Deploy RESTful Services in Your Database

Tuesday February 08, 1:10 PM MST

Robert Marz
Databees.

rmoug.org

Robert Marz – Independent Consultant

Primary Role

Senior Technical Architect
with database centric view of the world

DOAG
(German Oracle User Group)

Active Member of Database Community
Responsible for Cloud Topics

Databees.

ORACLE ACE

@RobbieDatabee

robbie.databee.org

robert.marz
@databee.org

ORACLE
ACE Program

500+ technical experts helping peers globally

The **Oracle ACE Program** recognizes and rewards community members for their technical contributions in the Oracle community

3 membership tiers

ORACLE ACE Director | ORACLE ACE | ORACLE ACE Associate

For more details on Oracle ACE Program:
bit.ly/OracleACEProgram

Oracle Groundbreakers

Nominate

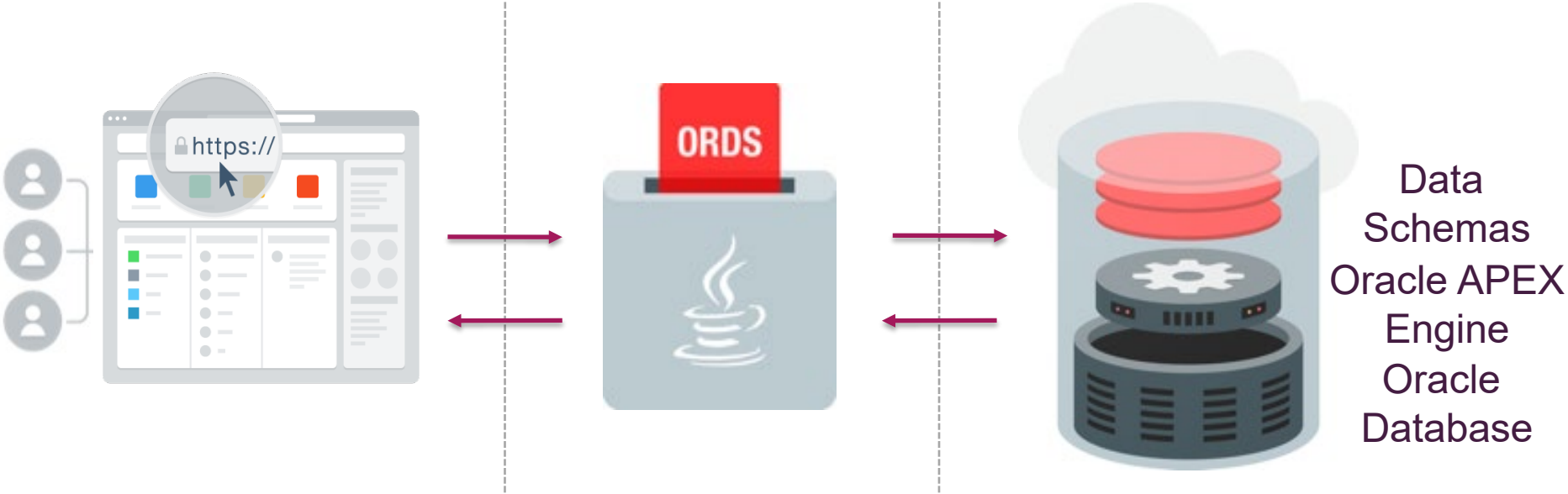
yourself or someone you know:

acenomination.oracle.com

Connect: **oracle-ace_ww@oracle.com** Facebook.com/oracleaces @oracleace

RESTful Apps & Oracle ORDS: An Overview

Oracle Application Express 3-Tier Architecture



Browser

Mid Tier

Database Tier

ORDS = Oracle REST Data Services

Oracle REST Data Services (ORDS)

