02
May 2024



Oracle REST Data Services Developer's Guide, Release 24.1

F91834-02

Copyright © 2011, 2024, Oracle and/or its affiliates.

Primary Authors: Mamata Basapur, Chuck Murray, Tulika Das

Contributors: Kris Rice, Jeff D. Smith, Colm Divilly, Peter J. Obrien, Dermot O'Neill, Elizabeth Saunders, Ashley Chen, Sharon Kennedy, Ganesh Pitchaiah, 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, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

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

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

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

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

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

Contents

Preface

Audience	xvii
Documentation Accessibility	xvii
Related Documents	xvii
Conventions	xvii

Changes in Release 24.1 Oracle REST Data Services Developer's Guide

Changes in Oracle REST Data Services 24.1	xix
---	-----

1 Introduction to Oracle REST Data Services

1.1 About Oracle REST Data Services	1-1
1.2 Features of Oracle REST Data Services	1-2

2 Developing Oracle REST Data Services Applications

2.1 Introduction to Relevant Software	2-2
2.1.1 Oracle APEX	2-2
2.1.2 REST APIs	2-2
2.2 Getting Started with RESTful Services	2-2
2.2.1 RESTful Services Terminology	2-3
2.2.2 ORDS RESTful Web Services Architecture Diagrams	2-3
2.2.3 About Request Path Syntax Requirements	2-5
2.2.4 "Getting Started" Documents Included in Installation	2-5
2.2.5 About cURL and Testing RESTful Services	2-6
2.3 Automatic Enabling of Schema Objects for REST Access (AutoREST)	2-6
2.3.1 Examples: Accessing Objects Using RESTful Services	2-8
2.3.1.1 Get Schema Metadata	2-8
2.3.1.2 Get Object Metadata	2-9
2.3.1.3 Get Object Data	2-10
2.3.1.4 Get Table Data Using Paging	2-12
2.3.1.5 Get Table Data Using Query	2-13
2.3.1.6 Get Table Row Using Primary Key	2-14

2.3.1.7	Insert Table Row	2-15
2.3.1.8	Update/Insert Table Row	2-16
2.3.1.9	Delete Using Filter	2-17
2.3.1.10	Post by Batch Load	2-17
2.3.2	Filtering in Queries	2-18
2.3.2.1	FilterObject Grammar	2-19
2.3.2.2	Examples: FilterObject Specifications	2-22
2.3.3	Auto PL/SQL	2-27
2.3.3.1	Method and Content Type Supported for Auto Enabling PL/SQL Objects	2-27
2.3.3.2	Auto-Enabling the PL/SQL Objects	2-28
2.3.3.3	Generating the PL/SQL Endpoints	2-31
2.3.3.4	Resource Input Payload	2-32
2.3.3.5	Resource Payload Response	2-32
2.3.3.6	Function Return Value	2-33
2.3.4	Support for JSON-Relational Duality View	2-33
2.3.4.1	Table AutoREST Versus JSON-Relational Duality View AutoREST	2-33
2.3.4.2	Support for Enhanced ETag Matching	2-34
2.3.4.3	Enhanced JSON QBE (Query by Example) Filtering	2-35
2.3.4.4	Enhanced JSON Batch Loading	2-35
2.4	Manually Creating RESTful Services Using SQL and PL/SQL	2-36
2.4.1	About Oracle REST Data Services Mechanisms for Passing Parameters	2-37
2.4.1.1	Using JSON to Pass Parameters	2-37
2.4.1.2	Using Route Patterns to Pass Parameters	2-42
2.4.1.3	Using Query Strings for Optional Parameters	2-46
2.4.2	Using SQL/JSON Database Functions	2-47
2.4.2.1	Inserting Nested JSON Objects into Relational Tables	2-47
2.4.2.2	Generating Nested JSON Objects from Hierarchical Relational Data	2-52
2.4.2.3	Testing the RESTful Services	2-55
2.5	Manually Creating RESTful Services Using Javascript	2-57
2.5.1	Allowed JavaScript Structures	2-57
2.5.2	Defining the REST Service and JavaScript Handler Using PL/SQL Function	2-59
2.5.3	About Executing SQL in Javascript	2-60
2.5.4	About Using the Fetch Function	2-62
2.5.5	Referencing MLE Environments	2-63
2.6	About Working with Dates Using Oracle REST Data Services	2-65
2.6.1	About Datetime Handling with Oracle REST Data Services	2-65
2.6.2	About Setting the Time Zone	2-66
2.6.3	Exploring the Sample RESTful Services in APEX (Tutorial)	2-67
2.7	Creating RESTful Web Services Using Database Actions	2-71
2.8	Configuring Secure Access to RESTful Services	2-72
2.8.1	Authentication	2-72
2.8.1.1	First Party Cookie-Based Authentication	2-72

2.8.1.2	Third Party OAuth 2.0-Based Authentication	2-72
2.8.2	About Privileges for Accessing Resources	2-73
2.8.3	About Users and Roles for Accessing Resources	2-74
2.8.4	About the File-Based User Repository	2-74
2.8.5	Tutorial: Protecting and Accessing Resources	2-74
2.8.5.1	OAuth Flows and When to Use Each	2-75
2.8.5.2	Assumptions for This Tutorial	2-75
2.8.5.3	Steps for This Tutorial	2-76
2.9	JWT Bearer Token Authentication and Authorization Using JWT Profile	2-86
2.9.1	About JSON Web Tokens (JWTs)	2-86
2.9.2	Prerequisites for JWT Authentication	2-88
2.9.3	Creating an ORDS JWT Profile	2-88
2.9.4	JWT Identity Provider Details	2-90
2.9.4.1	Parameters for Verifying JWT Signatures	2-90
2.9.4.2	JWT Scopes and ORDS Privileges	2-91
2.9.4.3	JWT Subject	2-91
2.9.5	Making Requests to ORDS Using a JWT Bearer Token	2-91
2.10	About Oracle REST Data Services User Roles	2-92
2.10.1	About Oracle APEX Users and Oracle REST Data Services Roles	2-92
2.10.1.1	Granting APEX Users Oracle REST Data Services Roles	2-92
2.10.1.2	Automatically Granting APEX Users Oracle REST Data Services Roles	2-93
2.10.2	Controlling RESTful Service Access with Roles	2-93
2.10.2.1	About Defining RESTful Service Roles	2-93
2.10.2.2	Associating Roles with RESTful Privileges	2-94
2.11	Authenticating Against WebLogic Server User Repositories	2-94
2.11.1	Authenticating Against WebLogic Server	2-94
2.11.1.1	Creating a WebLogic Server User	2-94
2.11.1.2	Verifying the WebLogic Server User	2-95
2.12	Integrating with Existing Group/Role Models	2-96
2.12.1	About role-mapping.xml	2-96
2.12.1.1	Parameterizing Mapping Rules	2-97
2.12.1.2	Dereferencing Parameters	2-97
2.12.1.3	Indirect Mappings	2-97
2.13	Integrating Oracle REST Data Services and WebLogic Server	2-98
2.13.1	Configuring ORDS to Integrate with WebLogic Server	2-98
2.14	Using the Oracle REST Data Services PL/SQL API	2-99
2.14.1	Creating a RESTful Service Using the PL/SQL API	2-100
2.14.2	Testing the RESTful Service	2-100
2.15	Oracle REST Data Services Database Authentication	2-102
2.15.1	Installing Sample Database Scripts	2-102
2.15.2	Enabling the Database Authentication	2-103
2.15.3	Configuring the Request Validation Function	2-104

2.15.4	Testing the Database Authenticated User	2-104
2.15.5	Uninstalling the Sample Database Schema	2-104
2.16	Overview of Pre-hook Functions	2-105
2.16.1	Configuring the Pre-hook Function	2-105
2.16.2	Using a Pre-hook Function	2-105
2.16.3	Processing of a Request	2-106
2.16.4	Identity Assertion of a User	2-106
2.16.5	Aborting Processing of a Request	2-106
2.16.6	Ensuring Pre-hook is Executable	2-107
2.16.7	Exceptions Handling by Pre-hook Function	2-107
2.16.8	Pre-hook Function Efficiency	2-107
2.16.9	Pre-Hook Examples	2-107
2.16.9.1	Installing the Examples	2-107
2.16.9.2	Uninstalling the Examples	2-111
2.17	Generating Hyperlinks	2-111
2.17.1	Primary Key Hyperlinks	2-111
2.17.1.1	Composite Primary Keys	2-113
2.17.2	Arbitrary Hyperlinks	2-113
2.17.2.1	About the related Link Relation	2-115
2.17.2.2	URL Resolution	2-115
2.18	About HTTP Error Responses	2-118
2.18.1	About error.responseFormat	2-118
2.18.1.1	HTML Mode	2-118
2.18.1.2	json Mode	2-118
2.18.1.3	auto Mode	2-119

3 Implicit Parameters

3.1	List of Implicit Parameters	3-1
3.1.1	About the :body parameter	3-4
3.1.2	About the :body_text Parameter	3-5
3.1.3	About the :content_type Parameter	3-6
3.1.4	About the :current_user Parameter	3-6
3.1.5	About the :status_code Parameter	3-6
3.1.6	About the :forward_location Parameter	3-6
3.1.7	About the Pagination Implicit Parameters	3-8
3.1.7.1	About the :page_offset Parameter	3-9
3.1.7.2	About the :page_size Parameter	3-9
3.1.7.3	About the :row_offset Parameter	3-9
3.1.7.4	About the :row_count Parameter	3-9
3.1.7.5	About the :fetch_offset Parameter	3-10
3.1.7.6	About the :fetch_size Parameter	3-10

3.1.7.7	About Automatic Pagination	3-10
3.1.7.8	About Manual Pagination	3-10

4 ORDS PL/SQL Package Reference

4.1	ORDS.CREATE_ROLE	4-1
4.2	ORDS.CREATE_SERVICE	4-1
4.3	ORDS.DEFINE_HANDLER	4-4
4.4	ORDS.DEFINE_MODULE	4-6
4.5	ORDS.DEFINE_PARAMETER	4-7
4.6	ORDS.DEFINE_PRIVILEGE	4-9
4.7	ORDS.DEFINE_SERVICE	4-11
4.8	ORDS.DEFINE_TEMPLATE	4-14
4.9	ORDS.DELETE_MODULE	4-15
4.10	ORDS.DELETE_PRIVILEGE	4-16
4.11	ORDS.DELETE_ROLE	4-16
4.12	ORDS.DROP_REST_FOR_SCHEMA	4-17
4.13	ORDS.ENABLE_OBJECT	4-17
4.14	ORDS.DROP_REST_FOR_OBJECT	4-18
4.15	ORDS.ENABLE_SCHEMA	4-19
4.16	ORDS.PUBLISH_MODULE	4-20
4.17	ORDS.RENAME_MODULE	4-20
4.18	ORDS.RENAME_PRIVILEGE	4-21
4.19	ORDS.RENAME_ROLE	4-22
4.20	ORDS.SET_MODULE_ORIGINS_ALLOWED	4-22
4.21	ORDS.SET_URL_MAPPING	4-23
4.22	ORDS.SET_SESSION_DEFAULTS	4-24
4.23	ORDS.RESET_SESSION_DEFAULTS	4-24
4.24	ORDS.SET_PROPERTY	4-25
4.25	ORDS.UNSET_PROPERTY	4-25

5 Oracle REST Data Services Administration PL/SQL Package Reference

5.1	ORDS_ADMIN.CREATE_ROLE	5-1
5.2	ORDS_ADMIN.DEFINE_HANDLER	5-2
5.3	ORDS_ADMIN.DEFINE_MODULE	5-4
5.4	ORDS_ADMIN.DEFINE_PARAMETER	5-5
5.5	ORDS_ADMIN.DEFINE_PRIVILEGE	5-7
5.6	ORDS_ADMIN.DEFINE_SERVICE	5-10
5.7	ORDS_ADMIN.DEFINE_TEMPLATE	5-13
5.8	ORDS_ADMIN.DELETE_MODULE	5-14
5.9	ORDS_ADMIN.DELETE_PRIVILEGE	5-15

5.10	ORDS_ADMIN.DELETE_ROLE	5-15
5.11	ORDS_ADMIN.DROP_REST_FOR_SCHEMA	5-16
5.12	ORDS_ADMIN.ENABLE_OBJECT	5-16
5.13	ORDS_ADMIN.DROP_REST_FOR_OBJECT	5-18
5.14	ORDS_ADMIN.ENABLE_SCHEMA	5-18
5.15	ORDS_ADMIN.PUBLISH_MODULE	5-19
5.16	ORDS_ADMIN.RENAME_MODULE	5-20
5.17	ORDS_ADMIN.RENAME_PRIVILEGE	5-21
5.18	ORDS_ADMIN.RENAME_ROLE	5-21
5.19	ORDS_ADMIN.SET_MODULE_ORIGINS_ALLOWED	5-22
5.20	ORDS_ADMIN.SET_URL_MAPPING	5-23
5.21	ORDS_ADMIN.ENABLE_HOUSEKEEPING_JOB	5-24
5.22	ORDS_ADMIN.DROP_HOUSEKEEPING_JOB	5-24
5.23	ORDS_ADMIN.PERFORM_HOUSEKEEPING	5-25
5.24	ORDS_ADMIN.SET_SESSION_DEFAULTS	5-25
5.25	ORDS_ADMIN.RESET_SESSION_DEFAULTS	5-26
5.26	ORDS_ADMIN.PROVISION_ADMIN_ROLE	5-26
5.27	ORDS_ADMIN.PROVISION_RUNTIME_ROLE	5-27
5.28	ORDS_ADMIN.UNPROVISION_ROLES	5-27
5.29	ORDS_ADMIN.CONFIG_PLSQL_GATEWAY	5-28
5.30	ORDS_ADMIN.SET_PROPERTY	5-29
5.31	ORDS_ADMIN.SET_PROPERTY	5-30
5.32	ORDS_ADMIN.UNSET_PROPERTY	5-31

6 OAUTH PL/SQL Package Reference

6.1	OAUTH.CREATE_CLIENT	6-1
6.2	OAUTH.DELETE_CLIENT	6-3
6.3	OAUTH.GRANT_CLIENT_ROLE	6-3
6.4	OAUTH.RENAME_CLIENT	6-4
6.5	OAUTH.REVOKE_CLIENT_ROLE	6-5
6.6	OAUTH.UPDATE_CLIENT	6-5
6.7	OAUTH.ROTATE_CLIENT_SECRET	6-7
6.8	OAUTH.UPDATE_CLIENT_SECRET	6-8
6.9	OAUTH.IMPORT_CLIENT	6-9
6.10	OAUTH.CREATE_JWT_PROFILE	6-11
6.11	OAUTH.DELETE_JWT_PROFILE	6-12

7 OAUTH_ADMIN PL/SQL Package Reference

7.1	OAUTH_ADMIN.CREATE_JWT_PROFILE	7-1
-----	--------------------------------	-----

7.2	OAUTH_ADMIN.DELETE_JWT_PROFILE	7-3
-----	--------------------------------	-----

8 Enabling ORDS Database API

8.1	Basic Setup to Enable ORDS Database API	8-1
8.2	Advanced Setup to Enable the ORDS Database API	8-2
8.2.1	Pluggable Database Lifecycle Management	8-3
8.2.2	Disabling PDB Lifecycle Management	8-3
8.3	Creating a Default Administrator	8-4
8.4	Configuration of Database API Environment Services	8-4
8.5	Configuration of Database API with Open Service Broker API Compatible Platforms	8-5

9 REST-Enabled SQL Service

9.1	REST-Enabled SQL Service Terminology	9-1
9.2	Configuring the REST-Enabled SQL Service	9-2
9.3	Using cURL with REST-Enabled SQL Service	9-2
9.4	Getting Started with the REST-Enabled SQL Service	9-3
9.4.1	REST-Enabling the Oracle Database Schema	9-4
9.4.2	REST-Enabled SQL Authentication	9-4
9.4.3	REST-Enabled SQL Endpoint	9-4
9.5	REST-Enabled SQL Service Examples	9-5
9.5.1	POST Requests Using application/sql Content-Type	9-5
9.5.1.1	Using a Single SQL Statement	9-6
9.5.1.2	Using a File with cURL	9-7
9.5.1.3	Using Multiple SQL Statements	9-8
9.5.2	POST Requests Using application/json Content-Type	9-11
9.5.2.1	Using a File with cURL	9-11
9.5.2.2	Specifying the Limit Value in a POST Request for Pagination	9-13
9.5.2.3	Specifying the Offset Value in a POST Request for Pagination	9-14
9.5.2.4	Defining Binds in a POST Request	9-16
9.5.2.5	Specifying Batch Statements in a POST Request	9-20
9.5.3	Example POST Request with DATE and TIMESTAMP Format	9-23
9.5.4	Data Types and Formats Supported	9-24
9.6	REST-Enabled SQL Request and Response Specifications	9-29
9.6.1	Request Specification	9-29
9.6.2	Response Specification	9-31
9.7	Supported SQL, SQL*Plus, and SQLcl Statements	9-36
9.7.1	Supported SQL Statements	9-36
9.7.2	Supported PL/SQL Statements	9-37
9.7.3	Supported SQL*Plus Statements	9-37
9.7.3.1	Set System Variables	9-38

9.7.3.2	Show System Variables	9-39
9.7.4	Supported SQLcl Statements	9-40
9.8	REST-Enabled SQL Service and MySQL Database	9-40
9.8.1	Examples	9-40

10 GraphQL in Oracle REST Data Services

10.1	GraphQL Terminology	10-1
10.2	Enabling GraphQL in Oracle REST Data Services	10-1
10.3	Enabling Objects for GraphQL	10-2
10.3.1	Accessing Protected REST-Enabled Objects	10-2
10.4	Accessing Objects Using GraphQL queries	10-2
10.4.1	Getting GraphQL Schema	10-3
10.4.2	Simple Query	10-4
10.4.3	Join Query	10-5
10.4.3.1	Circular Relationships Between Objects	10-11
10.5	Examples of Filtering in Queries	10-13
10.5.1	Supported Data Types	10-13
10.5.2	Filtering by Primary Key	10-13
10.5.2.1	Filtering by Composite Primary Key	10-14
10.5.3	Where Filter	10-14
10.5.3.1	Example: EQUALS (eq) operator	10-16
10.5.3.2	Example: Greater than (>) Operator and Date Data Type	10-17
10.5.3.3	Example: LIKE (like) operator	10-17
10.5.3.4	Example: IN (in) operator	10-18
10.5.3.5	Example: NOT (not) Operator	10-18
10.5.3.6	Example: AND (and) operator	10-20
10.5.3.7	Example: OR (or) operator	10-22
10.5.3.8	Example: Where Filter in Children Types	10-22
10.5.3.9	Working with Dates/Timestamps Using Filters	10-24
10.6	Sorting the Data	10-27
10.6.1	Example: Sorting by Multiple Columns	10-28
10.7	Keyset Pagination	10-29
10.7.1	Example: Pagination with Other Filters	10-30
10.7.2	Example: Pagination in Nested Types	10-30
10.8	Using Dynamic Arguments in Queries: Variables	10-31
10.9	GraphiQL	10-32

11 Extending ORDS Functionality with Plugins

11.1	Plugin Demonstration Example	11-1
11.2	Plugin Javascript	11-1

12 Migrating from mod_plsql to ORDS

12.1	Oracle HTTP Server mod_plsql Authentication	12-1
12.2	Example Oracle HTTP Server DAD file	12-1
12.3	Mapping mod_plsql Settings to ORDS	12-3
12.4	Example ORDS Configuration Files	12-7
12.4.1	Example Configuration File for Basic Authentication	12-7
12.4.2	Example Configuration File for Basic Dynamic Authentication	12-7
12.4.3	Example Configuration file for Custom Authentication	12-8
12.5	Example ORDS URL Mapping	12-9
12.6	Example ORDS Default Configuration	12-9
12.7	Oracle REST Data Services Functionality	12-10
12.7.1	Basic Authentication	12-10
12.7.2	Basic Dynamic Authentication	12-10
12.7.3	Custom Authentication	12-10
12.8	ORDS Features	12-11
12.8.1	Request Validation Function	12-11
12.8.2	Pre Process Feature	12-12
12.8.3	Post Process Feature	12-12
12.8.4	File Upload Feature	12-12
12.8.5	Cross-Origin Resource Sharing Feature	12-13
12.8.6	Procedure Allow List	12-13
12.8.6.1	Configuring ORDS PL/SQL Gateway Allow List	12-13
12.9	Modifying Synonyms	12-15

A Setting-up a PL/SQL Gateway User

B Oracle REST Data Services Database Type Mappings

B.1	Oracle Built-in Types	B-1
B.2	Handling Structural Database Types	B-3
B.3	Oracle Geospatial Encoding	B-5
B.4	Enabling Database Mapping Support	B-5

C Troubleshooting Oracle REST Data Services

C.1	Enabling Detailed Request Error Messages	C-1
C.2	ORDS User Defined Service	C-1

D Third-Party License Information

D.1	Hack 3.003	D-1
D.2	Monaco Editor 0.44.0	D-2
D.3	babel-polyfill 7.20.15	D-3
D.4	gridstack.js 10.1.0	D-3
D.5	d3-flame-graph 4.1.3	D-4
D.6	Dexie 3.2.4	D-13
D.7	Jetty 10.0.20	D-17
D.8	jackson-core 2.16.1	D-29
D.9	Jakarta Activation API (JAF) 2.1.2	D-35
D.10	jackson-annotations 2.16.1	D-37
D.11	jackson-databind 2.16.1	D-42
D.12	graphql-js 16.8.0	D-50
D.13	avsc 5.7.3	D-51
D.14	D3 7.8.4	D-51
D.15	long.js 5.2.0	D-67
D.16	SnappyJS 0.6.1	D-71
D.17	JavaScript Extension Toolkit (JET) 14.0.0	D-71
D.18	Commons FileUpload 1.5	D-86
D.19	opentelemetry-java 1.24.0	D-89
D.20	Google Guava 33.0.0	D-97
D.21	xml2js 0.6.2	D-104
D.22	Jansi 2.4.1	D-105
D.23	commons-io 2.15.1	D-109
D.24	Join Monster 3.3.1	D-113
D.25	SheetJS 0.20.1	D-115
D.26	graphql-compose 9.0.10	D-119
D.27	jackson-dataformat-xml 2.16.1	D-120
D.28	hotkeys-js 3.13.7	D-130
D.29	swagger-ui 5.1.1	D-130
D.30	Commons Compress 1.26.0	D-174

Index

List of Examples

2-1	Enabling the PL/SQL Function	2-28
2-2	Enabling the PL/SQL Procedure	2-29
2-3	Generating an Endpoint for the Stored Procedure	2-31
2-4	Package Procedure and Function Endpoints	2-31
2-5	Nested JSON Purchase Order with Nested Lineltems	2-49
2-6	PL/SQL Handler Code Used for a POST Request	2-50
2-7	GET Handler Code using Oracle REST Data Services Query on Relational Tables for Generating a Nested JSON object	2-54
2-8	PL/SQL API Call for Creating a New test/:id Template and GET Handler in the demo Module	2-55
2-9		2-57
2-10	ACL Rule in the Database	2-62
2-11	Setting the Duser.timezone Java Environment Variable in Standalone Mode	2-66
2-12	Setting the Duser.timezone Java Environment Variable in a Java Application Server	2-67
2-13	Setting Enabled for all Pools	2-103
3-1	Example	3-5
6-1	Example to Add Multiple Privileges	6-7
9-1	Example cURL Command	9-2
9-2	Binds in POST Request	9-16
9-3	Complex Bind in POST Request	9-18
9-4	Batch statements	9-20
9-5	Batch bind values	9-21
9-6	Oracle REST Data services Time Zone Set as Europe/London	9-23
9-7	PL/SQL Statement	9-37
9-8	Script	9-41
9-9	Query	9-42
9-10	Export	9-44
12-1	dads.conf file	12-2
12-2	ords_conf/ords/conf/basic_auth.xml	12-7
12-3	ords_conf/ords/conf/basic_dynamic_auth.xml	12-7
12-4	ords_confs/ords/conf/custom_auth.xml	12-8
12-5	ords_conf/ords/url-mapping.xml	12-9
12-6	ords_conf/ords/defaults.xml	12-9
12-7	security.requestValidationFunction	12-12
12-8	procedure.preProcess	12-12
12-9	procedure.postProcess	12-12
12-10	Table upload	12-12

12-11	Procedure upload	12-13
12-12	Curl command for file upload	12-13

List of Figures

1-1	ORDS Landing Page	1-2
2-1	Relationship Between Components of the ORDS RESTful Web Services	2-4
2-2	Architecture Diagram for a GET Operation	2-4
2-3	Selecting the Enable REST Service Option	2-30
2-4	Auto Enabling the PL/SQL Package Object	2-30
2-5	Adding an Anonymous PL/SQL Block to the Handler for the PUT Method	2-39
2-6	Setting the Bind Parameter l_salarychange to Pass for the PUT Method	2-40
2-7	Obtaining the URL to Call from the Details Tab	2-40
2-8	Displaying the Results from a SQL Query to Confirm the Execution of the PUT Method	2-41
2-9	Creating a Template Definition to Include a Route Pattern for Some Parameters or Bind Variables	2-43
2-10	Adding a SQL Query to the Handler	2-44
2-11	Using Browser to Show the Results of Using a Route Pattern to Send a GET Method with Some Required Parameter Values	2-45
2-12	Using Browser to Show the Results of Using a Query String to Send a GET Method with Some Parameter Name/Value Pairs	2-46
2-13	Complete Response Body in JSON Format	2-53
2-14	Generating Nested JSON Objects	2-57

List of Tables

2-1	Parameters for batchload	2-17
2-2	ORDS Request Object Properties	2-58
2-3	ORDS Response Object Functions	2-59
3-1	List of Implicit Parameters	3-1
3-2	Pagination Implicit Parameters	3-8
8-1	Open Service Broker Service Catalog	8-5
10-1	Supported Operators	10-15
12-1	Mappings of mod_plsql Directives to ORDS Settings	12-3
C-1	List of ORDS user defined service	C-2

Preface

Oracle REST Data Services Developer's Guide explains how to develop applications using Oracle REST Data Services. (Oracle REST Data Services was called *Oracle Application Express Listener* before Release 2.0.6.)

Topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

Audience

This document is intended for application developers who develop applications using 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.
<i>italic</i>	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 Release 24.1 Oracle REST Data Services Developer's Guide

Changes in Oracle REST Data Services 24.1

New Features

- Oracle Database Multilingual Engine (MLE) JavaScript functionality: See [Manually Creating RESTful Services Using Javascript](#)
- Updated `ORDS.CREATE_SERVICE`, `ORDS.DEFINE_HANDLER`, and `ORDS.DEFINE_SERVICE` functions with a new entry `source_type_mle_javascript`. See [ORDS PL/SQL Package Reference](#)

Other Changes

Updated the [Third-Party License Information](#) section in this release.

1

Introduction to Oracle REST Data Services

This chapter provides an overview of Oracle REST Data Services and its features.

Topics:

- [About Oracle REST Data Services](#)
- [Features of Oracle REST Data Services](#)

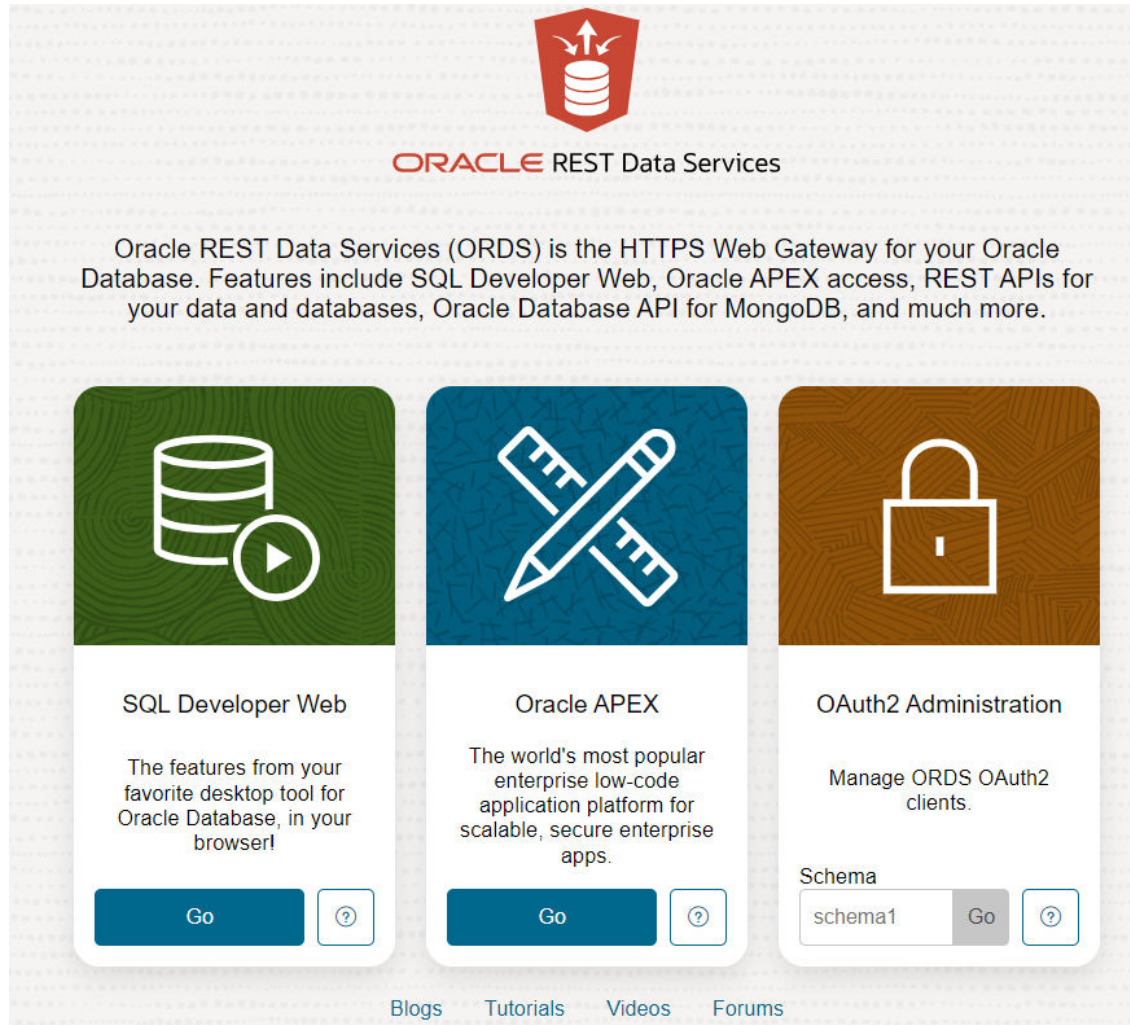
1.1 About Oracle REST Data Services

Oracle REST Data Services (ORDS) is the HTTPS Web Gateway for your Oracle Database, which includes features such as Oracle Database Actions, Oracle APEX access, REST APIs for your data and databases, Oracle Database API for MongoDB, and much more. 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 APEX architecture requires a web server to proxy requests between a web browser and the Oracle APEX engine. Oracle REST Data Services Meets the requirement but its use goes beyond that of Oracle APEX 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.

Starting with release 23.2, ORDS provides a default landing page. The landing page displays the main tools and also lets you know whether a particular tool is enabled or disabled. If a tool is disabled, then you can click the help button (?) to navigate to the corresponding documentation and get help to install or enable the tool.

Figure 1-1 ORDS Landing Page



1.2 Features of Oracle REST Data Services

This section lists the features of Oracle REST Data Services (ORDS).

Database Actions

Database Actions, is a web-based interface that provides development, data tools, administration, and monitoring features for Oracle Database. Additionally, ORDS is provided as a managed feature of the Oracle Autonomous Database Cloud Services.



See Also:

[Database Actions Home Page](#)

REST-Enabled SQL

REST-Enabled SQL is a REST API that allows for ad-hoc SQL and SQL Scripts to be executed. You can POST one or more SQL statements to the service. The service then runs

the SQL statements against Oracle Database and returns the results and output to the client in a JSON format.



Note:

[REST-Enabled SQL Service](#)

Database REST APIs

ORDS includes a collection of more than 500 REST APIs for performing operations such as monitoring and maintaining your Oracle Database, including PDB lifecycle management, performance, security, data dictionary, data pump.



See Also:

[Enabling ORDS Database API](#)

REST APIs

Provides the ability to define the REST APIs with SQL and PL/SQL. ORDS marshals SQL and PL/SQL types to and from JSON, auto-paginates the results of your SQL queries, supports GeoJSON for spatial, handles common database errors with appropriate HTTPS responses and much more. Users can also choose to REST enable tables, views, and stored procedures to take advantage of the AutoREST feature.



See Also:

[ORDS REST APIs](#)

PL/SQL Gateway

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 the new ORDS configuration files. The mod_plsql database resources such as before procedures, after procedures, request validation functions, owa_custom packages, document upload procedures and document tables require no change when you are migrating to ORDS. PL/SQL gateway enables you to access your APEX applications from an application server such as WebLogic or Tomcat.



See Also:

[About the Database Users Used by Oracle REST Data Services](#)

2

Developing Oracle REST Data Services Applications

This section explains how to develop applications that use Oracle REST Data Services (ORDS).

See Also:

If you want to get started quickly, you can try the tutorial in Oracle REST Data Services Quick Start Guide.

Note:

- Ensure that you have installed and configured both Oracle APEX 4.2 or later, and Oracle REST Data Services 3.0 or later, prior to attempting the examples discussed in this chapter.
- Install the Oracle REST APIs prior to using the Oracle REST APIs for JSON Data Persistence. See *Oracle REST Data Services SODA for REST Developer's Guide*
- Refer to the [Oracle APEX Documentation](#), if you are new to Oracle APEX.

Topics:

- [Introduction to Relevant Software](#)
- [Getting Started with RESTful Services](#)
- [Automatic Enabling of Schema Objects for REST Access \(AutoREST\)](#)
- [Manually Creating RESTful Services Using SQL and PL/SQL](#)
- [Manually Creating RESTful Services Using Javascript](#)
- [About Working with Dates Using Oracle REST Data Services](#)
- [Creating RESTful Web Services Using Database Actions](#)
- [Configuring Secure Access to RESTful Services](#)
- [JWT Bearer Token Authentication and Authorization Using JWT Profile](#)
- [About Oracle REST Data Services User Roles](#)
- [Authenticating Against WebLogic Server User Repositories](#)
- [Integrating with Existing Group/Role Models](#)
- [Integrating Oracle REST Data Services and WebLogic Server](#)
- [Using the Oracle REST Data Services PL/SQL API](#)

- [Oracle REST Data Services Database Authentication](#)
- [Overview of Pre-hook Functions](#)
- [Generating Hyperlinks](#)
- [About HTTP Error Responses](#)

2.1 Introduction to Relevant Software

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

Topics:

- [Oracle APEX](#)
- [REST APIs](#)

Related Topics

- [About Oracle REST Data Services](#)

2.1.1 Oracle APEX

ORDS makes your APEX applications available to the various application servers like WebLogic Server or Tomcat, through the PL/SQL Gateway feature. It is a fully-supported, no-cost option available with all editions of Oracle Database. Using only a web browser, you can develop and deploy professional applications that are both fast and secure.

2.1.2 REST APIs

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 REST 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.

ORDS provides a built-in web application, SQL Developer Web, which is used to build, test, document, and secure your REST APIs.

2.2 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](#)

2.2.1 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

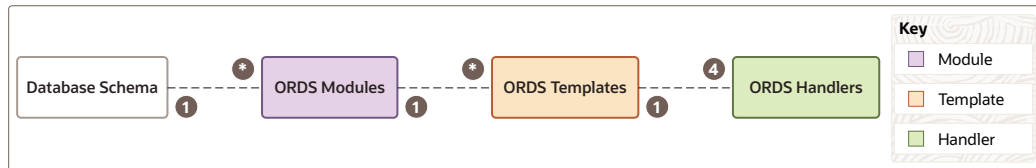
- [REST APIs](#)

2.2.2 ORDS RESTful Web Services Architecture Diagrams

This section describes the ORDS RESTful web services architecture diagrams.

The following diagram illustrates the relationship between the different components of the ORDS RESTful Web Services architecture:

Figure 2-1 Relationship Between Components of the ORDS RESTful Web Services



The Database Schema is the schema that you have REST-enabled. It can contain several resource modules. Similarly, a resource module, which is the top-level container for the REST Services offered by ORDS, can contain several resource templates. The resource templates are represented by the trailing part of the URL. Every resource template can contain four resource handlers, namely, GET, POST, PUT, and DELETE.

