

ORACLE®

**ORACLE
OPEN
WORLD**

Oracle REST Data Services

CONN6662 Oracle Database REST APIs

Colm Divilly
Consulting Member of Technical Staff
Oracle, Database Tools

October 3, 2017

October 1–5, 2017
SAN FRANCISCO, CA



ORACLE

Safe Harbor Statement

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

Oracle REST Data Services

CON6667 Securing Your RESTful Services

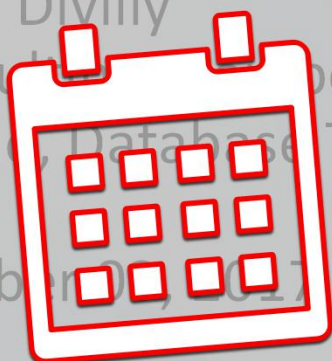
October 1–5, 2017
SAN FRANCISCO, CA

Colm Divilly

Consulting Member of Technical Staff

Oracle, Database Tools

October 1, 2017



*Don't miss Part II
Right after this session
@12.45 in this room*



Oracle Database Exadata Express Cloud Service

Cloning & Lifecycle Management

Tuesday, 3:45-4:30
Moscone West,
Room #3012

ORACLE
OPEN
WORLD

October 1–5, 2017
SAN FRANCISCO, CA



ORACLE®

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

CON6662 - Oracle Database REST APIs - Agenda

- 1 Overview of Oracle REST Data Services
- 2 Demo: REST Enabling a Database Table
- 3 Demo: REST Enabling a Database View
- 4 New Feature: Enabling a Database Stored Procedure
- 5 New Feature: Introducing REST Enabled SQL
- 6 Future Direction

Oracle REST Data Services Overview

Why do we need REST?

What is the point of REST ?

- REST is really all about just one thing:
 - Usability!
- REST is a way to make APIs:
 - Uniform
 - Self Describing
 - Navigable
- An API with these traits will be easy to use.

What is Oracle REST Data Services?

- ORDS is a Java application that connects Oracle Database to the Web
- Provides a means to:
 - **Declaratively enable database tables, views and procedures** to make them accessible in a REST style over the web
 - Declaratively define REST APIs that execute **user defined SQL queries or PL/SQL statements**
 - **Execute arbitrary SQL/PLSQL** over the Web. Think JDBC/ODBC for the Web!
- All of this is done **securely** with a declarative security model

New Release Model

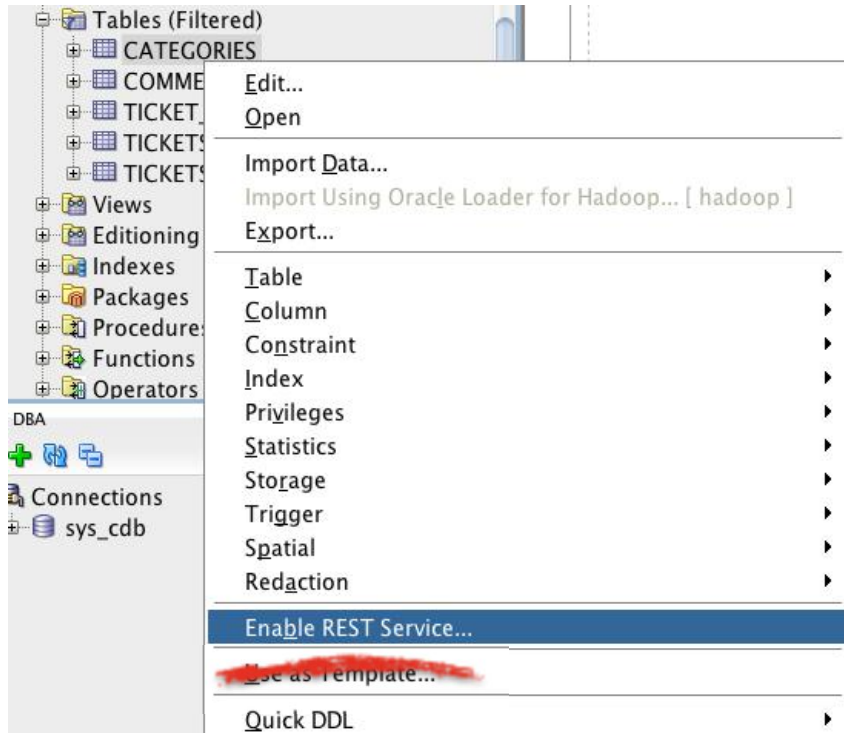
Aligning with Oracle Cloud Release Model