Evolved

from APEX Listener

Deploy in Application Server

- Tomcat
- Glassfish (deprecated)
- WebLogic

ords.war

Java Web Archive

Standalone mode

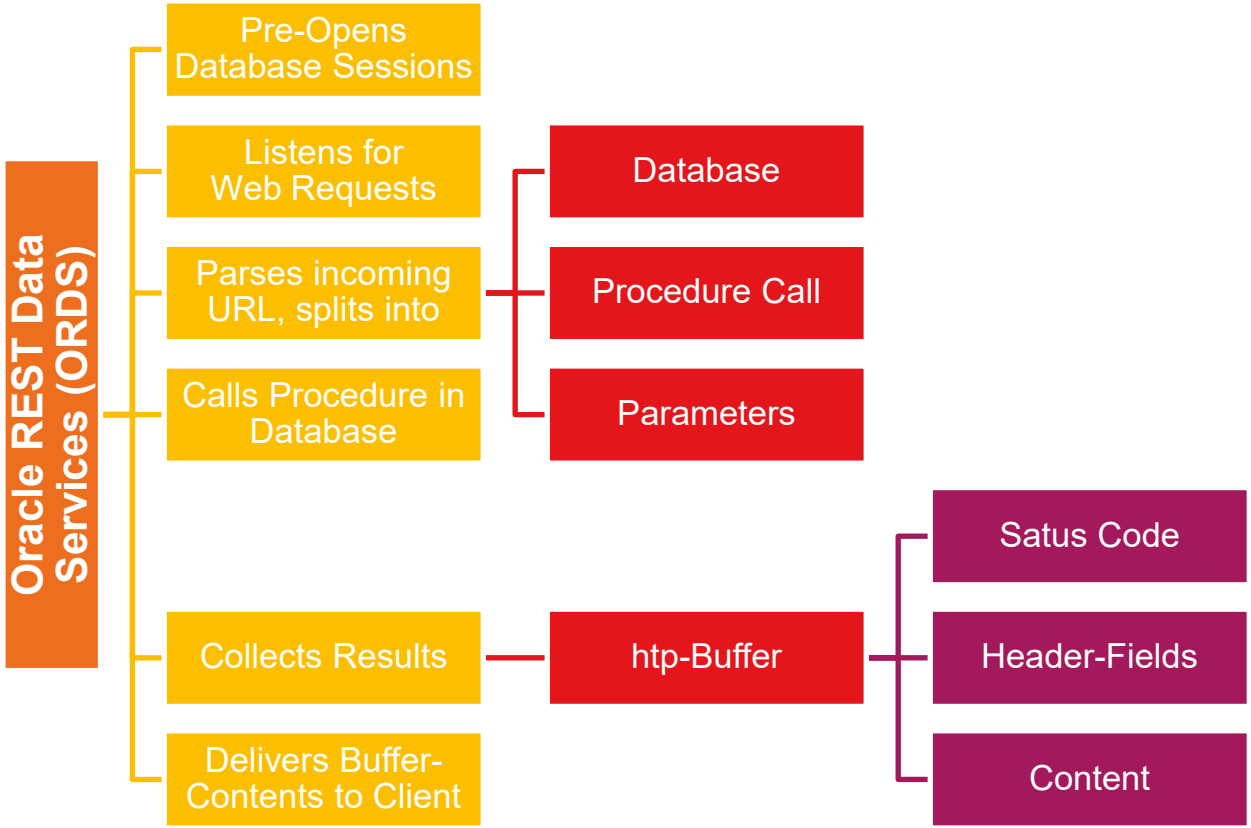
Brings own http-server
Supported for Production

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

What is REST

REST	Representational state transfer	doctoral dissertation by Roy Fielding, 2000
	programming paradigm	distributed systems Web services.
RESTful Applications	implements 6 constraints	**Uniform Interface (API via URIs)** **Stateless**, Client-Server, Layered System **Cacheable**, Code on Demand
Implementation	Transport protocol	http(s)
	content	JSON Documents

What is ORDS' job?