After you create a RESTful Web Service, you can test it by entering the following URL in your browser:

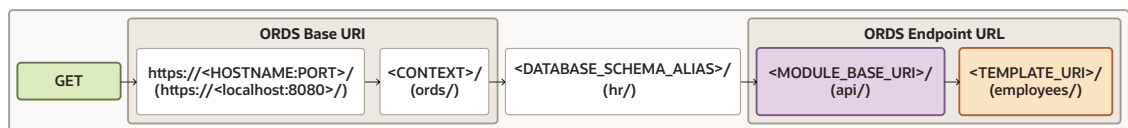
```
https://<HOSTNAME:PORT>/<CONTEXT>/<DATABASE_SCHEMA_ALIAS>/<MODULE_BASE_URI>/<TEMPLATE_URI>/
```

Where:

- **HOSTNAME:PORT/CONTEXT:** Specifies the address at which ORDS is running. You can also refer to it as the ORDS Base URI.
- **DATABASE_SCHEMA_ALIAS:** Specifies the name that you provided while REST-enabling your database schema. By default, it is the name of the schema in lowercase.
- **MODULE_BASE_URI:** Specifies the URI of the module.
- **TEMPLATE_URI:** Specifies the URI of the template. This value, along with the **MODULE_BASE_URI**, comprises the ORDS Endpoint URL.

The following diagram illustrates how a GET operation is performed:

Figure 2-2 Architecture Diagram for a GET Operation



In this case, you will enter the following URL in your browser to perform the GET operation:

```
https://localhost:8080/ords/hr/api/employees/
```

2.2.3 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.

2.2.4 "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.)

2.2.5 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 APEX \(Tutorial\)](#)

See Also:

`curl` - command line tool and library
The example in this section uses cURL with the services mentioned in [Exploring the Sample RESTful Services in APEX \(Tutorial\)](#)

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

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 lose some flexibility and customizability if you use the AutoREST feature, but it reduces your time and effort to a significant extent. 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.

If you are using Database Actions or SQL Developer, you can AUTOREST enable the database objects with convenient wizards. REST Data Services also provides an ORDS PL/SQL package that can be used to enable objects for REST.

 **Note:**

This feature is only available for Oracle REST Data Services enabled schemas and not for Oracle APEX workspaces.

 **See Also:**

[ORDS.ENABLE_OBJECT](#)

To enable Oracle REST Data Services access to one or more specified tables, views, or PL/SQL programs, you can 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**. Once the schema is enabled, you can use that schema or user to login to SQL Developer Web and REST Enable objects in your schema using the web interface.
(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**.
3. **Schema Alias:** You can alias the schema in the URIs for your REST APIs. This prevents your API consumers from knowing your database user accounts.
4. **Authorization Required:** This protects the API Catalog endpoints for your schema. If you enable this option, then the requests to the metadata-catalog endpoint on your schema will require authorization.

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

2.3.1 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](#)

2.3.1.1 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.

2.3.1.2 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/"
    }
  ]
}
```

2.3.1.3 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/"
    }
  ]
}
```

```
]
}
```

2.3.1.4 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"
    }
  ]
}

```

2.3.1.5 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"
      }
    ]
  }
}

```

2.3.1.6 Get Table Row Using Primary Key

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

Note:

- If a table does not have a primary key, then ORDS uses the ROWID to uniquely address the rows.
- The primary keys are not compatible with a REST interface if they meet any of the following characteristics:
 - End with a period
 - Contain // or \\
 - Begin with /
 - Contains two or more periods in sequence (For example: ..., ...)
 - Contains any of the following characters: "<", ">", ".", ":", "!", "?", "*", "#", ";", or "%" Requests that contain such primary keys returns *HTTP 400 Bad Request* as a response. If the primary keys contain any of the preceding incompatible characters, then it is recommended to have a secondary key that does not conflict with the link generation rules.

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/"
    }
  ]
}
```

2.3.1.7 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 the POST request can include the primary key value in the body. Or, if the table has an IDENTITY CLAUSE, sequence or trigger, then the primary key column may be omitted. If the table does not have a primary key, then the ROWID of the row is 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/"
  }
]
}

```

2.3.1.8 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/"
}
]
```

2.3.1.9 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
}
```

2.3.1.10 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 2-1](#).

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

Parameters:

Table 2-1 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 <code>errorsMax</code> (the service option) or by <code>errors</code> (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 <code>errorsMax</code> value), specify UNLIMITED (-1) .

Table 2-1 (Cont.) Parameters for batchload

Parameter	Description
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 <code>errorsMax</code> (the service option) or by <code>errors</code> (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 (<code>\r</code> , <code>\r\n</code> or <code>\n</code>), then <code>lineEnd</code> does not need to be specified.
lineMax	Sets a maximum line length for identifying lines/rows in the data stream. A <code>lineMax</code> value will prevent reading an entire stream as a single line when the incorrect <code>lineEnd</code> character is being used. The default is unlimited.
locale	Sets the locale.
responseEncoding	Sets the encoding for the response stream.
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. <code>False</code> (the default) does not delete table data before the load; <code>True</code> causes table data to be deleted with the DELETE SQL statement; <code>Truncate</code> 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
```

2.3.2 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"}
```

2.3.2.1 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
  columnName : sortingNulls
  columnName : sortingValues

sortingValues
  [sortingValue]
  [sortingNulls]
  [sortingValue, sortingNulls]
  [sortingNulls, sortingValue]

sortingNulls
  "NULLS FIRST"
  "NULLS LAST"

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.

2.3.2.2 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}
}
```

Order by with nulls first

```
{
  "$orderby": {"SALARY": ["ASC", "NULLS FIRST"]}
}
```

Order by with nulls last

```
{
  "$orderby": {"SALARY": ["ASC", "NULLS LAST"]}
}
```

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. Escape 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%"} } ] }  
}
```

2.3.3 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.

2.3.3.1 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`.

2.3.3.2 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](#)

2.3.3.2.1 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 2-1 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 2-2 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;
/
```

2.3.3.2.2 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.

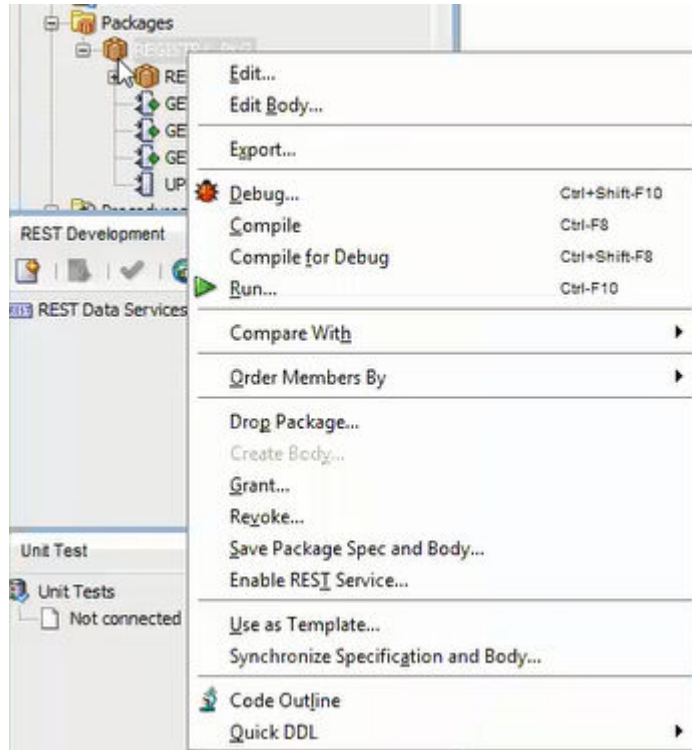
To enable the PL/SQL objects (for example, package) using SQL Developer, perform the following steps:

 **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.

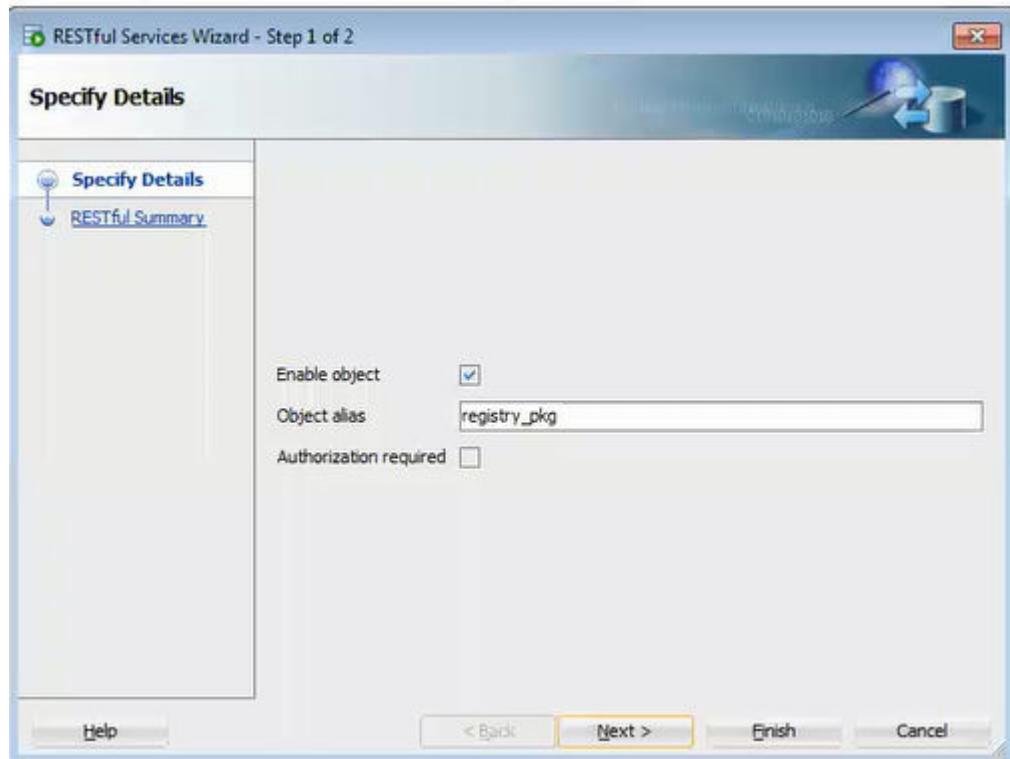
1. In SQL Developer, right-click on a package as shown in the following figure:

Figure 2-3 Selecting the Enable REST Service Option



2. Select **Enable RESTful Services** to display the following wizard page:

Figure 2-4 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**.

2.3.3.3 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 namespace.

Example 2-3 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 2-4 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.

2.3.3.4 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.

2.3.3.5 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:

```
{  
  
}
```

2.3.3.6 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"  
}
```

2.3.4 Support for JSON-Relational Duality View

ORDS supports AutoREST enabling of JSON-relational duality view functionality. This functionality is supported only with Oracle Database 23c or later.

JSON-relational duality view is a revolutionary Oracle Database feature that combines the benefits of relational databases and NoSQL JSON document stores. This feature allows the storage of normalized data in relational tables while exposing it to applications in JSON. Multiple JSON-relational duality views can be created on the same relational data to address different use cases. In other words, the same relational data can have different JSON representations.

Note:

For best performance, configure the Oracle REST Data Services (ORDS) metadata cache.

See Also:

- [Configuring ORDS Metadata Cache](#)
- [Understanding Configurable Settings](#)
- [JSON-Relational Duality Developer's Guide](#)

2.3.4.1 Table AutoREST Versus JSON-Relational Duality View AutoREST

A JSON-relational duality view is classified as a VIEW in Oracle Database, so it can be AutoRest enabled like any relational view. This section provides a comparison between the AutoREST functionality of JSON-relational duality views with relational tables:

Similarities:

- Exposes the same set of endpoints and methods (GET, PUT, POST, DELETE, and HEAD)
- Uses the same comma-separated primary key identifier format as that of the associated root table
- Supports the same Read, Create, Upsert, or Delete semantics
- Generates the same HTTP `If-None-Match` header ETag digest, where multiple items are processed.
- Injects the `links` hyperlinks field into the response payload

Differences:

- Supported only with Oracle Database 23c or later
- Passes the JSON payload directly between the request or response and the JSON-relational duality view DATA column.
- Uses the JSON-relational duality view ETag value for HTTP `If-Match` and `If-None-Match` header conditional matching, where a single item is processed (GET, PUT, and DELETE methods).
- Uses the SODA extended Query by Example (QBE) syntax for rich filtering and ordering
- Uses a JSON-friendly `batchload` format

2.3.4.2 Support for Enhanced ETag Matching

Oracle REST Data Services (ORDS) integrates with the JSON-relational duality view ETag feature to support optimistic locking and client caching.

HTTP ETag Matching

ORDS uses the JSON-relational duality view generated ETag instead of its own digest value when evaluating matching headers for single item operations such a GET (`If-None-Match`) and PUT/DELETE (`If-Match`).

Match Header	HTTP False Response	Header Example
<code>If-None-Match</code>	304 - "Not Modified"	<code>If-None-Match: "536001F31A8718819AEEF28EC 20D8677"</code>
<code>If-Match</code>	412 - "Precondition Failed"	<code>If-Match: "536001F31A8718819AEEF28EC 20D8677"</code>



Note:

The double-quotes around the ETag value are mandatory.

Database ETag Matching

The Oracle Database also performs ETag matching for `UPDATE` operations where an ETag is available in the `_metadata` object of the request payload, otherwise this field is ignored in all other cases.

Content Example	HTTP 'False' Response
<pre>{ ... "_metadata": { "etag": "536001F31A8718819AEEF28EC20D8677", "asof": "00000000002BECD5" }, ... }</pre>	412 - "Precondition Failed"

2.3.4.3 Enhanced JSON QBE (Query by Example) Filtering

Oracle REST Data Services (ORDS) exposes the same QBE filtering syntax that Simple Oracle Document Access (SODA) uses, providing the user with a robust set of JSON operators and functionality that are more appropriate for processing JSON.

Although, the syntax currently only applies to JSON-relational duality views, it is specified in the `q` URL parameter, similar to the relational tables and views.

The following example filters the content of the `race_dv` JSON-relational duality view, where the `points` field is greater than 40:

```
curl http://localhost:8080/ords/janus/race_dv/?q={"points":{"$gt":40}}
```

The following example adds ordering on the `points` field to the preceding example:

```
curl http://localhost:8080/ords/janus/race_dv/?q={"$query":{"points":{"$gt":40}}, "$orderby":[{"path":"points","datatype":"number"}]}
```



See Also:

[Simple Oracle Document Access \(SODA\)](#)

2.3.4.4 Enhanced JSON Batch Loading

As the JSON-relational duality view `DATA` column is mapped directly to the request payload, the same approach should be applied to batch loading. Therefore, ORDS provides an optimized `batchload` endpoint that accepts one of the following JSON content types:

Header Content-Type	Description
application/json	<p>Freely formatted JSON array of JSON documents payload. For example:</p> <pre>[{ "x":1, "y":1 }, { "x":2, "y":2 }]</pre>
application/json; boundary=LF	<p>Linefeed delimited list of JSON documents. Payload example:</p> <pre>{"x":1,"y":1} {"x":2,"y":2}</pre>

Each JSON document is passed to the ORDS batch load service as a row and can be fine-tuned with the query parameters in the same way as in any table.

For example, `batchesPerCommit`, `batchRows`, and `truncate` can be used to optimize the batch loading process.

The following example shows the batch loading of the `points_dv` JSON-relational duality view in batches of 25 rows of JSON document:

```
curl -i -X POST --data-binary @points.json -H "Content-Type: application/json"
      http://localhost:8080/ords/ordstest/points_dv/batchload?batchRows=25
```

The familiar batch process result is returned in the response as shown in the following code snippet:

```
HTTP/1.1 200 OK
...
#INFO Number of rows processed: 2
#INFO Number of rows in error: 0
#INFO Last row processed in final committed batch: 2
SUCCESS: Processed without errors
```

2.4 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](#)

2.4.1 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](#)

2.4.1.1 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;
```

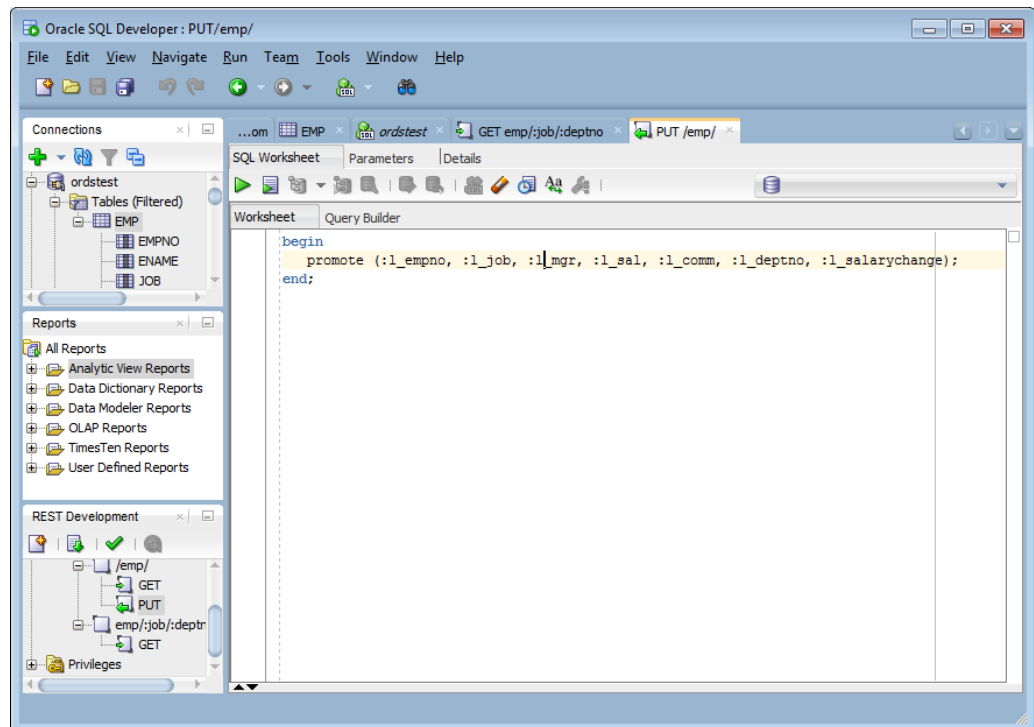
As a privileged `ordstest` user, connect to the `ordstest` schema and create the `promote` stored procedure.

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;
```

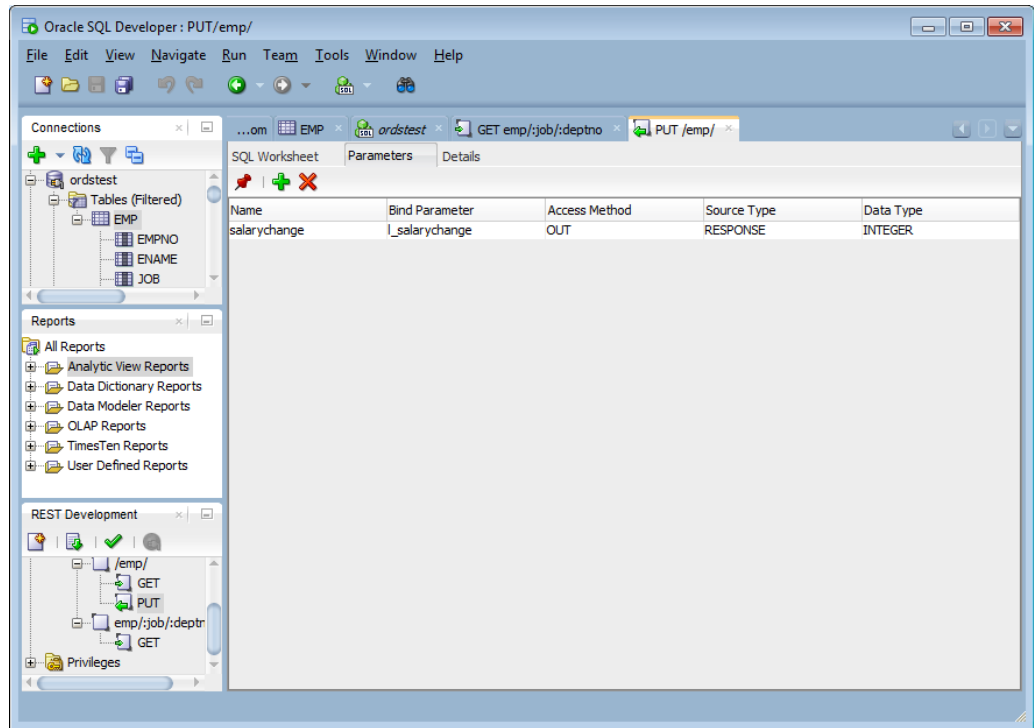
as shown in the following figure.

Figure 2-5 Adding an Anonymous PL/SQL Block to the Handler for the PUT Method



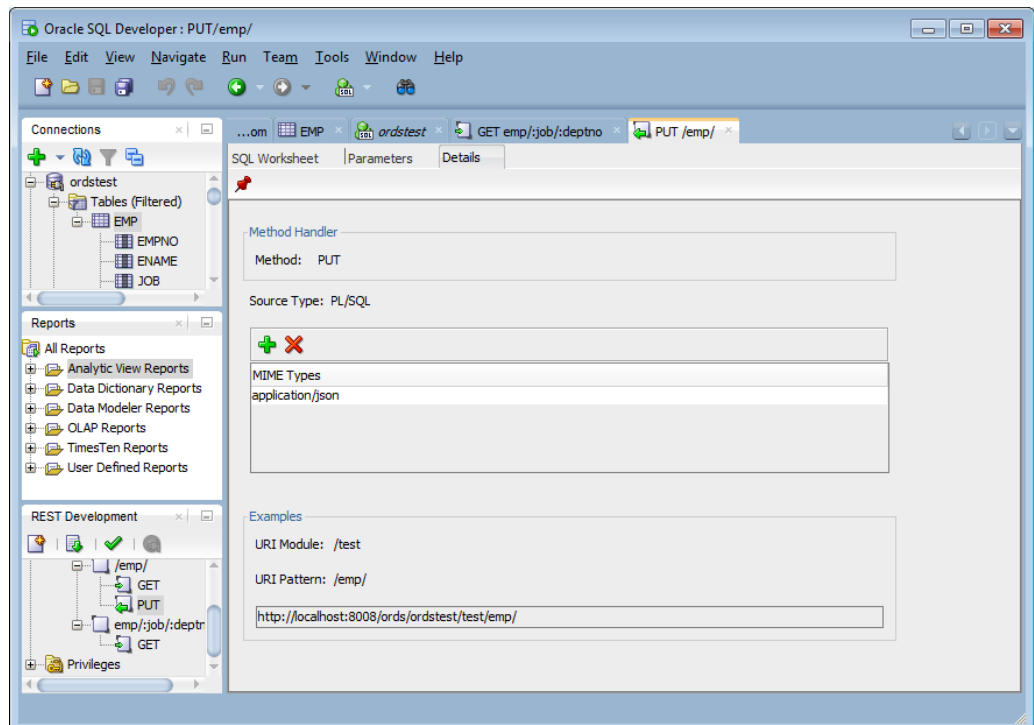
- d. Click the **Parameters** tab to set the **Bind Parameter** as `l_salarychange`, the **Access Method** as an **OUT** parameter, the **Source Type** as `RESPONSE`, and **Data Type** as `INTEGER` as shown in the following figure. This is the promote procedure's output which is an integer value equal to the change in salary in a JSON name/value format.

Figure 2-6 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 2-7 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}'
```

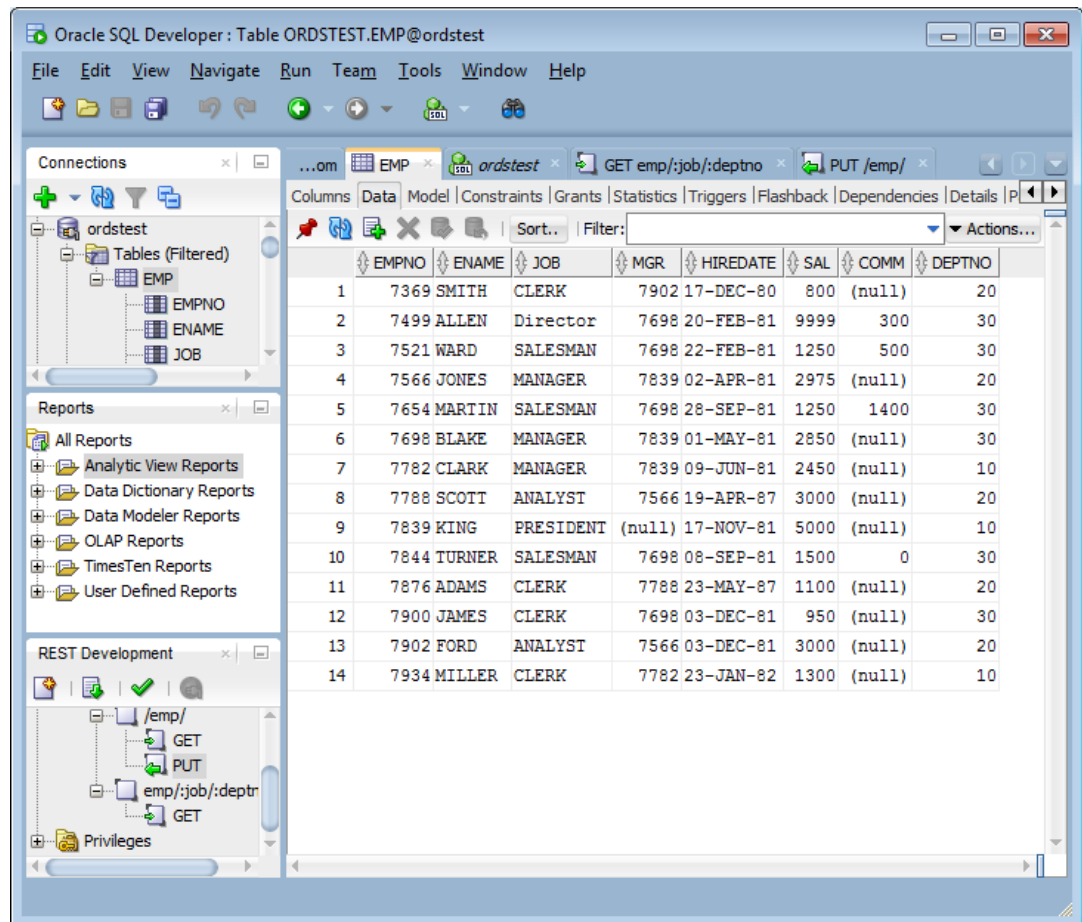
 **Note:**
You can also use any REST client available to test the RESTful service.

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 2-8 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.

2.4.1.2 Using Route Patterns to Pass Parameters

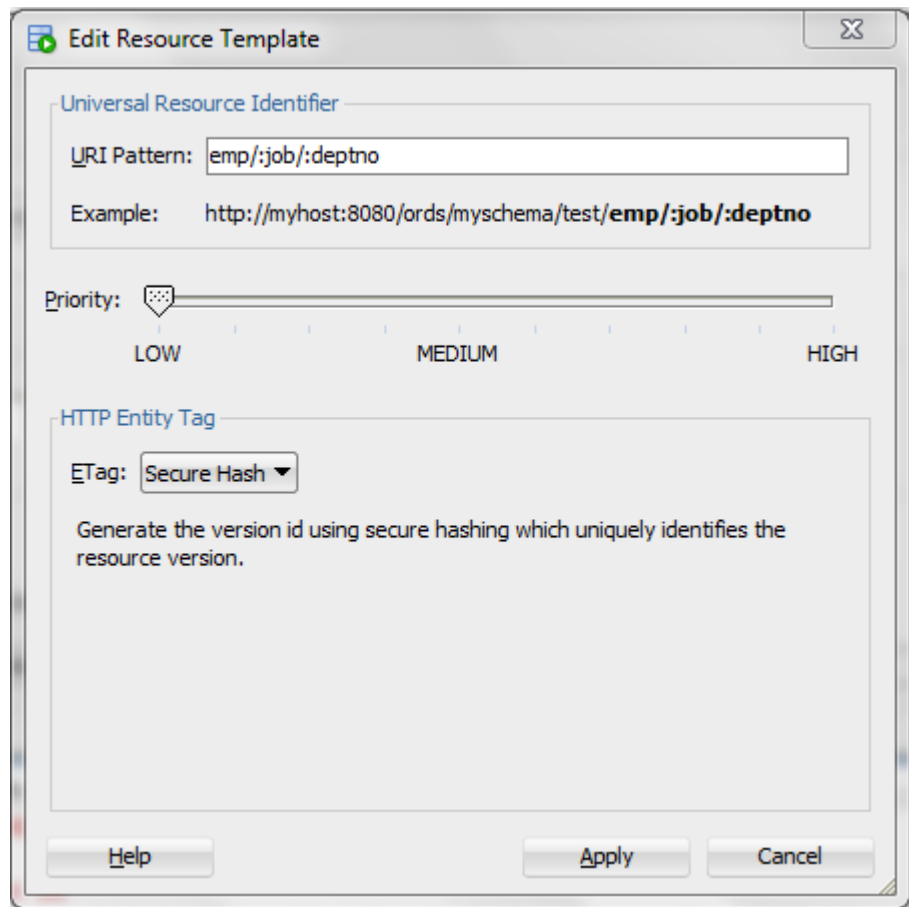
This section describes how to use route patterns in the URI to specify parameters for REST requests, such as with the `GET` method, which does not have a body.

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:

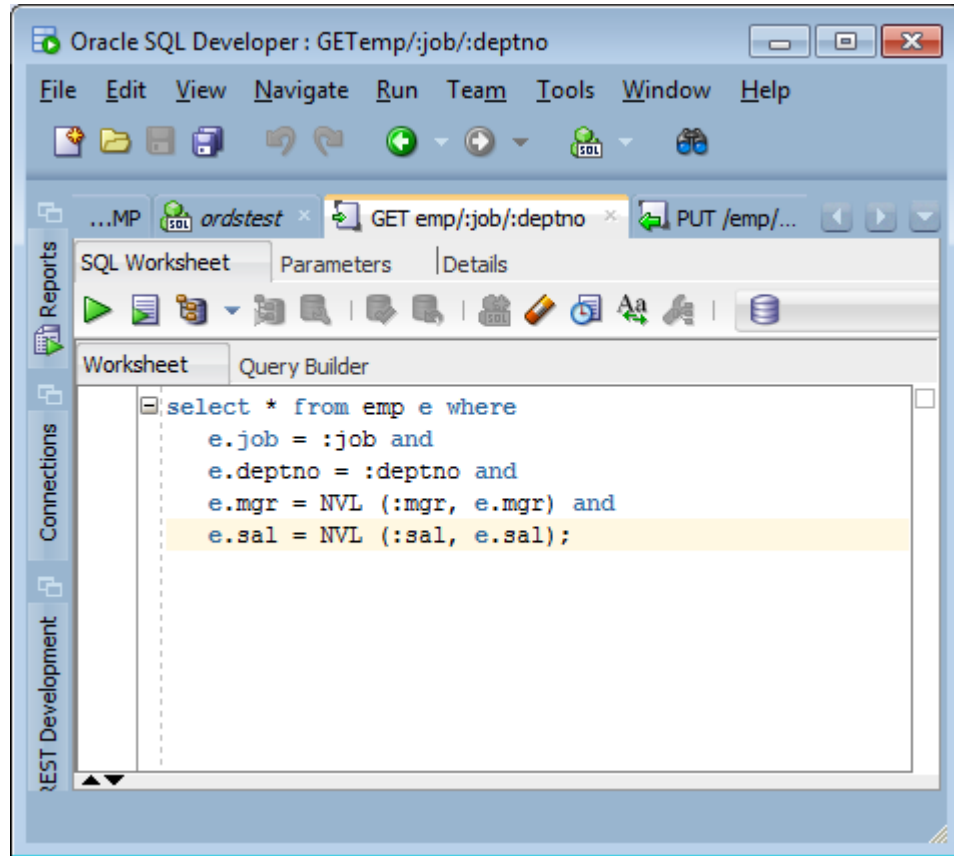
1. In SQL Developer, right click on the test module and select **Add Template** to create a new template that calls `emp`; however, in this case the template definition includes a route pattern for the parameters or bind variables that is included in the URI rather than in the body of the method. To 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 2-9 Creating a Template Definition to Include a Route Pattern for Some Parameters or Bind Variables

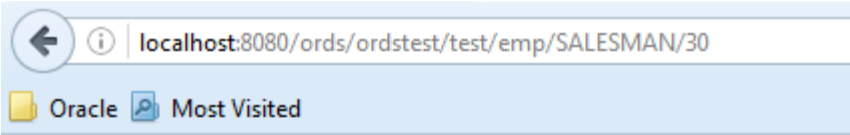


2. Click **Next** to go to **REST Data Services — Step 2 of 3**, and click **Next** to go to **REST Data Services — Step 3 of 3**, then click **Finish** to complete the template.
3. Right click on the `emp/:job/:deptno` template and select **Add Handler** for the `GET` method.
4. Right click on the `GET` 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 2-10 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 2-11 Using Browser to Show the Results of Using a Route Pattern to Send a GET Method with Some Required Parameter Values

```
{
  items: [
    {
      empno: 7521,
      ename: "WARD",
      job: "SALESMAN",
      mgr: 7698,
      hiredate: "1981-02-21T18:30:00Z",
      sal: 1250,
      comm: 500,
      deptno: 30
    },
    {
      empno: 7654,
      ename: "MARTIN",
      job: "SALESMAN",
      mgr: 7698,
      hiredate: "1981-09-27T18:30:00Z",
      sal: 1250,
      comm: 1400,
      deptno: 30
    },
    {
      empno: 7844,
      ename: "TURNER",
      job: "SALESMAN",
      mgr: 7698,
      hiredate: "1981-09-07T18:30:00Z",
      sal: 1500,
      comm: 0,
      deptno: 30
    }
  ],
  hasMore: false,
  limit: 25,
  offset: 0,
  count: 3,
  ...
}
```

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](#)

2.4.1.3 Using Query Strings for Optional Parameters

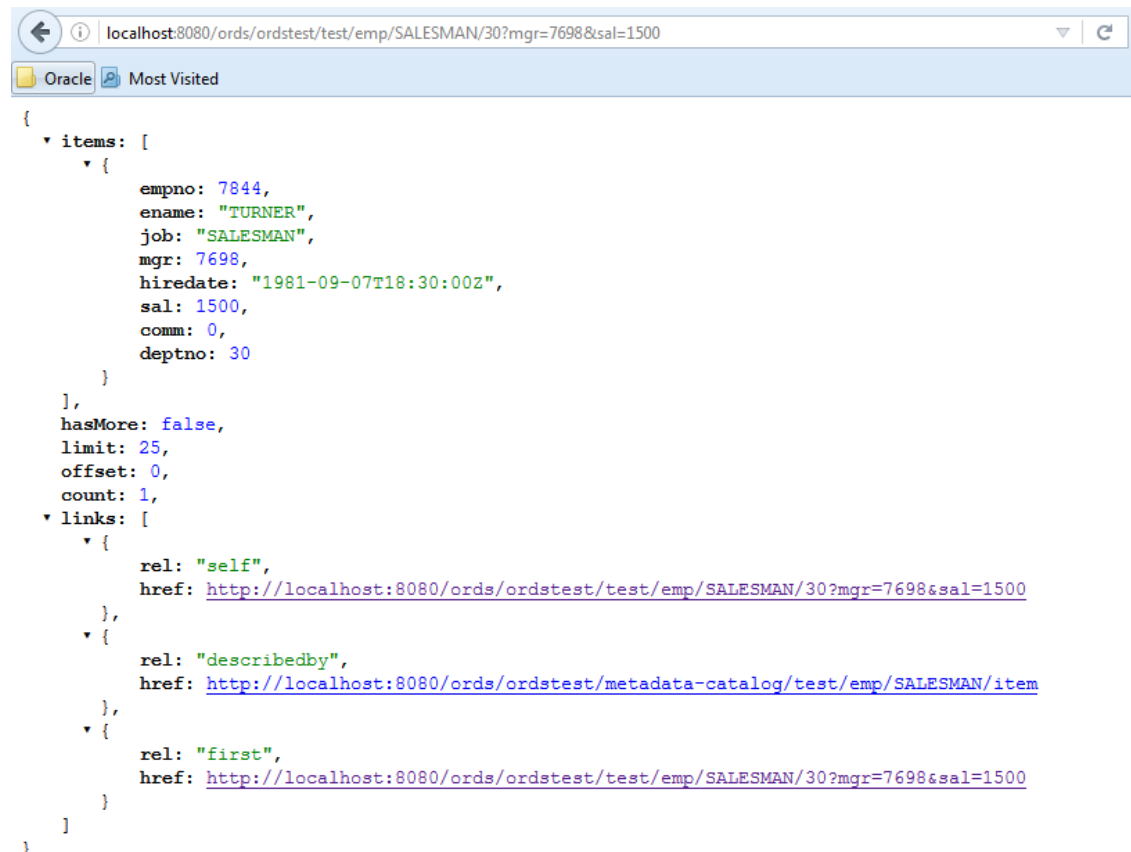
This section describes how to use query strings in the URI to specify parameters for REST requests like the GET method, which does not have a body. You can use query strings for any of the other optional bind variables in the query as you choose.

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 2-12 Using Browser to Show the Results of Using a Query String to Send a GET Method with Some Parameter Name/Value Pairs