- The next version of ORDS will be **17.3/17.4**
- After that we plan **quarterly releases**, so 18.1, 18.2, 18.3, 18.4
- If there is a critical issue we may release an **intermediate release** so 18.1.1 etc.
- This aligns with **Oracle Cloud**, enabling us to iterate faster, and **bring new features to market sooner**

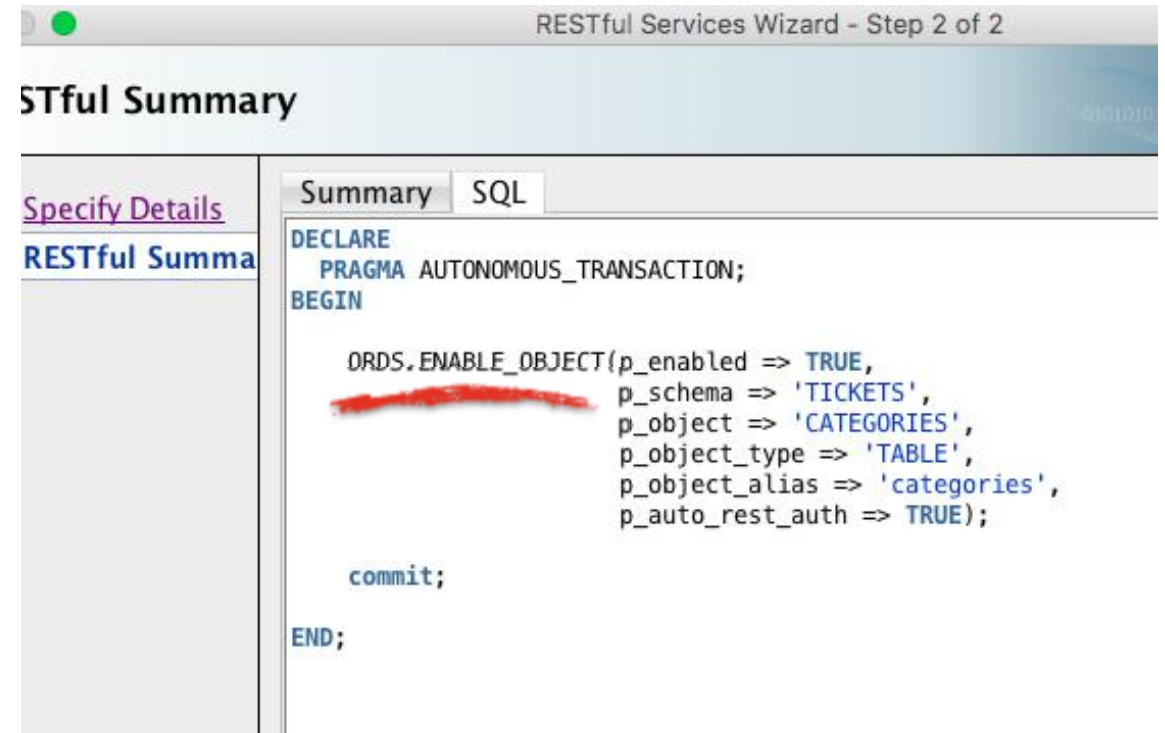
A man with a beard and mustache, wearing a dark suit, light blue shirt, and dark tie, is looking at a tablet. The background is a bokeh of lights, suggesting an indoor setting at night. The image is overlaid with a teal geometric pattern.

REST Enabling a Database Table

REST Enable a Database Table



In SQL Developer, right click on a table, and choose 'Enable REST Service...'