Demo – Build a RESTfull Service in 100 seconds

Prerequisites

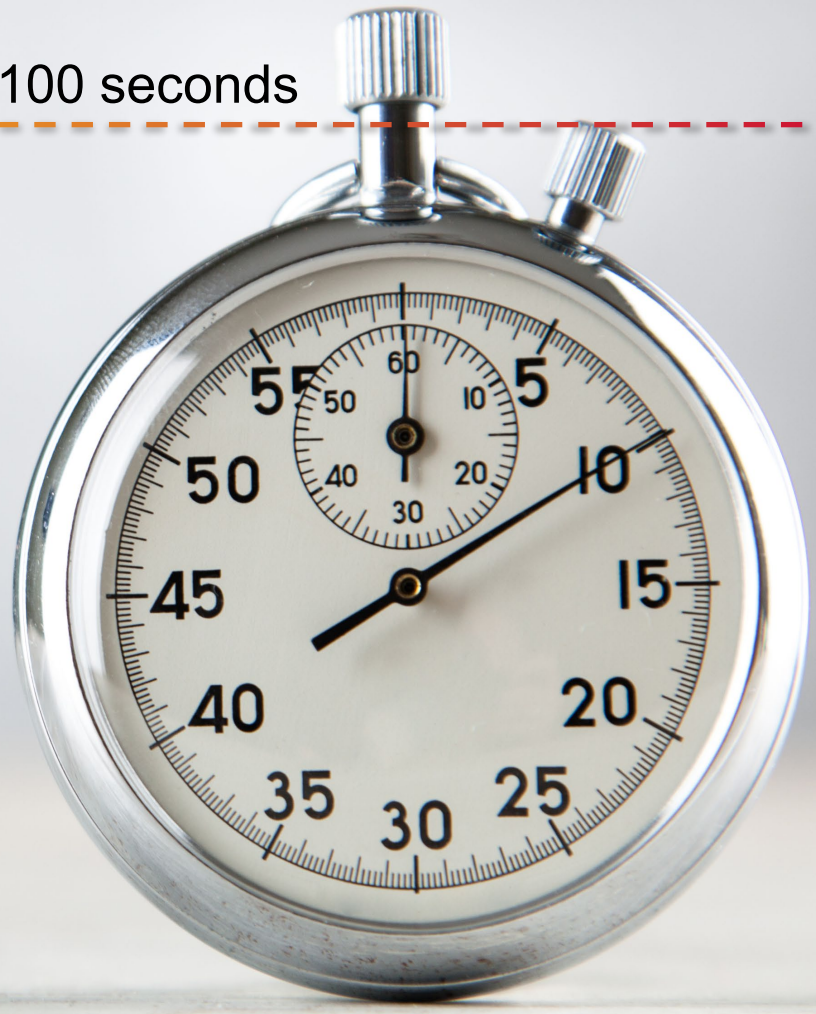
- Running Oracle DB 11gR2 or higher
  - Schema with Table & Data
  - DBA Access
- Installed & Configured ORDS

Tools

- Insomnia REST Client
- SQL Developer
- Oracle Visual Builder Add-in for Excel

Demo Sequence

- REST enable Schema
- REST enable Table
- Use REST Client to test
- Show Visual Builder Add-In



Build a RESTfull Service in 100 seconds

ORDS Autorest – REST enable Tables or Procedures

Autorest

- REST enable Table or Procedure very easy
- Read / Write
- Stateless / no transaction
- SQLDeveloper Wizards
- PL/SQL API

ORDS PL/SQL API: enable Schema (simple)

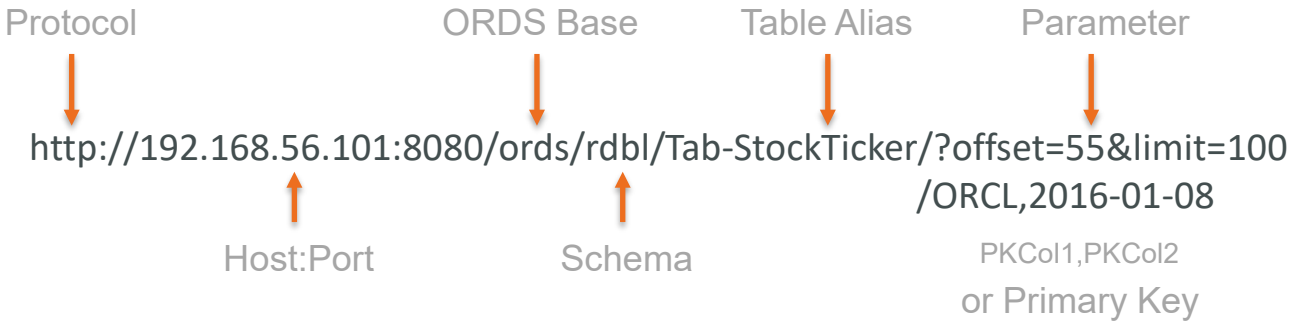
```
begin
   ords.enable_schema;
   commit;  -- This commit is important!
end;
/
```