```

{
  items: [
    {
      empno: 7844,
      ename: "TURNER",
      job: "SALESMAN",
      mgr: 7698,
      hiredate: "1981-09-07T18:30:00Z",
      sal: 1500,
      comm: 0,
      deptno: 30
    }
  ],
  hasMore: false,
  limit: 25,
  offset: 0,
  count: 1,
  links: [
    {
      rel: "self",
      href: http://localhost:8080/ords/ordstest/test/emp/SALESMAN/30?mgr=7698&sal=1500
    },
    {
      rel: "describedby",
      href: http://localhost:8080/ords/ordstest/metadata-catalog/test/emp/SALESMAN/item
    },
    {
      rel: "first",
      href: http://localhost:8080/ords/ordstest/test/emp/SALESMAN/30?mgr=7698&sal=1500
    }
  ]
}

```

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:

- [Lab 4 of the ORDS Oracle By Example \(OBE\)](#)
- [Database Application Development Virtual Image](#)

2.4.2 Using SQL/JSON Database Functions

This section describes how to use the SQL/JSON database functions available in Oracle Database 19c Release or later 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 Hierarchical Relational Data](#)

2.4.2.1 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 21c

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:**

- JSON in the Oracle Database Technology
- Ordinality Column

2.4.2.1.1 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.

- The `:body_text` bind variable is a CLOB datatype and you can assign it only to a CLOB variable.
- If you use either `:body` or `:body_text`, then you cannot reference individual JSON attributes through the ORDS `:bind` variables.

 **See Also:**

Database SQL Language Reference

2.4.2.1.2 Example of JSON Purchase Order with Nested Lineltems

This section shows an example that takes the JSON Purchase Order with Nested Lineltems and inserts it into a row of the `PurchaseOrder` table and rows of the `Lineltem` table.

Example 2-5 Nested JSON Purchase Order with Nested Lineltems

```

{"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}]]'

```

2.4.2.1.3 Table Definitions for PurchaseOrder and Lineltems 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));

```

2.4.2.1.4 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 2-6 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;

```

2.4.2.1.5 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.

2.4.2.1.6 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](#)
- [ORDS PL/SQL Package Reference](#)

2.4.2.2 Generating Nested JSON Objects from Hierarchical 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:

- SQL/JSON functions are available with Oracle Database. 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.

2.4.2.2.1 Bypassing JSON Generation for Relational Data

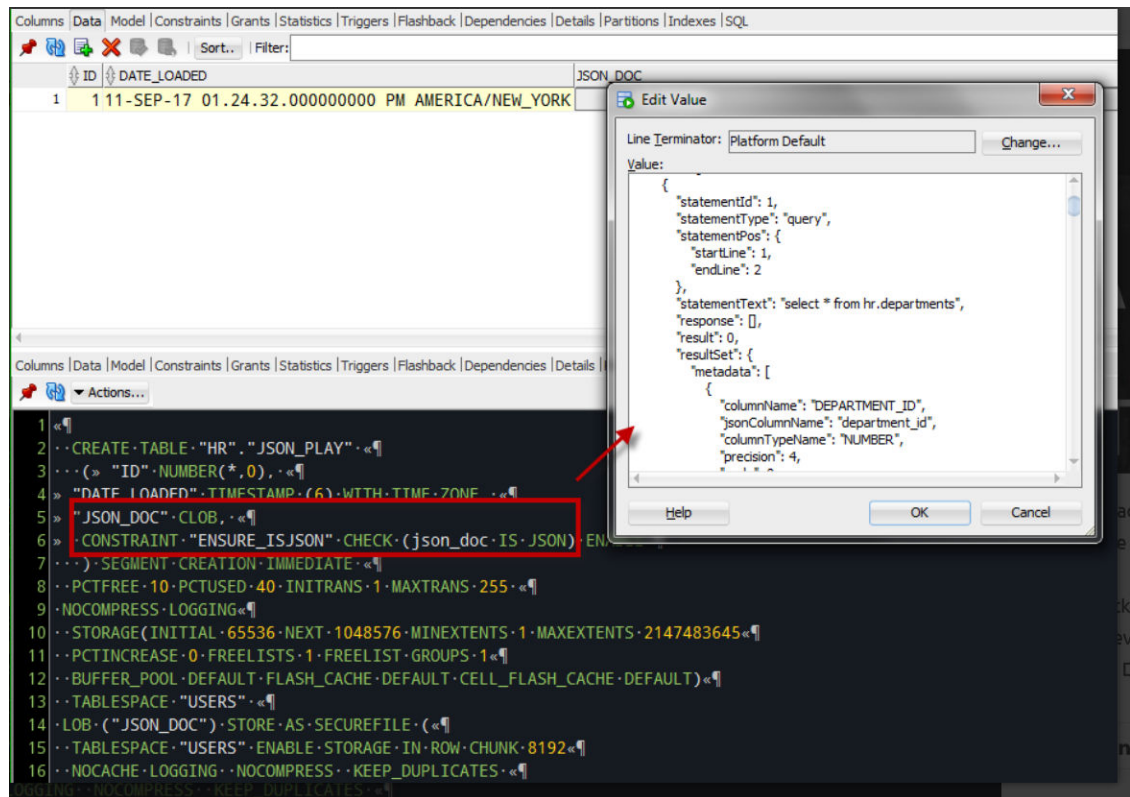
This section describes and provides solutions for handling responses that are already in a JSON format.

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 2-13 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.

2.4.2.2.2 Example to Generate Nested JSON Objects from the Hierarchical 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 2-7 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
```

2.4.2.2.3 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 2-8 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;

```

2.4.2.3 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](#)

2.4.2.3.1 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 PO Number 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

```

2.4.2.3.2 Generating JSON Object from the Database

This section shows the results of a GET method to fetch the JSON object from the database..

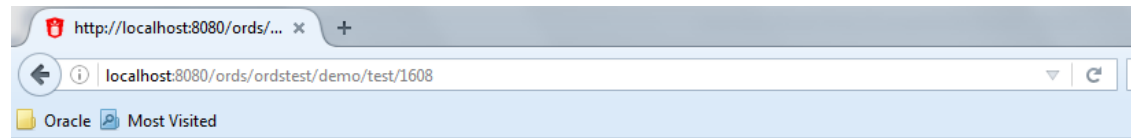
Method: GET

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 2-14 Generating Nested JSON Objects



The screenshot shows a web browser window with the address bar containing `localhost:8080/ords/ordstest/demo/test/1608`. The main content area displays a JSON object with the following structure:

```
{
  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: 20,
        UPPCode: 1313109289
      },
      Quantity: 9
    },
    {
      ItemNumber: 2,
      Part: {
        Description: "Lethal Weapon",
        UnitPrice: 20,
        UPPCode: 8539162892
      },
      Quantity: 5
    },
    {
      ItemNumber: 1,
      Part: {
        Description: "One Magic Christmas",
        UnitPrice: 20,
        UPPCode: 1313109289
      },
      Quantity: 9
    }
  ]
}
```

2.5 Manually Creating RESTful Services Using Javascript

This section describes how to manually create the RESTful Services using JavaScript that runs in Oracle Database Release 23ai or later.

2.5.1 Allowed JavaScript Structures

The defined JavaScript code must be inside an anonymous function that receives the following two parameters:

- ORDS request object
- ORDS response object

Example 2-9

```
(req, resp) => {}
```

ORDS provides utility properties and functions in those parameters that allow reading and manipulating the request and response.

Table 2-2 ORDS Request Object Properties

Property name	Description	Handler Code Example
uri	Specifies the path of the current handler.	<pre>(req, resp) => { const x = req.uri; }</pre>
body	Specifies the payload of the HTTP request. If the payload is a JSON structure, then the corresponding JavaScript object is created. Otherwise this is a string.	<pre>(req, resp) => { const x = req.body; }</pre>
content_type	Specifies the content-type of the request.	<pre>(req, resp) => { const x = req.content_type; }</pre>
query_parameters	Specifies a JavaScript object with the query parameter key/value pairs.	<pre>(req, resp) => { const x = req.query_parameters.myparameter; }</pre>
uri_parameters	Specifies a JavaScript object with the uri parameter key/value pairs if available.	<pre>(req, resp) => { const x = req.uri_parameters.myuri param; }</pre>
current_user	Specifies the ORDS authenticated user who is doing the call.	<pre>(req, resp) => { const x = req.current_user; }</pre>

Table 2-3 ORDS Response Object Functions

Function	Description	Example
append	This function appends a string into the body HTTP response body.	<pre>(req, resp) => { resp.append('Test'); }</pre>
end	Stops further processing of the manipulation of the HTTP response. Any other function called after end() function, does not alter the state of the the HTTP response.	<pre>(req, resp) => { resp.end(); }</pre>
send	Overrides anything appended before and writes the string provided into the HTTP response body and commits the response.	<pre>(req, resp) => { resp.send('Test'); }</pre>
json	Overrides anything appended before and writes the JSON representation of the provided JavaScript object into the HTTP response body and commits the response.	<pre>(req, resp) => { resp.json({test_key: 'Test'}); }</pre>
content_type	Sets the content-type of the HTTP response. Regardless of what is set, the content-type of the response also includes the charset=UTF-8.	<pre>(req, resp) => { resp.content_type('text/ plain'); }</pre>
status	Sets the HTTP status code with the provided integer value.	<pre>(req, resp) => { resp.status(200); }</pre>

If the response is not manipulated, then the default content-type of the request is application/json; charset=utf=8 and the default status code is 200.

2.5.2 Defining the REST Service and JavaScript Handler Using PL/SQL Function

This section shows how to define the REST Service with a JavaScript handler for a GET call using the Oracle REST Data Services PL/SQL function.

```
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 => 'demojs',
    p_base_path   => '/demojs/',
    p_items_per_page => 25,
    p_status      => 'PUBLISHED',
    p_comments    => NULL);
ORDS.DEFINE_TEMPLATE(
    p_module_name => 'demojs',
    p_pattern     => 'test/',
    p_priority    => 0,
    p_etag_type   => 'HASH',
    p_etag_query  => NULL,
    p_comments    => NULL);
ORDS.DEFINE_HANDLER(
    p_module_name => 'demojs',
    p_pattern     => 'test/',
    p_method      => 'GET',
    p_source_type => 'mle/javascript',
    p_items_per_page => 0,
    p_mimes_allowed => '',
    p_comments    => NULL,
    p_mle_env_name => NULL,
    p_source      =>
q'~
    (req, resp) => {
        resp.content_type('text/plain');
        resp.status(200);
        resp.send('success');
    }
~'
    );

    COMMIT;
END;
/

```

The preceding handler can be invoked using the following cURL command:

```
curl -i -X GET 'https://example.com/ords/ordstest/demojs/test/'
```

2.5.3 About Executing SQL in Javascript

This section describes how to execute SQL in JavaScript.

JavaScript handlers depend on Oracle Database Multilingual Engine (MLE), this gives the ability to interact with the Oracle Database.

The following example returns a fixed salary hike for the requested employee based on the original salary:

```
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     => 'demojssql',
    p_base_path       => '/demojssql/',
    p_items_per_page  => 25,
    p_status           => 'PUBLISHED',
    p_comments        => NULL);
  ORDS.DEFINE_TEMPLATE(
    p_module_name     => 'demojssql',
    p_pattern         => 'test/:emp_id',
    p_priority        => 0,
    p_etag_type       => 'HASH',
    p_etag_query      => NULL,
    p_comments        => NULL);
  ORDS.DEFINE_HANDLER(
    p_module_name     => 'demojssql',
    p_pattern         => 'test/:emp_id',
    p_method          => 'GET',
    p_source_type     => 'mle/javascript',
    p_items_per_page  => 0,
    p_mimes_allowed  => '',
    p_comments        => NULL,
    p_mle_env_name   => NULL,
    p_source          =>
q'~
  (req, resp) => {
    const query = 'select employee_id, first_name, salary from employees
where employee_id = :1';
    const options = { fetchInfo: { SALARY: { type:
oracledb.ORACLE_NUMBER } } };
    const res = session.execute(query, [req.uri_parameters.emp_id], options);

    const l_raise_calc = res.rows[0].SALARY.mul(new OracleNumber(0.15));

    resp.content_type('application/json');
    resp.json({raise: l_raise_calc.toNumber()});
  }
~'
  );
  COMMIT;
END;
/
```

The preceding handler can be invoked using the following cURL command, assuming that there is an entry for employee 151:

```
curl -i -X GET \  
'https://example.com/ords/ordstest/demojssql/test/151'
```

2.5.4 About Using the Fetch Function

This section describes how to use Oracle Database Multilingual Engine (MLE) fetch function.

JavaScript handlers also have the ability to do HTTP requests through the built in MLE fetch function. For this, Access Control List (ACL) rules must be defined in Oracle Database.



See Also:

[MLE Fetch API polyfill](#)

Example 2-10 ACL Rule in the Database

```
BEGIN  
  DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(  
    host => 'mydomain.com',  
    ace => xs$ace_type(  
      privilege_list => xs$name_list('http'),  
      principal_name => 'ORDSTEST',  
      principal_type => xs_acl.p_type_db  
    )  
  );  
END;  
/
```

You can create the following handler that gets information from the external service if the preceding rule is defined in the database :

```
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 => 'demojsfetch',  
    p_base_path   => '/demojsfetch/',  
    p_items_per_page => 25,  
    p_status       => 'PUBLISHED',  
    p_comments     => NULL);  
  ORDS.DEFINE_TEMPLATE(  
    p_module_name => 'demojsfetch',  
    p_pattern     => 'test/',  
    p_priority    => 0,
```

```

        p_etag_type      => 'HASH',
        p_etag_query     => NULL,
        p_comments      => NULL);
ORDS.DEFINE_HANDLER(
    p_module_name      => 'demojsfetch',
    p_pattern          => 'test/',
    p_method           => 'GET',
    p_source_type      => 'mle/javascript',
    p_items_per_page  => 0,
    p_mimes_allowed   => '',
    p_comments        => NULL,
    p_mle_env_name    => NULL,
    p_source           =>
q'~
    (req, resp) => {
        const fetch_response = await fetch(
            'http://mydomain.com/sample_service.json',
            {
                credentials: "include"
            }
        );
        if (! fetch_response.ok) {
            throw Error(`An error occurred: ${fetch_response.status}`);
        }
        const data = await fetch_response.json();
        resp.json(data);
    }
~'
    );
COMMIT;
END;
/

```

The preceding handler can be invoked using the following cURL command:

```
curl -i -X GET \ 'https://example.com/ords/ordstest/demojsfetch/test/'
```

2.5.5 Referencing MLE Environments

JavaScript handlers can take advantage of the MLE environment objects defined in the database.

Note:

Following are the restrictions when you are adding the MLE environments:

- The MLE environment must have simple unquoted names.
- The MLE environment must belong to the enabled schema.

**See Also:**[CREATE MLE ENV](#)

```

CREATE MLE MODULE IF NOT EXISTS po_module LANGUAGE JAVASCRIPT AS

export function addTwo(item) {
    return item+2;
}
/

CREATE OR REPLACE MLE ENV
    po_env
IMPORTS (
    'po_module' MODULE PO_MODULE
);
/

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      => 'demojsenv',
        p_base_path        => '/demojsenv/',
        p_items_per_page   => 25,
        p_status           => 'PUBLISHED',
        p_comments         => NULL);

    ORDS.DEFINE_TEMPLATE(
        p_module_name      => 'demojsenv',
        p_pattern          => 'test/',
        p_priority         => 0,
        p_etag_type        => 'HASH',
        p_etag_query       => NULL,
        p_comments         => NULL);

    ORDS.DEFINE_HANDLER(
        p_module_name      => 'demojsenv',
        p_pattern          => 'test/',
        p_method           => 'GET',
        p_source_type      => 'mle/javascript',
        p_items_per_page   => 0,
        p_mimes_allowed    => '',
        p_comments         => NULL,
        p_mle_env_name     => 'PO_ENV',
        p_source           =>
q'~
        (req, resp) => {
            const po = await import('po_module');
            resp.content_type('text/plain');
            resp.status(200);

```

```
resp.send(po.addTwo(6).toString());
}
~'
);

COMMIT;
END;
/
```

The preceding handler can be invoked using the following cURL command:

```
curl -i -X GET 'https://example.com/ords/ordstest/demojsenv/test/'
```

2.6 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](#)

2.6.1 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 Also:**

[Internet Date/Time Format](#)

2.6.2 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 set to UTC when running ORDS in standalone. This can be overridden by setting the environment variable `JVM_TIMEZONE` before running the `ords serve` command. 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 2-11 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 2-12 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 time zone in a Java Application server — Tomcat 8.

```
CATALINA_TIMEZONE="-Duser.timezone=America/New_York"  
CATALINA_OPTS="$CATALINA_OPTS $CATALINA_TIMEZONE"
```

2.6.3 Exploring the Sample RESTful Services in APEX (Tutorial)

Oracle highly recommends to develop Oracle REST Data Services application using SQL Developer Web because it supports the most recent Oracle REST Data Services releases, that is, 3.0.X. APEX 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 APEX 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 starts 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/` are 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 APEX is deployed
- `port` is the port the server is listening on
- `workspace` is the name of the Oracle APEX 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
- JSON Formatter

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 indices 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 Reference](#)

2.7 Creating RESTful Web Services Using Database Actions

You can create RESTful web services using the `Modules`, `Templates` and `Handlers` pages available in Database Actions.

See Also:

[Creating RESTful Web Services](#)

2.8 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 or can be provided as a scope in a valid JWT bearer token.

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](#)

2.8.1 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](#)

2.8.1.1 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.

2.8.1.2 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:**

The OAuth 2.0 Authorization Framework

2.8.1.2.1 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 of which resources each client is authorized to access.

Additionally, ORDS supports integration with Identity Providers that can issue JWT access tokens to the party calling the RESTful API for the purposes of accessing the RESTful API. A JWT Profile can be created for a REST-Enabled Schema to define how to validate JWT bearer tokens.

2.8.2 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](#)

2.8.3 About Users and Roles for Accessing Resources

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. The user responsible for deploying Oracle REST Data Services on an application server must 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.

2.8.4 About the File-Based User Repository

Oracle REST Data Services provides 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:

```
ords config user --help
```

Format:

```
ords config user add <name> <roles>
```

Example:

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

Arguments:

- `<user>` is the user ID of the user.
- `<roles>` is the list of roles that the user has. Use a comma to separate multiple roles in the list.

Related Topics

- [Tutorial: Protecting and Accessing Resources](#)

2.8.5 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](#)

2.8.5.1 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.

2.8.5.2 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/>

2.8.5.3 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":"1982-01-23T00:00:00Z","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"}
  ]
}
```

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

```
ords config user add 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.

- 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

- 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": "2YotnFZFEjrlzCsicMWpAA",
  "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":"1982-01-23T00:00:
      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"
      }
    ]
  }
}
```

- 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_orcls_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 `uniqueRandomValue` 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 `cliendId` 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=D5doeTSIDgxbWiWkPl9UpA..&type=bearer&expires_in=3600&state=uniqueRandom
Value
```

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":"1982-01-23T00:00:
    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"}  
  ]  
}
```

Related Topics

- [Using the Oracle REST Data Services PL/SQL API](#)



See Also:

[Managing OAuth Clients](#)

2.9 JWT Bearer Token Authentication and Authorization Using JWT Profile

ORDS release 23.3 introduces support for JSON Web Token (JWT). JWT bearer tokens enable the ORDS developers to delegate authentication and authorization to any OAuth2-compliant Identity Provider to issue a JWT access token that ORDS can validate to provide access to ORDS protected resources.

ORDS acts as a resource server in a typical OpenID connect or OAuth2 flow, making it convenient for the developers to access the ORDS APIs from their web applications.

You can create a JWT Profile for any REST-Enabled schema to provide ORDS with a mechanism to validate JWT bearer tokens. If a JWT bearer token is validated, then ORDS accepts the following:

- The JWT subject claim as the authenticated user making the request
- The JWT scope claims as the REST-Enabled schemas ORDS privileges that the user has consented to the application using the privileges on their behalf

Topics:

- [About JSON Web Tokens \(JWTs\)](#)
- [Prerequisites for JWT Authentication](#)
- [Creating an ORDS JWT Profile](#)
- [JWT Identity Provider Details](#)
- [Making Requests to ORDS Using a JWT Bearer Token](#)

2.9.1 About JSON Web Tokens (JWTs)

This section introduces you to the JSON Web Tokens.

A JSON Web Token (JWT) is a compact, URL-safe means of representing claims to be transferred between two parties. The claims in a JWT are encoded as a JSON object. ORDS supports the use of any OAuth2-compliant identity providers such as, OCI IAM with Identity Domains, Oracle Identity Cloud Service (IDCS), Auth0, and Okta. If a JWT is required to access a resource, ORDS validates the JWT using a corresponding public verification key provided by the authorization server.

A JWT comprises of the following:

- A header, that identifies the type of token and the cryptographic algorithm used to generate the signature.
 - The header is required to contain the following reserved claims.

 **Note:**

A claim is a key value pair, where the key is the name of the claim.

- * alg (algorithm)
 - * kid (key id)
- The header can optionally contain the following reserved claims that ORDS takes into account
 - * x5t (x.509 certificate thumbprint)
 - * typ (type)
- The header can also contain custom claims with user-defined names.
- A payload containing claims about the identity of the end user, and the properties of the JWT.
 - A payload is required to contain the following reserved names of the claims:
 - * sub (subject)
 - * aud (audience)
 - * iss (issuer)
 - * iat (issued at)
 - * exp (expiration time)
 - The payload can optionally contain the following reserved claims that ORDS takes into account
 - * scope or scp
 - * nbf (not before)
 - A payload can also contain custom claims with user-defined names
- A signature, to validate the authenticity of the JWT (derived by base64 encoding the header and the payload).
When using JWTs to control access to the target schema APIs or resources, the JWT Profile in the REST-Enabled schema specifies that the reserved claims in the payload of the JWT must have particular values before ORDS considers the JWT to be valid.
ORDS only accepts the following:
 - alg (algorithm) values of RS256, RS384 and RS512
 - kid (key id) value that can be matched to a corresponding public verification key
 - x5t (x.509 certificate thumbprint) if present to a corresponding public verification key
 - typ (type) if present, requires the value to be JWT
 - aud (audience) that matches the target schemas JWT Profile audience
 - iss (issuer) that matches the target schema JWT Profile issuer

- `iat` (issued at) identifies the time when the JWT was issued and is not be accepted before this time. This claim is used to determine the age of the JWT and enforce the JWT Profile allowed age if it is set.
- `exp` (expiration time) identifies the expiration time when or after which the JWT is not accepted for processing.
- `nbf` (not before) if present, identifies the time before which the JWT is not accepted for processing.

When a JWT is validated and the payload of JWT contains the scope claim, the ORDS privilege name protecting the resource is verified as being provided in the scope claim before processing.

2.9.2 Prerequisites for JWT Authentication

This section lists the prerequisites for JWT authentication.

Before ORDS can accept authentication and authorization using JWTs:

- An OAuth2-compliant identity provider (for example, OCI IAM with Identity Domains, Oracle Identity Cloud Service (IDCS), Auth0) must have already been set up to issue JWTs for users who are allowed to access the ORDS resources.
- If you want to use custom claims in authorization policies, the identity provider must be set up to add the custom claims to the JWTs that it issues.

See Also:

- [Managing Applications](#)
- [Oracle Identity Cloud Service](#)
- [Auth0, an identity platform to manage access to your applications.](#)

To validate a JWT using a corresponding public verification key provided by the issuing identity provider:

- the signing algorithm used to generate the signature of JWT must be one of `RS256`, `RS384`, or `RS512`
- the public verification key must have a minimum length of 2048 bits and must not exceed 4096 bits
- the public verification key must be specified using the JSON Web Key (JWK) format and ORDS can access it without authentication

The JWK URI

- The URI must be routable from the subnet containing ORDS
- Certain key parameters must be present in the JWKS to verify the signature of the JWT. See [Parameters for Verifying JWT Signatures](#).
- By default, the JWKS can be up to 10000 bytes in size

2.9.3 Creating an ORDS JWT Profile

This section explains how to create an ORDS JWT Profile.

A JWT Profile can be created within a REST-Enabled schema using the `OAUTH.CREATE_JWT_PROFILE` procedure. Alternatively, `OAUTH_ADMIN.CREATE_JWT_PROFILE` can be used to create a JWT Profile in other REST-Enabled schemas as long as the user has the `ORDS_ADMINISTRATOR` role.

 **Note:**

Only one JWT Profile can be defined per schema. To update an existing JWT Profile, the existing JWT Profile must be deleted before creating a new one.

Example:

```
BEGIN
OAUTH.CREATE_JWT_PROFILE (
  p_issuer => 'https://identity.oraclecloud.com/',
  p_audience => 'ords/myapplication/api' ,
  p_jwk_url =>'https://
idcs-10a10a10a10a10a10a10a10a10a10a10a10a.identity.oraclecloud.com/admin/v1/SigningCert/
jwk'
);
COMMIT;
END;
/
```

This JWT Profile specifies the issuer, audience, and the JWK URL.

Additionally, an allowed skew and age can be specified. The `p_issuer` must be a non null value and must match the `iss` claim in the JWT bearer token. The `p_audience` must be a non null value and must match with the `aud` claim in the JWT bearer token.

The `p_jwk_url` must be a non null value starting with `https://` and identify the public verification key provided by the authorization server in a JSON Web Key (JWK) format.

Once the JWT Profile has been created, requests made to the schema protected resources can be accessed by providing a valid JWT bearer token with the scope to access the protected resource.

 **Note:**

A JWT scope claim is a JSON string containing a space-separated list of scopes. A protected ORDS resource is protected with a named ORDS privilege. To access the protected ORDS resource, the JWT scope claim must contain a scope with the same name as the protecting ORDS privilege. The scope of an ORDS privilege are case sensitive.

 **See Also:**

[OAUTH PL/SQL Package Reference](#)

2.9.4 JWT Identity Provider Details

The identity provider that issued the JWT, determines the values that are allowed to specify for the issuer (*iss*), and the audience (*aud*) claims in the JWT. The identity provider that issued the JWT also determines the URI from where to retrieve the JSON Web Key Set (JWKS) to verify the signature of the JWT.

Identity Provider	Issuer (<i>iss</i>) claim	Audience (<i>aud</i>) Claim	Format of URI from which to Retrieve the JWKS
Okta	<code>https://<your-okta-tenant-name>.com</code>	Customer-specific. The audience configured for the Authorization Server in the Okta Developer Console.	<code>https://<your-okta-tenant-name>.com/oauth2/<auth-server-id>/v1/keys</code>
IDCS	<code>https://identity.oraclecloud.com/</code>	Customer-specific. Refer to "Validating Access Tokens" section in Oracle Identity Cloud Service documentation.	<code>https://<tenant-base-url>/admin/v1/SigningCert/jwk</code> To obtain the JWKS without logging in to Oracle Identity Cloud Service, refer to "Change Default Settings" in Oracle Identity Cloud Service documentation.
OCI IAM with Identity Domains	<code>https://identity.oraclecloud.com</code>	Customer-specific. See "Managing Applications" section in OCI IAM with Identity Domains documentation.	<code>https://<tenant-base-url>/admin/v1/SigningCert/jwk</code>
Auth0	<code>https://<your-account-name>.auth0.com/</code>	Customer-specific.	<code>https://<your-account-name>.auth0.com/.well-known/jwks.json</code>

See Also:

- [Validating Access Tokens](#) in Oracle Identity Cloud Service documentation.
- [Change Default Settings](#) in Oracle Identity Cloud Service documentation.
- [Managing Applications](#) in OCI IAM with Identity Domains documentation.

2.9.4.1 Parameters for Verifying JWT Signatures

This section lists the key parameters required to verify the JWT signatures.

To verify the signature on a JWT, ORDS requires that the key parameters are present in the JWKS returned from an URI.

Key Parameter	Notes
kid	The identifier of the key used to sign the JWT. The value must match the <code>kid</code> claim in the JWT header. For example, <code>master_key</code> .
kti	The type of the key used to sign the JWT. Note that RSA is currently the only supported key type.
n	The public key modulus.
e	The public key exponent.
alg	The signing algorithm (if present) must be set to one of RS256, RS384 or RS512.

2.9.4.2 JWT Scopes and ORDS Privileges

You must configure the identity provider that issued the JWT, so as to provide the scope that matches the desired ORDS privilege. If a resource is protected in ORDS using an ORDS privilege, then that privilege name must be defined as a scope. The scope is then available for the application to request on behalf of the user. The issued JWT must then provide that as a scope claim.

Typically, identity providers allow APIs, resources, or scopes to be defined for a particular audience. For example: ORDS REST-Enabled schema defined API. These APIs, resources, or scopes can then be made available to specific applications or clients. The application can then request access tokens on behalf of an authenticated user for that audience and scope.

More than one scope can be requested and provided in the JWT. The protected ORDS resource is accessible as long as one of the scopes matches the ORDS privilege protecting the resource.

2.9.4.3 JWT Subject

ORDS accepts the subject (`sub`) claim in a valid JWT bearer token as the unique identifier for the user who consented for the application to access their data.

The value of the subject claim in a valid JWT bearer token is bound to the `:current_user` implicit parameter and the `REMOTE_IDENT OWA CGI` environment variable.

2.9.5 Making Requests to ORDS Using a JWT Bearer Token

Once a JWT Profile has been created for a REST-Enabled schema, the protected ORDS resources in that schema can be accessed by providing a valid JWT bearer token with the request.

Request to an ORDS protected resource is made from a third party application on behalf of a user. The third party application has configured its authentication using an Identity Provider. The same Identity Provider can be configured to issue JWT access tokens for ORDS. After the third party application has acquired a JWT access token from the Identity Provider, it can include the JWT as a bearer token in requests to ORDS. Third party application can request suitable JWT access tokens with the required scope to access the ORDS resource.

```
curl -X GET http://localhost:8080/ords/myapplication/api/sales / --header
"Authorization: Bearer
  $JWT"
```

2.10 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*.
- `SQL Administrator` - This role is for the Database API and is required for the pdb lifecycle management operations.

Topics:

- [About Oracle APEX Users and Oracle REST Data Services Roles](#)
- [Controlling RESTful Service Access with Roles](#)

2.10.1 About Oracle APEX Users and Oracle REST Data Services Roles

By default, Oracle APEX users do not have any of the Oracle REST Data Services predefined user roles. This means that, by default, APEX users cannot:

- Invoke protected RESTful Services
- Register OAuth 2.0 applications
- Use Oracle SQL Developer to develop RESTful services.

This applies to all APEX users, including APEX developers and administrators. It is therefore important to remember to follow the steps below to add APEX users to the appropriate user groups, so that they can successfully perform the above actions.

Topics:

- [Granting APEX Users Oracle REST Data Services Roles](#)
- [Automatically Granting APEX Users Oracle REST Data Services Roles](#)

2.10.1.1 Granting APEX Users Oracle REST Data Services Roles

To give an APEX User any of the roles above, the user must be added to the equivalent APEX 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**.

2.10.1.2 Automatically Granting APEX Users Oracle REST Data Services Roles

Adding APEX 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 APEX 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.

2.10.2 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](#)

2.10.2.1 About Defining RESTful Service Roles

A RESTful Service **role** is an APEX 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**.

2.10.2.2 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.

2.11 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](#)

2.11.1 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](#)

2.11.1.1 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.

2.11.1.2 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).

2.12 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`](#)

2.12.1 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](#)

2.12.1.1 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 (`{}`).

2.12.1.2 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`.

2.12.1.3 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.

2.13 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.

2.13.1 Configuring ORDS to Integrate with WebLogic Server

This section explains how to configure ORDS to work with WebLogic server for authentication.

To configure ORDS to work with WebLogic server authentication, use the `--weblogic-auth` option as shown in the following command when you are generating the deployable `ords.war` file:

```
ords war --weblogic-auth <path for new war file>.
```

Specify the `--help` option to get help on the `ords war` command:

```
ords war --help.
```

Using the `--weblogic-auth` option with the `ords war` command, the `--weblogic-auth` option re-configures the `web.xml` deployment descriptor in the generated web application file that helps the WebLogic server to pass any established user identity to ORDS. After executing the preceding command, the generated web application 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](#)

2.14 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 all the operations. The available subprograms are included in the following PL/SQL packages:

- ORDS, documented in [ORDS PL/SQL Package Reference](#)
- OAUTH, documented in [OAUTH PL/SQL Package Reference](#)

To use the Oracle REST Data Services PL/SQL API:

Note:

You must be logged in as the user to the schema that you want to enable or to the ORDS services to be published when using the ORDS package. ORDS is granted `EXECUTE` privileges for public, which means any user can REST enable their schema and publish REST APIs. You may revoke this public grant if that is undesirable for your environments. If you want to work on another schema, then use the `ORDS_ADMIN` package, which requires the ORDS Administrator database role.

- 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 \(AutoREST\)](#)

2.14.1 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](#)

2.14.2 Testing the RESTful Service

To test the RESTful service that you created, start Oracle REST Data Services if it is not already started:

```
ords -c \path\to\ords\config serve
```

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.

2.15 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.

2.15.1 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:

```
examples\db_auth $ cd sql/
sql $ sql system/<password>

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.

2.15.2 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 assuming `ords/bin` is in `$PATH`, run the following command:

```
ords -c c:\ords\config config --db-pool default set jdbc.auth.enabled true
```

Output:

```
ORDS: Production Release 22.1 on Mon Mar 07 17:01:52 2022
```

```
Copyright (c) 2010, 2022, Oracle. All rights reserved.
```

```
Configuration:  
/C:/ords/config/
```

```
The setting named: jdbc.auth.enabled was set to: true in configuration:  
default
```

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 2-13 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.

2.15.3 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 APEX).

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. Navigate to the `<Current Configuration directory>/global` directory.
2. Open the `settings.xml` file, which stores the Oracle REST Data Services configuration information.
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

2.15.4 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..`

2.15.5 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 system/<password>
```

```
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
```

2.16 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](#)

2.16.1 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.

2.16.2 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).

2.16.3 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.

2.16.4 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.

2.16.5 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.

2.16.6 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.

2.16.7 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 interprets 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.

2.16.8 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.

2.16.9 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.

2.16.9.1 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 system/<password>
```

```
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.

2.16.9.1.1 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 `settings.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.

2.16.9.1.2 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.

2.16.9.1.3 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 `settings.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"}
```

2.16.9.2 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 system/<password>

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
```

2.17 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.

2.17.1 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;
```

2.17.1.1 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.N`, where N is the position of the column in the key. ORDS combines such values to form the relative path of the item URL.

Example:

```
SELECT
      ID1 "$.id.1",
      ID2 "$.id.2",
      ID3 "$.id.3",
      ...
```

Related Topics

- [Route Patterns Specification](#)

2.17.2 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/"
    }
]
```

2.17.2.1 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](#)

2.17.2.2 URL Resolution

This section describes how ORDS resolves column values using URI resolution algorithm.

Related Topics

- [rfc3986](#)

2.17.2.2.1 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"
}
```

2.17.2.2.2 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/"
}
```

2.17.2.2.3 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.

2.17.2.2.4 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.

2.17.2.2.5 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.

2.18 About HTTP Error Responses

ORDS can now generate HTTP error responses in JSON or HTML format. Prior to ORDS release 20.4, only HTML responses were supported. To preserve the backward compatibility, by default, ORDS attempts to automatically determines the best format to render the error responses.

You can configure `error.responseFormat` setting and force ORDS to always render the error responses in either HTML or JSON format.

2.18.1 About `error.responseFormat`

The `error.responseFormat` setting is a global setting that supports the following values:

- `html` - Force all error responses to be in HTML format.
- `json` - Force all error responses to be in JSON format.
- `auto` (default value) - Automatically determine most appropriate format for a request.

2.18.1.1 HTML Mode

When `error.responseFormat` value is set to `html`, all the error responses are rendered in HTML format. This setting can be used to match the behaviour of ORDS 20.3.1 and prior releases. The HTML format displays properly in web-browsers. However, for non-human clients, HTML format is verbose and challenging to parse.


2.18.1.2 json Mode

When `error.responseFormat` value is set to `json`, all the error responses are rendered in JSON format. The JSON format complies with the [Problem Details for HTTP APIs](#) standard. The JSON format is terse, and straightforward for non-human clients to parse. However, it does not display properly in browsers and is not user friendly for non-technical users.

2.18.1.3 auto Mode

The default value for `error.responseFormat` is `auto`. When this value is configured, ORDS applies the following rules and automatically chooses the most appropriate format to use:

- If the client supplies an `Accept` request header, where `application/json` or `application/problem+json` is the most preferred media type, then the response must be in JSON format.
- If the client supplies an `Accept` request header where `text/html` is the most preferred media type, then the response must be in HTML format.
- If the client supplies a `X-Requested-With` header, then the response must be in JSON format. Presence of this header indicates that the request is initiated from the JavaScript code and so JSON would be the appropriate response format.
- If the client supplies an `Origin` header, then the response must be in JSON format. Presence of this header indicates that the request is initiated from the JavaScript code and so JSON would be the appropriate response format.
 - There is one exception to this rule, if the request method is `POST` and the `Content-Type` of the request is `application/x-www-form-urlencoded`, then the response will be in HTML format.
- If the client supplies a `User-Agent` header whose value starts with `curl/`, then the response must be in JSON format. `cURL` is a popular command line tool for making the HTTP requests. The terser JSON format is more readable in a command line environment. If none of the preceding rules apply, then the response will be in HTML format.

 **See Also:**
[cURL](#)

3

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.

3.1 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 3-1 List of Implicit Parameters

Name	Type	Access Mode	HTTP Header	Description	Introduced
<code>:body</code>	BLOB	IN	N/A	Specifies the body of the request as a temporary BLOB.	2.0
<code>:body_text</code>	CLOB	IN	N/A	Specifies the body of the request as a temporary CLOB.	18.3
<code>:content_type</code>	VARCHAR	IN	Content-Type	Specifies the MIME type of the request body, as indicated by the Content-Type request header.	2.0

Table 3-1 (Cont.) List of Implicit Parameters

Name	Type	Access Mode	HTTP Header	Description	Introduced
: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
: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

Table 3-1 (Cont.) List of Implicit Parameters

Name	Type	Access Mode	HTTP Header	Description	Introduced
: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
: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

Table 3-1 (Cont.) List of Implicit Parameters

Name	Type	Access Mode	HTTP Header	Description	Introduced
: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
:status_code	NUMBER	OUT	X-ORDS-STATUS-CODE	Specifies the HTTP status code for the request.	18.3

3.1.1 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 3-1 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. If you need this value more than once, then assign it to a local variable, and dereference the local variable instead.

You can use either one of the implicit parameters `:body` or `:body_text`. Otherwise, the PL/SQL block displays an error message "Duplicate steam parameter".

If you use either `:body` or `:body_text`, then you cannot use `:bind` notation to read attributes of the JSON payload of the request.

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 tab1(content) values (l_content);
  insert into tab2(content) values (l_content);
end;
```

3.1.2 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.