A Wizard guides you through enabling a table, click the SQL tab on the last screen to see the PL/SQL used to enable the table

REST Enabling a View

Views > Tables

REST Enable a View

Columns Data | Grants | Dependencies | Details | Triggers | SQL | Errors

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID
TICKET_ID	NUMBER	Yes	(null)	1
\$related	VARCHAR2(51)	Yes	(null)	2
TITLE	VARCHAR2(255)	Yes	(null)	3
ACTIVITY	VARCHAR2(255)	Yes	(null)	4
CREATED_BY	VARCHAR2(255)	Yes	(null)	5
CREATED_ON	DATE	Yes	(null)	6

RESTful Summary

Specify Details RESTful Summary

```
DECLARE
PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
  ORDS.ENABLE_OBJECT(p_enabled => TRUE,
                    p_schema => 'TICKETS',
                    p_object => 'TICKETS_FEED',
                    p_object_type => 'VIEW',
                    p_object_alias => '$related',
                    p_auto_rest_auth => TRUE);

  commit;
END;
```

- You can enable a view in the same manner as a table
- In practice tables are **most often too granular** to be REST enabled, views can aggregate tables together into useful web resources.
- Also you can alias view column names to turn them into **hyperlinks** (see "\$related" in this example)

A man with a beard and mustache, wearing a dark suit, light blue shirt, and dark tie, is looking at a tablet. The background is a bokeh of lights, suggesting an indoor setting at night. The image is overlaid with a teal geometric pattern.

Enabling a Stored Procedure

Enable a Procedure

The screenshot shows the Oracle SQL Developer interface. On the left, a tree view displays database objects, with 'DO_SOMETHING' selected under 'Procedures'. The main window is titled 'RESTful Services Wizard - Step 2 of 2' and shows the 'RESTful Summary' tab. The 'Code' tab displays the following SQL:

```
1 create or replace procedure do_something
2 (
3   an_argument in varchar2
4   , another_argument in number
5 ) as
6 begin
7   null;
8 end do_something;
```