ORDS PL/SQL API: enable Object

```
begin
  ords.enable_object (
    P_ENABLED      => true,
    P_SCHEMA       => 'RESTDBLINKSPROV',
    P_OBJECT       => 'STOCKTICKER',
    P_OBJECT_TYPE  => 'TABLE',
    P_OBJECT_ALIAS => 'tab-StockTicker',
    P_AUTO_REST_AUTH => false );
  commit;  -- This commit is important, too!
end;
/
```

Case Sensitive

Anatomy of a ORDS AutoREST URL



HTTP Method	ORDS AutoREST Action
GET	Retrieve Data – Single Row or Rowset
PUT	Insert or Modify Row
POST	Bulk Insert csv-data
DELETE	Delete Row

Interpreting the ORDS AutoREST Responses (1)


```
{
    "items": [
        {
            "symbol": "TDC",
            "id1": 56,
            "tstamp": "2017-05-06T23:27:00Z",
            "price": 20.625,
            "links": [
                {
                    "rel": "self",
                    "href": "http://127.0.0.1:8080/ords/rdbl/Tab-StockTicker/56"
                }
            ]
        },
        {
            "symbol": "ORCL",
            "id1": 57,
            "tstamp": "2017-05-06T23:28:00Z",
            "price": 42,
```


Table Rows

Columns & Values

Interpreting the ORDS AutoREST Responses (2)


```
{
    "items": [↔],
    "hasMore": true,
    "limit": 2,
    "offset": 55,
    "count": 2,
    "links": [
        {↔},
        {↔},
        {↔},
        {↔},
        {
            "rel": "next",
            "href": "http://127.0.0.1:8080/ords/rdbl/Tab-StockTicker/?offset=57&limit=2"
        },
        {↔}
    ]
}
```


Hypermedia **a**s **t**he **E**ngine **o**f **A**pplication **S**tate (HATEOAS)

AutoREST PL/SQL

PL/SQL

Procedures & Functions
Standalone or in Packages
No Overloading

POST Method only

More like RPC than RESTful WebService
JSON Document required (can be empty)

JSON Response

All OUT & IN OUT-Parameter Values
Function Return Value

AutoREST PL/SQL vs ORDS REST Module

Similar effort, but
REST Module offers way more flexibility

Designing a REST API

APIs should be human readable

“Programs must be written for people to read, and only incidentally for machines to execute.”

Harold Abelson, Structure and Interpretation of Computer Programs, 1984

This applies to APIs, as well.

API Grammar

Nouns / What?

Your API Objects
  e.g. contracts, cars, VirtualMachines, …
  GET /products
  GET /VirtualMachines/4711

Verbs / How?

http Methods
  e.g GET, POST, PUT, DELETE

Relations

Sub-resources
  e.g DELETE /VirtualMachines/4711/VMDiskMapping/5

HTTP methods: The Verbs

HTTP Methods

Actions	part of http-request
Common Methods	GET, POST, PUT, DELETE
	OPTIONS, HEAD, TRACE
	CONNECT
Expandable	Make up your own

The HTTP-Protocol - methods

Server

```
# Listens on Port 8080 like a Webserver
nc -l 8080

GET /ords/VM/4711 HTTP/1.1
Host: localhost:8080
User-Agent: curl/7.58.0
Accept: */*


POST /ords/VM/4711 HTTP/1.1
Host: localhost:8080
…

TRALALLA /ords/VM/4711 HTTP/1.1
...
```

Client

```
curl \
    http://localhost:8080/ords/VM/4711


curl --request POST \
    http://localhost:8080/ords/VM/4711


curl --request TRALALLA \
     http://localhost:8080/ords/VM/4711
```