 **Note:**

`:body_text` implicit parameter must only be dereferenced once inside the entire PL/SQL block. If you need this value more than once, assign it to a local variable, and dereference the local variable instead.

You can use either one of the implicit parameters `:body` or `:body_text`. Otherwise, the PL/SQL block displays an error message "Duplicate steam parameter".

It is recommended to use `:body_text` (a character representation) rather than `:body` (a binary representation) particularly where the PL/SQL block uses JSON functions to process the request body efficiently.

3.1.3 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.

3.1.4 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.

3.1.5 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.

3.1.6 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.

3.1.7 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 3-2 Pagination Implicit Parameters

Name	Description	Status
<code>:page_offset</code>	Specifies the zero based page offset in a pagination request.	Deprecated
<code>:page_size</code>	Specifies the maximum number of rows to be retrieved on a page.	Deprecated
<code>:row_offset</code>	Specifies the index of the first row to be displayed in a pagination request.	Not Recommended

Table 3-2 (Cont.) Pagination Implicit Parameters

Name	Description	Status
: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

3.1.7.1 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.

3.1.7.2 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.

3.1.7.3 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.

3.1.7.4 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.

3.1.7.5 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.

3.1.7.6 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.

3.1.7.7 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.

3.1.7.8 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;
```


4

ORDS PL/SQL Package Reference

The ORDS 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](#)

4.1 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');
```

4.2 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
- `source_type_mle_javascript`. Minimum Database Oracle Release version 23ai or later is required. Executes an anonymous `javascript` function that accepts a request and response parameter.

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;
/

```

4.3 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
- `source_type_mle_javascript`. Minimum Database Oracle Release version 23ai or later is required. Executes an anonymous `javascript` function that accepts a request and response parameter.

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;
/

```

4.4 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;
/
```

4.5 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`, `RESULTSET`.

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;
/

```

4.6 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;
/

```

4.7 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

- `source_type_mle_javascript`. Minimum Database Oracle Release version 23ai or later is required. Executes an anonymous `javascript` function that accepts a request and response parameter.

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;
/

```

4.8 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;
/
```

4.9 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');
```

4.10 ORDS.DELETE_PRIVILEGE

Format

```
ORDS.DELETE_PRIVILEGE(  
    p_name IN sec_privileges.name%type);
```

Description

DELETE_PRIVILEGE deletes a privilege.

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');
```

4.11 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');
```

4.12 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](#)

4.13 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 function, materialized view, package, procedure, 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. Valid values: FUNCTION, MVIEW, PACKAGE, PROCEDURE, 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;
/
```

4.14 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 current user `CATEGORIES` table.

```
BEGIN
  ORDS.DROP_REST_FOR_OBJECT (
    p_object=>'CATEGORIES'
  );
END;
/
```

4.15 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;
```

4.16 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');
```

4.17 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;
/
```

4.18 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;
/
```

4.19 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;
/
```

4.20 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;
/
```

4.21 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;  
/
```

4.22 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;  
/
```

4.23 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;
/
```

4.24 ORDS.SET_PROPERTY

Format

```
ORDS.SET_PROPERTY(
    p_key          IN ords_prop_facts.key%type,
    p_value        IN ords_prop_values.value%type);
```

Description

`SET_PROPERTY` sets the value of the SCHEMA scoped property for the current enabled schema. The value must not be `NULL`.

Parameters

p_key

The property key.

p_value

The new property value.

Examples

The following example sets a property value:

```
BEGIN
    ORDS.SET_PROPERTY(
        p_key => 'a.key',
        p_value => 'a value');
END;
/
```

4.25 ORDS.UNSET_PROPERTY

Format

```
ORDS.UNSET_PROPERTY(
    p_key IN ords_prop_facts.key%type);
```

Description

`UNSET_PROPERTY` unsets the value of the SCHEMA scoped property for the current enabled schema.

Parameters

p_key

The property key.

Examples

The following example unsets a property value:

```
BEGIN
  ORDS.UNSET_PROPERTY (
    p_key => 'a.key');
END;
/
```

5

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_ADMINISTRATOR_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

- [ORDS PL/SQL Package Reference](#)

5.1 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;
/
```

5.2 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;
/

```

5.3 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;
/
```

5.4 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;
/

```

5.5 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;
/
```

5.6 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;
/
```

5.7 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

The 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;
/
```

5.8 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;
/
```


5.9 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;
/
```

5.10 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;
/

```

5.11 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;
/

```

5.12 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 function, materialized view, package, procedure, 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. Valid values: FUNCTION, MVIEW, PACKAGE, PROCEDURE, 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;
/
```

5.13 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;  
/
```

5.14 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;
/
```

5.15 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;
/
```

5.16 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;
/
```

5.17 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;
/
```

5.18 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;
/
```

5.19 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;
/

```

5.20 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;
/

```

5.21 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;
```

5.22 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;
```

5.23 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;
```

5.24 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;  
/
```

5.25 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;  
/
```

5.26 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;
```

```
);  
END;  
/
```

5.27 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;  
/
```

5.28 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;
/
```

5.29 ORDS_ADMIN.CONFIG_PLSQL_GATEWAY

Format

```
ORDS_ADMIN.CONFIG_PLSQL_GATEWAY(
  p_runtime_user      IN varchar2 DEFAULT NULL,
  p_plsql_gateway_user IN varchar2,
  p_comments          IN varchar2 DEFAULT NULL);
```

Description

Configures the database proxy user that must be used for PL/SQL Gateway calls serviced by the specified runtime user.

Parameters

p_runtime_user

Name of the runtime user to be configured.

p_plsql_gateway_user

Name of the proxy user.

p_comments

Comment text.

Usage Notes

When `p_runtime_user` is NULL, then the value provided through `ORDS_ADMIN.SET_SESSION_DEFAULTS` is used. Otherwise, `ORDS_PUBLIC_USER` is used. When

`p_plsql_gateway_user` is NULL, then the PL/SQL Gateway for the runtime user is unconfigured.

Examples

The following example configures the PL/SQL Gateway for `ORDS_PUBLIC_USER` runtime user:

```
BEGIN
  ords_admin.config_plsql_gateway(
    p_runtime_user => 'ORDS_PUBLIC_USER',
    p_plsql_gateway_user => 'GATEWAY_USER'
  );
END;
/
```

The following example unconfigures the PL/SQL Gateway for `ORDS_PUBLIC_USER` runtime user:

```
BEGIN
  ords_admin.config_plsql_gateway(
    p_runtime_user => 'ORDS_PUBLIC_USER',
    p_plsql_gateway_user => NULL
  );
END;
/
```

5.30 ORDS_ADMIN.SET_PROPERTY

Format

```
ORDS_ADMIN.SET_PROPERTY (
  p_schema          IN ords_schemas.parsing_schema%type,
  p_key             IN ords_prop_facts.key%type,
  p_value           IN ords_prop_values.value%type);
```

Description

`SET_PROPERTY` sets the value of the SCHEMA scoped property for the specified enabled schema. The value must not be NULL.

Parameters

p_schema

The name of the owning enabled schema. This parameter is mandatory.

p_key

The property key.

p_value

The new property value.

Usage Notes

(None.)

Examples

The following example sets a property value:

```
BEGIN
  ORDS_ADMIN.SET_PROPERTY(
    p_schema => 'tickets',
    p_key => 'a.key',
    p_value => 'a value'
  );
END;
/
```

5.31 ORDS_ADMIN.SET_PROPERTY

Format

```
ORDS_ADMIN.SET_PROPERTY (
  p_key          IN ords_prop_facts.key%type,
  p_value        IN ords_prop_values.value%type);
```

Description

`SET_PROPERTY` sets the value of the non-SCHEMA scoped property. The value must not be NULL.

Parameters

p_key

The property key.

p_value

The new property value.

Usage Notes

(None.)

Examples

The following example sets a property value:

```
BEGIN
  ORDS_ADMIN.SET_PROPERTY(
    p_key => 'a.key',
    p_value => 'a value'
  );
END;
/
```


5.32 ORDS_ADMIN.UNSET_PROPERTY

Format

```
ORDS_ADMIN.UNSET_PROPERTY(  
    p_schema          IN ords_schemas.parsing_schema%type,  
    p_key             IN ords_prop_facts.key%type);
```

Description

UNSET_PROPERTY unsets the value of the SCHEMA scoped property for the specified enabled schema.

Parameters

p_schema

The name of the owning enabled schema. This parameter is mandatory.

p_key

The property key.

Usage Notes

(None.)

Examples

The following example unsets a property value:

```
BEGIN  
    ORDS_ADMIN.UNSET_PROPERTY(  
        p_schema => 'tickets',  
        p_key => 'a.key'  
    );  
END;  
/
```

6

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](#)

6.1 OAUTH.CREATE_CLIENT

Format

```
OAUTH.CREATE_CLIENT(  
  p_name           IN VARCHAR2,  
  p_grant_type     IN VARCHAR2,  
  p_owner         IN VARCHAR2 DEFAULT NULL,  
  p_description    IN VARCHAR2 DEFAULT NULL,  
  p_origins_allowed IN VARCHAR2 DEFAULT NULL,  
  p_redirect_uri   IN VARCHAR2 DEFAULT NULL,  
  p_support_email  IN VARCHAR2 DEFAULT NULL,  
  p_support_uri    IN VARCHAR2 DEFAULT NULL,  
  p_privilege_names IN VARCHAR2  
  p_token_duration IN NUMBER,  
  p_refresh_duration IN NUMBER,  
  p_code_duration  IN NUMBER)
```

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_origins_allowed

A comma-separated list of URL prefixes. If the list is empty, then 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.

p_token_duration

Duration of the access token in seconds. `NULL` duration fallback to the value in the ORDS instance. By default, it can be set through a property or set to 3600 seconds.

p_refresh_duration

Duration of refresh token in seconds. `NULL` duration fallback to the value in the ORDS instance. By default, it can be set through a property or set to 86400 seconds.

p_code_duration

Duration of the code token in seconds applicable only when `grant_type` value is `authorization_code`. If the value is set to `NULL` or the `grant_type` value is not `authorization_code`, then the lifetime is the one defined in the ORDS instance. By default, the value is 300.

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',
    NULL,
    NULL,
    NULL
  );
  COMMIT;
END;
/
```

6.2 OAUTH.DELETE_CLIENT

Format

```
OAUTH.DELETE_CLIENT(  
  p_name IN VARCHAR2);
```

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;  
/
```

6.3 OAUTH.GRANT_CLIENT_ROLE

Format

```
OAUTH.GRANT_CLIENT_ROLE(  
  p_client_name IN VARCHAR2,  
  p_role_name   IN VARCHAR2);
```

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;
/
```

6.4 OAUTH.RENAME_CLIENT

Format

```
OAUTH.RENAME_CLIENT(
  p_name      IN VARCHAR2,
  p_new_name  IN VARCHAR2);
```

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;
/
```

6.5 OAUTH.REVOKE_CLIENT_ROLE

Format

```
OAUTH.REVOKE_CLIENT_ROLE(  
  p_client_name  IN VARCHAR2,  
  p_role_name    IN VARCHAR2);
```

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;  
/
```

6.6 OAUTH.UPDATE_CLIENT

Format

```
OAUTH.UPDATE_CLIENT(  
  p_name                IN VARCHAR2,  
  p_description          IN VARCHAR2,  
  p_origins_allowed     IN VARCHAR2,  
  p_redirect_uri        IN VARCHAR2,  
  p_support_email       IN VARCHAR2,  
  p_support_uri         IN VARCHAR2,  
  p_privilege_names     IN t_orcs_vchar_tab DEFAULT NULL,  
  p_token_duration      IN NUMBER,  
  p_refresh_duration    IN NUMBER,  
  p_code_duration       IN NUMBER  
);
```

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_origins_allowed

A comma-separated list of URL prefixes. If the list is empty, then any existing origins are removed.

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.

p_token_duration

Duration of the access token in seconds. `NULL` duration fallback to the value in the ORDS instance. By default, it can be set through a property or set to 3600 seconds.

p_refresh_duration

Duration of refresh token in seconds. `NULL` duration fallback to the value in the ORDS instance. By default, it can be set through a property or set to 86400 seconds.

p_code_duration

Duration of the code token in seconds applicable only when `grant_type` is `authorization_code`. If the value is set to `NULL` or the `grant_type` is not `authorization_code`, then the lifetime is the one defined in the ORDS instance. By default, the value is 300.

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 Update 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,
    p_token_duration => null,
    p_refresh_duration => null,
    p_code_duration => null);
  COMMIT;
END;
/

```

Example 6-1 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_description => 'Description altered.',
    p_origins_allowed => NULL,
    p_redirect_uri => '/abc/efg/',
    p_privilege_names => my_privs,
    p_token_duration => NULL,
    p_refresh_duration => NULL,
    p_code_duration => NULL);
  commit;
end;

```

Related Topics

- [OAUTH.RENAME_CLIENT](#)

6.7 OAUTH.ROTATE_CLIENT_SECRET

Format

```

OAUTH.ROTATE_CLIENT_SECRET(
  p_client_id      IN NUMBER,
  p_editing_user   IN VARCHAR2,
  p_revoke_sessions IN BOOLEAN DEFAULT TRUE);

```

Description

ROTATE_CLIENT_SECRET regenerates a new client secret and deletes all existing client sessions by default.

Parameters**p_client_id**

The ID of the client modified.

p_editing_user

The user requesting this change.

p_revoke_sessions

Controls if the approval for the existing client sessions must be revoked. Default value is `TRUE`.

Example

The following example rotates a client secret:

```
BEGIN
  OAUTH.ROTATE_CLIENT_SECRET(
    p_client_id => 1234567890,
    p_editing_user => 'USERA',
    p_revoke_sessions => TRUE
  );
END;
/
```

6.8 OAUTH.UPDATE_CLIENT_SECRET

Format

```
OAUTH.UPDATE_CLIENT_SECRET(
  p_client_name IN VARCHAR2,
  p_editing_user IN VARCHAR2,
  p_client_secret IN VARCHAR2);
```

Description

`UPDATE_CLIENT_SECRET` sets a new value for the secret of the client. By default, it deletes all the existing client sessions.

Parameters**p_client_name**

The name of the client in the current schema.

p_editing_user

The user requesting this change.

p_client_secret

The value of the new secret for the client.

Usage Notes

For the operation to take effect, use the `COMMIT` statement after calling this procedure.

Example

The following example updates the secret of a particular client:

```

BEGIN
  OAUTH.UPDATE_CLIENT_SECRET(
    p_client_name => 'CLIENT_TEST',
    p_editing_user => 'USERA ',
    p_client_secret => 'RaFhM690PA6cN1ffpkNx3Q..');
END;
/

```

6.9 OAUTH.IMPORT_CLIENT

Format

```

OAUTH.IMPORT_CLIENT(
  p_name           IN VARCHAR2,
  p_client_id      IN VARCHAR2,
  p_client_secret  IN VARCHAR2 DEFAULT NULL,
  p_grant_type     IN VARCHAR2,
  p_owner          IN VARCHAR2 DEFAULT NULL,
  p_description    IN VARCHAR2 DEFAULT NULL,
  p_origins_allowed IN VARCHAR2 DEFAULT NULL,
  p_redirect_uri   IN VARCHAR2 DEFAULT NULL,
  p_support_email  IN VARCHAR2 DEFAULT NULL,
  p_support_uri    IN VARCHAR2 DEFAULT NULL,
  p_privilege_names IN VARCHAR2,
  p_token_duration IN NUMBER DEFAULT NULL,
  p_refresh_duration IN NUMBER DEFAULT NULL,
  p_code_duration  IN NUMBER DEFAULT NULL);

```

Description

Imports an existing client into this schema, preserving the identifier and optionally a secret. If the secret is not provided, then a new one is generated.

Parameters

p_name

Name for the client displayed to the end user during the approval phase of three-legged OAuth. The name must be unique.

p_client_id

A unique client identifier.

p_client_secret

Optional parameter. If not provided, then a random secret is generated.

p_grant_type

The value 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. Can be null if p_grant_type value is client_credentials. Otherwise, it must not be null.

p_origins_allowed

A comma-separated list of URL prefixes.

p_redirect_uri

Client-controlled URI with a redirect containing an OAuth access token or error is sent. Can be a null if the value of p_grant_type is client_credentials. Otherwise, it must not be null.

p_support_email

The email where the end users can contact the client for support.

p_support_uri

The URI where the end users can contact the client for support.

Example URI: `http://www.myclientdomain.com/support/`

p_privilege_names

List of comma-separated privileges that the client wants to access.

p_token_duration

Duration of the access token in seconds. NULL duration fallback to the value in the ORDS instance. By default, it can be set through a property or set to 3600 seconds.

p_refresh_duration

Duration of refresh token in seconds. NULL duration fallback to the value in the ORDS instance. By default, it can be set through a property or set to 86400 seconds.

p_code_duration

Duration of the code token in seconds is applicable only when grant_type value is authorization code. If the value is set to NULL or if the value of grant_type is not authorization_code, then the lifetime is the one defined in the ORDS instance. By default, the value is 300.

Usage Notes

For this operation to take effect, use the COMMIT statement after calling this procedure.

Example

The following example, imports an OAuth client without custom durations or origins:

```
BEGIN
  OAUTH.IMPORT_CLIENT (
    p_name           => 'CLIENT_TEST',
    p_client_id      => 'awVMtPlqullIqPXhAwh4zA..',
    p_grant_type     => 'authorization_code',
    p_owner          => 'RESTEASY',
    p_description    => 'This is a test description.',
    p_origins_allowed => NULL,
    p_redirect_uri   => 'https://example.org/my_redirect/',
    p_support_email  => 'test@example.org',
    p_support_uri    => 'https://example.org/help/')
```

```
        p_privilege_names      => 'MyPrivilege');

    COMMIT;
END;
/
```

6.10 OAUTH.CREATE_JWT_PROFILE

Format

```
OAUTH.CREATE_JWT_PROFILE (
    p_issuer      IN VARCHAR2,
    p_audience   IN VARCHAR2,
    p_jwk_url     IN VARCHAR2,
    p_description IN VARCHAR2 DEFAULT NULL,
    p_allowed_skew IN NUMBER DEFAULT NULL,
    p_allowed_age IN NUMBER DEFAULT NULL
)
```

Description

Creates a new JWT Profile for the schema if it does not already exist. If a JWT Profile already exists, then it must be deleted first.

Parameters

p_issuer

The issuer of acceptable JWT access tokens. This value must match the `iss` claim provided in the JWT.

p_audience

The audience of acceptable JWT access tokens. This value must match the `aud` claim provided in the JWT.

p_jwk_url

This is the url to the jwk(s) used to validate acceptable JWT access tokens. It must start with "https://"

p_description

A description of the JWT Profile. This value can be null.

p_allowed_skew

The number of seconds allowed to skew time claims provided in the JWT. This can help mediate issues with differences in the clock used by ORDS and the token issuer. The default value of null, specifies that the ORDS global setting `security.jwt.allowed.skew` is taken. A value less than or equal to 0 means, it is disabled. A max of 60 seconds can be specified.

p_allowed_age

The maximum allowed age of a JWT in seconds, regardless of expired claim. The age of the JWT is taken from the JWT issued at claim. The default value of null means the ORDS global setting of `security.jwt.allowed.age` is taken. A value less than or equals to 0 means, it is disabled.

Usage Notes

For this operation to take effect, use the `COMMIT` statement after calling this procedure.

Example

The following example, deletes any existing JWT Profile for the schema and creates a new JWT Profile for the schema. Any requests made to the resources in this schema can use a JWT bearer token for authorization. The JWT token must be signed and its signature must be verifiable using a public key provided by `p_jwk_url`. The JWTs issuer and audience claims must also match the `p_issuer` and `p_audience` values. The JWT must provide a scope that matches the ORDS Privilege protected by the resource.

```
BEGIN
  OAUTH.DELETE_JWT_PROFILE();
  OAUTH.CREATE_JWT_PROFILE(
    p_issuer => 'https://identity.oraclecloud.com/',
    p_audience => 'ords/myapplication/api' ,
    p_jwk_url => 'https://
idcs-10a10a10a10a10a10a10a10a10a10a10a.identity.oraclecloud.com/admin/v1/SigningCert/
jwk'
  );
  COMMIT;
END;
/
```

6.11 OAUTH.DELETE_JWT_PROFILE

Format

```
OAUTH.DELETE_JWT_PROFILE ()
```

Description

Deletes the JWT Profile for the schema if one exists.

Usage Notes

For this operation to take effect, use the `COMMIT` statement after calling this procedure.

Example

The following example, deletes any existing JWT Profile for the schema:

```
BEGIN
  OAUTH.DELETE_JWT_PROFILE();
  COMMIT;
END;
/
```

JWT bearer tokens are not be accepted when authorizing requests to the protected resources.

7

OAUTH_ADMIN PL/SQL Package Reference

The `OAUTH_ADMIN` PL/SQL package contains subprograms (procedures and functions) for implementing OAuth authentication using Oracle REST Data Services for a privileged user.

Before a database user can invoke the `OAUTH_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_ADMINISTRATOR_ROLE TO ADMIN;
```

The `OAUTH_ADMIN` package is defined with the `AUTHID CURRENT_USER` right and each method requires a `p_schema` parameter where the target schema must be specified.



See also:

[Using the Oracle REST Data Services PL/SQL API](#)

7.1 OAUTH_ADMIN.CREATE_JWT_PROFILE

Format

```
OAUTH_ADMIN.CREATE_JWT_PROFILE (  
    p_schema      IN VARCHAR2,  
    p_issuer      IN VARCHAR2,  
    p_audience   IN VARCHAR2,  
    p_jwk_url     IN VARCHAR2,  
    p_description IN VARCHAR2 DEFAULT NULL,  
    p_allowed_skew IN NUMBER DEFAULT NULL,  
    p_allowed_age IN NUMBER DEFAULT NULL  
)
```

Description

Creates a new JWT Profile for the specified schema, if one does not already exist. If a JWT Profile already exists, it must be deleted first.

Parameters

p_schema

Name of the schema. This parameter is mandatory.

p_issuer

The issuer of acceptable JWT access tokens. This value must match the `iss` claim provided in the JWT.

7.2 OAUTH_ADMIN.DELETE_JWT_PROFILE

Format

```
OAUTH_ADMIN.DELETE_JWT_PROFILE ( p_schema          IN VARCHAR2) ;
```

Description

Deletes the JWT Profile for the specified schema, if it exists.

Parameters

p_schema

Name of the schema. This parameter is mandatory.

Usage Notes

For this operation to take effect, use the `COMMIT` statement after calling this procedure.

Example

The following example, deletes any existing JWT Profile for the schema `HR`:

```
BEGIN
  OAUTH_ADMIN.DELETE_JWT_PROFILE(p_schema=>'HR');
  COMMIT;
END;
/
```

JWT bearer tokens are not accepted while authorizing requests to the protected resources in the `HR` schema.

8

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.

8.1 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:

```
ords config set 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:

```
ords config set 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](#)

8.2 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

8.2.1 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.

```
ords config set db.cdb.adminUser "C##DBAPI_CDB_ADMIN as SYSDBA"
ords config secret db.cdb.adminUser.password
```

The ORDS role, SQL Administrator must be used to access the `https://<server>/ords/_/db-api/stable/database/pdbs/ services`.

8.2.2 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:

```
ords config set database.api.management.services.disabled true
```

This produces a response, `HTTP 503 Service Unavailable` with an explanatory reason.

8.3 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 snippet:

```
ords config set db.adminUser C##_DBAPI_DEFAULT_ADMIN  
ords config secret db.adminUser.password
```

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.

8.4 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:

```
ords config set database.api.management.services.disabled true
```

8.5 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 8-1 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.

Table 8-1 (Cont.) Open Service Broker Service Catalog

Database Type	Service	Plans	Prerequisites
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:

```
ords config --db-pool <pool-name> set 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.

9

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 a 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 Examples](#)
- [REST-Enabled SQL Request and Response Specifications](#)
- [REST-Enabled SQL Request and Response Specifications](#)
- [REST-Enabled SQL Service and MySQL Database](#)

9.1 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.

9.2 Configuring the REST-Enabled SQL Service

By default, the REST- Enabled SQL service is turned off. To configure the REST- Enabled SQL service settings, see Configuring REST Enabled SQL Service Settings.

9.3 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 9-1 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",
          "columnName":"DATE",
          "precision":0,
          "scale":0,
          "isNullable":1
        }
      ],
      "items":[
        {
          "sysdate":"2017-07-21T08:06:44Z"
        }
      ],
      "hasMore":false,
      "limit":1500,
      "offset":0,
      "count":1
    }
  ]
}

```

9.4 Getting Started with the REST-Enabled SQL Service

The REST- Enabled SQL service is provided only through HTTPS POST method.

Note:

In ORDS, a RESTful service is stateless. In a stateless environment, each HTTPS request from a client maps to a new database session. Therefore, a session begins and ends with every SQL statement or script execution, that is, the worksheet is auto-committed.

As the session state is not maintained, session attributes do not persist and commands such as ROLLBACK and COMMIT do not apply. If a SQL statement or script executes successfully, an implicit commit is performed. If it executes with an error, an implicit rollback is performed. Therefore, when you need, include the ROLLBACK and COMMIT commands or session attributes in the PL/SQL code block that is sent to the database for a session.

Topics:

- [REST-Enabling the Oracle Database Schema](#)
- [REST-Enabled SQL Authentication](#)
- [REST-Enabled SQL Endpoint](#)

9.4.1 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](#)

9.4.2 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.

9.4.3 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

9.5 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](#)

9.5.1 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"

9.5.1.1 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",
            "columnName":"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.

9.5.1.2 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",
                    "columnName":"NUMBER",
                    "precision":0,
                    "scale":-127,
                    "isNullable":1
                }
            ],
            "items":[
                {
                    "10":10
                }
            ],
            "hasMore":false,
            "limit":1500,
            "offset":0,
            "count":1
        }
    ]
}

```

9.5.1.3 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",
                "columnName":"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",
                "columnName":"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
}
]
}

```

9.5.2 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](#)

9.5.2.1 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')",
            "columnName":"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
      }
    }
  ]
}
```

```

    }
  ]
}

```

9.5.2.2 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 `defaults.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",
          "columnName":"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
    }
  }
]
}

```

9.5.2.3 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",
            "columnName":"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
}
}
]
}

```

9.5.2.4 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 9-2 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 9-3 Complex Bind in POST Request**File**complex_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,

```

```

    ]
  }
]
}

```

9.5.2.5 Specifying Batch Statements in a POST Request

This section shows the examples with batch statements and batch bind values in a POST request.

Example 9-4 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 9-5 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"
          ]
        }
      ]
    }
  ]
}
```

```

    ]
  }
]
}

```

9.5.3 Example POST Request with DATE and TIMESTAMP Format

Example 9-6 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",
            "columnName":"DATE",
            "precision":0,
            "scale":0,
            "isNullable":1
          },
          {
            "columnName":"SUMMER",
            "jsonColumnName":"summer",
            "columnName":"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
      }
    }
  ]
}

```

9.5.4 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"
    }
}
]
```

9.6 REST-Enabled SQL Request and Response Specifications

The following sections provide REST-Enabled SQL request and response specifications:

- [Request Specification](#)
- [Response Specification](#)

9.6.1 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
\$.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 coll=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 defaults.xml, is ignored.	500	misc.pagination.maxRows	Between 0 to misc.pagination.maxRows.

JSONPath	Type	Description	Example	Default Value	Possible Values
\$.binds	Array	Specifies an array of objects specifying the bind information.	<pre>"binds": [{ "name": "mybind1", "data_type": "NUMBER", "mode": "out" }, { "name": "mybind2", "data_type": "NUMBER", "value": 7 }]</pre>	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
\$.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: <ul style="list-style-type: none"> • 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]

JSONPath	Type	Description	Example	Default Value	Possible Values
\$.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
\$.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

9.6.2 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

JSONPath	Data type	Description	Example Values	Possible values
\$.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"]
\$.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 coll=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

JSONPath	Data type	Description	Example Values	Possible values
<code>\$.items[*].resultSet</code>	Object	Specifies information about the result set generated from a query.	Not applicable	Not applicable
<code>\$.items[*].resultSet.metadata</code>	Array	Specifies each object in the array provides information about the metadata of a column.	Not applicable	Not applicable
<code>\$.items[*].resultSet.metadata[*].columnName</code>	String	Specifies the name of the column used in the Oracle Database.	Not applicable	Not applicable
<code>\$.items[*].resultSet.metadata[*].jsonColumnName</code>	String	Specifies the name of the column used in <code>\$.items[*].resultSet.items[*].<columnname></code>	Not applicable	Not applicable
<code>\$.items[*].resultSet.metadata[*].columnTypeName</code>	String	Specifies the Oracle Database data type of the column.	Not applicable	Not applicable
<code>\$.items[*].resultSet.metadata[*].precision</code>	Number	Specifies the precision of the column.	Not applicable	Not applicable
<code>\$.items[*].resultSet.metadata[*].scale</code>	Number	Specifies the scale of the column.	Not applicable	Not applicable
<code>\$.items[*].resultSet.metadata[*].isNullable</code>	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
<code>\$.items[*].resultSet.items</code>	Array	Specifies the list of all rows returned in the result set.	Not applicable	Not applicable
<code>\$.items[*].resultSet.items[*].<columnname></code>	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 defaults.xml file are ignored.	500	Between 0 to misc.pagination.maxRows
\$.items[*].binds	Array	Specifies an array of objects specifying the bind information.	<pre>"binds": [{ "name" :"mybind1" , "data_type ":"NUMBER" , "mode":"out" } , { "name":" mybind2", "data_type ":"NUMBER" , "value":7 }]</pre>	Not applicable

JSONPath	Data type	Description	Example Values	Possible values
<code>\$.items[*].binds[*].name</code>	String	Specifies the name of the bind, when you are using named notation.	"mybind"	Not applicable
<code>\$.items[*].binds[*].index</code>	Number	Specifies the index of bind, when you are using positional notation.	1	1 - n
<code>\$.items[*].binds[*].data_type</code>	String	Specifies the Oracle data type of the bind.	"NUMBER"	For more information, refer to Oracle Built-in Types
<code>\$.items[*].binds[*].value</code>	Any type	Specifies the value of the bind.	"value to insert"	Can be one of the following data types: <ul style="list-style-type: none"> • Number • String • Array For more information, refer to Oracle Built-in Types
<code>\$.items[*].binds[*].result</code>	Any type	Specifies the result of an OUT bind.	Not applicable	Not applicable
<code>\$.items[*].binds[*].mode</code>	String	Specifies the mode in which the bind is used.	"out"	["in" , "inout", "out"]
<code>\$.items[*].binds[*].batch</code>	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 <code>\$binds[*]</code> value must be an array of values.	true	[true, false]
<code>\$.items[*].binds[*].type_name</code>	String	Required when using <code>\$binds[*].data_type = "PL/SQL TABLE"</code> . Currently, only an empty string is accepted as the value.	""	Not applicable

JSONPath	Data type	Description	Example Values	Possible values
<code>\$.items[*].binds[*].type_subname</code>	String	Required when using <code>\$binds[*].data_type = "PL/SQL TABLE"</code> . Currently, only an empty string is accepted as the value.	""	Not applicable
<code>\$.items[*].binds[*].type_components</code>	Array	Array of data types in the PL/SQL TABLE. Required when using <code>\$binds[*].data_type = "PL/SQL TABLE"</code> .	<code>[{"data_type": "NUMBER"}]</code>	Not applicable
<code>\$.items[*].binds[*].type_components[*].data_type</code>	String	The Oracle data type of a column in the PL/SQL TABLE. Required when using <code>\$binds[*].data_type = "PL/SQL TABLE"</code>	"NUMBER"	For more information, refer to Oracle Built-in Types

9.7 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](#)

9.7.1 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](#)

9.7.2 Supported PL/SQL Statements

The REST- Enabled SQL service supports PL/SQL statements and blocks.

Example 9-7 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](#)

9.7.3 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 statements.

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](#)

9.7.3.1 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](#)

9.7.3.2 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 TIME[NG]
- SHOW USER
- SHOW VER[IFY]
- SHOW XQUERY

Related Topics

- [show_command](#)

9.7.4 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

9.8 REST-Enabled SQL Service and MySQL Database

This section describes an ORDS feature that is supported only with MySQL databases running on Oracle Cloud Infrastructure.

You can use the REST-Enabled SQL Service with MySQL database 8.0 or later, hosted in Oracle Cloud infrastructure. For MySQL database, you do not need to install any ORDS-specific software, but must specify the configuration details about how to connect to the database over JDBC through a connection pool. The ORDS distribution includes the MySQL connector/J JDBC driver.

The endpoints for REST-Enabled SQL Service and the corresponding export service end with `/_/sql` and `/_/sql/export` respectively.

ORDS returns data in a well-formed JSON structure. The MySQL data types JSON and GEOMETRY are returned as a JSON object in the response. Any binary data, such as BLOB data types, is returned as a BASE64 encoded string. The supported export format types are CSV, HTML, JSON, and XML.

9.8.1 Examples

This section describes how to configure a sample MySQL database and perform a few common operations.

The examples described in this section refers to the MySQL `sakila` sample database. The connection pool called `mysql` is configured to connect to the MySQL database instance with `db.credentials`. The source is set to `REQUEST` and MySQL database user in this example is `francis` and the password is set as `frank`.

Example 9-8 Script

This example shows how to list the schemas in the database instance.

Request

```
curl --user francis:frank --request POST 'http://localhost:8080/ords/mysql/_/
sql' \
--header 'Content-Type: application/sql' \
--data 'show databases'
```

Response

```
{
  "env" : {
    "defaultTimeZone" : "UTC"
  },
  "items" : [
    {
      "response" : [
        "Database",
        "\n",
        "-----",
        "\n",
        "information_schema",
        "\n",
        "mysql",
        "\n",
        "performance_schema",
        "\n",
        "sakila",
        "\n",
        "sys",
        "\n"
      ],
      "result" : 0,
      "statementId" : 1,
      "statementPos" : {
        "endLine" : 1,
        "startLine" : 1
      },
      "statementText" : "show databases",
      "statementType" : "sqlplus"
    }
  ]
}
```



```
]
}
```

Example 9-9 Query

This example shows how to query the `film` table in the `sakila` schema, using bind variables and limit in the query.



Note:

All bind variables are `VARCHAR` data type and are mapped to the appropriate data type for the referenced column.

Request

```
curl --user francis:frank --request POST 'http://localhost:8080/ords/mysql/_/
sql' \
--header 'Content-Type: application/json' \
--data-raw '{
  "statementText": "select film.title, film.release_year from sakila.film
film where film.rating = :var1 and film.release_year between :lowDate
and :highDate order by release_year",
  "offset": 0,
  "limit": 5,
  "binds": [
    {
      "name": "var1",
      "data_type": "VARCHAR",
      "value": "G"
    },
    {
      "name": "highDate",
      "data_type": "VARCHAR",
      "value": "2006-01-01T00:00:00Z"
    },
    {
      "name": "lowDate",
      "data_type": "VARCHAR",
      "value": "2005-01-01T00:00:00Z"
    }
  ]
}'
```

Response