The 'Summary' tab shows the following SQL:

```
DECLARE
PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
  ORDS.ENABLE_OBJECT(p_enabled => TRUE,
                    p_schema => 'TICKETS',
                    p_object => 'DO_SOMETHING',
                    p_object_type => 'PROCEDURE',
                    p_object_alias => 'do_somethi',
                    p_auto_rest_auth => TRUE);

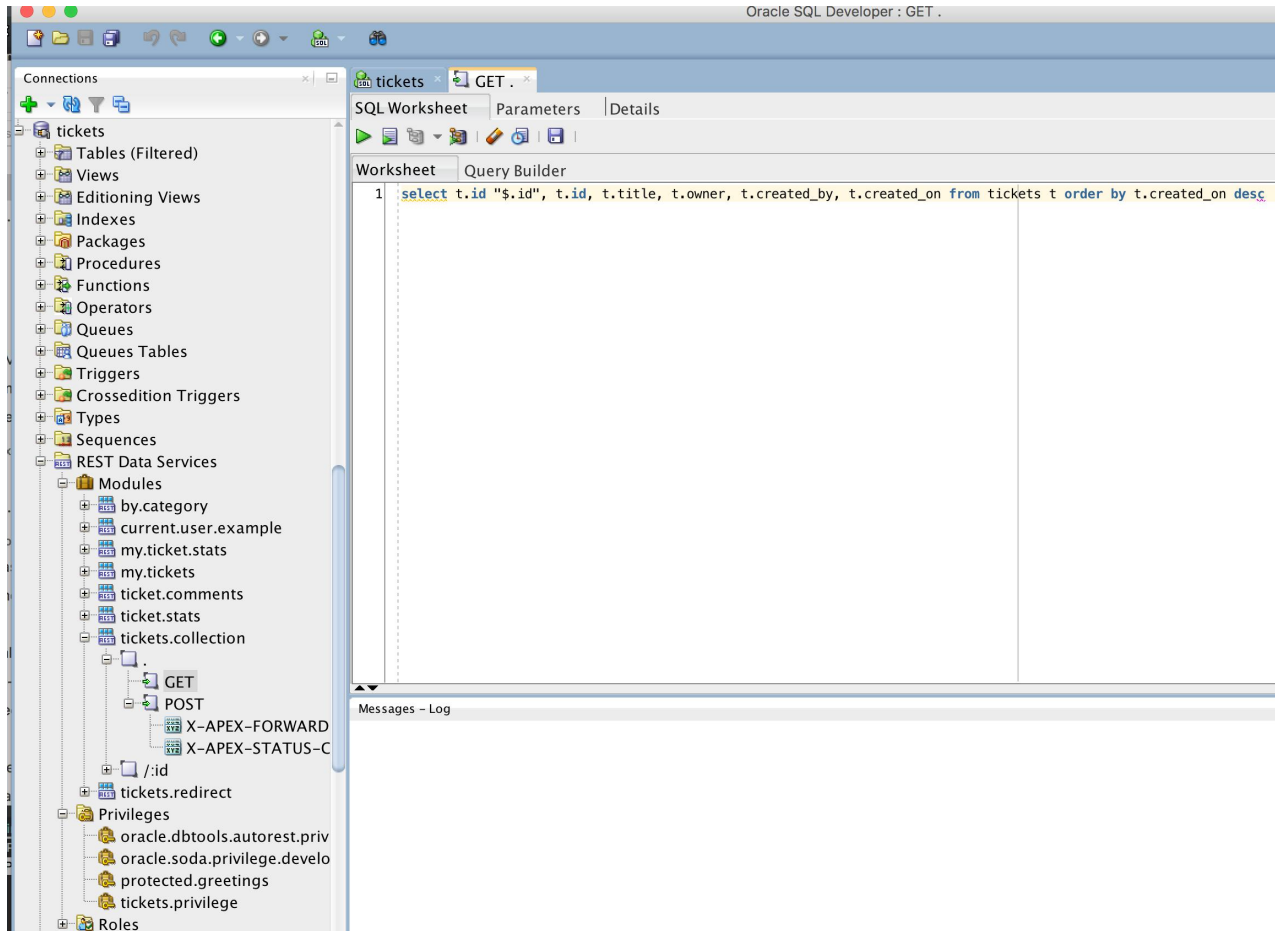
  commit;
END;
```

- You can enable a procedure in a similar manner
- Note the database has much less metadata about what a procedure does. ORDS can't tell if procedure is idempotent or safe, so it can only be invoked via a **POST method**
- This makes this facility much less RESTful than an enabled view. It's more like **RPC over HTTPS+JSON**
- You can also enable **functions** and **packages**

A man and a woman are sitting at a desk in a meeting. The woman is holding a small red and white striped cone. The man is looking at the cone. The background is a blurred office setting with a laptop and a stack of papers. The entire image is overlaid with a red gradient.

Resource Modules

Resource Modules



- Gives most control/expressiveness over the behaviour and structure of the REST API
- Bind Parameterized SQL and PL/SQL to URLs and HTTP methods
- Define hyperlinks by prefixing column name with '\$'
- Generate arbitrary responses with 'Media Resource' Source Type

REST Enabling SQL



REST Enabled SQL

Sometimes pre-defined REST endpoints aren't enough

- REST Enabled SQL lets you POST sql scripts to be evaluated by ORDS and the results are returned in a rich well structured JSON document.
- Many use cases:
 - Complex queries not expressible using URL query string
 - Batch Operations
 - Tailoring output based on channel/client in use
 - Ad-hoc queries and statements during development phase of lifecycle

Future Directions



Future Directions

- REST Enabled SQL will empower a range of interesting applications
 - Web based version of SQL Developer
 - JDBC Driver that communicates over HTTP
 - Remote Data Access in Application Express 5.2
- In addition we want to provide REST APIs for all database operations
 - Create, Remove, plug, unplug a PDB
 - Data Import/Export
 - Etc etc.

Integrated Cloud

Applications & Platform Services

ORACLE®