HTTP Status Codes

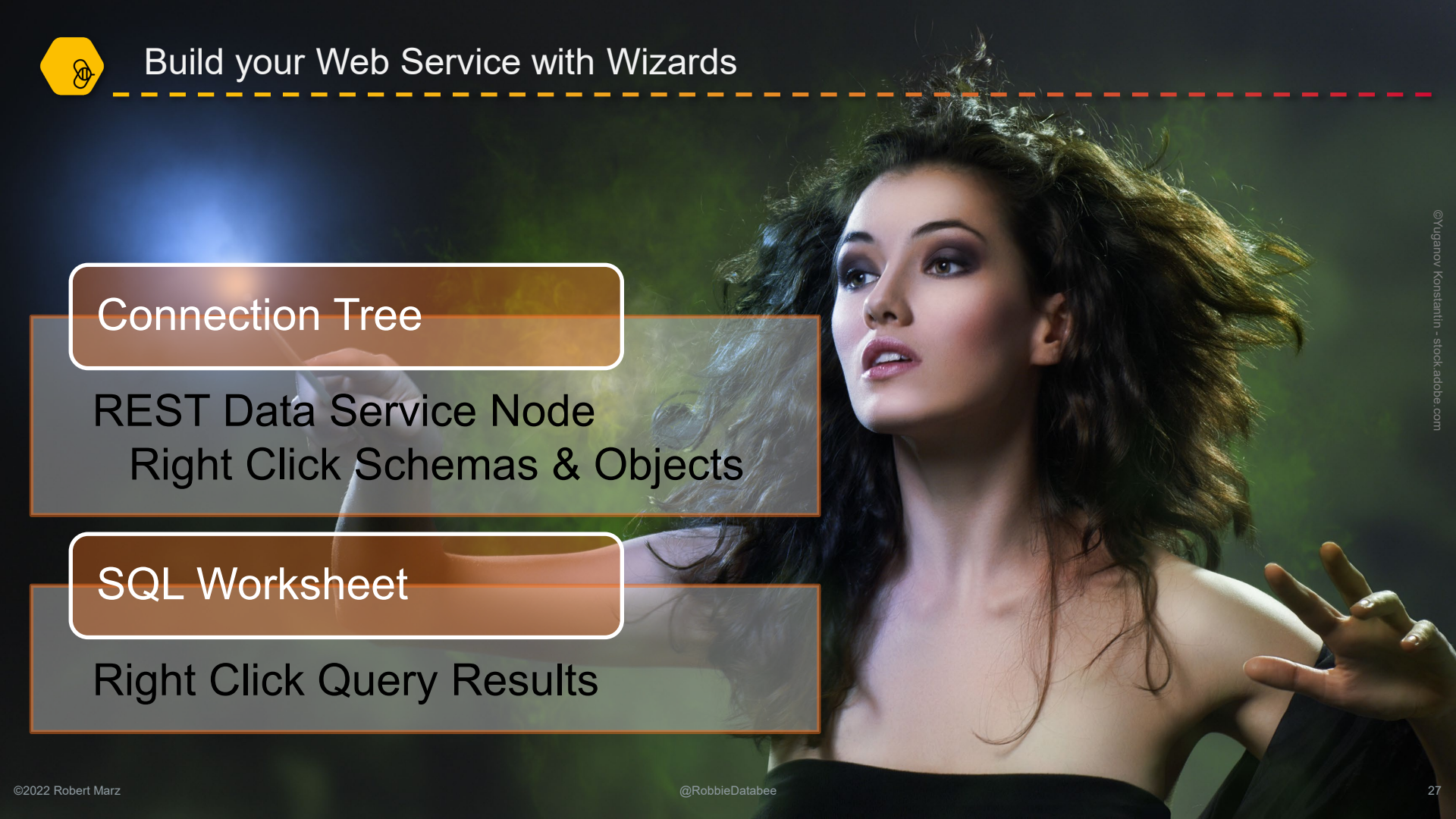
1xx Informational „Hold on“	100 Continue
	101 Switching Protocols
	102 Processing
2xx Success „Here you go“	200 OK
	201 Created
	208 Already Reported
3xx Redirection „Go away“	301 Moved Permanently
	304 Not modified
	307 Temporary Redirect
4xx Client Error „You fucked up“	400 Bad Request
	401 Unauthorized
	404 Not Found
5xx Server Error „I fucked up“	500 Internal Server Error
	502 Bad Gateway
	503 Service Unavailable