```
{
  "env" : {
    "defaultTimeZone" : "Europe/Dublin"
  },
  "items" : [
    {
      "binds" : [
        {
```

```
        "data_type" : "VARCHAR",
        "name" : "var1",
        "value" : "G"
    },
    {
        "data_type" : "VARCHAR",
        "name" : "highDate",
        "value" : "2006-01-01T00:00:00Z"
    },
    {
        "data_type" : "VARCHAR",
        "name" : "lowDate",
        "value" : "2005-01-01T00:00:00Z"
    }
],
"response" : [],
"result" : 0,
"resultSet" : {
    "count" : 5,
    "hasMore" : true,
    "items" : [
        {
            "release_year" : "2006-01-01T00:00:00Z",
            "title" : "ACE GOLDFINGER"
        },
        {
            "release_year" : "2006-01-01T00:00:00Z",
            "title" : "AFFAIR PREJUDICE"
        },
        {
            "release_year" : "2006-01-01T00:00:00Z",
            "title" : "AFRICAN EGG"
        },
        {
            "release_year" : "2006-01-01T00:00:00Z",
            "title" : "ALAMO VIDEOTAPE"
        },
        {
            "release_year" : "2006-01-01T00:00:00Z",
            "title" : "AMISTAD MIDSUMMER"
        }
    ],
    "limit" : 5,
    "metadata" : [
        {
            "columnName" : "title",
            "columnClassName" : "java.lang.String",
            "columnTypeName" : "VARCHAR",
            "isNullable" : 0,
            "jsonColumnName" : "title",
            "precision" : 128,
            "scale" : 0
        },
        {
            "columnName" : "release_year",
            "columnClassName" : "java.sql.Date",
        }
    ]
}
```

```
        "columnName" : "release_year",
        "columnTypeName" : "YEAR",
        "isNullable" : 1,
        "jsonColumnName" : "release_year",
        "precision" : 4,
        "scale" : 0
    }
  ],
  "offset" : 0
},
"statementId" : 1,
"statementPos" : {
  "endLine" : 2,
  "startLine" : 1
},
"statementText" : "select film.title, film.release_year from
sakila.film film where film.rating = :var1 and film.release_year
between :lowDate and :highDate order by release_year",
"statementType" : "query"
}
]
}
```

Example 9-10 Export

This example shows how to export the rows from the `film` table in CSV format to a file `film.csv`.

Request

```
curl --user francis:frank --location --output film.csv --request
      POST 'http://localhost:8080/ords/mysql/_/sql/export' --header
'Content-Type: application/x-www-form-urlencoded' --data-urlencode
'data={"statementText":"select * from sakila.film",
      "formatDetails":{"format":"CSV", "header": true, "lineTerminator":
      "\n"}}'
```

10

GraphQL in Oracle REST Data Services

This section introduces GraphQL functionality in Oracle REST Data Services.

The GraphQL feature in Oracle REST Data Services enables you to fetch the data from an Oracle REST Data Services enabled schema using GraphQL queries.

Topics:

- [GraphQL Terminology](#)
- [Enabling GraphQL in Oracle REST Data Services](#)
- [Enabling Objects for GraphQL](#)
- [Accessing Objects Using GraphQL queries](#)
- [Examples of Filtering in Queries](#)
- [Sorting the Data](#)
- [Keyset Pagination](#)
- [Using Dynamic Arguments in Queries: Variables](#)
- [GraphiQL](#)

10.1 GraphQL Terminology

This section describes the common terms used in this section.

Following are the common terms used in this section:

- **GraphQL Schema Definition Language (SDL):** Sometimes it is simply referred to as GraphQL schema language. It is a language with a simple syntax that allows to define a schema.
- **Schema:** A schema in the GraphQL context refers to a collection of GraphQL types.
- **Type:** Represents a kind of object that you can fetch from your service. Each REST-Enabled table or view object in Oracle REST Data Services represents a GraphQL type.
- **Field:** A GraphQL type contains a set of fields that you can fetch in a query. Every column of a table or view object in Oracle REST Data Services represents a field.

10.2 Enabling GraphQL in Oracle REST Data Services

This section describes how to enable GraphQL.

To enable GraphQL, Oracle REST Data Services is required to run in a GraalVM runtime environment with the Java Script component enabled.



See Also:

[System Requirements](#)

10.3 Enabling Objects for GraphQL

This section explains how to enable the objects for GraphQL.

Any REST-Enabled table or view of an Oracle REST Data Services enabled schema can be accessed through GraphQL queries. For a REST-Enabled object to be mapped into a GraphQL type, it is necessary that it has one or multiple primary keys associated to the object. If this condition is not satisfied, then the ROWID pseudo column is used to guarantee that the objects obtained in a query are unique and are not a duplicate derived from a join.



Note:

The use of ROWID as an identifier has some limitations.

GraphQL endpoint syntax:

```
http://<HOST>:<PORT>/ords/<Schema>/_/graphql
```



Note:

This feature is available only for Oracle REST Data Services enabled schemas.

10.3.1 Accessing Protected REST-Enabled Objects

Any privilege or role defined by the user can protect the REST-Enabled objects that require authorization. For example, if a REST-enabled object is protected by the autoREST default privilege or role it requires the following roles and privileges to access such object::

- `oracle.dbtools.autoREST.any.schema`
- `oracle.dbtools.role.autoREST.<SCHEMANAME>.<OBJECTNAME>`
- `oracle.dbtools.autoREST.privilege.<SCHEMANAME>.<OBJECTNAME>`

This means that, GraphQL request must have proper authorization in order to have access to the protected objects. This protection is not limited to the auto REST privileges and roles listed above since the GraphQL feature honors URI pattern protection.



See Also:

[About Oracle REST Data Services User Roles](#)

10.4 Accessing Objects Using GraphQL queries

This section provides examples for using GraphQL queries against tables and views after REST-enabling the tables and views.

Following examples are discussed in this section:

- [Getting GraphQL Schema](#)
- [Simple Query](#)
- [Join Query](#)

10.4.1 Getting GraphQL Schema

The GraphQL schema is auto generated and it contains the REST-enabled objects (tables and views) of the rest enabled user database schema.

The generated schema includes the following:

- Each REST-enabled object represented as a GraphQL type with its columns represented as fields and the relationships between the objects.
- The resolvers for all the REST-enabled objects
- Supported data types

To get the GraphQL schema, run the following query:

Syntax:

```
GET 'http://<HOST>:<PORT>/ords/<Schema>/_/graphql'
```

Example query:

```
GET 'http://localhost:8080/ords/hr/_/graphql'
```

Response:

```
{ "schemaName": "HR", "description": "the SDL representation of the 'HR' GraphQL Schema", "SDL": "type Query {  \"\"Generic resolver for EMPLOYEES type.\"\"  employees(primaryKey: JSON, where: JSON, sort: JSON, limit: Int, offset: Int): [EMPLOYEES]  \"\"Generic resolver for COUNTRIES type.\"\"  countries(primaryKey: JSON, where: JSON, sort: JSON, limit: Int, offset: Int): [COUNTRIES]  }  }  \"\"The 'Date' scalar type represents date values as specified by the ISO 8601 format in UTC time zone (YYYY-MM-DDThh:mm:ssZ).\"\"  Date  \"\"The `Float` scalar type represents signed double-precision fractional values as specified by [IEEE 754] (https://en.wikipedia.org/wiki/IEEE\_floating\_point).\"\"  Float  \"\"The `Int` scalar type represents non-fractional signed whole numeric values. Int can represent values between  $-(2^{31})$  and  $2^{31} - 1$ .\"\"  Int  \"\"The `JSON` scalar type represents JSON values as specified by [ECMA-404] (http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf).\"\"  JSON  \"\"The `String` scalar type represents textual data, represented as UTF-8 character sequences. The String type is most often used by GraphQL to
```

```

represent free-form
  human-readable text.\n\"\"\n\nscalar String\n\nntype COUNTRIES {\n
country_id: String!\n
  country_name: String\n  region_id: Int\n}\n\nntype EMPLOYEES {\n
employee_id: Int!\n
  manager_id: Int\n  phone_number: String\n  commission_pct: Float\n
department_id: Int\n
  salary: Float\n  first_name: String\n  email: String!\n  job_id: String!\n
  hire_date:
    Date!\n  last_name: String!\n\n  \"\"\n\n  The relationship between the
EMPLOYEES type
  and the EMPLOYEES type on EMPLOYEES.MANAGER_ID =
EMPLOYEES.EMPLOYEE_ID\n  \"\"\n\n
  manager_id_employees(primaryKey: JSON, where: JSON, sort: JSON, limit:
Int, offset: Int):
  [EMPLOYEES]\n\n  \"\"\n\n  The relationship between the EMPLOYEES type
and the EMPLOYEES
  type on EMPLOYEES.EMPLOYEE_ID = EMPLOYEES.MANAGER_ID\n  \"\"\n\n
  employees_manager_id(primaryKey: JSON, where: JSON, sort: JSON, limit:
Int, offset: Int):
  [EMPLOYEES]\n\n}\"}

```

10.4.2 Simple Query

A simple query retrieves the data in a type present in the GraphQL Schema.

This example query fetches the `employee_id`, `first_name`, `last_name`, `job_id`, and `salary` in the `employees` type from the HR schema.

```

query Employees {
  employees {
    employee_id
    first_name
    last_name
    job_id
    salary
  }
}

```

Example cURL command:

```

curl --location 'http://localhost:8080/ords/hr/_/graphql' \
--header 'Content-Type: application/json' \
--data '{
  "query": "{employees { employee_id first_name last_name job_id salary }}"
}'

```

Response:

```

{
  "data": {
    "employees": [
      {
        "employee_id": 100,

```

```
    "first_name": "Steven",
    "last_name": "King",
    "job_id": "AD_PRES",
    "salary": 24000
  },
  {
    "employee_id": 101,
    "first_name": "Neena",
    "last_name": "Kochhar",
    "job_id": "AD_VP",
    "salary": 17000
  },
  {
    "employee_id": 103,
    "first_name": "Alexander",
    "last_name": "Hunold",
    "job_id": "IT_PROG",
    "salary": 9000
  },
  {
    "employee_id": 104,
    "first_name": "Bruce",
    "last_name": "Ernst",
    "job_id": "IT_PROG",
    "salary": 6000
  },
  {
    "employee_id": 105,
    "first_name": "David",
    "last_name": "Austin",
    "job_id": "IT_PROG",
    "salary": 4800
  },
  ...
}
```

10.4.3 Join Query

A join query retrieves the data from one or more relationships between existing types present in the GraphQL Schema.

Example 1:

The following query fetches all the cities associated with a location as well as the departments in each city and the employees who work in each one of the departments.

```
query Locations{
  locations{
    city
    departments_location_id{
      department_name
      employees_department_id{
        first_name
        last_name
        salary
      }
    }
  }
}
```



```
    }  
  }  
}
```

Example cURL Command:

```
curl --location 'http://localhost:8080/ords/hr/_/graphql' \  
--header 'Content-Type: application/json' \  
--data '{  
  "query": "query Locations{ locations{ city  
departments_location_id{ department_name employees_department_id{first_name  
last_name salary} } } }"  
}'
```

Response:

```
{  
  "data": {  
    "locations": [  
      {  
        "city": "Seattle",  
        "departments_location_id": [  
          {  
            "department_name": "Executive",  
            "employees_department_id": [  
              {  
                "first_name": "Steven",  
                "last_name": "King",  
                "salary": 24000  
              },  
              {  
                "first_name": "Neena",  
                "last_name": "Kochhar",  
                "salary": 17000  
              },  
              {  
                "first_name": "Lex",  
                "last_name": "De Haan",  
                "salary": 17000  
              }  
            ]  
          }  
        ],  
      },  
      {  
        "department_name": "Finance",  
        "employees_department_id": [  
          {  
            "first_name": "Nancy",  
            "last_name": "Greenberg",  
            "salary": 12000  
          },  
          {  
            "first_name": "Daniel",  
            "last_name": "Faviet",  
            "salary": 9000  
          }  
        ],  
      }  
    ]  
  }  
}
```

```
{
  "first_name": "John",
  "last_name": "Chen",
  "salary": 8200
},
{
  "first_name": "Ismael",
  "last_name": "Sciarra",
  "salary": 7700
},
{
  "first_name": "Jose Manuel",
  "last_name": "Urman",
  "salary": 7800
},
{
  "first_name": "Luis",
  "last_name": "Popp",
  "salary": 6900
}
]
},
{
  "department_name": "Purchasing",
  "employees_department_id": [
    {
      "first_name": "Den",
      "last_name": "Raphaely",
      "salary": 11000
    },
    {
      "first_name": "Alexander",
      "last_name": "Khoo",
      "salary": 3100
    },
    {
      "first_name": "Shelli",
      "last_name": "Baida",
      "salary": 2900
    },
    {
      "first_name": "Sigal",
      "last_name": "Tobias",
      "salary": 2800
    },
    {
      "first_name": "Guy",
      "last_name": "Himuro",
      "salary": 2600
    },
    {
      "first_name": "Karen",
      "last_name": "Colmenares",
      "salary": 2500
    }
  ]
}
```

```
},
{
  "department_name": "Administration",
  "employees_department_id": [
    {
      "first_name": "Jennifer",
      "last_name": "Whalen",
      "salary": 4400
    }
  ]
},
{
  "department_name": "Accounting",
  "employees_department_id": [
    {
      "first_name": "Shelley",
      "last_name": "Higgins",
      "salary": 12000
    },
    {
      "first_name": "William",
      "last_name": "Gietz",
      "salary": 8300
    }
  ]
},
{
  "department_name": "IT Support",
  "employees_department_id": []
},
{
  "department_name": "Operations",
  "employees_department_id": []
},
{
  "department_name": "Payroll",
  "employees_department_id": []
},
{
  "department_name": "Construction",
  "employees_department_id": []
},
{
  "department_name": "Government Sales",
  "employees_department_id": []
},
{
  "department_name": "Retail Sales",
  "employees_department_id": []
},
{
  "department_name": "Contracting",
  "employees_department_id": []
},
{
  "department_name": "Recruiting",
```

```

        "employees_department_id": []
      },
      {
        "department_name": "Control And Credit",
        "employees_department_id": []
      },
      {
        "department_name": "NOC",
        "employees_department_id": []
      },
      {
        "department_name": "Treasury",
        "employees_department_id": []
      },
      {
        "department_name": "Manufacturing",
        "employees_department_id": []
      },
      {
        "department_name": "Corporate Tax",
        "employees_department_id": []
      },
      {
        "department_name": "IT Helpdesk",
        "employees_department_id": []
      },
      {
        "department_name": "Shareholder Services",
        "employees_department_id": []
      },
      {
        "department_name": "Benefits",
        "employees_department_id": []
      }
    ]
  }
]
}

```

Example 2:

The following example query fetches all the employees from the HR schema and the department in which they work:

```

query Employees {
  employees {
    employee_id
    first_name
    last_name
    departments_department_id {
      department_id
      department_name
    }
  }
}

```

```
    }
  }
}
```

Example cURL command:

```
curl --location 'http://localhost:8080/ords/hr/_/graphql' \
--header 'Content-Type: application/json' \
--data '{
  "query": "{employees { employee_id first_name last_name
departments_department_id{ department_id department_name } }}"
}'
```

Response:

```
{
  "data": {
    "employees": [
      {
        "employee_id": 200,
        "first_name": "Jennifer",
        "last_name": "Whalen",
        "departments_department_id": [
          {
            "department_id": 10,
            "department_name": "Administration"
          }
        ]
      },
      {
        "employee_id": 201,
        "first_name": "Michael",
        "last_name": "Hartstein",
        "departments_department_id": [
          {
            "department_id": 20,
            "department_name": "Marketing"
          }
        ]
      },
      {
        "employee_id": 202,
        "first_name": "Pat",
        "last_name": "Fay",
        "departments_department_id": [
          {
            "department_id": 20,
            "department_name": "Marketing"
          }
        ]
      },
      ...
    ]
  }
}
```

 **Note:**

GraphQL nesting depth is limited to a maximum of five levels. Any query with more than five nested joins returns an error.

 **See Also:**

Understanding Configurable Settings

10.4.3.1 Circular Relationships Between Objects

This section explains with an example a circular relationship.

A table or view can have a circular relationship and GraphQL can be used to query the data.

Following is an example showing a circular relationship in the HR schema.

The `employees` table has a constraint defined between `manager_id` and `employee_id` columns.

The following example query fetches all the employees from the HR schema along with their respective managers:

```
query Employees {
  employees {
    employee_id
    first_name
    last_name
    manager_id
    manager_id_employees {
      first_name
      last_name
      employee_id
    }
  }
}
```

Example cURL Command:

```
curl --location 'http://localhost:8080/ords/hr/_/graphql' \
--header 'Content-Type: application/json' \
--data '{
  "query": "{ employees { employee_id first_name last_name manager_id
employees_manager_id{ first_name last_name employee_id } } }"
```

Response:

```
{
  "data": {
    "employees": [
      {
```

```
"employee_id": 101,  
"first_name": "Neena",  
"last_name": "Kochhar",  
"manager_id": 100,  
"employees_manager_id": [  
  {  
    "first_name": "Steven",  
    "last_name": "King",  
    "employee_id": 100  
  }  
]  
},  
{  
  "employee_id": 114,  
  "first_name": "Den",  
  "last_name": "Raphaely",  
  "manager_id": 100,  
  "employees_manager_id": [  
    {  
      "first_name": "Steven",  
      "last_name": "King",  
      "employee_id": 100  
    },  
    {  
      "first_name": "Eleni",  
      "last_name": "Zlotkey",  
      "employee_id": 149  
    }  
  ]  
},  
{  
  "employee_id": 120,  
  "first_name": "Matthew",  
  "last_name": "Weiss",  
  "manager_id": 100,  
  "employees_manager_id": [  
    {  
      "first_name": "Steven",  
      "last_name": "King",  
      "employee_id": 100  
    },  
    {  
      "first_name": "John",  
      "last_name": "Russell",  
      "employee_id": 145  
    },  
    {  
      "first_name": "Karen",  
      "last_name": "Partners",  
      "employee_id": 146  
    }  
  ]  
}  
}  
}
```

10.5 Examples of Filtering in Queries

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

To filter in a query, include the parameter `<filterName>`: GraphQLJSON, where GraphQLJSON is a JSON like object that represents the custom selection to be applied to the resource. Each filter has its own predefined GraphQLJSON syntax.

10.5.1 Supported Data Types

This section lists the supported data types for filters.

Data Type	Description
String	The <code>string</code> scalar type represents a textual data, represented as UTF-8 character sequences. The string type is most often used by GraphQL to represent free-form human-readable text.
Int	The <code>int</code> scalar type represents non-fractional signed whole numeric values. Int can represent values between $-(2^{31})$ and $2^{31} - 1$.
Float	The <code>float</code> scalar type represents signed double-precision fractional values as specified by IEEE 754.
Date	The <code>date</code> scalar type represents date values as specified by the ISO 8601 format in UTC time zone (YYYY-MM-DDThh:mm:ssZ).
Timestamp	The <code>timestamp</code> scalar type represents timestamp values as specified by the ISO 8601 format in UTC time zone (YYYY-MM-DDThh:mm:ss.sssZ).
Boolean	The <code>boolean</code> scalar type represents <code>true</code> or <code>false</code> .

10.5.2 Filtering by Primary Key

Filtering by primary key enables you to retrieve the data by specifying its identifying key value or key values.

Primary Key Syntax:

```
value = String | Int | Float | Date | Timestamp
primaryKeyPair = <fieldName> : <value>
primaryKeyExp = { primaryKeyPair1, ... , primaryKeyPairN }
```

The following query includes a filter that restricts the `employee_id` field to 100:

```
query {
  employees(primaryKey: {employee_id: 100}){
    employee_id
    first_name
    last_name
    job_id
    salary
  }
}
```



```

    }
  }
}

```

Example cURL command:

```

curl --location 'http://localhost:8080/ords/hr/_/graphql' \
--header 'Content-Type: application/json' \
--data '{
  "query": "{ employees(primaryKey : {employee_id :100}) { first_name
last_name department_id job_id } } "
}'

```

Response:

```

{
  "data": {
    "employees": [
      {
        "first_name": "Steven",
        "last_name": "King",
        "department_id": 90,
        "job_id": "AD_PRES"
      }
    ]
  }
}

```

10.5.2.1 Filtering by Composite Primary Key

Filtering by primary key enables you to retrieve the data from the tables that have a composite primary key by adding a list of primary keys to the filter.

```

query {
  compositeTable(primaryKey: { <fieldName> : <value>, <fieldName> : <value>}){
    data
  }
}

```

10.5.3 Where Filter

Filtering using a `where` condition enables you to query the data and specify a valid condition or conditions that the fields present in the requested types should satisfy.

Where Filter Syntax:

```

fieldName = stringValue = String | Int | Float | Date | Timestamp operator =
eq | neq | gt | lt | gte | lte | like | nlike | in | nin | btwn | nbtwn
| nullbooleanOperator = and | orvalidFilter = { <fieldName> :
{ <operator> : <value> } }booleanExp = { <booleanOperator> : [ <ValidFilter1
| BooleanExp1>, ..., <ValidFilterN
| BooleanExpN> ] }whereExp = { where : <validFilter | booleanExp> }

```

Table 10-1 Supported Operators

Operator	GraphQLJSON Syntax	Description	Supported Data Types
=	{ column : { eq : value } }	Equality	String Int Float Date Timestamp
!=, <>	{ column : { neq : value } }	Inequality	String Int Float Date Timestamp
>	{ column : { gt : value } }	Greater than	String Int Float Date Timestamp
<	{ column : { lt : value } }	Less than	String Int Float Date Timestamp
>=	{ column : { gte : value } }	Greater than or equal to	String Int Float Date Timestamp
<=	{ column : { lte : value } }	Less than or equal to	String Int Float Date Timestamp
LIKE	{ column : { like : pattern } }	Operator used for pattern matching	String
NOT LIKE	{ column : { nlike : pattern } }	Operator used for pattern matching	String
IN	{ column : { in : [value1, ... , value_n] } }	Equal to any value in a list of values	String Int Float Date Timestamp
NOT IN	{ column : { nin : [value_1, ... ,value_n] } }	Not equal to any value in a list of values	String Int Float Date Timestamp
BETWEEN	{ column : { btwn : [value_1, value_2] } }	Equivalent to >= n and <= y	String Int Float Date Timestamp
NOT BETWEEN	{ column : { nbtwn : [value_1, value_2] } }	Equivalent to NOT >= n and <= y	String Int Float Date Timestamp
IS NULL	{ column : { null : [Boolean] } }	NULL test	Boolean
OR	{ or : [{ GraphQL expression 1 } , ... , { GraphQL expression n }] }	Logical operator, returns true if any expression is true.	Not Applicable
NOT	{ NOT : { GraphQL expression} }	Logical operator, negates the logical value of the expression on which it operates.	Not Applicable

Table 10-1 (Cont.) Supported Operators

Operator	GraphQLJSON Syntax	Description	Supported Data Types
AND	<pre>{ and : [{ GraphQL expression 1 }, , { GraphQL expression n }] }</pre>	Logical operator, returns true if both expressions are true.	Not Applicable

10.5.3.1 Example: EQUALS (eq) operator

The following query includes a filter that restricts the the `job_id` field to `IT_PROG`.

```
query {
  employees(where : {job_id: {eq : "IT_PROG"}}){
    employee_id
    first_name
    last_name
    job_id
    salary
  }
}
```

Example cURL command:

```
curl --location 'http://localhost:8080/ords/hr/_/graphql' \
--header 'Content-Type: application/json' \
--data '{
  "query": "{ employees(where : {job_id : {eq : \"IT_PROG\"}}) { employee_id
first_name last_name job_id salary } } "
}'
```

Response:

```
{
  "data": {
    "employees": [
      {
        "employee_id": 103,
        "first_name": "Alexander",
        "last_name": "Hunold",
        "job_id": "IT_PROG",
        "salary": 9000
      },
      {
        "employee_id": 104,
        "first_name": "Bruce",
        "last_name": "Ernst",

```

```

        "job_id": "IT_PROG",
        "salary": 6000
    },
    {
        "employee_id": 105,
        "first_name": "David",
        "last_name": "Austin",
        "job_id": "IT_PROG",
        "salary": 4800
    },
    {
        "employee_id": 106,
        "first_name": "Valli",
        "last_name": "Pataballa",
        "job_id": "IT_PROG",
        "salary": 4800
    },
    {
        "employee_id": 107,
        "first_name": "Diana",
        "last_name": "Lorentz",
        "job_id": "IT_PROG",
        "salary": 4200
    }
]
}
}

```

10.5.3.2 Example: Greater than (>) Operator and Date Data Type

The following query includes a filter that restricts the `hire_date` field to be greater than 01 Jan 2006.

```

query {
  employees(where : { hire_date : { gt : "2006-01-01T00:00:00Z" } }) {
    employee_id
    first_name
    last_name
    hire_date
  }
}

```

10.5.3.3 Example: LIKE (like) operator

The following query includes a filter that restricts the `first_name` field to match the pattern S%:

```

query {
  employees(where : { first_name : { like : "S%" } }) {
    employee_id
    first_name
    last_name
  }
}

```

10.5.3.4 Example: IN (in) operator

The following query includes a filter that restricts the the `job_id` field to `IT_PROG` or `FI_ACCOUNT` using the `in` operator:

```
query {
  employees(where : { job_id : { in : ["IT_PROG", "FI_ACCOUNT"] } }) {
    employee_id
    first_name
    last_name
    job_id
    salary
  }
}
```

10.5.3.5 Example: NOT (not) Operator

The following query includes a filter that negates the result of restricting the salary field to be between 2000 and 10000.

```
query Employees {
  employees(where : {not : {salary : {btwn : [2000, 10000]}}}) {
    employee_id
    first_name
    last_name
    job_id
    salary
  }
}
```

Request:

```
curl --location 'http://localhost:8080/ords/hr/_/graphql' \
--header 'Content-Type: application/json' \
--data '{
  "query": "{employees(where : {not : {salary : {btwn : [2000, 10000]}}}){
    employee_id first_name last_name job_id salary } } "
```

Response:

```
{
  "data": {
    "employees": [
      {
        "employee_id": 100,
        "first_name": "Steven",
        "last_name": "King",
        "job_id": "AD_PRE",
        "salary": 24000
      },
      {
```

```
    "employee_id": 101,  
    "first_name": "Neena",  
    "last_name": "Kochhar",  
    "job_id": "AD_VP",  
    "salary": 17000  
  },  
  {  
    "employee_id": 102,  
    "first_name": "Lex",  
    "last_name": "De Haan",  
    "job_id": "AD_VP",  
    "salary": 17000  
  },  
  {  
    "employee_id": 108,  
    "first_name": "Nancy",  
    "last_name": "Greenberg",  
    "job_id": "FI_MGR",  
    "salary": 12008  
  },  
  {  
    "employee_id": 114,  
    "first_name": "Den",  
    "last_name": "Raphaely",  
    "job_id": "PU_MAN",  
    "salary": 11000  
  },  
  {  
    "employee_id": 145,  
    "first_name": "John",  
    "last_name": "Russell",  
    "job_id": "SA_MAN",  
    "salary": 14000  
  },  
  {  
    "employee_id": 146,  
    "first_name": "Karen",  
    "last_name": "Partners",  
    "job_id": "SA_MAN",  
    "salary": 13500  
  },  
  {  
    "employee_id": 147,  
    "first_name": "Alberto",  
    "last_name": "Errazuriz",  
    "job_id": "SA_MAN",  
    "salary": 12000  
  },  
  {  
    "employee_id": 148,  
    "first_name": "Gerald",  
    "last_name": "Cambrault",  
    "job_id": "SA_MAN",  
    "salary": 11000  
  },  
  {  
    }
```

```
        "employee_id": 149,  
        "first_name": "Eleni",  
        "last_name": "Zlotkey",  
        "job_id": "SA_MAN",  
        "salary": 10500  
    },  
    {  
        "employee_id": 162,  
        "first_name": "Clara",  
        "last_name": "Vishney",  
        "job_id": "SA_REP",  
        "salary": 10500  
    },  
    {  
        "employee_id": 168,  
        "first_name": "Lisa",  
        "last_name": "Ozer",  
        "job_id": "SA_REP",  
        "salary": 11500  
    },  
    {  
        "employee_id": 174,  
        "first_name": "Ellen",  
        "last_name": "Abel",  
        "job_id": "SA_REP",  
        "salary": 11000  
    },  
    {  
        "employee_id": 201,  
        "first_name": "Michael",  
        "last_name": "Hartstein",  
        "job_id": "MK_MAN",  
        "salary": 13000  
    },  
    {  
        "employee_id": 205,  
        "first_name": "Shelley",  
        "last_name": "Higgins",  
        "job_id": "AC_MGR",  
        "salary": 12008  
    }  
  ]  
}  
}
```

10.5.3.6 Example: AND (and) operator

The following query includes a filter that restricts the the `job_id` field to `IT_PROG` and the `salary` field to be between 4000 and 6000:

```
query Employees {  
  employees(where : { and : [  
    {job_id : { eq : "IT_PROG" }},  
    {salary : { btwn : [4000, 6000] }}  
  ]}){
```

```
    employee_id
    first_name
    last_name
    job_id
    salary
  }
}
```

Request:

```
query Employees {
  employees(where : { and : [
    {job_id : { eq : "IT_PROG" }},
    {salary : { btwn : [4000, 6000] }}
  ]}){
    employee_id
    first_name
    last_namecurl --location 'http://localhost:8080/ords/hr/_/graphql' \
--header 'Content-Type: application/json' \
--data '{
  "query": "{employees(where : { and : [ {job_id : { eq : \"IT_PROG\" }},
{salary : { btwn : [4000, 6000] }} ] }){
    employee_id first_name last_name job_id salary } } "
}'
    job_id
    salary
  }
}
```

Response:

```
{
  "data": {
    "employees": [
      {
        "employee_id": 104,
        "first_name": "Bruce",
        "last_name": "Ernst",
        "job_id": "IT_PROG",
        "salary": 6000
      },
      {
        "employee_id": 105,
        "first_name": "David",
        "last_name": "Austin",
        "job_id": "IT_PROG",
        "salary": 4800
      },
      {
        "employee_id": 106,
        "first_name": "Valli",
        "last_name": "Pataballa",
        "job_id": "IT_PROG",
        "salary": 4800
      },
    ],
  },
}
```



```

        {
            "employee_id": 107,
            "first_name": "Diana",
            "last_name": "Lorentz",
            "job_id": "IT_PROG",
            "salary": 4200
        }
    ]
}

```

10.5.3.7 Example: OR (or) operator

The following query includes a filter that restricts the the `job_id` field to `IT_PROG` or `FI_ACCOUNT` using `or` operator:

```

query Employees {
  employees(where : { or : [
    {job_id : { eq : "IT_PROG" }},
    {job_id : { eq : "FI_ACCOUNT" }}
  ]}){
    employee_id
    first_name
    last_name
    job_id
    salary
  }
}

```

10.5.3.8 Example: Where Filter in Children Types

All the filters described in the preceding sections can be applied to nested types in a query, that enables you to widen the range of fields that can be filtered in a single query.

The following query retrieves all employees that are managers of employees whose `job_id` is equal to `IT_PROG`:

```

query{
  employees{
    employee_id
    first_name
    last_name
    job_id
    salary
    employees_manager_id(where : {job_id : {eq : "IT_PROG"}}){
      employee_id
      first_name
      last_name
      job_id
      salary
    }
  }
}

```

Request:

```
curl --location 'http://localhost:8080/ords/hr/_/graphql' \
--header 'Content-Type: application/json' \
--data '{
  "query": "query{ employees{ employee_id first_name last_name job_id
salary employees_manager_id( where : { job_id :
  { eq :      \"IT_PROG\" } } )}{employee_id first_name last_name job_id
salary} } }"
}'
```

Response:

```
{
  "data": {
    "employees": [
      {
        "employee_id": 102,
        "first_name": "Lex",
        "last_name": "De Haan",
        "job_id": "AD_VP",
        "salary": 17000,
        "employees_manager_id": [
          {
            "employee_id": 103,
            "first_name": "Alexander",
            "last_name": "Hunold",
            "job_id": "IT_PROG",
            "salary": 9000
          }
        ]
      },
      {
        "employee_id": 103,
        "first_name": "Alexander",
        "last_name": "Hunold",
        "job_id": "IT_PROG",
        "salary": 9000,
        "employees_manager_id": [
          {
            "employee_id": 104,
            "first_name": "Bruce",
            "last_name": "Ernst",
            "job_id": "IT_PROG",
            "salary": 6000
          },
          {
            "employee_id": 105,
            "first_name": "David",
            "last_name": "Austin",
            "job_id": "IT_PROG",
            "salary": 4800
          }
        ],
        {
          "employee_id": 106,
```

```

        "first_name": "Valli",
        "last_name": "Pataballa",
        "job_id": "IT_PROG",
        "salary": 4800
    },
    {
        "employee_id": 107,
        "first_name": "Diana",
        "last_name": "Lorentz",
        "job_id": "IT_PROG",
        "salary": 4200
    }
]
}
]
}
}
}

```

10.5.3.9 Working with Dates/Timestamps Using Filters

Most of the filters described in the previous sections, can be applied on fields whose type is `Date` or `Timestamp`. To apply these filters on fields whose type is `Date`, you must use the format: `YYYY-MM-DDThh:mm:ssZ`. For the date fields, `YYYY-MM-DD` format can also be used. To apply these filters on fields whose type is `Timestamp`, you must use the format: `YYYY-MM-DDThh:mm:ss.sssZ`. The following query includes a filter that restricts the `hire_date` field to be inbetween the range 01 Jan 2006 and 01 Jun 2006:

```

query{
  employees(where : {hire_date : {btwn : ["2006-01-01", "2006-06-01"]}){
    employee_id
    first_name
    last_name
    job_id
    salary
    hire_date
  }
}

```

Request:

```

curl --location 'http://localhost:8080/ords/hr/_graphql' --header 'Content-Type:
application/json' --data '{  "query": "query{ employees(where :
{hire_date : {btwn : [\"2006-01-01\",
    \"2006-06-01\"]})}{employee_id first_name last_name job_id salary
hire_date}
}"}'

```

Response:

```

{
  "data": {
    "employees": [
      {

```

```
"employee_id": 103,  
"first_name": "Alexander",  
"last_name": "Hunold",  
"job_id": "IT_PROG",  
"salary": 9000,  
"hire_date": "2006-01-03T00:00:00Z"  
},  
{  
  "employee_id": 106,  
  "first_name": "Valli",  
  "last_name": "Pataballa",  
  "job_id": "IT_PROG",  
  "salary": 4800,  
  "hire_date": "2006-02-05T00:00:00Z"  
},  
{  
  "employee_id": 112,  
  "first_name": "Jose Manuel",  
  "last_name": "Urman",  
  "job_id": "FI_ACCOUNT",  
  "salary": 7800,  
  "hire_date": "2006-03-07T00:00:00Z"  
},  
{  
  "employee_id": 139,  
  "first_name": "John",  
  "last_name": "Seo",  
  "job_id": "ST_CLERK",  
  "salary": 2700,  
  "hire_date": "2006-02-12T00:00:00Z"  
},  
{  
  "employee_id": 140,  
  "first_name": "Joshua",  
  "last_name": "Patel",  
  "job_id": "ST_CLERK",  
  "salary": 2500,  
  "hire_date": "2006-04-06T00:00:00Z"  
},  
{  
  "employee_id": 143,  
  "first_name": "Randall",  
  "last_name": "Matos",  
  "job_id": "ST_CLERK",  
  "salary": 2600,  
  "hire_date": "2006-03-15T00:00:00Z"  
},  
{  
  "employee_id": 153,  
  "first_name": "Christopher",  
  "last_name": "Olsen",  
  "job_id": "SA_REP",  
  "salary": 8000,  
  "hire_date": "2006-03-30T00:00:00Z"  
},  
{
```

```
    "employee_id": 169,  
    "first_name": "Harrison",  
    "last_name": "Bloom",  
    "job_id": "SA_REP",  
    "salary": 10000,  
    "hire_date": "2006-03-23T00:00:00Z"  
  },  
  {  
    "employee_id": 170,  
    "first_name": "Tayler",  
    "last_name": "Fox",  
    "job_id": "SA_REP",  
    "salary": 9600,  
    "hire_date": "2006-01-24T00:00:00Z"  
  },  
  {  
    "employee_id": 176,  
    "first_name": "Jonathon",  
    "last_name": "Taylor",  
    "job_id": "SA_REP",  
    "salary": 8600,  
    "hire_date": "2006-03-24T00:00:00Z"  
  },  
  {  
    "employee_id": 177,  
    "first_name": "Jack",  
    "last_name": "Livingston",  
    "job_id": "SA_REP",  
    "salary": 8400,  
    "hire_date": "2006-04-23T00:00:00Z"  
  },  
  {  
    "employee_id": 180,  
    "first_name": "Winston",  
    "last_name": "Taylor",  
    "job_id": "SH_CLERK",  
    "salary": 3200,  
    "hire_date": "2006-01-24T00:00:00Z"  
  },  
  {  
    "employee_id": 181,  
    "first_name": "Jean",  
    "last_name": "Fleaur",  
    "job_id": "SH_CLERK",  
    "salary": 3100,  
    "hire_date": "2006-02-23T00:00:00Z"  
  },  
  {  
    "employee_id": 196,  
    "first_name": "Alana",  
    "last_name": "Walsh",  
    "job_id": "SH_CLERK",  
    "salary": 3100,  
    "hire_date": "2006-04-24T00:00:00Z"  
  },  
  {  
    }
```

```

        "employee_id": 197,
        "first_name": "Kevin",
        "last_name": "Feeney",
        "job_id": "SH_CLERK",
        "salary": 3000,
        "hire_date": "2006-05-23T00:00:00Z"
    }
  ]
}
}

```

10.6 Sorting the Data

Sorting enables you to sort the data in an ascending or descending order by one or more fields.

Sort Query Syntax:

```

sortValue = "asc" | "desc" | "ASC" | "DESC"
sortExp = [{<fieldName> : sortValue}, ... ,{<fieldName> : sortValue} ]
sort : <sortExp>

```

The following query specifies `sort` filter to order the `employee_id` field in a descending order:

```

query {
  employees(sort : [ { employee_id : "desc" } ] ){
    employee_id
    first_name
    last_name
    salary
  }
}

```

Request:

```

curl --location 'http://localhost:8080/ords/hr/_/graphql' \
--header 'Content-Type: application/json' \
--data '{
  "query": "query { employees(sort : [ { employee_id : \"desc\" } ] )
  { employee_id first_name last_name salary } }"
}'

```

Response:

```

{
  "data": {
    "employees": [
      {
        "employee_id": 206,
        "first_name": "William",
        "last_name": "Gietz",
        "salary": 8300
      },
      {

```

```
        "employee_id": 205,  
        "first_name": "Shelley",  
        "last_name": "Higgins",  
        "salary": 12008  
    },  
    {  
        "employee_id": 204,  
        "first_name": "Hermann",  
        "last_name": "Baer",  
        "salary": 10000  
    },  
    {  
        "employee_id": 203,  
        "first_name": "Susan",  
        "last_name": "Mavris",  
        "salary": 6500  
    },  
    {  
        "employee_id": 202,  
        "first_name": "Pat",  
        "last_name": "Fay",  
        "salary": 6000  
    },  
    {  
        "employee_id": 201,  
        "first_name": "Michael",  
        "last_name": "Hartstein",  
        "salary": 13000  
    },  
    {  
        "employee_id": 200,  
        "first_name": "Jennifer",  
        "last_name": "Whalen",  
        "salary": 4400  
    },  
    ...  
  ]  
}
```

10.6.1 Example: Sorting by Multiple Columns

The following query includes a sort filter that orders the data in a descending order by `department_id` field and in an ascending order by `salary` field:

```
query {  
  employees(sort : [ { department_id : "desc" } , { salary : "asc" } ] ){  
    employee_id  
    first_name  
    last_name  
    salary  
    department_id  
  }  
}
```

10.7 Keyset Pagination

Keyset pagination enables you to specify a `limit` and `offset` to paginate the data received from any given query. If sorting expression is not specified, then `ROWID` is used by default as a sort argument to uniquely address the rows.

The following query specifies the `offset` and `limit` parameters:

```
query Employees {
  employees(limit: 3, offset: 5) {
    employee_id
    first_name
    last_name
    email
  }
}
```

Request:

```
curl --location 'http://localhost:8080/ords/hr/_/graphql' \
--header 'Content-Type: application/json' \
--data '{
  "query": "query { employees( limit: 3, offset: 5 ){ employee_id
first_name    last_name    email } }"
}'
```

Response:

```
{
  "data": {
    "employees": [
      {
        "employee_id": 105,
        "first_name": "David",
        "last_name": "Austin",
        "email": "DAUSTIN"
      },
      {
        "employee_id": 106,
        "first_name": "Valli",
        "last_name": "Pataballa",
        "email": "VPATABAL"
      },
      {
        "employee_id": 107,
        "first_name": "Diana",
        "last_name": "Lorentz",
        "email": "DLORENTZ"
      }
    ]
  }
}
```


10.7.1 Example: Pagination with Other Filters

The following query specifies the `offset` and `limit` parameters and orders the results in a descending order by `employee_id` field:

```
query {
  employees(sort : [ { employee_id : "DESC" } ], limit: 3, offset: 2){
    employee_id
    first_name
    last_name
    salary
    department_id
  }
}
```

10.7.2 Example: Pagination in Nested Types

The following query specifies the `limit` parameter both in `employees` and in the nested type `employees_manager_id` and limits the number of employees returned in the nested object to two:

```
query{
  employees(limit : 1){
    employee_id
    first_name
    last_name
    job_id
    salary
    employees_manager_id(limit : 2){
      employee_id
      first_name
    }
  }
}
```

Request:

```
curl --location 'http://localhost:8080/ords/hr/_/graphql' \
--header 'Content-Type: application/json' \
--data '{
  "query": "query {employees(limit : 1){employee_id first_name last_name
job_id salary employees_manager_id(limit : 2){employee_id first_name}}}"
}'
```

Response:

```
{
  "data": {
    "employees": [
      {
        "employee_id": 100,
        "first_name": "Steven",
```

```

    "last_name": "King",
    "job_id": "AD_PRES",
    "salary": 24000,
    "employees_manager_id": [
      {
        "employee_id": 101,
        "first_name": "Neena"
      },
      {
        "employee_id": 102,
        "first_name": "Lex"
      }
    ]
  }
]
}
}
}

```

10.8 Using Dynamic Arguments in Queries: Variables

To replace variables with static values in GraphQL queries, perform the following steps:

1. Replace the static value with `$variableName`
2. Declare `$variableName` as one of the variables accepted by the query and then specify the data type
3. Pass variables dictionary separately

The following query uses variables to use dynamic values in the filters:

```

query Employees($job_id : String, $min_salary : Int, $max_salary : Int){
  employees (where : { and : [
    {job_id : { eq : $job_id }},
    {salary : { btwn : [$min_salary, $max_salary] }} ]}){
    employee_id
    manager_id
    phone_number
    commission_pct
    department_id
    salary
    first_name
    email
    job_id
    hire_date
    last_name
  }
}

```

Variables Dictionary:

```

{
  "job_id" : "IT_PROG",
  "min_salary" : 4000,

```

```
    "max_salary" : 6000
  }
```

Request:

```
curl --location 'http://localhost:8080/ords/hr/_/graphql' \
--header 'Content-Type: application/json' \
--data '{
  "query": "query Employees($job_id : String, $min_salary :
Int, $max_salary : Int){ employees (where : { and : [\n  {job_id :
{ eq : $job_id }}, {salary : { btwn : [$min_salary, $max_salary] }} ]})
{ employee_id manager_id phone_number commission_pct department_id salary
first_name email job_id hire_date last_name }}",
  "operationName": "Employees",
  "variables": {
    "job_id": "IT_PROG",
    "min_salary": 4000,
    "max_salary": 6000
  }
}'
```

10.9 GraphiQL

Oracle REST Data Services includes GraphiQL, an in-browser IDE for exploring GraphQL. Use the following endpoint and login with the Rest-enabled user database schema credentials:

```
http://<HOST>:<PORT>/ords/<SCHEMANAME>/_/graphql
```

11

Extending ORDS Functionality with Plugins

This chapter explains and provides examples on using ORDS plugin framework.

ORDS has a plugin framework that allows you to add your own custom functionality into the ORDS web application. Plugins can be added to the ORDS runtime by placing the jar files in the `lib/ext` directory. The ORDS distribution contains the source for example plugins. The plugin examples can be built using Apache `ant`, a software tool used for automating the build processes.

11.1 Plugin Demonstration Example

This section shows how you can locate and build a plugin demonstration example..

The plugin-demonstraion example is at `examples/plugins/plugin-demo` location and contains the source for a `HttpServlet` that gets a database connection injected at runtime. The servlet uses that JDBC database connection to run a query in the database and return a response at runtime.

Perform the following steps to build and use the demonstration example:

1. Change the directory to `examples/plugins/plugin-demo`
2. Run `ant` to build the `examples/plugins/plugin-demo/built/plugin-demo.jar` file
3. Copy the `plugin-demo.jar` to the ORDS distribution `lib/ext` directory and start an ORDS instance.
4. Invoke the servlet using the following URL pattern: `http://server/ords/schema/demos/plugin?who=somebody`
 - a. For example: `http://localhost:8080/ords/hr/demos/plugin?who=scott` where ORDS is configured with a default pool and `HR` is an alias for a REST Enabled Schema in that database.

The details of developing and deploying Java based plugins is available in the *Oracle REST Data Services Java API Reference* book.

See Also:

- [Getting Started Guide](#)
- [Developer Guide](#)
- [Route Patterns Specification](#)

11.2 Plugin Javascript

ORDS provides a JavaScript as a service framework for customers to define a JavaScript that can be executed in the ORDS instance on request. This is similar to the conventional RESTful

services concept used to develop the applications. The framework is based on the module, template, and handler architecture. See [Developing Oracle REST Data Services Applications](#). Rather than defining the modules, templates, and handlers in the database, they are specified in an XML representation that is read from `lib/ext/` directory as a plugin.

The ORDS examples directory contains a `plugin-javascript` example and the source can be found in the `examples/plugins/plugin-javascript` directory. This section describes the key elements of the plugin.



Note:

GraalVM with JS component is required for JavaScript plugin ORDS feature to work.

GraalVM with JS component is required for this ORDS feature to work. See GraalVM Configuration for more information.

The example contains a number of inline and external definitions for JavaScript source. References to external JavaScript source are to the files that are found in the classpath.

File	Description
<code>build.xml</code>	The ant build project.
<code>src/js/example.js</code>	An example external JavaScript file. External here means, not defined in, but referred to from, the XML Resource Module file.
<code>src/META-INF/manifest.json</code>	A plugin configuration metadata file that ORDS reads at startup to register XML Resource Modules.
<code>src/META-ING/modules/javascript.xml</code>	An XML Resource Module file that defines an example module with a number of templates and handlers.

Perform the following steps to build and use the example:

1. Change the directory to `examples/plugins/plugin-javascript`.
2. Run `ant` to build `examples/plugins/plugin-javascript/built/plugin-javascript.jar` file.
3. Copy the `plugin-javascript.jar` file to the ORDS distribution `lib/ext` directory and start the ORDS instance using a supported GraalVM with JS component.
4. Invoke the defined handlers using the URL pattern: `http://server/ords/javascript-examples/{template pattern}`.
 - a. For example: `http://localhost:8080/ords/javascript-examples/now` where the current time is returned.



Note:

Unlike the ORDS REST Services, the JavaScript as a service implementation does not require or use a database connection.

11.2.1 Example Services Purpose and Use

This section provides the information on the purpose and use of the example services.

Purpose	Request	Action	Response
An example of inline Javascript that returns the current UTC time as application/json.	/ords/javascript-examples/now	GET	{ "now": "2023-08-31T16:08:55.471Z" }
An example of inline Javascript that accepts a parameter.	/ords/javascript-examples/future?days=7	GET	{ "now": "2023-08-31T16:08:55.471Z", "future": "2023-09-07T16:08:55.471Z", "days": 7 }
An example of inline Javascript that accepts various parameters from different sources.	/ords/javascript-examples/hello?name=Ted	GET	Hello Ted Hello Test
	curl --location 'ords/javascript-examples/hello' \ --header 'Agent: Test'		
An example of external Javascript file that accepts a parameter.	/ords/javascript-examples/fibonacci?length=50	GET	{fib: 12586269025}
An example of inline Javascript that uses implicit parameters content_type and body_text for getting the request values as well as using ords_response to invoke setStatus and setContentType on HttpServletResponse	curl --location '/ords/hr/javascript-examples/countwords' \ --header 'Content-Type: application/json' \ --data '{"text": "How many words are here?"}'	POST	{"text": "How many words are here?","count": 5}

12

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](#)
- [Oracle REST Data Services Functionality](#)
- [ORDS Features](#)
- [Modifying Synonyms](#)

12.1 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.

12.2 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 12-1 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
  PlsqlDatabaseConnectionString 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
  PlsqlDatabaseConnectionString 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
  PlsqlDatabaseConnectionString 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>
```


12.3 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 12-1 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.
PlsqlDatabaseConnectionString	Multiple Settings such as: <ul style="list-style-type: none"> db.hostname db.port db.servicename db.sid 	Specifies the connection to an Oracle database. ORDS and mod_plsql are equivalent.
PlsqlAuthenticationMode	security.requestAuthenticationFunction	Specifies the authentication mode to use to allow access. <p>When <code>security.requestAuthenticationFunction</code> is not specified, ORDS behavior is same as Basic mode of mod_plsql.</p> <p>When <code>security.requestAuthenticationFunction</code> is specified, ORDS can perform the same action as example dad directive <code>PlsqlAuthenticationMode CustomOwaof mod_plsql</code>.</p> <p>Example ORDS equivalent configuration parameter:</p> <pre><entry key="security.requestAuthenticationFunction">privileged_user.owa_custom.authorize</entry></pre> <p>ORDS and mod_plsql are equivalent.</p>

Table 12-1 (Cont.) Mappings of mod_plsql Directives to ORDS Settings

mod_plsql Setting	ORDS Setting	Description
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.
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.

Table 12-1 (Cont.) Mappings of mod_plsql Directives to ORDS Settings

mod_plsql Setting	ORDS Setting	Description
PlsqlExclusionList	security.exclusionList	Specifies a pattern for procedures, packages, or schema names which are forbidden to be directly run from a browser. ORDS and mod_plsql are equivalent. See Understanding Configurable Parameters.
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.
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: <code>dbms_session.modify_package_state(dbms_session.reinitialize)</code> 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.

Table 12-1 (Cont.) Mappings of mod_plsql Directives to ORDS Settings

mod_plsql Setting	ORDS Setting	Description
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 <code>owa_util.get_page</code> or <code>owa_util.get_page_raw</code> . ORDS materializes results as a 32K VARCHAR or CLOB if results are greater than 32K, so not applicable.
PlsqlNLSLanguage	N/A	Specifies the NLS_LANG variable. ORDS, Java, and JDBC use unicode.
PlsqlTransferMode	N/A	<code>PlsqlTransferMode</code> 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.

Table 12-1 (Cont.) Mappings of mod_plsql Directives to ORDS Settings

mod_plsql Setting	ORDS Setting	Description
PlsqlPathAliasProcedure	N/A	ORDS has no equivalent.
PlsqlUploadAsLongRaw	N/A	ORDS has no equivalent.

12.4 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](#)

12.4.1 Example Configuration File for Basic Authentication

Example 12-2 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.validationF
unc</entry>
  <entry key="owa.docTable">sample_plsql_app.doc_table</entry>
</properties>
```

12.4.2 Example Configuration File for Basic Dynamic Authentication

Example 12-3 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">!passwordFOR$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.validationF
unc</entry>
</properties>

```

12.4.3 Example Configuration file for Custom Authentication

Example 12-4 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">!passwordFOR$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.validationF
unc</entry>
    <entry
key="security.requestAuthenticationFunction">privileged_user.owa_custom.author
ize</entry>
</properties>

```

12.5 Example ORDS URL Mapping

This section shows the example mapping between base-path url and the configuration files.

Example 12-5 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>

```

12.6 Example ORDS Default Configuration

This section shows the example default configuration setting for ORDS.

The defaults.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 12-6 ords_conf/ords/defaults.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>
```

12.7 Oracle REST Data Services Functionality

Oracle REST Data Services is a J2EE-based servlet which offers increased functionality including a web-based configuration, enhanced security, and file caching.

The following sections explain the different functionalities equivalent to the `mod_plsql` module.

12.7.1 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.

12.7.2 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 the 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.

12.7.3 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.authorize  
</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 ;  
/
```

12.8 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](#)
- [Cross-Origin Resource Sharing Feature](#)
- [Procedure Allow List](#)

12.8.1 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 12-7 security.requestValidationFunction

```
<entry
key="security.requestValidationFunction">sample_plsql_app_metadata.validationF
unc</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.
```

12.8.2 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 12-8 procedure.preProcess

Following example code snippet shows a use case for logging in:

```
<entry key="procedure.preProcess">sample_plsql_app_metadata.beforeProc</entry>
```

12.8.3 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 12-9 procedure.postProcess

Following example code snippet shows a use case for logging out:

```
<entry key="procedure.postProcess">sample_plsql_app_metadata.afterProc</entry>
```

12.8.4 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 12-10 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 12-11 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 12-12 Curl command for file upload

```

curl -i -X POST -F 'filename=@helloworld.txt' "http://localhost:8086/ords/pls/
basic_auth/example_user1.upload

```

12.8.5 Cross-Origin Resource Sharing Feature

This section describes the Cross-Origin Resource Sharing (CORS) feature.

By default ORDS does not allow cross-origin calls to its PL/SQL gateway.

Trusted origins can be configured through the `security.externalSessionTrustedOrigins` configuration parameter that defines a comma separated list of origins that are trusted to make CORS request. If this parameter is empty or not configured, then no CORS requests are allowed for the PL/SQL gateway and results in a 403 Unauthorized status.

```

<entry key="security.externalSessionTrustedOrigins">http://example.com, https://
example.com:8443</entry>

```

12.8.6 Procedure Allow List

This section describes the Allow List feature for PL/SQL Gateway procedures.

Oracle REST Data Services (ORDS) provides an Allow List feature for PL/SQL Gateway procedures. You can authorize execution of custom procedures by adding them to the Allow List.

12.8.6.1 Configuring ORDS PL/SQL Gateway Allow List

This section describes how to configure the ORDS PL/SQL Gateway Allow List.

Ensure that your PL/SQL Gateway pool is configured to use the ORDS validation function named `ords_util.authorize_plsql_gateway`.

```

./ords config --db-pool <plsql_pool> get security.requestValidationFunction

```

If PL/SQL Gateway pool is empty, then it is setup using the following command:

```
./ords config --db-pool <plsql_pool> set security.requestValidationFunction
ords_util.authorize_plsql_gateway
```

 **Note:**

To use the feature ORDS PL/SQL Gateway Allow list with APEX, the APEX must be installed in a Pluggable Database.

12.8.6.1.1 Authorizing Procedures

This section describes how to authorize the stored procedures.

To authorize the stored procedures you must add them to the PL/SQL Gateway Allow List using the `ords_admin.add_plsql_gateway_procedure` procedure. You are required to have `ORDS_ADMINISTRATOR_ROLE` role to execute the procedure.

```
BEGIN
    ords_admin.add_plsql_gateway_procedure(
        p_owner => 'MY_SCHEMA',
        p_package_name => 'MY_PACKAGE', /* Can be null if not a
package procedure*/
        p_procedure_name => 'MY_STORED_PROCEDURE',
        p_comments => 'Enabling access to Project 1'); /*
Optional comments*/
END;
```

The Allow List stores resolved procedure names. Procedures are resolved before adding them to the list.

12.8.6.1.2 Removing Stored Procedures

This section describes how to remove the stored procedures from the PL/SQL Gateway Allow List.

To remove the stored procedures from the PL/SQL Gateway Allow List use `ords_admin.remove_plsql_gateway_procedure` procedure. You are required to have `ORDS_ADMINISTRATOR_ROLE` role to execute the procedure.

```
BEGIN
    ords_admin.remove_plsql_gateway_procedure(
        p_owner => 'MY_SCHEMA', /* Schema owning the
stored procedure */
        p_package_name => 'MY_PACKAGE', /* Can be null if not
a package procedure*/
        p_procedure_name => 'MY_STORED_PROCEDURE');
END;
```

12.8.6.1.3 Removing Stored Procedures in Bulk

This section describes how to remove the stored procedures in bulk from the PL/SQL Gateway Allow List.

To remove the stored procedures from the PL/SQL Gateway Allow List use `ords_admin.clear_plsql_gateway_procedures` procedure. You are required to have `ORDS_ADMINISTRATOR_ROLE` role to execute the procedure.

```
BEGIN
    ords_admin.clear_plsql_gateway_procedures(
        p_owner => 'MY_SCHEMA'); /* Remove all
procedures owned by this schema. */
END;
```

12.9 Modifying Synonyms

When you are invoking synonyms through PL/SQL Gateway, ORDS executes the procedure they point to. If an existing synonym is redefined to point to a second procedure, then revoke the EXECUTE privilege from the first procedure to force the synonym to reload and ensure that ORDS executes the second procedure.

A

Setting-up a PL/SQL Gateway User

This section explains how to set-up a PL/SQL gateway user.

To set-up a PL/SQL gateway user, perform the following steps:

1. Unzip the ords*.zip file.
2. Execute the script that provides the password:
Example:

```
SQL> @install <password>
```

```
install.sql  
set define '^'  
set termout on
```

```
define PWD          = '^1'
```

```
-- Create the schema to hold the stored proc. This account is not directly  
accessible  
create user sample_plsql_app identified by L0ck3dAcc0unt password expire  
account lock;
```

```
-- create the application users  
create user example_user1 identified by ^PWD;  
create user example_user2 identified by ^PWD;  
grant connect to example_user1;  
grant connect to example_user2;
```

```
alter session set current_schema=sample_plsql_app;
```

```
-- define the stored procedure  
create or replace procedure sample_proc as  
  l_user varchar(255) := owa_util.get_cgi_env('REMOTE_USER');  
begin  
  http.prn('<h1>Hello ' || l_user || '!</h1>');  
end;  
/
```

```
-- authorize users to access stored proc  
grant execute on sample_plsql_app.sample_proc to example_user1;  
grant execute on sample_plsql_app.sample_proc to example_user2;
```

```
quit
```

Preceding sample creates three database users:

- SAMPLE_PLSQL_APP - A database schema where the protected SAMPLE_PROC is installed
- EXAMPLE_USER1 - A database user granted with execute privilege on SAMPLE_PLSQL_APP.SAMPLE_PROC

- `EXAMPLE_USER2` - A second database user granted with execute privilege on `SAMPLE_PLSQL_APP.SAMPLE_PROC`

Use the non-interactive install command and include the options `--gateway-user <database user>` and `--gateway-mode proxied` to indicate that the PL/SQL gateway user is a proxied user.

Configuring a PL/SQL Gateway User

Non-Interactive Example:

```
./bin/ords --config /path/to/test/config install-cli --db-pool pdb2 --admin-user  
SYS --proxy-user --db-hostname localhost --db-port 1521 --db-servicename pdb1 --  
feature-sdw true --gateway-user EXAMPLE_USER1 --gateway-mode proxied --log-  
folder /path/to/logs < password.txt
```

Assuming ORDS is running in a standalone mode on localhost on port 8080, access the following URL in your web browser: `http://localhost:8080/ords/pdb2/sample_plsql_app.sample_proc`. The browser displays the following text

Hello EXAMPLE_USER1!

B

Oracle REST Data Services Database Type Mappings

This appendix describes the REST Data Services database type mappings along with the structural database types.

B.1 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" : "SEVMTE8gV09STE Qh"	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" : "AAAGq9AAEAAAA0 bAAA"	Output as the native Oracle textual representation. For example, equivalent to the following conversion: SELECT ROWIDTOCHAR(id) id FROM DUAL.
UROWID	string	Custom	"uid" : "AAAGq9AAEAAAA0 bAAA"	Output as the native Oracle textual representation. For example, equivalent to the following conversion: SELECT CAST(uid as VARCHAR(4000)) id FROM DUAL.
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.

Data Type	JSON Data Type	REST Version	Value Example	Description
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" : "1995-06-02T04:29:11.002Z" "when" : "1995-06-02T04:29:11.002Z"	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" : { "SEVMTE8gV09S TEQh"	
BLOB	string	Custom	"bin" : { "base64_value" : "bGVhc3VyZS4="	
CLOB	string	Custom	"text" : { "value" : "Hello World!" " }	

Data Type	JSON Data Type	REST Version	Value Example	Description
BFILE	Object	Custom	<pre>"file" : { "locator" : "TARGET_DIR", "filename" : "myfile" }</pre>	
BOOLEAN	true false	v1	<pre>"right" : true "wrong" : false</pre>	

B.2 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.

B.3 Oracle Geospatial Encoding

Oracle Geospatial 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](#)

B.4 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>
```

C

Troubleshooting Oracle REST Data Services

This appendix contains information on troubleshooting Oracle REST Data Services.

Topics:

- [Enabling Detailed Request Error Messages](#)
- [Configuring Oracle APEX Static Resources with Oracle REST Data Services](#)

C.1 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 production systems due to the risk of sensitive information being revealed to an attacker.

C.2 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	>curl -i
200	-X GET --
BEGIN	user
	DEMO:demo
ORDS.defi	-k
ne_servic	http://
e(localhost
	:8082/
p_module_	ords/
name	demo/
=>	test1/ok/
'test1',	HTTP/1.1
	200 OK
p_base_pa	Date:
th	Thu, 19
=>	Mar 2020
'test1/',	17:18:05
	GMT
p_pattern	Content-
	Type:
=> 'ok/',	applicati
	on/json
p_method	ETag:
=> 'GET',	"BLNTmyd/
	ZM889Q0G1
	gJ1t7lkSY
p_source_	o2kpAVIv4
type	CY5dvtp9N
=>	I/
ORDS.sour	Em1DJRzpm
ce_type_c	E5Bg/
ollection	4GiKifewt
_feed,	zuJA6i+YC
	gdxETWWQ=
p_source	="
=>	Transfer-
'SELECT	Encoding:
* FROM	chunked
dual',	
p_items_p	
er_page	
=> 0);	
COMMIT;	
END;	
/	

Table C-1 (Cont.) List of ORDS user defined service

Service	Response
--HTTP 200 , p_source_ type => ORDS.sour ce_type_c ollection _feed, BEGIN	>curl -- head -i -X GET --user DEMO:demo -k http:// localhost :8082/ ords/ demo/ test2/ norows/ HTTP/1.1
p_module_ name => 'test2',	200 OK Date: Thu, 19 Mar 2020 17:18:28
p_base_pa th => 'test2/',	GMT Content- Type: applicati on/json
p_pattern => 'norows/' ,	ETag: "aZVsHTwe wrbbkl6wH NcTa3RFFd EsbdtDRBT
p_method => 'GET',	S1R93r/ vBmDvVsgu d2rFqLDI6 5UKxzSEln
p_source_ type => ORDS.sour ce_type_c ollection _feed,	AAMQdlBj/ sB9ywWqQ= ="
p_source => 'SELECT * FROM dual where 1 = 2', p_items_p er_page	Transfer- Encoding: chunked

Table C-1 (Cont.) List of ORDS user defined service

Service	Response
	<pre>=> 0); COMMIT; END; /</pre>

Table C-1 (Cont.) List of ORDS user defined service

Service	Response
create table no_rows (coll int);	>curl --head -i -X GET --user DEMO:demo --HTTP -k http://localhost:8082/ords/demo/test2b/norows/HTTP/1.1
200 ,	200 OK
p_source_type => ORDS.source_type_collection_feed,	Date: Thu, 19 Mar 2020 17:18:34 GMT
BEGIN	Content-Type: application/json
ORDS.define_service('test2b',	ETag: "Ns/g/hFxVWYPHUYZT53HN16EMV1QUXD5
p_module_name => 'test2b',	wmz3eoO15dlY6nSVkk
	2FX3sNw3Yvq87SdLYA1CLeuqb4N
p_base_path => 'test2b/'	4DQrcy+0Q=="
,	Transfer-Encoding: chunked
p_pattern => 'norows/'	
,	
p_method => 'GET',	
p_source_type => ORDS.source_type_collection_feed,	
p_source => 'SELECT	

Table C-1 (Cont.) List of ORDS user defined service

Service	Response
<pre>* FROM no_rows', p_items_p er_page => 0); COMMIT; END; /</pre>	

Table C-1 (Cont.) List of ORDS user defined service

Service	Response
--HTTP 404 , p_source_ type => ORDS.sour ce_type_c ollection _item, BEGIN	>curl -- head -i -X GET --user DEMO:demo -k http:// localhost :8082/ ords/ demo/ test2c/ norows/ HTTP/1.1
p_module_ name => 'test2c',	404 Not Found Content- Type: text/html
p_base_pa th => 'test2c/'	Content- Length: 16127
,	
p_pattern => 'norows/'	
,	
p_method => 'GET',	
p_source_ type => ORDS.sour ce_type_c ollection _item,	
p_source => 'SELECT * FROM dual where 1 = 2',	
p_items_p	

Table C-1 (Cont.) List of ORDS user defined service

Service	Response
er_page	<pre>=> 0); COMMIT; END; /</pre>

Table C-1 (Cont.) List of ORDS user defined service

Service	Response
--HTTP 404 BEGIN ORDS.defi ne_servic e(p_module_ name => 'test3', p_base_pa th => 'test3/', p_pattern => 'doesnote xist/', p_method => 'GET', p_source_ type => ORDS.sour ce_type_c ollection _feed, p_source => 'SELECT 10 as A FROM doesnotex ist', p_items_p er_page => 0); COMMIT; END; /	>curl -- head -i -X GET --user DEMO:demo -k http:// localhost :8082/ ords/ demo/ test3/ doesnotex ist/ HTTP/1.1 403 Forbidden Content- Type: text/html Error- Reason: error="mi ssing.obj ect"; error_des cription* =UTF-8' 'The request could not be processed because a table or view reference d by the SQL statement being evaluated is not accessibl e or does not exist Content-

Table C-1 (Cont.) List of ORDS user defined service

Service	Response
	Length: 16327

Table C-1 (Cont.) List of ORDS user defined service

Service	Response
--HTTP	>curl --
555	head -i
BEGIN	-X GET
	--user
ORDS.defi	DEMO:demo
ne_servic	-k
e(http://
	localhost
p_module_	:8082/
name =>	ords/
'test4',	demo/
	test4/
p_base_pa	badsyntax
th =>	/
'test4/',	HTTP/1.1
	500
	Server
p_pattern	Error
=>	Content-
'badsynta	Type:
x/',	text/html
	Error-
p_method	Reason:
=>	error="re
'GET',	source.ge
	nerator.e
p_source_	valuation
type =>	";
ORDS.sour	error_des
ce_type_c	cription*
ollection	=UTF-8'
_feed,	'The
	request
p_source	could
=>	not be
'SELECT	processed
10',	because
	an error
p_items_p	occurred
er_page	whilst
=> 0);	attemptin
	g
COMMIT;	to
END;	evaluate
/	the SQL
	statement
	associate
	d with

Table C-1 (Cont.) List of ORDS user defined service

Service	Response
	<p>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</p>

C.3 Configuring Oracle APEX 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 APEX page, for example, when attempting to display `https://example/ords/`. This problem is caused by an improper configuration of Oracle APEX Express static resources, which causes the JavaScript and CSS resources required by APEX not to be found and the APEX page not to render correctly.

The specific cause can be any of the following:

- Forgetting to ensure that the APEX static images are located on the same server as the Oracle REST Data Services instance
- Forgetting to deploy a web application for the static APEX images to Apache Tomcat or WebLogic Server.
- 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 `--apex-images` option
- Upgrading to a new version of APEX 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 APEX 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.

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 APEX 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 APEX static resources to match the APEX 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.



See Also:

Configuring Oracle Application Express Images

D

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.

D.1 Hack 3.003

The work in the Hack project is Copyright 2018 Source Foundry Authors and licensed under the MIT License

The work in the DejaVu project was committed to the public domain.

Bitstream Vera Sans Mono Copyright 2003 Bitstream Inc. and licensed under the Bitstream Vera License with Reserved Font Names "Bitstream" and "Vera"

MIT License

Copyright (c) 2018 Source Foundry Authors

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.

BITSTREAM VERA LICENSE

Copyright (c) 2003 by Bitstream, Inc. All Rights Reserved. Bitstream Vera is a trademark of Bitstream, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of the fonts accompanying this license ("Fonts") and associated documentation files (the "Font Software"), to reproduce and distribute the Font Software, including without

limitation the rights to use, copy, merge, publish, distribute, and/or sell copies of the Font Software, and to permit persons to whom the Font Software is furnished to do so, subject to the following conditions:

The above copyright and trademark notices and this permission notice shall be included in all copies of one or more of the Font Software typefaces.

The Font Software may be modified, altered, or added to, and in particular the designs of glyphs or characters in the Fonts may be modified and additional glyphs or characters may be added to the Fonts, only if the fonts are renamed to names not containing either the words "Bitstream" or the word "Vera".

This License becomes null and void to the extent applicable to Fonts or Font Software that has been modified and is distributed under the "Bitstream Vera" names.

The Font Software may be sold as part of a larger software package but no copy of one or more of the Font Software typefaces may be sold by itself.

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL BITSTREAM OR THE GNOME FOUNDATION BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

Except as contained in this notice, the names of Gnome, the Gnome Foundation, and Bitstream Inc., shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Font Software without prior written authorization from the Gnome Foundation or Bitstream Inc., respectively. For further information, contact: fonts at gnome dot org.

D.2 Monaco Editor 0.44.0

The MIT License (MIT)

Copyright (c) 2016 - present Microsoft Corporation

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.

D.3 babel-polyfill 7.20.15

<https://github.com/babel/babel/blob/main/LICENSE>

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.

Note - The component has NO 4th party components

D.4 gridstack.js 10.1.0

MIT License

== Copyright

Copyright (c) 2019-2023 Alain Dumesny. v0.4.0 and older (c) 2014-2018 Pavel Reznikov, Dylan Weiss

Copyright (c) 2021-2022 Alain Dumesny

Copyright (c) 2022 Alain Dumesny - see GridStack root license

----- Licenses

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.

D.5 d3-flame-graph 4.1.3

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.

4th party Dependencies


```

-----
LIBRARY NAME      LICENSE (see license text below)
-----
d3-array          ISC License
internmap         ISC License
d3-dispatch       ISC License
d3-ease           BSD-3-Clause
d3-format         ISC License
d3-hierarchy      ISC License
d3-scale          ISC License
d3-interpolate    ISC License
d3-color          ISC License
d3-time-format    ISC License
d3-time           ISC License
d3-selection      ISC License
d3-transition     ISC License
d3-timer          ISC License
-----

```

d3-array:

Copyright 2010-2021 Mike Bostock

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.

internmap:

Copyright 2021 Mike Bostock

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.

d3-dispatch:

Copyright 2010-2021 Mike Bostock

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.

d3-ease:

Copyright 2010-2021 Mike Bostock

Copyright 2001 Robert Penner

All rights reserved.

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.
- * Neither the name of the author nor the names of 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.

d3-format:

Copyright 2010-2021 Mike Bostock

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.

d3-hierarchy:

Copyright 2010-2021 Mike Bostock

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.

d3-scale:

Copyright 2010-2021 Mike Bostock

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.

d3-interpolate:

Copyright 2010-2021 Mike Bostock

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.

d3-color:

Copyright 2010-2021 Mike Bostock

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.

d3-time-format:

Copyright 2010-2021 Mike Bostock

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.

d3-time:

Copyright 2010-2021 Mike Bostock

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.

d3-selection:

Copyright 2010-2021 Mike Bostock

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.

d3-transition:

Copyright 2010-2021 Mike Bostock

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.

d3-timer:

Copyright 2010-2021 Mike Bostock

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.

D.6 Dexie 3.2.4

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.

D.7 Jetty 10.0.20

MAIN component is licensed under EPL and Apache 2.0 License

LICENSE for MAIN component

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 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.

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.

SPDX-License-Identifier: EPL-2.0 OR Apache-2.0

END OF LICENSE for MAIN component

NOTICE.txt for MAIN component

Notices for Eclipse Jetty

=====

This content is produced and maintained by the Eclipse Jetty project.

Project home: <https://www.eclipse.org/jetty/>

Trademarks

Eclipse Jetty, and Jetty are trademarks of the Eclipse Foundation.

Copyright

All contributions are the property of the respective authors or of entities to which copyright has been assigned by the authors (eg. employer).

Declared Project Licenses

This artifacts of this project are made available under the terms of:

- * the Eclipse Public License v. 1.0
<http://www.eclipse.org/legal/epl-v10.html>
SPDX-License-Identifier: EPL-1.0

or

- * the Apache License, Version 2.0
<https://www.apache.org/licenses/LICENSE-2.0>
SPDX-License-Identifier: Apache-2.0

The following dependencies are EPL.

- * org.eclipse.jetty.orbit:org.eclipse.jdt.core

If ALPN is used to negotiate HTTP/2 connections, then the following distribution may be included in the distribution or downloaded when ALPN module is selected. These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
<http://openjdk.java.net/legal/gplv2+ce.html>

- * java.sun.security.ssl

The following dependencies 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

The following dependencies are ASL2 licensed.

- * org.apache.taglibs:taglibs-standard-spec
- * org.apache.taglibs:taglibs-standard-impl

The following dependencies 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
```

The following artifacts are CDDL + GPLv2 with classpath exception.
<https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html>

```
* org.eclipse.jetty.toolchain:jetty-schemas
```

Cryptography

Content may contain encryption software. The country in which you are currently may have restrictions on the import, possession, and use, and/or re-export to another country, of encryption software. BEFORE using any encryption software, please check the country's laws, regulations and policies concerning the import, possession, or use, and re-export of encryption software, to see if this is permitted.