API Design Best Practices

- Try it, Test it
- Be redundant
- Use nouns, but no verbs, Nouns are plural
- GET method should never alter states
- Use HTTP headers
- Use **H**ypermedia **a**s **t**he **E**ngine **o**f **A**pplication **S**tate (HATEOAS)
- Provide Filtering, Sorting, Field Selection & Paging


Building
ORDS RESTful Services

Build your Web Service with Wizards



Connection Tree

REST Data Service Node
Right Click Schemas & Objects

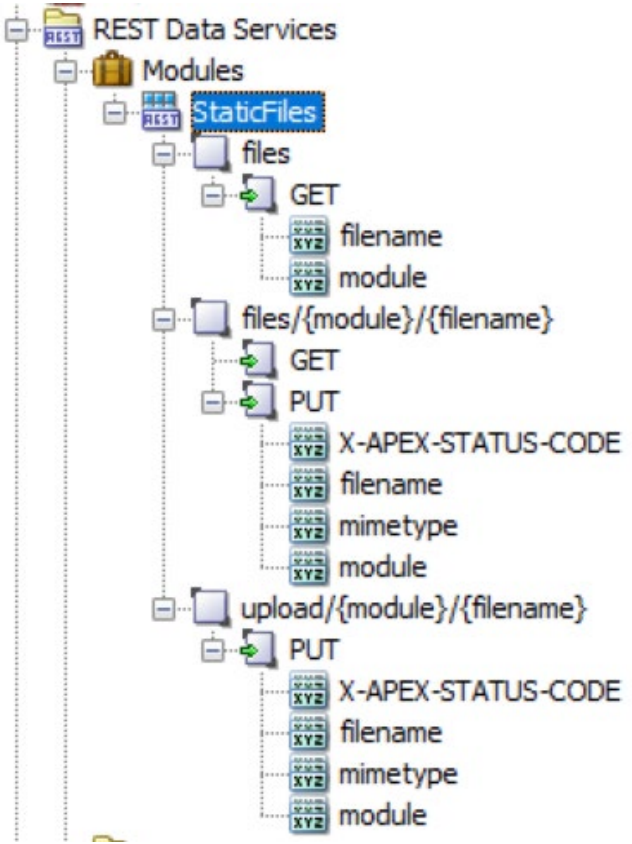
SQL Worksheet

Right Click Query Results

Modules, Template & Handlers

REST Data Service

Module	logically groups a set of URLs	Like a PL/SQL Package
	Name	Include API Version# In Name & URI Prefix
	URI Prefix	Part of the URL
Template	URI Pattern	following URI Prefix from Module may contain parameters
Handler	http-Methods	GET, POST PUT, DELETE one per Template
	Parameters	http-header URI
	Your Code	goes here



Handlers & Parameters

Handler: Source Type	All Handlers:	PL/SQL
	GET Handler (Output Format varies)	Collection Query [Item]; Query [One Row]; Feed
		Media Ressource
Handler: Results	All Handler	JSON
	GET Source-Type	Query: JSON or CSV
		Media Ressource: Binary
Parameter	Types	In
		Out, In / Out
passed as Bind Variables	Source Type	http-Header
		URI

Choose your Option: Basic or OAUTH

Always use https for prod

Credentials and tokens are easy prey

Basic

Let the http-Server do the Job

OAuth

Let someone else authenticate and bear a token

The Idea behind OAUTH

Register your app with someone you trust

- eg. Facebook, Google or GitHub

Redirect Client to external service

Let them do the authentication

Client receives a “bearer” token

Token authorizes your app/API

OAUTH with ORDS

Define

Roles
Privileges

Assign Privileges

by Module
by Resource (URI Pattern)

OAuth Token

Metadata in ORDS-Tables

API Documentation

What is Swagger?