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.

```
##### END OF NOTICE.txt for MAIN component #####
```

```
#####License for 4th party component ASM #####
#####License for 4th party component ASM-COOMON #####
```

<https://asm.ow2.io/license.html>

License

ASM is released under the following 3-Clause BSD 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.

```
#####License.txt of 4th party component SLF4J-API#####
```

Copyright (c) 2004-2022 QOS.ch Sarl (Switzerland)
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.

=====

D.8 jackson-core 2.16.1

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 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.

Copyright

Copyright 2007-, Tatu Saloranta (tatu.saloranta@iki.fi)

Licensing

Jackson 2.x core and extension components are licensed under Apache License 2.0
To find the details that apply to this artifact see the accompanying LICENSE file.

Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

FastDoubleParser

jackson-core bundles a shaded copy of FastDoubleParser <<https://github.com/wrandelshofer/FastDoubleParser>>. That code is available under an MIT license <<https://github.com/wrandelshofer/FastDoubleParser/blob/main/LICENSE>> under the following copyright.

Copyright © 2023 Werner Randelshofer, Switzerland. MIT License.

See FastDoubleParser-NOTICE for details of other source code included in FastDoubleParser and the licenses and copyrights that apply to that code.

== FastDoubleParser License

MIT License

Copyright (c) 2023 Werner Randelshofer, Switzerland.

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.

== FastDoubleParser Notice

FastDoubleParser

This is a Java port of Daniel Lemire's fast_float project. This project provides parsers for double, float, BigDecimal and BigInteger values.

Copyright

Copyright © 2023 Werner Randelshofer, Switzerland.

Licensing

This code is licensed under MIT License.
<https://github.com/wrandelshofer/FastDoubleParser/blob/522be16e145f43308c43b23094e31d5efcaa580e/LICENSE>
 (The file 'LICENSE' is included in the sources and classes Jar files that are released by this project
 - as is required by that license.)

Some portions of the code have been derived from other projects.

All these projects require that we include a copyright notice, and some require that we also include some text of their license file.

fast_double_parser, Copyright (c) 2022 Daniel Lemire. BSL License.
https://github.com/lemire/fast_double_parser
https://github.com/lemire/fast_double_parser/blob/07d9189a8fb815fe800cb15ca022e7a07093236e/LICENSE.BSL
 (The file 'thirdparty-LICENSE' is included in the sources and classes Jar files that are released by this project
 - as is required by that license.)

fast_float, Copyright (c) 2021 The fast_float authors. MIT License.
https://github.com/fastfloat/fast_float
https://github.com/fastfloat/fast_float/blob/ccl1e01e9eee74128e48d51488a6b1df4a767a810/LICENSE-MIT
 (The file 'thirdparty-LICENSE' is included in the sources and classes Jar files that are released by this project
 - as is required by that license.)

bigint, Copyright 2020 Tim Bukt. 2-clause BSD License.
<https://github.com/tbukt/bigint/tree/floatfft>
<https://github.com/tbukt/bigint/blob/617c8cd8a7c5e4fb4d919c6a4d11e2586107f029/LICENSE>
<https://github.com/wrandelshofer/FastDoubleParser/blob/39e123b15b71f29a38a087d16a0bc620fc879aa6/bigint-LICENSE>
 (We only use those portions of the bigint project that can be licensed under 2-clause BSD License.)
 (The file 'thirdparty-LICENSE' is included in the sources and classes Jar files that are released by this project
 - as is required by that license.)

D.9 Jakarta Activation API (JAF) 2.1.2

Copyright (c) 2018 Oracle and/or its affiliates. All rights reserved.

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.
- Neither the name of the Eclipse Foundation, Inc. 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.

=====

Notices for Jakarta Activation

This content is produced and maintained by the Jakarta Activation project.

* Project home: <https://projects.eclipse.org/projects/ee4j.jaf>

Trademarks

Jakarta Activation is a trademark of the Eclipse Foundation.

Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at <https://www.eclipse.org/legal/epl-2.0>, or the Eclipse Distribution License v1.0 which is available at <https://www.eclipse.org/org/documents/edl-v10.php>. 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: (secondary) GPL-2.0 with Classpath-exception-2.0 which is available at <https://openjdk.java.net/legal/gplv2+ce.html>.

SPDX-License-Identifier: EPL-2.0 OR BSD-3-Clause OR GPL-2.0-only with Classpath-exception-2.0

Source Code

The project maintains the following source code repositories:

- * <https://github.com/jakartaee/jaf-api>
- * <https://github.com/jakartaee/jaf-tck>

Third-party Content

This project leverages the following third party content.

Apache Ant (1.9.6)

- * License: Apache License, 2.0, W3C License, Public Domain

Apache Ant (1.9.6)

```
* License: Apache License, 2.0, W3C License, Public Domain

Apache commons-lang (3.5)

* License: Apache-2.0

font-awesome (4.7.0)

* License: OFL-1.1 AND MIT

jsoup (1.10.2)

* License: MIT

JTHarness (5.0)

* License: (GPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0)
* Project: https://wiki.openjdk.java.net/display/CodeTools/JT+Harness
* Source: http://hg.openjdk.java.net/code-tools/jtharness/

JUnit (4.12)

* License: Eclipse Public License

normalize.css (3.0.2)

* License: MIT
* Project: http://necolas.github.io/normalize.css/
* Source: http://necolas.github.io/normalize.css/

SigTest (4.0)

* License: GPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0
* Project: https://wiki.openjdk.java.net/display/CodeTools/sigtest
* Source: http://hg.openjdk.java.net/code-tools/sigtest/file/c57f97e2ac2f

## Cryptography

Content may contain encryption software. The country in which you are
currently
may have restrictions on the import, possession, and use, and/or re-export to
another country, of encryption software. BEFORE using any encryption software,
please check the country's laws, regulations and policies concerning the
import,
possession, or use, and re-export of encryption software, to see if this is
permitted.
```

D.10 jackson-annotations 2.16.1

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 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.

Copyright

Copyright 2007-, Tatu Saloranta (tatu.saloranta@iki.fi)

Licensing

Jackson 2.x core and extension components are licensed under Apache License 2.0
To find the details that apply to this artifact see the accompanying LICENSE file.

Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included

in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

D.11 jackson-databind 2.16.1

Fourth parties -

```
com.fasterxml.jackson.core:jackson-annotations:2.16.1
com.fasterxml.jackson.core:jackson-core:2.16.1
```

```
----- Top-level license -----
https://github.com/FasterXML/jackson-databind/blob/2.16/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.

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.

----- Copyright notices -----

<https://github.com/FasterXML/jackson-databind/blob/2.16/src/main/resources/META-INF/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.

Copyright

Copyright 2007-, Tatu Saloranta (tatu.saloranta@iki.fi)

Licensing

Jackson 2.x core and extension components are licensed under Apache License 2.0
To find the details that apply to this artifact see the accompanying LICENSE file.

Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

----- Fourth-party information -----

== jackson-annotations 2.16.1
<https://github.com/FasterXML/jackson-annotations>
 == Apache-2.0 license
 == Copyright Notices
<https://github.com/FasterXML/jackson-annotations/blob/jackson-annotations-2.16.1/src/main/resources/META-INF/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.

## Copyright

Copyright 2007-, Tatu Saloranta (tatu.saloranta@iki.fi)

## Licensing

Jackson 2.x core and extension components are licensed under Apache License
2.0
To find the details that apply to this artifact see the accompanying LICENSE
file.

## Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.

----- (separator) -----

== jackson-core 2.16.1
   https://github.com/FasterXML/jackson-core
== Apache-2.0 license
== Copyright Notices
   https://github.com/FasterXML/jackson-core/blob/jackson-core-2.16.1/src/
   main/resources/META-INF/jackson-core-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.

## Copyright

Copyright 2007-, Tatu Saloranta (tatu.saloranta@iki.fi)

## Licensing

Jackson 2.x core and extension components are licensed under Apache License
2.0
To find the details that apply to this artifact see the accompanying LICENSE
file.

## Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.

## FastDoubleParser

jackson-core bundles a shaded copy of FastDoubleParser <https://github.com/
```

wrandelshofer/FastDoubleParser>.
That code is available under an MIT license <<https://github.com/wrandelshofer/FastDoubleParser/blob/main/LICENSE>>
under the following copyright.

Copyright © 2023 Werner Randelshofer, Switzerland. MIT License.

See FastDoubleParser-NOTICE for details of other source code included in
FastDoubleParser
and the licenses and copyrights that apply to that code.

----- (separator) -----

```
== FastDoubleParser 1.0.0
   https://github.com/wrandelshofer/FastDoubleParser
== MIT License
   https://github.com/wrandelshofer/FastDoubleParser/blob/v1.0.0/LICENSE
```

```
== Copyright Notices
   https://github.com/wrandelshofer/FastDoubleParser/blob/v1.0.0/NOTICE
```

FastDoubleParser

This is a Java port of Daniel Lemire's fast_float project.
This project provides parsers for double, float, BigDecimal and BigInteger
values.

Copyright

Copyright © 2023 Werner Randelshofer, Switzerland.

Licensing

MIT License

Copyright (c) 2023 Werner Randelshofer, Switzerland.

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.

fast_double_parser, Copyright (c) 2022 Daniel Lemire. BSL License.
https://github.com/lemire/fast_double_parser
Copyright (c) Daniel Lemire

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

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, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

fast_float, Copyright (c) 2021 The fast_float authors. MIT License.
https://github.com/fastfloat/fast_float
MIT License

Copyright (c) 2021 The fast_float authors

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 NON-INFRINGEMENT. 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.

bigint, Copyright 2020 Tim Bukt. 2-clause BSD License.
<https://github.com/tbukt/bigint/tree/floatfft>

Copyright 2022 Tim Bukt

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.

D.12 graphql-js 16.8.0

MIT License

Copyright (c) GraphQL 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.

D.13 avsc 5.7.3

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.

D.14 D3 7.8.4

Copyright 2010-2023 Mike Bostock

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.

Depends on commander, license follows
(The MIT License)

Copyright (c) 2011 TJ Holowaychuk <tj@vision-media.ca>

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.

Depends on d3-array, license follows
Copyright 2010-2023 Mike Bostock

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.

Depends on d3-axis, license follows
Copyright 2010-2021 Mike Bostock

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.

Depends on d3-brush, license follows
Copyright 2010-2021 Mike Bostock

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.

Depends on d3-chord, license follows
Copyright 2010-2021 Mike Bostock

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.

Depends on d3-color, license follows
Copyright 2010-2022 Mike Bostock

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.

Depends on d3-contour, license follows
Copyright 2012-2023 Mike Bostock

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.

Depends on d3-delaunay, license follows
Copyright 2018-2021 Observable, Inc.
Copyright 2021 Mapbox

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.

Depends on d3-dispatch, license follows
Copyright 2010-2021 Mike Bostock

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.

Depends on d3-drag, license follows
Copyright 2010-2021 Mike Bostock

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.

Depends on d3-dsv, license follows
Copyright 2013-2021 Mike Bostock

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.

Depends on d3-ease, license follows
Copyright 2010-2021 Mike Bostock
Copyright 2001 Robert Penner
All rights reserved.

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.
- * Neither the name of the author nor the names of 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.

Depends on d3-fetch, license follows
Copyright 2016-2021 Mike Bostock

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.

Depends on d3-force, license follows
Copyright 2010-2021 Mike Bostock

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.

Depends on d3-format, license follows
Copyright 2010-2021 Mike Bostock

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.

Depends on d3-geo, license follows
Copyright 2010-2022 Mike Bostock

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.

This license applies to GeographicLib.

Copyright 2008-2012 Charles Karney

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.

Depends on d3-hierarchy, license follows
Copyright 2010-2021 Mike Bostock

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.

Depends on d3-interpolate, license follows
Copyright 2010-2021 Mike Bostock

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.

Depends on d3-path, license follows
Copyright 2015-2022 Mike Bostock

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.

Depends on d3-polygon, license follows
Copyright 2010-2021 Mike Bostock

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.

Depends on d3-quadtrees, license follows
Copyright 2010-2021 Mike Bostock

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.

Depends on d3-random, license follows
Copyright 2010-2021 Mike Bostock

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.

Depends on d3-scale, license follows
Copyright 2010-2021 Mike Bostock

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.

Depends on d3-scale-chromatic, license follows
Copyright 2010-2021 Mike Bostock

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.

Apache-Style Software License for ColorBrewer software and ColorBrewer Color Schemes

Copyright 2002 Cynthia Brewer, Mark Harrower, and The Pennsylvania State University

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.

Depends on d3-selection, license follows
Copyright 2010-2021 Mike Bostock

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.

Depends on d3-shape, license follows
Copyright 2010-2022 Mike Bostock

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.

Depends on d3-time, license follows
Copyright 2010-2022 Mike Bostock

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.

Depends on d3-time-format, license follows
Copyright 2010-2021 Mike Bostock

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.

Depends on d3-timer, license follows
Copyright 2010-2021 Mike Bostock

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.

Depends on d3-transition, license follows
Copyright 2010-2021 Mike Bostock

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.

Depends on d3-zoom, license follows
Copyright 2010-2021 Mike Bostock

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.

Depends on delaunator, license follows
ISC License

Copyright (c) 2017, Mapbox

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.

Depends on iconv-lite, license follows
Copyright (c) 2011 Alexander Shtuchkin

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.

Depends on internmap, license follows
Copyright 2021 Mike Bostock

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.

Depends on robust-predicates, license follows
This is free and unencumbered software released into the public domain.

Anyone is free to copy, modify, publish, use, compile, sell, or distribute this software, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.

In jurisdictions that recognize copyright laws, the author or authors of this software dedicate any and all copyright interest in the

software to the public domain. We make this dedication for the benefit of the public at large and to the detriment of our heirs and successors. We intend this dedication to be an overt act of relinquishment in perpetuity of all present and future rights to this software under copyright law.

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 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.

For more information, please refer to <<http://unlicense.org>>

Depends on rw, license follows
Copyright (c) 2014-2016, Michael Bostock
All rights reserved.

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.
- * The name Michael Bostock may not 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 MICHAEL BOSTOCK 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.

Depends on safer-buffer, license follows
MIT License

Copyright (c) 2018 Nikita Skovoroda <chalkerx@gmail.com>

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.

D.15 long.js 5.2.0

```
/**
 * @license
 * Copyright 2009 The Closure Library Authors.
 * Copyright 2020 Daniel Wirtz / The long.js Contributors.
 * SPDX-License-Identifier: Apache-2.0
 */
```

<https://github.com/dcodeIO/long.js/blob/master/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.

D.16 SnappyJS 0.6.1

<https://github.com/zhipeng-jia/snappyjs/blob/master/LICENSE>

The MIT License (MIT)

Copyright (c) 2016 Zhipeng Jia

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.

D.17 JavaScript Extension Toolkit (JET) 14.0.0

Oracle JET 14.0.0

You may not use the identified files except in compliance with the Universal Permissive License (UPL), Version 1.0 (the "License.")

You may obtain a copy of the License at <https://opensource.org/licenses/UPL>. A copy of the license is also reproduced below.

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 (c) 2014, 2023 Oracle and/or its affiliates
The Universal Permissive License (UPL), Version 1.0

Subject to the condition set forth below, permission is hereby granted to any

person obtaining
a copy of this software, associated documentation and/or data (collectively
the "Software"),
free of charge and under any and all copyright rights in the Software, and
any and all patent
rights owned or freely licensable by each licensor hereunder covering either
(i) the unmodified
Software as contributed to or provided by such licensor, or (ii) the Larger
Works (as defined below),
to deal in both

(a) the Software, and (b) any piece of software and/or hardware listed in the
lrgwrks.txt file if
one is included with the Software (each a Larger Work to which the Software
is contributed by such licensors),
without restriction, including without limitation the rights to copy, create
derivative works of,
display, perform, and distribute the Software and make, use, sell, offer for
sale, import, export,
have made, and have sold the Software and the Larger Work(s), and to
sublicense the foregoing rights
on either these or other terms.

This license is subject to the following condition:

The above copyright notice and either this complete permission notice or at a
minimum a reference
to the UPL must 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.

DO NOT TRANSLATE OR LOCALIZE.

**
Oracle elects to use only the GNU Lesser General Public License version 2.1
(LGPL) for any software where a choice of LGPL/GPL license versions are made
available with the language indicating that LGPLv2.1/GPLv2 or any later
version
may be used, or where a choice of which version of the LGPL/GPL is applied is
unspecified.

**

```

THIRD-PARTY COMPONENT FILE
LICENSE
(path in the installation)          (see license text reproduced
below)
-----
--
js/libs/chai/
chai                                MIT
js/libs/hammer/hammer                MIT
js/libs/js-signals/signals.          MIT
js/libs/jquery/jquery
MIT
js/libs/jquery/jquery-ui.custom.     MIT
js/libs/jquery/jqueryui-amd/core     MIT
js/libs/jquery/jqueryui-amd/draggable MIT
js/libs/jquery/jqueryui-amd/mouse    MIT
js/libs/jquery/jqueryui-amd/position MIT
js/libs/jquery/jqueryui-amd/sortable MIT
js/libs/jquery/jqueryui-amd/widget   MIT
js/libs/knockout/knockout            MIT
js/libs/knockout/knockout-mapping-latest.js
MIT
js/libs/oj/v14.0.0/min/ojcspexpressionevaluator (cspexpressionevaluator.js)
MIT
js/libs/oj/v14.0.0/min/ojexpparser    MIT
js/libs/oj/v14.0.0/min/ojknockout     MIT
js/libs/oj/v14.0.0/min/ojselectcombobox. Apache 2.0
js/libs/oj/v14.0.0/ojtree             MIT
js/libs/oj/v14.0.0/ojL10n            MIT
js/libs/proj4js/dist/proj4
Proj4js
js/libs/require/require               MIT
js/libs/require/text                 MIT
js/libs/require-css/css.min          MIT
scss/oj/v14.0.0/3rdparty/normalize/normalize.scss MIT
js/libs/touchr/touchr.js
MIT
js/libs/preact/dist/preact.umd.js
MIT

```

Chai
<https://github.com/chaijs/chai>
 Copyright (c) 2017 Chai.js Assertion Library

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.

expression-eval
<https://github.com/donmccurdy/expression-eval>
Copyright (c) 2017 Don McCurdy

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.

jsep -
<https://github.com/soney/jsep>
Copyright (c) 2013 Stephen Oney, <https://ericmekens.github.io/jsep/>

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.

Knockout Fast Foreach

By: Brian M Hunt (C) 2015 | License: 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.

proj4js

<http://proj4js.org/>

Copyright (C) 2014 Mike Adair, Richard Greenwood, Didier Richard, Stephen Irons, Olivier Terral and Calvin Metcalf;
Licensed under the Proj4js license

require-css

<https://github.com/guybedford/require-css>

Copyright (C) 2013 Guy Bedford

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.

Hammer.JS
<http://hammerjs.github.io/>
Copyright (C) 2011-2017 by Jorik Tangelder (Eight Media)

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.

Foundation Responsive Library
<http://foundation.zurb.com>
Copyright 2014, ZURB

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.

Normalize.scss
Copyright © Nicolas Gallagher and Jonathan Neal

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.

RequireJS i18n

<http://github.com/requirejs/i18n> for details

Copyright (c) 2010-2011, The Dojo 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.

jsTree <http://jstree.com/>

Copyright (c) 2012 Ivan Bozhanov (<http://vakata.com>)

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.

select2.js

<https://github.com/select2/select2>
Copyright 2012 Igor Vaynberg
This software is licensed under the Apache License, Version 2.0 (the "Apache License") or the GNU General Public License version 2 (the "GPL License"). You may choose either license to govern your use of this software only upon the condition that you accept all of the terms of either the Apache License or the GPL License.

You may obtain a copy of the Apache License and the GPL License at:

<http://www.apache.org/licenses/LICENSE-2.0>
<http://www.gnu.org/licenses/gpl-2.0.html>

Unless required by applicable law or agreed to in writing, software distributed under the Apache License or the GPL License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the Apache License and the GPL License for the specific language governing permissions and limitations under the Apache License and the GPL License.

jQuery UI -
<http://jqueryui.com>
Includes: core.js, widget.js, mouse.js, position.js, draggable.js, sortable.js

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.

jQuery JavaScript Library
<http://jquery.com/>
Copyright OpenJS Foundation and other contributors, <https://openjsf.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 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.

JS Signals <<http://millermedeiros.github.com/js-signals/>>
Author: Miller Medeiros

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.

RequireJS text
<http://github.com/requirejs/text>
Copyright jQuery Foundation and other contributors, <https://jquery.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 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 NON-INFRINGEMENT. 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.

RequireJS

<http://github.com/jrburke/requirejs>

Copyright jQuery Foundation and other contributors, <https://jquery.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 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 NON-INFRINGEMENT. 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.

Knockout JavaScript library

Copyright (c) 2010 Steven Sanderson, the Knockout.js team, and other contributors

<http://knockoutjs.com/>

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 NON-INFRINGEMENT. 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.

Preact

The MIT License (MIT)
Copyright (c) 2015-present Jason Miller

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.

===== Apache-2.0 =====

The following applies to all products licensed under the Apache 2.0 License:

You may not use the identified files except in compliance with the Apache License, Version 2.0 (the "License.")

You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>.

A copy of the license is also reproduced below.

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.

===== MIT =====

The MIT License

Copyright (c) __YEARS__, __NAMES__

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.

===== Proj4js =====

Mike Adair madairATdmsolutions.ca
Richard Greenwood richATgreenwoodmap.com
Didier Richard didier.richardATign.fr
Stephen Irons stephen.ironsonclear.net.nz
Olivier Terral oterralATgmail.com
Calvin Metcalf cmetcalfATappgeo.com

Copyright (c) 2014, Mike Adair, Richard Greenwood, Didier Richard, Stephen Irons, Olivier Terral and Calvin Metcalf

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.

D.18 Commons FileUpload 1.5

Top Level Component : commons-fileupload
Top Level Component License : Apache 2.0
Top Level Component Copyright : Copyright

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:

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

Top Level Component Copyright : Copyright

Apache Commons FileUpload
Copyright 2002-2023 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<http://www.apache.org/>).

```
=====
Fourth Party Dependency #1 Name : commons-io
Fourth Party Dependency #1 License : Apache 2.0
Fourth Party Dependency #1 Copyright : Copyright
```

Apache Commons IO
Copyright 2002-2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

D.19 opentelemetry-java 1.24.0

OpenTelemetry Java (io.opentelemetry:*)

Copyright The OpenTelemetry Authors
Copyright 2000-2021 JetBrains s.r.o.
Copyright 2019 LINE Corporation
Copyright 2013-2020 The OpenZipkin Authors

Copyright 2015 The gRPC Authors
Copyright Rafael Winterhalter
Copyright (C) 2008 The Guava Authors
Copyright 2019 LINE Corporation

SPDX-License-Identifier: 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.

```
-----  
  
# LICENSES FOR THIRD-PARTY COMPONENTS  
  
=====
```

Third Party Runtime Dependencies

```
=====
```

```
# io.opentelemetry:opentelemetry-sdk (Apache-2.0)
```

Copyright The OpenTelemetry Authors
SPDX-License-Identifier: Apache-2.0

io.opentelemetry:opentelemetry-sdk depends on the following technologies/
licenses.

A copy of each license is included accordingly.

- io.opentelemetry:opentelemetry-sdk-logs (Apache-2.0)
- io.opentelemetry:opentelemetry-api (Apache-2.0)
- io.opentelemetry:opentelemetry-sdk-common (Apache-2.0)
- io.opentelemetry:opentelemetry-sdk-trace (Apache-2.0)
- io.opentelemetry:opentelemetry-sdk-metrics (Apache-2.0)

-
io.opentelemetry:opentelemetry-sdk-logs (Apache-2.0)

Copyright The OpenTelemetry Authors
SPDX-License-Identifier: Apache-2.0

io.opentelemetry:opentelemetry-sdk-logs depends on the following technologies/
licenses.

A copy of each license is included accordingly.

- io.opentelemetry:opentelemetry-sdk-common (Apache-2.0)

-
io.opentelemetry:opentelemetry-sdk-common (Apache-2.0)

Copyright The OpenTelemetry Authors
SPDX-License-Identifier: Apache-2.0

Includes work from:

Copyright 2019 LINE Corporation

LINE Corporation licenses this file to you 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:

<https://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.

io.opentelemetry:opentelemetry-sdk-common depends on the following

technologies/licenses.
A copy of each license is included accordingly.

- io.opentelemetry:opentelemetry-semconv (Apache-2.0)
- io.opentelemetry:opentelemetry-api Apache-2.0

-

io.opentelemetry:opentelemetry-semconv (Apache-2.0)

Copyright The OpenTelemetry Authors
SPDX-License-Identifier: Apache-2.0

io.opentelemetry:opentelemetry-semconv depends on the following technologies/
licenses.

A copy of each license is included accordingly.

- io.opentelemetry:opentelemetry-api (Apache-2.0)

-

io.opentelemetry:opentelemetry-api (Apache-2.0)

Copyright The OpenTelemetry Authors
SPDX-License-Identifier: Apache-2.0

Includes work from:

Copyright 2000-2021 JetBrains s.r.o.
Copyright (C) 2008 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:

<https://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.

io.opentelemetry:opentelemetry-api depends on the following technologies/
licenses.

A copy of each license is included accordingly.

- io.opentelemetry:opentelemetry-context (Apache-2.0)

-


```
##### io.opentelemetry:opentelemetry-context (Apache-2.0)
```

```
Copyright The OpenTelemetry Authors  
SPDX-License-Identifier: Apache-2.0
```

Includes work from:

```
Copyright 2013-2020 The OpenZipkin Authors  
Copyright 2015 The gRPC Authors  
Copyright Rafael Winterhalter
```

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.

```
-----  
-
```

```
#### io.opentelemetry:opentelemetry-api (Apache-2.0)
```

```
-----  
-
```

```
## io.opentelemetry:opentelemetry-api (Apache-2.0)
```

```
-----  
-
```

```
## io.opentelemetry:opentelemetry-sdk-common (Apache-2.0)
```

```
-----  
-
```

```
## io.opentelemetry:opentelemetry-sdk-trace (Apache-2.0)
```

```
Copyright The OpenTelemetry Authors  
SPDX-License-Identifier: Apache-2.0
```

io.opentelemetry:opentelemetry-sdk-trace depends on the following technologies/licenses.

A copy of each license is included accordingly.

- io.opentelemetry:opentelemetry-semconv (Apache-2.0)
- io.opentelemetry:opentelemetry-api (Apache-2.0)
- io.opentelemetry:opentelemetry-sdk-common (Apache-2.0)

```
-----
```

```
-  
## io.opentelemetry:opentelemetry-sdk-metrics (Apache-2.0)  
  
Copyright The OpenTelemetry Authors  
SPDX-License-Identifier: Apache-2.0  
  
io.opentelemetry:opentelemetry-sdk-metrics depends on the following  
technologies/licenses.  
A copy of each license is included accordingly.  
  
- io.opentelemetry:opentelemetry-api (Apache-2.0)  
- io.opentelemetry:opentelemetry-sdk-common (Apache-2.0)  
  
-----  
-
```

D.20 Google Guava 33.0.0

Copyright (C) 2020 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.

Apache License Version 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.

=====

+--- 4th party: com.google.guava:failureaccess

Copyright (C) 2018 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.

< Apache License Version 2.0 >

=====

+--- 4th party: com.google.guava:listenablefuture

Copyright (C) 2018 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.

< Apache License Version 2.0 >

=====

+--- 4th party: com.google.code.findbugs:jsr305
Copyright: JSR305 expert group
License: BSD 3-Clause

Copyright (c) 2007-2009, JSR305 expert group
All rights reserved.

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.
- * Neither the name of the JSR305 expert group 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.

== jcip-annotations relicensed to Oracle under BSD 3-clause license

Copyright (c) 2005, Brian Goetz and Tim Peierls

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.
- * Neither the name of the JSR305 expert group 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.

=====
+--- 4th party: com.google.errorprone:error_prone_annotations

Copyright 2015 The Error Prone 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.

< Apache License Version 2.0 >

+--- 4th party: com.google.j2objc:j2objc-annotations

Google Inc.
Daniel Connelly

Copyright 2012 Google Inc. All Rights Reserved.

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 >

=====
+--- 4th party: org.checkerframework:checker-qual

Copyright 2004-present by the Checker Framework developers

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.

=====

D.21 xml2js 0.6.2

xml2js - MIT license
Copyright (c) 2010-2022 Mathias Bynens <<https://mathiasbynens.be/>>

Copyright 2010, 2011, 2012, 2013. 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.

sax

The ISC License

Copyright (c) 2010-2022 Isaac Z. Schlueter and Contributors

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.

====

xmlbuilder

The MIT License (MIT)

Copyright (c) 2013 Ozgur Ozcitak

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.

D.22 Jansi 2.4.1

jansi (org.fusesource.jansi:jansi)
Copyright (C) 2009-2023 the original author(s).

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.

Fourth Party Dependencies

HawtJNI

Copyright (C) 2009-2011 FuseSource Corp.

Copyright (c) 2004, 2008 IBM Corporation and others.

This product includes software developed by FuseSource Corp. <http://fusesource.com>

This product includes software developed at Progress Software Corporation and/or its subsidiaries or affiliates.

This product includes software developed by IBM Corporation and others.

Apache License Version 2.0

D.23 commons-io 2.15.1

Apache Commons IO

Copyright 2002-2023 The Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (<https://www.apache.org/>).

--

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>

D.24 Join Monster 3.3.1

Join Monster
License MIT

MIT License

Copyright (c) 2016 Stem Disintermedia, Inc.

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.

=====
Fourth Party Dependency #1 - stem/nesthydrationjs
License - 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.

=====
=====

Fourth Party Dependency #2 - debug
License - MIT
(The MIT License)

Copyright (c) 2014-2017 TJ Holowaychuk <tj@vision-media.ca>
Copyright (c) 2018-2021 Josh Junon

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.

=====
=====

Fourth Party Dependency #3 - deprecate
author Brian M. Carlson
license : MIT

MIT Listed in package.json, but not enumerated in a file.

=====
=====

Fourth Party Dependency #4 - generatorics
author Andrew Carlson <acarl005@ucla.edu>
License MIT

MIT Listed in package.json, but not enumerated in a file.

=====
=====

Fourth Party Dependency #5 - graphql-relay
MIT License

Copyright (c) GraphQL 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.

=====

D.25 SheetJS 0.20.1

SheetJS is licensed under Apache 2.0. License Text below.

Copyright (C) 2012-present SheetJS LLC

Copyright (C) 2013-present SheetJS LLC

Copyright (C) 2019-present SheetJS LLC

=====

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.

Fourth Party Licenses (same as above)

Adler-32

CFB

codepage

crc-32

ssf

wmf

word

Frac

cfb

=====

Except this below

JSDOM

Copyright (c) 2010 Elijah Insua

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.

D.26 graphql-compose 9.0.10

The MIT License (MIT)

Copyright (c) 2016-present Pavel Chertorogov

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 dependency
graphql-type-json
MIT License
The MIT License (MIT)

Copyright (c) 2016 Jimmy Jia

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.

D.27 jackson-dataformat-xml 2.16.1

Top Level Component: jackson-dataformat-xml

Top Level Component License: Apache License

Apache License
Version 2.0, January 2004
<https://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

<https://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.

Top Level Component Copyright Notices:

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.

Separator-----

Fourth Party Component # com.fasterxml.jackson.core:jackson-annotations
Fourth Party Component # License: Apache 2.0
Fourth Party Component # Copyright Notices:
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.

Copyright

Copyright 2007-, Tatu Saloranta (tatu.saloranta@iki.fi)

Licensing

Jackson 2.x core and extension components are licensed under Apache License 2.0
To find the details that apply to this artifact see the accompanying LICENSE file.

Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

Separator-----

Fourth Party Component # com.fasterxml.jackson.core:jackson-core
Fourth Party Component # License: Apache 2.0
Fourth Party Component # Copyright Notices:

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.

Copyright

```

Copyright 2007-, Tatu Saloranta (tatu.saloranta@iki.fi)

## Licensing

Jackson 2.x core and extension components are licensed under Apache
License 2.0
To find the details that apply to this artifact see the accompanying
LICENSE file.

## Credits

A list of contributors may be found from CREDITS(-2.x) file, which is
included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.

## FastDoubleParser

jackson-core bundles a shaded copy of FastDoubleParser <https://
github.com/wrandelshofer/FastDoubleParser>.
That code is available under an MIT license <https://github.com/
wrandelshofer/FastDoubleParser/blob/main/LICENSE>
under the following copyright.

Copyright © 2023 Werner Randelshofer, Switzerland. MIT License.

See FastDoubleParser-NOTICE for details of other source code included in
FastDoubleParser
and the licenses and copyrights that apply to that code.

-----
Separator-----

Fourth Party Component # com.fasterxml.jackson.core:jackson-databind
Fourth Party Component # License: Apache 2.0
Fourth Party Component # Copyright Notices:

# 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.

## Copyright

Copyright 2007-, Tatu Saloranta (tatu.saloranta@iki.fi)

## Licensing

Jackson 2.x core and extension components are licensed under Apache
License 2.0
To find the details that apply to this artifact see the accompanying
LICENSE file.

```

Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

Separator-----

Fourth Party Component # org.codehaus.woodstox:stax2-api
Fourth Party Component # License: BSD Clause-2

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.

Fourth Party Component # Copyright Notice:

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:

<COPYRIGHT HOLDER> = FasterXML.com

<YEAR> = 2010-2022

Separator-----

Fourth Party Component # com.fasterxml.woodstox:woodstox-core
Fourth Party Component # License: Apache 2.0
Fourth Party Component # Copyright Notices:

This copy of Jackson JSON processor databind module 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>

```
/* Woodstox XML processor
 *
 * Copyright (c) 2004- Tatu Saloranta, tatu.saloranta@iki.fi
 *
 * Licensed under the License specified in the file LICENSE which is
 * included with the source code.
 * You may not use this file except in compliance with the License.
 *
 * 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 © 2020 FasterXML. All rights reserved.

Separator-----

Fourth Party Component # ch.randelshofer » fastdoubleparser
Fourth Party Component # License:

MIT License

Copyright (c) 2023 Werner Randelshofer, Switzerland.

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.

Fourth Party Dependency # Copyright Notices:

FastDoubleParser

This is a Java port of Daniel Lemire's fast_float project. This project provides parsers for double, float, BigDecimal and BigInteger values.

Copyright

Copyright © 2023 Werner Randelshofer, Switzerland.

Licensing

This code is licensed under MIT License.
<https://github.com/wrandelshofer/FastDoubleParser/blob/522be16e145f43308c43b23094e31d5efcaa580e/LICENSE>
(The file 'LICENSE' is included in the sources and classes Jar files that are released by this project
- as is required by that license.)

Some portions of the code have been derived from other projects. All these projects require that we include a copyright notice, and some require that we also include some text of their license file.

fast_double_parser, Copyright (c) 2022 Daniel Lemire. Apache 2.0 License.

https://github.com/fastfloat/fast_float
https://github.com/fastfloat/fast_float/blob/dc88f6f882ac7eb8ec3765f633835cb76afa0ac2/LICENSE-APACHE

fast_float, Copyright (c) 2021 The fast_float authors. Apache 2.0 License.

https://github.com/fastfloat/fast_float
https://github.com/lemire/fast_double_parser/blob/07d9189a8fb815fe800cb15ca022e7a07093236e/LICENSE

bigint, Copyright 2020 Tim Bukt. 2-clause BSD License.
<https://github.com/tbukt/bigint/tree/floatfft>
<https://github.com/tbukt/bigint/blob/617c8cd8a7c5e4fb4d919c6a4d11e2586107f029/LICENSE>
<https://github.com/wrandelshofer/FastDoubleParser/blob/>

```

39e123b15b71f29a38a087d16a0bc620fc879aa6/bigint-LICENSE
  (We only use those portions of the bigint project that can be licensed
  under 2-clause BSD License.)
  (The file 'bigint-LICENSE' is included in the sources and classes Jar
  files that are released by this project
  - as is required by that license.)

-----
Separator-----

```

D.28 hotkeys-js 3.13.7

```

MIT License
Copyright (c) 2015-present, Kenny Wong.
Copyright (c) 2024 kenny wong <wowohoo@qq.com>
Copyright (c) 2011-2013 Thomas Fuchs (https://github.com/madrobby/keymaster)
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.

```

D.29 swagger-ui 5.1.1

```

Top Level Component : Swagger UI
-----
Top Level Component 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.

Top Level Component Copyright: NOTICE file

swagger-ui
Copyright 2020-2021 SmartBear Software Inc.

=====
=====
4th Party Component #1 : base64-js

4th Party Component #1 License : MIT

4th Party Component #1 Copyright:

Copyright (c) 2014 Jameson Little

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 Component #2 : classnames

4th Party Component #2 License : MIT

4th Party Component #2 Copyright:

Copyright (c) 2018 Jed Watson

=====
=====
4th Party Component #3 : css.escape

4th Party Component #3 License : MIT

4th Party Component #3 Copyright:

Copyright Mathias Bynens

=====
=====

4th Party Component #4 : deep-extend

4th Party Component #4 License : MIT

4th Party Component #4 Copyright:

Copyright (c) 2013-2018, Viacheslav Lotsmanov

=====
=====

4th Party Component #5 : dompurify

4th Party Component #5 License : Apache 2.0, MPL 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.

Mozilla Public License, version 2.0

1. Definitions

1.1. "Contributor"

means each individual or legal entity that creates, contributes to the
creation of, or owns Covered Software.

1.2. "Contributor Version"

means the combination of the Contributions of others (if any) used by a
Contributor and that particular Contributor's Contribution.

1.3. "Contribution"

means Covered Software of a particular Contributor.

1.4. "Covered Software"

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. "Incompatible With Secondary Licenses"

means

- a. that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or
- b. that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. "Executable Form"

means any form of the work other than Source Code Form.

1.7. "Larger Work"

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. "License"

means this document.

1.9. "Licensable"

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. "Modifications"

means any of the following:

- a. any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or
- b. any new file in Source Code Form that contains any Covered Software.

1.11. "Patent Claims" of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of

either its Contributions or its Contributor Version.

1.12. "Secondary License"

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. "Source Code Form"

means the form of the work preferred for making modifications.

1.14. "You" (or "Your")

means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

a. under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or

as

part of a Larger Work; and

b. under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its

Contributions

or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become

effective for each Contribution on the date the Contributor first distributes

such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this

License. No additional rights or licenses will be implied from the distribution

or licensing of Covered Software under this License. Notwithstanding
Section

2.1(b) above, no patent license is granted by a Contributor:

- a. for any code that a Contributor has removed from Covered Software; or
- b. for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- c. under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code

Form

of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- a. such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and
- b. You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support,

indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an "as is" basis, without

warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

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.

Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0

4th Party Component #5 Copyright:

DOMPurify
Copyright 2015 Mario Heiderich

DOMPurify is free software; you can redistribute it and/or modify it under the terms of either:

- a) the Apache License Version 2.0, or
- b) the Mozilla Public License Version 2.0

=====
=====

4th Party Component #6 : ieee754
4th Party Component #6 License : BSD 3-Clause "New" or "Revised" License

4th Party Component #6 Copyright:

Copyright 2008 Fair Oaks Labs, Inc.

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

=====
=====

4th Party Component #7 : immutable

4th Party Component #7 License : MIT

4th Party Component #7 Copyright:

Copyright (c) 2014-present, Facebook, Inc.

=====
=====

4th Party Component #8 : js-file-download

4th Party Component #8 License : MIT

4th Party Component #8 Copyright:

Copyright 2017 Kenneth Jiang

=====
=====
4th Party Component #9 : js-yaml

4th Party Component #9 License : MIT

4th Party Component #9 Copyright:

Copyright (C) 2011-2015 by Vitaly Puzrin

=====
=====
4th Party Component #10 : lodash

4th Party Component #10 License : MIT

4th Party Component #10 Copyright:

Copyright OpenJS Foundation and other contributors

Based on Underscore.js, copyright Jeremy Ashkenas,
DocumentCloud and Investigative Reporters & Editors

This software consists of voluntary contributions made by many
individuals. For exact contribution history, see the revision history
available at <https://github.com/lodash/lodash>

Copyright and related rights for sample code are waived via CC0. Sample
code is defined as all source code displayed within the prose of the
documentation.

CC0: <http://creativecommons.org/publicdomain/zero/1.0/>

====

Files located in the node_modules and vendor 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.

=====
=====
4th Party Component #11 : prop-types

4th Party Component #11 License : MIT

4th Party Component #11 Copyright:

Copyright (c) 2013-present, Facebook, Inc.

=====
=====
4th Party Component #12 : randexp

4th Party Component #12 License : MIT

4th Party Component #12 Copyright:

Copyright (C) 2011 by fent

=====
=====
4th Party Component #13 : randombytes

4th Party Component #13 License : MIT

4th Party Component #13 Copyright:

Copyright (c) 2017 crypto-browserify

=====
=====
4th Party Component #14 : react

4th Party Component #14 License : MIT

4th Party Component #14 Copyright:

Copyright (c) Facebook, Inc. and its affiliates.

=====
=====
4th Party Component #15 : react-copy-to-clipboard

4th Party Component #15 License : MIT

4th Party Component #15 Copyright:

Copyright (c) 2016 Nik Butenko

=====
=====
4th Party Component #16 : react-debounce-input

4th Party Component #16 License : MIT

```
4th Party Component #16 Copyright:
-----
Copyright (c) 2016 Nik Butenko

=====
=====
4th Party Component #17 : react-dom

4th Party Component #17 License : MIT
-----

4th Party Component #17 Copyright:
-----
Copyright (c) Facebook, Inc. and its affiliates.

=====
=====
4th Party Component #18 : react-immutable-proptypes

4th Party Component #18 License : MIT
-----

4th Party Component #18 Copyright:
-----
Copyright (c) 2015 James Burnett

=====
=====
4th Party Component #19 : react-immutable-pure-component

4th Party Component #19 License : MIT
-----

4th Party Component #19 Copyright:
-----
Copyright (c) 2017 Piotr Tomasz Monarski

=====
=====
4th Party Component #20 : react-inspector

4th Party Component #20 License : MIT
-----

4th Party Component #20 Copyright:
-----
Copyright (c) 2017 Xiaoyi Chen

=====
=====
4th Party Component #21 : react-redux

4th Party Component #21 License : MIT
-----

4th Party Component #21 Copyright:
```

```
-----  
Copyright (c) 2015 Dan Abramov  
  
=====
```

4th Party Component #22 : react-syntax-highlighter

4th Party Component #22 License : MIT

```
-----
```

4th Party Component #22 Copyright:

```
-----
```

Copyright (c) 2019 Conor Hastings

```
=====
```

4th Party Component #23 : redux

4th Party Component #23 License : MIT

```
-----
```

4th Party Component #23 Copyright:

```
-----
```

Copyright (c) 2015-present Dan Abramov

```
=====
```

4th Party Component #24 : redux-immutable

4th Party Component #24 License : BSD 3-Clause

```
-----
```

4th Party Component #24 Copyright:

```
-----
```

Copyright (c) 2016, Gajus Kuizinas (<http://gajus.com/>)
All rights reserved.

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.
- * Neither the name of the Gajus Kuizinas (<http://gajus.com/>) 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 ANUARY 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.

=====
=====

4th Party Component #25 : remarkable

4th Party Component #25 License : MIT

4th Party Component #25 Copyright:

Copyright (c) 2014-2016, Jon Schlinkert
Copyright (c) 2014 Jon Schlinkert, Vitaly Puzrin.

=====
=====

4th Party Component #26 : reselect

4th Party Component #26 License : MIT

4th Party Component #26 Copyright:

Copyright (c) 2015-2018 Reselect Contributors

=====
=====

4th Party Component #27 : serialize-error

4th Party Component #27 License : MIT

4th Party Component #27 Copyright:

Copyright (c) Sindre Sorhus (sindresorhus.com)

=====
=====

4th Party Component #28 : sha.js

4th Party Component #28 License : MIT / BSD-3

4th Party Component #28 Copyright:

Copyright (c) 2013-2018 sha.js contributors

Copyright (c) 1998 - 2009, Paul Johnston & Contributors
All rights reserved.

=====
=====

4th Party Component #29 : swagger-client

```
4th Party Component #29 License : Apache 2.0
-----

4th Party Component #29 Copyright: NOTICE file
-----
swagger-js/swagger-client
Copyright 2020-2021 SmartBear Software Inc.

=====
=====
4th Party Component #30 : url-parse

4th Party Component #30 License : MIT
-----

4th Party Component #30 Copyright:
-----
Copyright (c) 2015 Unshift.io, Arnout Kazemier, the Contributors.

=====
=====
4th Party Component #31 : xml

4th Party Component #31 License : MIT
-----

4th Party Component #31 Copyright:
-----
Copyright (c) 2011-2017 Dylan Greene <dylang@gmail.com>

=====
=====
4th Party Component #32 : xml-but-prettier

4th Party Component #32 License : MIT
-----

4th Party Component #32 Copyright:
-----
Copyright (c) 2015 Jonathan Persson

=====
=====
4th Party Component #33 : zenscroll

4th Party Component #33 License :
-----

4th Party Component #33 Copyright:
-----
This is free and unencumbered software released into the public domain.

Anyone is free to copy, modify, publish, use, compile, sell, or
distribute this software, either in source code form or as a compiled
binary, for any purpose, commercial or non-commercial, and by any
means.
```

In jurisdictions that recognize copyright laws, the author or authors of this software dedicate any and all copyright interest in the software to the public domain. We make this dedication for the benefit of the public at large and to the detriment of our heirs and successors. We intend this dedication to be an overt act of relinquishment in perpetuity of all present and future rights to this software under copyright law.

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 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.

For more information, please refer to <<http://unlicense.org>>

```
=====
=====
```

```
4th Party Component #34 : @babel/babel-runtime-corejs3, @babel/runtime
```

```
4th Party Component #34 License : MIT
-----
```

```
4th Party Component #34 Copyright:
-----
```

```
Copyright (c) 2014-present Sebastian McKenzie and other contributors
```

```
=====
=====
```

```
4th Party Component #35 : core-js-pure
```

```
4th Party Component #35 License : MIT
-----
```

```
4th Party Component #35 Copyright:
-----
```

```
Copyright (c) 2014-2022 Denis Pushkarev
```

```
=====
=====
```

```
4th Party Component #36 : regenerator-runtime
```

```
4th Party Component #36 License : MIT
-----
```

```
4th Party Component #36 Copyright:
-----
```

```
Copyright (c) 2014-present, Facebook, Inc.
```

```
=====
=====
```

```
4th Party Component #37 : @braintree/sanitize-url
```

4th Party Component #37 License : MIT

4th Party Component #37 Copyright:

Copyright (c) 2017 Braintree

=====
=====

4th Party Component #38 : argparse

4th Party Component #38 License : PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020 Python Software Foundation; All Rights Reserved" are retained in Python alone or in any derivative version prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.

4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote

products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0

BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.

3. BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at <http://www.pythonlabs.com/logos.html> may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.
2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the Internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the Internet using the following URL: <http://hdl.handle.net/1895.22/1013>".
3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.
4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.
5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.
6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.
7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the

law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam, The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, 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.

4th Party Component #38 Copyright:

Copyright (C) 2010-2020 Python Software Foundation
Copyright (C) 2020 argparse.js authors

=====
=====

4th Party Component #39 : loose-envify

4th Party Component #39 License : MIT

4th Party Component #39 Copyright:

Copyright (c) 2015 Andres Suarez <zertosh@gmail.com>

=====

```
=====
4th Party Component #40 : js-tokens

4th Party Component #40 License : MIT
-----

4th Party Component #40 Copyright:
-----
Copyright (c) 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021 Simon Lydell

=====
=====
4th Party Component #41 : object-assign

4th Party Component #41 License : MIT
-----

4th Party Component #41 Copyright:
-----
Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

=====
=====
4th Party Component #42 : react-is

4th Party Component #42 License : MIT
-----

4th Party Component #42 Copyright:
-----
Copyright (c) Facebook, Inc. and its affiliates.

=====
=====
4th Party Component #43 : drange

4th Party Component #43 License : MIT
-----

4th Party Component #43 Copyright:
-----
Copyright (c) 2014 David Tudury

=====
=====
4th Party Component #44 : ret

4th Party Component #44 License : MIT
-----

4th Party Component #44 Copyright:
-----
Copyright (C) 2011 by fent

=====
=====
```

4th Party Component #45 : safe-buffer

4th Party Component #45 License : MIT

4th Party Component #45 Copyright:

Copyright (c) Feross Aboukhadijeh

=====
=====

4th Party Component #46 : copy-to-clipboard

4th Party Component #46 License : MIT

4th Party Component #46 Copyright:

Copyright (c) 2017 sudodoki <smd.deluzion@gmail.com>

=====
=====

4th Party Component #47 : toggle-selection

4th Party Component #47 License : MIT

4th Party Component #47 Copyright:

Copyright (c) 2017 sudodoki <smd.deluzion@gmail.com>

=====
=====

4th Party Component #48 : lodash.debounce

4th Party Component #48 License : MIT

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: <http://creativecommons.org/publicdomain/zero/1.0/>

4th Party Component #48 Copyright:

Copyright JS Foundation and other contributors <<https://js.foundation/>>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <<http://underscorejs.org/>>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at <https://github.com/lodash/lodash>

=====

```
=====
4th Party Component #49 : scheduler

4th Party Component #49 License : MIT
-----

4th Party Component #49 Copyright:
-----
Copyright (c) Facebook, Inc. and its affiliates.

=====
=====
4th Party Component #50 : invariant

4th Party Component #50 License : MIT
-----

4th Party Component #50 Copyright:
-----
Copyright (c) 2013-present, Facebook, Inc.

=====
=====
4th Party Component #51 : is-dom

4th Party Component #51 License : MIT
-----

4th Party Component #51 Copyright:
-----
Copyright (c) 2014

=====
=====
4th Party Component #52 : is-object

4th Party Component #52 License : MIT
-----

4th Party Component #52 Copyright:
-----
Copyright (c) 2013 Colingo.

=====
=====
4th Party Component #53 : is-window

4th Party Component #53 License : MIT
-----

4th Party Component #53 Copyright:
-----
Copyright (c) 2016 W.Y.

=====
=====
```

```
4th Party Component #54 : @types/react-redux

4th Party Component #54 License : MIT
-----

4th Party Component #54 Copyright:
-----
// Type definitions for react-redux 7.1
// Project: https://github.com/reduxjs/react-redux
// Definitions by: Qubo <https://github.com/tkqubo>
//                 Curits Layne <https://github.com/claynell>
//                 Frank Tan <https://github.com/tansongyang>
//                 Nicholas Boll <https://github.com/nicholasboll>
//                 Dibyo Majumdar <https://github.com/mdibyo>
//                 Valentin Descamps <https://github.com/vall1984>
//                 Johann Rakotoharisoa <https://github.com/jrakotoharisoa>
//                 Anatoli Papirovski <https://github.com/apapirovski>
//                 Boris Sergeev <https://github.com/surgeboris>
//                 Søren Bruus Frank <https://github.com/soerenbf>
//                 Jonathan Ziller <https://github.com/mrwolfz>
//                 Dylan Vann <https://github.com/dylanvann>
//                 Yuki Ito <https://github.com/Lazyuki>
//                 Kazuma Ebina <https://github.com/kazuma1989>
//                 Michael Lebedev <https://github.com/megazazik>
//                 jun-sheaf <https://github.com/jun-sheaf>
//                 Lenz Weber <https://github.com/phryneas>
//                 Mark Erikson <https://github.com/mark Erikson>
// Definitions: https://github.com/DefinitelyTyped/DefinitelyTyped
// TypeScript Version: 3.0

=====
=====
4th Party Component #55 : @types/hoist-non-react-statics

4th Party Component #55 License : MIT
-----

4th Party Component #55 Copyright:
-----
// Definitions by: JounQin <https://github.com/JounQin>, James Reggio
<https://github.com/jamesreggio>

=====
=====
4th Party Component #56 : @types/react

4th Party Component #56 License : MIT
-----

4th Party Component #56 Copyright:
-----
// Definitions by: Asana <https://asana.com>
//                 AssureSign <http://www.assuresign.com>
//                 Microsoft <https://microsoft.com>
//                 John Reilly <https://github.com/johnnyreilly>
//                 Benoit Benezech <https://github.com/bbenezech>
```

```

//      Patricio Zavolinsky <https://github.com/pzavolinsky>
//      Eric Anderson <https://github.com/ericanderson>
//      Dovydas Navickas <https://github.com/DovydasNavickas>
//      Josh Rutherford <https://github.com/therutherford4d>
//      Guilherme Hübner <https://github.com/guilhermehubner>
//      Ferdy Budhidharma <https://github.com/ferdaber>
//      Johann Rakotoharisoa <https://github.com/jrakotoharisoa>
//      Olivier Pascal <https://github.com/pascaloliv>
//      Martin Hochel <https://github.com/hotell>
//      Frank Li <https://github.com/franklixuefei>
//      Jessica Franco <https://github.com/Jessidhia>
//      Saransh Kataria <https://github.com/saranshkataria>
//      Kanitkorn Sujautra <https://github.com/lukyth>
//      Sebastian Silbermann <https://github.com/epsllon>
//      Kyle Scully <https://github.com/zieka>
//      Cong Zhang <https://github.com/dancerphil>
//      Dimitri Mitropoulos <https://github.com/dimitropoulos>
//      JongChan Choi <https://github.com/disjukr>
//      Victor Magalhães <https://github.com/vhfmag>
//      Dale Tan <https://github.com/hellatan>
//      Priyanshu Rav <https://github.com/priyanshurav>

=====
=====
4th Party Component #57 : @types/prop-types

4th Party Component #57 License : MIT
-----

4th Party Component #57 Copyright:
-----
// Definitions by: DovydasNavickas <https://github.com/DovydasNavickas>
//      Ferdy Budhidharma <https://github.com/ferdaber>
//      Sebastian Silbermann <https://github.com/epsllon>
=====
=====
4th Party Component #58 : @types/scheduler

4th Party Component #58 License : MIT
-----

4th Party Component #58 Copyright:
-----
// Definitions by: Nathan Bierema <https://github.com/Methuselah96>
//      Sebastian Silbermann <https://github.com/epsllon>

=====
=====
4th Party Component #59 : csstype

4th Party Component #59 License : MIT
-----

4th Party Component #59 Copyright:
-----
Copyright (c) 2017-2018 Fredrik Nicol

```



```
=====
=====
4th Party Component #60 : hoist-non-react-statics

4th Party Component #60 License : BSD-3
-----

4th Party Component #60 Copyright:
-----
Copyright (c) 2015, Yahoo! Inc. All rights reserved.

=====
=====
4th Party Component #61 : repeat-string

4th Party Component #61 License : MIT
-----

4th Party Component #61 Copyright:
-----
Copyright (c) 2014-2015, Jon Schlinkert.

=====
=====
4th Party Component #62 : highlight.js

4th Party Component #62 License : BSD-3
-----

4th Party Component #62 Copyright:
-----
Copyright (c) 2006, Ivan Sagalaev.
All rights reserved.

=====
=====
4th Party Component #63 : lowlight

4th Party Component # License : MIT
-----

4th Party Component # Copyright:
-----
Copyright (c) 2016 Titus Wormer <tituswormer@gmail.com>

=====
=====
4th Party Component #63 : fault

4th Party Component #63 License : MIT
-----

4th Party Component #63 Copyright:
-----
```

Copyright (c) 2015 Titus Wormer <tituswormer@gmail.com>

=====
=====

4th Party Component #64 : format

4th Party Component #64 License : MIT

4th Party Component #64 Copyright:

Copyright 2010 - 2016 Sami Samhuri sami@samhuri.net

=====
=====

4th Party Component #65 : prismjs

4th Party Component #65 License : MIT

4th Party Component #65 Copyright:

Copyright (c) 2012 Lea Verou

=====
=====

4th Party Component #66 : refractor

4th Party Component #66 License : MIT

4th Party Component #66 Copyright:

Copyright (c) 2017 Titus Wormer <tituswormer@gmail.com>

=====
=====

4th Party Component #67 : hastscript

4th Party Component #67 License : MIT

4th Party Component #67 Copyright:

Copyright (c) 2016 Titus Wormer <tituswormer@gmail.com>

=====
=====

4th Party Component #68 : @types/hast

4th Party Component #68 License : MIT

4th Party Component #68 Copyright:

// Definitions by: lukeggchapman <<https://github.com/lukeggchapman>>

```
// Junyoung Choi <https://github.com/rokt33r>
// Christian Murphy <https://github.com/ChristianMurphy>

=====
=====
4th Party Component #69 : @types/unist

4th Party Component #69 License : MIT
-----

4th Party Component #69 Copyright:
-----
// Definitions by: bizen241 <https://github.com/bizen241>
// Jun Lu <https://github.com/lujun2>
// Hernan Rajchert <https://github.com/hrajchert>
// Titus Wormer <https://github.com/wooorm>
// Junyoung Choi <https://github.com/rokt33r>
// Ben Moon <https://github.com/GuiltyDolphin>
// JounQin <https://github.com/JounQin>

=====
=====
4th Party Component #70 : comma-separated-tokens

4th Party Component #70 License : MIT
-----

4th Party Component #70 Copyright:
-----
Copyright (c) 2016 Titus Wormer <tituswormer@gmail.com>

=====
=====
4th Party Component #71 : hast-util-parse-selector

4th Party Component #71 License : MIT
-----

4th Party Component #71 Copyright:
-----
Copyright (c) 2016 Titus Wormer <tituswormer@gmail.com>

=====
=====
4th Party Component #72 : property-information

4th Party Component #72 License : MIT
-----

4th Party Component #72 Copyright:
-----
Copyright (c) 2015 Titus Wormer <mailto:tituswormer@gmail.com>

=====
=====
4th Party Component #73 : xtend
```

4th Party Component #73 License : MIT

4th Party Component #73 Copyright:

Copyright (c) 2012-2014 Raynos.

=====
=====
4th Party Component #74 : space-separated-tokens

4th Party Component #74 License : MIT

4th Party Component #74 Copyright:

Copyright (c) 2016 Titus Wormer <tituswormer@gmail.com>

=====
=====
4th Party Component #75 : parse-entities

4th Party Component #75 License : MIT

4th Party Component #75 Copyright:

Copyright (c) 2015 Titus Wormer <mailto:tituswormer@gmail.com>

=====
=====
4th Party Component #76 : character-entities-legacy

4th Party Component #76 License : MIT

4th Party Component #76 Copyright:

Copyright (c) 2015 Titus Wormer <tituswormer@gmail.com>

=====
=====
4th Party Component #77 : character-entities

4th Party Component #77 License : MIT

4th Party Component #77 Copyright:

Copyright (c) 2015 Titus Wormer <tituswormer@gmail.com>

=====
=====
4th Party Component #78 : character-reference-invalid

```
4th Party Component #78 License : MIT
-----

4th Party Component #78 Copyright:
-----
Copyright (c) 2015 Titus Wormer <tituswormer@gmail.com>

=====
=====
4th Party Component #79 : is-alphanumeric

4th Party Component #79 License : MIT
-----

4th Party Component #79 Copyright:
-----
Copyright (c) 2016 Titus Wormer <tituswormer@gmail.com>

=====
=====
4th Party Component #80 : is-alphabetical

4th Party Component #80 License : MIT
-----

4th Party Component #80 Copyright:
-----
Copyright (c) 2016 Titus Wormer <tituswormer@gmail.com>

=====
=====
4th Party Component #81 : is-decimal

4th Party Component #81 License : MIT
-----

4th Party Component #81 Copyright:
-----
Copyright (c) 2016 Titus Wormer <tituswormer@gmail.com>

=====
=====
4th Party Component #82 : is-hexadecimal

4th Party Component #82 License : MIT
-----

4th Party Component #82 Copyright:
-----
Copyright (c) 2016 Titus Wormer <tituswormer@gmail.com>

=====
=====
4th Party Component #83 : sprintf-js

4th Party Component #83 License : BSD-3
```

```
-----  
4th Party Component #83 Copyright:  
-----  
Copyright (c) 2007-present, Alexandru Mărășteanu <hello@alexei.ro>  
All rights reserved.  
  
=====
```

```
4th Party Component #84 : autolinker  
  
4th Party Component #84 License : MIT  
-----  
  
4th Party Component #84 Copyright:  
-----  
Copyright (c) 2014 Gregory Jacobs (http://greg-jacobs.com)  
  
=====
```

```
4th Party Component #85 : tslib  
  
4th Party Component #85 License : BSD-0  
-----  
Permission to use, copy, modify, and/or distribute this software for any  
purpose with or without fee is hereby granted.  
  
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.  
  
4th Party Component #85 Copyright:  
-----  
Copyright (c) Microsoft Corporation.  
  
=====
```

```
4th Party Component #86 : type-fest  
  
4th Party Component #86 License : MIT or CC0-1.0  
-----  
Creative Commons Legal Code  
  
CC0 1.0 Universal  
  
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:

- i. the right to reproduce, adapt, distribute, perform, display, communicate, and translate a Work;
- ii. moral rights retained by the original author(s) and/or performer(s);
- iii. publicity and privacy rights pertaining to a person's image or likeness depicted in a Work;
- iv. rights protecting against unfair competition in regards to a Work, subject to the limitations in paragraph 4(a), below;
- v. rights protecting the extraction, dissemination, use and reuse of data in a Work;
- vi. 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
- vii. 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.

- a. No trademark or patent rights held by Affirmer are waived, abandoned, surrendered, licensed or otherwise affected by this document.
- b. 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.
- c. 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.
- d. 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.

4th Party Component #86 Copyright:


```
-----  
Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (https://  
sindresorhus.com)  
  
=====
```

4th Party Component #87 : inherits

4th Party Component #87 License : ISC

```
-----  
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.  
  
4th Party Component #87 Copyright:  
-----  
Copyright (c) 2011-2022 Isaac Z. Schlueter  
  
=====
```

4th Party Component #88 : btoa

4th Party Component #88 License : Dual-licensed MIT and Apache-2.0

```
-----  
4th Party Component #88 Copyright:  
-----  
Code copyright 2012-2018 AJ O'Neal  
  
=====
```

4th Party Component #89 : cookie

4th Party Component #89 License : MIT

```
-----  
4th Party Component #89 Copyright:  
-----  
Copyright (c) 2012-2014 Roman Shtylman <shtylman@gmail.com>  
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>  
  
=====
```

4th Party Component #90 : cross-fetch

4th Party Component #90 License : MIT

```
-----
```

```
4th Party Component #90 Copyright:
-----
Copyright (c) 2017 Leonardo Quixadá

=====
4th Party Component #91 : node-fetch

4th Party Component #91 License : MIT
-----

4th Party Component #91 Copyright:
-----
Copyright (c) 2016 - 2020 Node Fetch Team

=====
4th Party Component #92 : whatwg-url

4th Party Component #92 License : MIT
-----

4th Party Component #92 Copyright:
-----
Copyright (c) 2015-2016 Sebastian Mayr

=====
4th Party Component #93 : tr46

4th Party Component #93 License : MIT
-----

4th Party Component #93 Copyright:
-----

Copyright (c) 2016 Sebastian Mayr
=====
4th Party Component #94 : webidl-conversions

4th Party Component #94 License : BSD-2
-----
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 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.

4th Party Component #94 Copyright:

Copyright (c) 2014, Domenic Denicola All rights reserved.

=====
=====

4th Party Component #95 : deepmerge

4th Party Component #95 License : MIT

4th Party Component #95 Copyright:

Copyright (c) 2012 James Halliday, Josh Duff, and other contributors

=====
=====

4th Party Component #96 : fast-json-patch

4th Party Component #96 License : MIT

4th Party Component #96 Copyright:

Copyright (c) 2013, 2014, 2020 Joachim Wester

=====
=====

4th Party Component #97 : form-data-encoder

4th Party Component #97 License : MIT

4th Party Component #97 Copyright:

Copyright (c) 2021-present Nick K.

=====
=====

4th Party Component #98 : formdata-node

4th Party Component #98 License : MIT

4th Party Component #98 Copyright:

Copyright (c) 2017-present Nick K.

=====
=====

4th Party Component #99 : node-domexception

4th Party Component #99 License : MIT

4th Party Component #99 Copyright:

Copyright (c) 2021 Jimmy Wärting

=====
=====

4th Party Component #100 : web-streams-polyfill

4th Party Component #100 License : MIT

4th Party Component #100 Copyright:

Copyright (c) 2020 Mattias Buelens

Copyright (c) 2016 Diwank Singh Tomer

=====
=====

4th Party Component #101 : is-plain-object

4th Party Component #101 License : MIT

4th Party Component #101 Copyright:

Copyright (c) 2014-2017, Jon Schlinkert.

=====
=====

4th Party Component #102 : qs

4th Party Component #102 License : BSD-3

4th Party Component #102 Copyright:

Copyright (c) 2014, Nathan LaFreniere and other contributors All rights reserved.

=====
=====

4th Party Component #103 : side-channel

4th Party Component #103 License : MIT

4th Party Component #103 Copyright:

```
-----  
Copyright (c) 2019 Jordan Harband  
  
=====
```

```
4th Party Component #104 : call-bind  
  
4th Party Component #104 License : MIT  
-----  
  
4th Party Component #104 Copyright:  
-----  
Copyright (c) 2020 Jordan Harband  
  
=====
```

```
4th Party Component #105 : function-bind  
  
4th Party Component #105 License : MIT  
-----  
  
4th Party Component #105 Copyright:  
-----  
Copyright (c) 2013 Raynos.  
  
=====
```

```
4th Party Component #106 : get-intrinsic  
  
4th Party Component #106 License : MIT  
-----  
  
4th Party Component #106 Copyright:  
-----  
Copyright (c) 2020 Jordan Harband  
  
=====
```

```
4th Party Component #107 : has-symbols  
  
4th Party Component #107 License : MIT  
-----  
  
4th Party Component #107 Copyright:  
-----  
Copyright (c) 2016 Jordan Harband  
  
=====
```

```
4th Party Component #108 : has  
  
4th Party Component #108 License : MIT  
-----  
  
4th Party Component #108 Copyright:
```

```
-----  
Copyright (c) 2013 Thiago de Arruda  
  
=====
```

```
4th Party Component #109 : object-inspect  
  
4th Party Component #109 License : MIT  
-----  
  
4th Party Component #109 Copyright:  
-----  
Copyright (c) 2013 James Halliday  
  
=====
```

```
4th Party Component #110 : traverse  
  
4th Party Component #110 License : MIT  
-----  
  
4th Party Component #110 Copyright:  
-----  
Copyright 2010 James Halliday (mail@substack.net)  
  
=====
```

```
4th Party Component #111 : url  
  
4th Party Component #111 License : MIT  
-----  
  
4th Party Component #111 Copyright:  
-----  
Copyright 2014 Joyent, Inc. and other Node contributors.  
  
=====
```

```
4th Party Component #112 : punycode  
  
4th Party Component #112 License : MIT  
-----  
  
4th Party Component #112 Copyright:  
-----  
Copyright Mathias Bynens <https://mathiasbynens.be/>  
  
=====
```

```
4th Party Component #113 : querystring  
  
4th Party Component #113 License : MIT  
-----  
  
4th Party Component #113 Copyright:  
-----
```

Copyright 2012 Irakli Gozalishvili

=====
=====

4th Party Component #114 : url-parse

4th Party Component #114 License : MIT

4th Party Component #114 Copyright:

Copyright (c) 2015 Unshift.io, Arnout Kazemier, the Contributors.

=====
=====

4th Party Component #115 : querystringify

4th Party Component #115 License : MIT

4th Party Component #115 Copyright:

Copyright (c) 2015 Unshift.io, Arnout Kazemier, the Contributors.

=====
=====

4th Party Component #116 : requires-port

4th Party Component #116 License : MIT

4th Party Component #116 Copyright:

Copyright (c) 2015 Unshift.io, Arnout Kazemier, the Contributors.

=====
=====

D.30 Commons Compress 1.26.0

Commons Compress

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.

Notice:

Apache Commons Compress
Copyright 2002-2023 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

=====
=====

4th party dependency license : commons-io

License: Apache License

====

Copyright:
Apache Commons IO
Copyright 2002-2023 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

=====
=====

4th party dependency license : commons-lang3

License: Apache License

Copyright:
Apache Commons Lang
Copyright 2001-2024 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (<https://www.apache.org/>).

Index

A

authentication
 against WebLogic user repositories, [2-94](#)

C

CREATE_CLIENT procedure, [6-1](#)
CREATE_ROLE procedure, [4-1](#), [5-1](#)
CREATE_SERVICE procedure (deprecated), [4-1](#)
cURL, [2-6](#)

D

defaults.xml
 enabling detailed request error messages,
 [C-1](#)
DEFINE_HANDLER procedure, [4-4](#), [5-2](#)
DEFINE_MODULE procedure, [4-6](#), [5-4](#)
DEFINE_PARAMETER procedure, [4-7](#), [5-5](#)
DEFINE_PRIVILEGE procedure, [4-9](#), [5-7](#)
DEFINE_SERVICE procedure, [4-11](#), [5-10](#)
DEFINE_TEMPLATE procedure, [4-14](#), [5-13](#)
DELETE_CLIENT procedure, [6-3](#)
DELETE_MODULE procedure, [4-15](#), [5-14](#)
DELETE_PRIVILEGE procedure, [4-16](#), [5-15](#)
DELETE_ROLE procedure, [4-16](#), [5-15](#)
DROP_REST_FOR_SCHEMA procedure, [4-17](#),
 [5-16](#)

E

ENABLE_OBJECT procedure, [4-17](#), [5-16](#), [5-24](#),
 [5-25](#)
ENABLE_SCHEMA procedure, [4-19](#), [5-18](#)

G

GRANT_CLIENT_ROLE procedure, [6-3](#)

J

JSON
 using to pass parameters, [2-37](#)

O

OAuth package
 CREATE_CLIENT, [6-1](#)
 DELETE_CLIENT, [6-3](#)
 GRANT_CLIENT_ROLE, [6-3](#)
 RENAME_CLIENT, [6-4](#)
 REVOKE_CLIENT_ROLE, [6-5](#)
 UPDATE_CLIENT, [6-5](#)
Oracle REST Data Services
 PL/SQL API
 PL/SQL API for Oracle REST Data
 Services, [2-99](#)
Oracle REST Data Services configuration file
 enabling detailed request error messages,
 [C-1](#)
Oracle REST Data Services package
 CREATE_SERVICE (deprecated), [4-1](#)
ORDS package
 CREATE_ROLE, [4-1](#), [5-1](#)
 DEFINE_HANDLER, [4-4](#), [5-2](#)
 DEFINE_MODULE, [4-6](#), [5-4](#)
 DEFINE_PARAMETER, [4-7](#), [5-5](#)
 DEFINE_PRIVILEGE, [4-9](#), [5-7](#)
 DEFINE_SERVICE, [4-11](#), [5-10](#)
 DEFINE_TEMPLATE, [4-14](#), [5-13](#)
 DELETE_MODULE, [4-15](#), [5-14](#)
 DELETE_PRIVILEGE, [4-16](#), [5-15](#)
 DELETE_ROLE, [4-16](#), [5-15](#)
 DROP_REST_FOR_SCHEMA, [4-17](#), [5-16](#)
 ENABLE_OBJECT, [4-17](#), [5-16](#), [5-24](#), [5-25](#)
 ENABLE_SCHEMA, [4-19](#), [5-18](#)
 PUBLISH_MODULE, [4-20](#), [5-19](#)
 RENAME_MODULE, [4-20](#), [5-20](#)
 RENAME_PRIVILEGE, [4-21](#), [5-21](#)
 RENAME_ROLE, [4-22](#), [5-21](#)
 SET_MODULE_ORIGINS_ALLOWED, [4-22](#),
 [5-22](#)
 SET_URL_MAPPING, [4-23](#), [5-23](#)

P

passing parameters
 using JSON, [2-37](#)
 using query strings, [2-46](#)
 using route patterns, [2-42](#)

PUBLISH_MODULE procedure, [4-20](#), [5-19](#)

R

RENAME_CLIENT procedure, [6-4](#)
RENAME_MODULE procedure, [4-20](#), [5-20](#)
RENAME_PRIVILEGE procedure, [4-21](#), [5-21](#)
RENAME_ROLE procedure, [4-22](#), [5-21](#)
resource handler, [2-3](#)
resource module, [2-3](#)
resource template, [2-3](#)
REST APIs
 getting started with, [2-2](#)
RESTful services
 about, [2-2](#)
 integrating with existing group/role models,
 [2-96](#)
 sample services, [2-67](#)
 terminology, [2-3](#)
 user roles, [2-92](#)
 using cURL, [2-6](#)
REVOKE_CLIENT_ROLE procedure, [6-5](#)
role-mapping.xml file, [2-96](#)
route pattern, [2-3](#)

S

SET_MODULE_ORIGINS_ALLOWED procedure,
 [4-22](#), [5-22](#)
SET_URL_MAPPING procedure, [4-23](#), [5-23](#)

T

troubleshooting, [C-1](#)
 enabling detailed request error messages,
 [C-1](#)

U

UPDATE_CLIENT procedure, [6-5](#)
upsert operation, [2-16](#)
URI pattern, [2-3](#)
URI template, [2-3](#)
user roles for RESTful services, [2-92](#)
using query strings
 to pass optional parameters, [2-46](#)
using route patterns
 for passing required parameters, [2-42](#)