Swagger

Cloud Platform	Swagger Editor
	Swagger UI
	Swagger Codegen
	Swagger Hub
	Swagger Inspector
Documentation Format	donated to Linux Foundation
	renamed to OpenAPI Specification
	JSON or YAML file

Swagger™
Supported by SMARTBEAR

OPENAPI
INITIATIVE

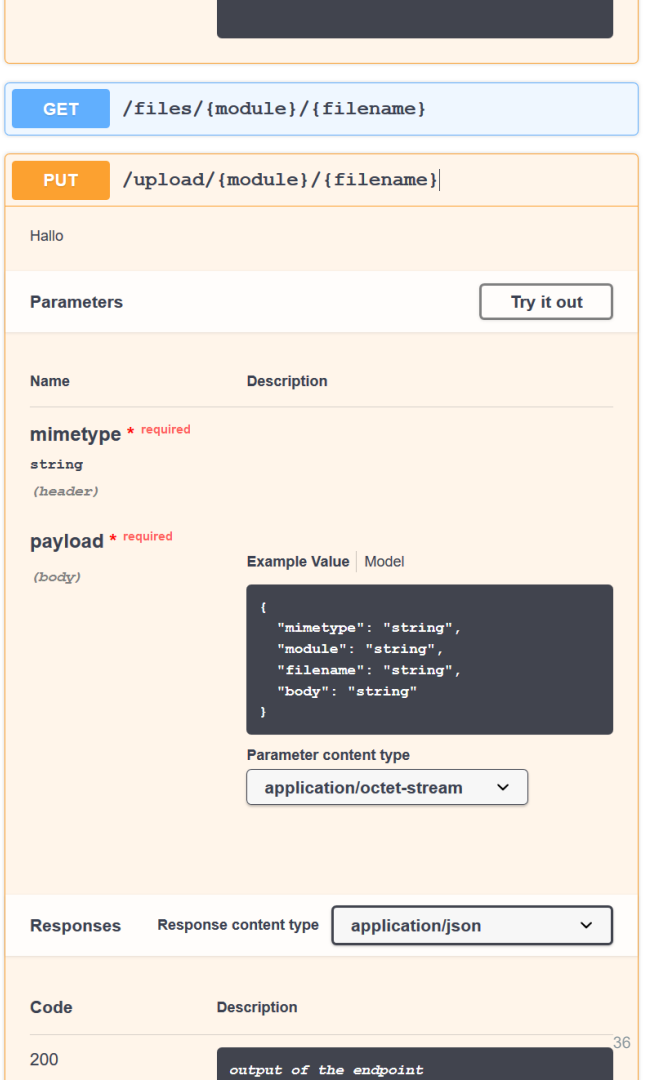
Generate Swagger Doc

URL:

<ords-base>/<schema-alias>/open-api-catalog/<module>/

Example:

```
curl http://localhost:8080/ords/util/open-api-catalog/stati

{"swagger":"2.0","info":{"title":"ORDS generated API for
StaticFiles","version":"1.0.0"},"host":"localhost:8080","ba
chemes":["http"],"produces":["application/json"],"paths":{"
"responses":{"200":{"description":"output of the
endpoint","schema":{"type":"object","properties":{}}}},"con
stream"],"parameters":

[...]
```



Backup & Version Control

ORDS Metadata

ORDS RESTful Services

Metadata only	Rows in Tables
	Dictionary Views
Maintained	PL/SQL APIs
	SQL Developer Wizards
Reporting	Metadata Views
	SQL Developer, SQLcl

Generate PL/SQL-Files with SQLcl

Service Definitions

Pure Metadata

Export

SQL Developer
- Wizard

SQLcl
- REST export

Version Control

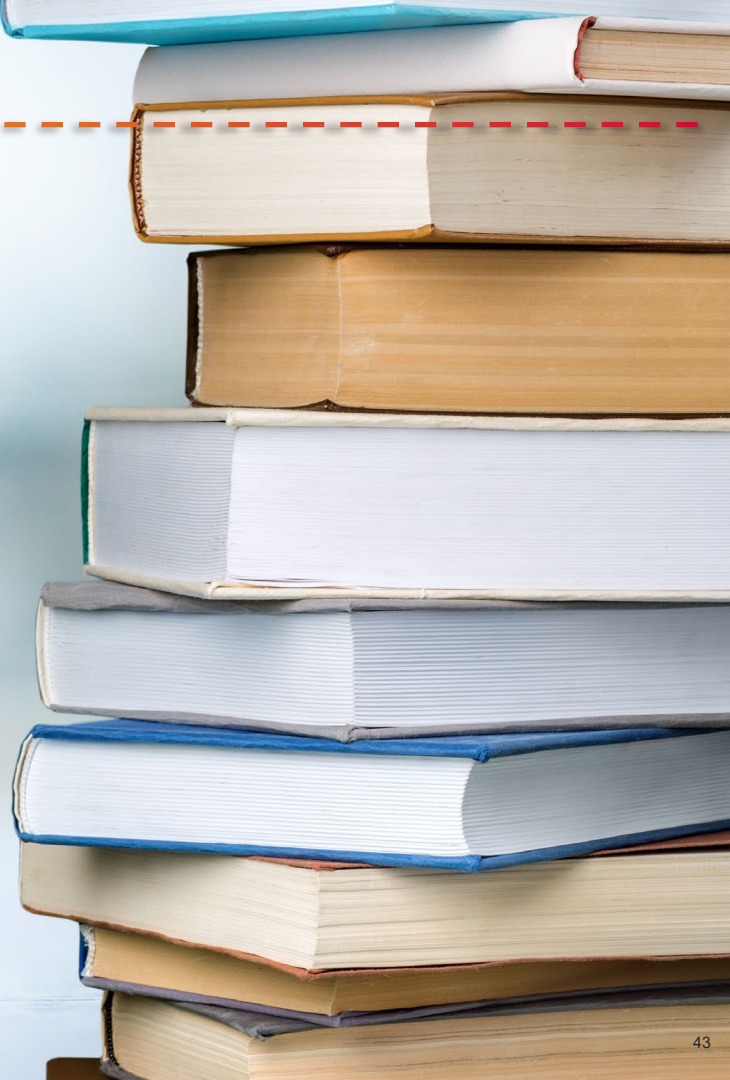
Check into your SCCM (e.g. git)

```
SQLcl: Release 18.3 Production on Sun Mar 17 21:38:17 2019

Copyright (c) 1982, 2019, Oracle.  All rights reserved.

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options


SQL> help rest
REST
------

REST allows to export ORDS 3.X services.

            REST export                   -      All modules
            REST export <module_name>     -      Export a specific module
            REST export <module_prefix>   -      Export a specific module related to the given prefix
            REST modules                  -      List the available modules
            REST privileges               -      List the existing privileges
            REST schemas                  -      List the available schemas
SQL> rest schemas
PARSING_SCHEMA PATTERN STATUS
-------------- ------- -------
APEX_USER      util    ENABLED

SQL> rest modules
NAME         PREFIX   STATUS    ITEMS_PER_PAGE
------------ -------- --------- --------------
StaticFiles /static/ PUBLISHED 25

SQL> rest export
-- Generated by SQLcl REST Data Services 18.3.0.0
-- Exported REST Definitions from ORDS Schema Version 18.4.0.r3541002
-- Schema: APEX_USER   Date: Sun Mar 17 21:39:00 CET 2019
--
BEGIN
  ORDS.ENABLE_SCHEMA(
      p_enabled             => TRUE,
      p_schema              => 'APEX_USER',
      p_url_mapping_type    => 'BASE_PATH',
      p_url_mapping_pattern => 'util',
      p_auto_rest_auth      => FALSE);

  ORDS.DEFINE_MODULE(
      p_module_name    => 'StaticFiles',
```

REST Data Services

Conclusion

Ressources



Oracle REST Data Services:

ORDS Download
ORDS Documentation
Oracle Visual Builder Add-in for Excel

Miscellaneous:

Swagger
Insomnia REST Client
Media Type application/vnd.oracle.resource+json
Oracle DB Tools GitHub ORDS examples

Blogs:

Jeff Smith (Oracle Product Manager)
Kris Rice (Oracle VP Development)

ORDS RESTful Services – Powerful, Flexible & Simple