



Build RESTful Services with ORDS

Robert Marz



Robert Marz – Independent Consultant



Client

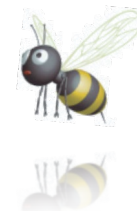
Senior Technical Architect
with database centric view of the world



ORACLE
ACE

its-people

Portfolio Manager Database Technologies
Blog Editor



DOAG (German Oracle User Group)

Active Member of Database Community
Responsible for Cloud Topics



@RobbieDatabee



blog.its-people.de

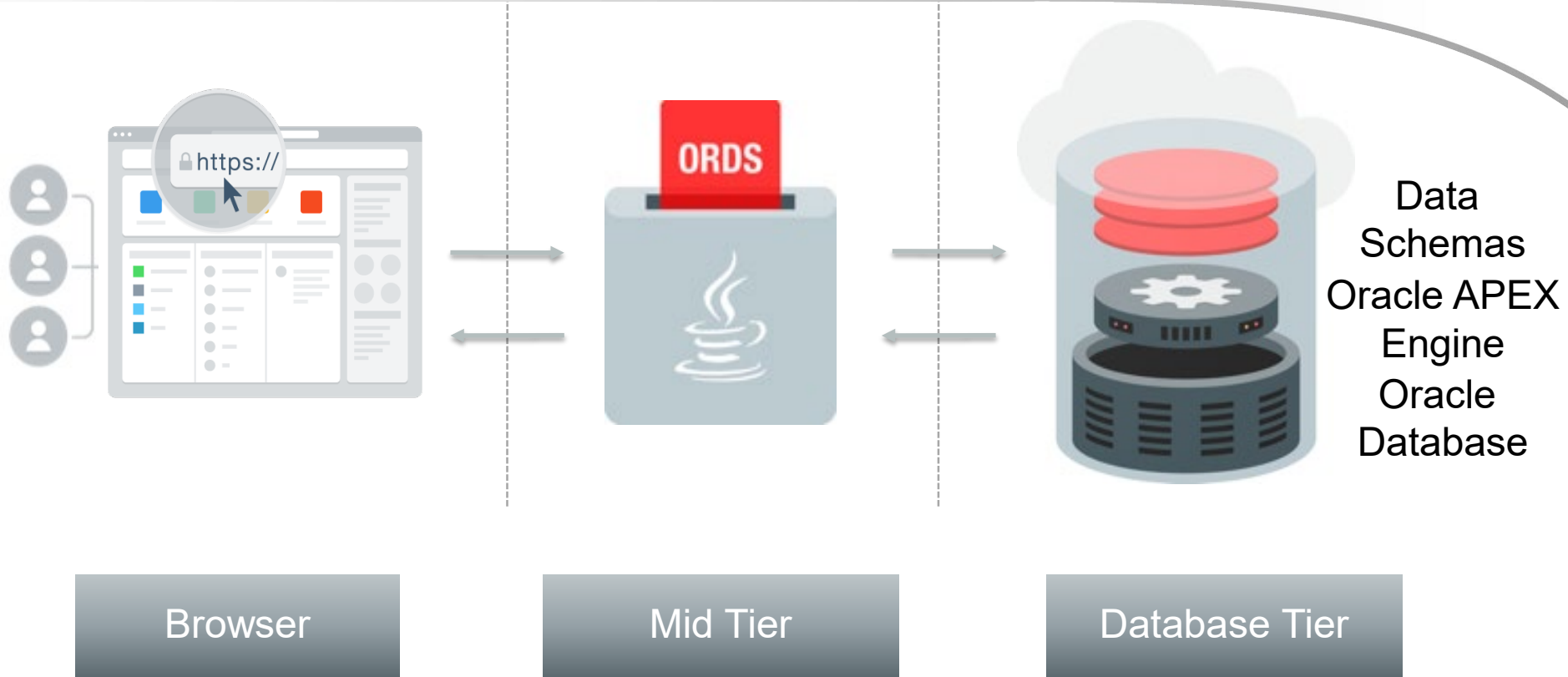


Robert.Marz
@its-people.de

A woman with dark hair in a braid, wearing a black and grey striped long-sleeved shirt, is climbing a grey metal ladder. She is holding binoculars to her eyes and looking upwards. The background is a high-angle view of a city with many skyscrapers, partially obscured by a layer of white clouds. The sky is bright and hazy. A red banner with white text is overlaid on the left side of the image.

RESTful Apps & Oracle ORDS: An Overview

Oracle Application Express 3-Tier Architecture



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`

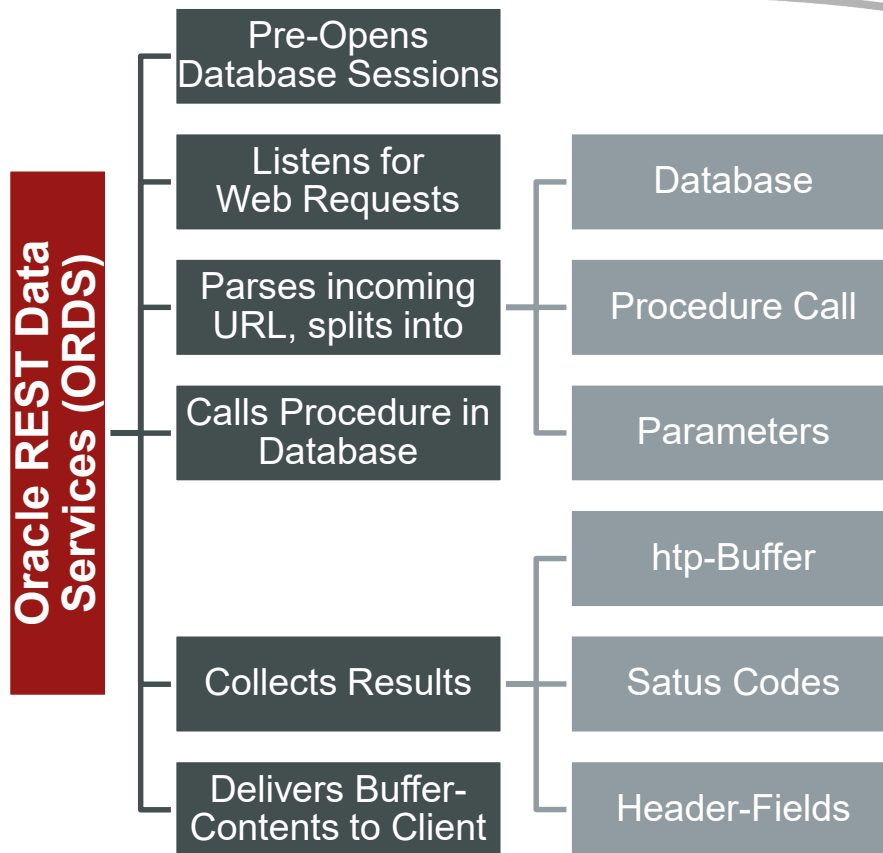


What is REST



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?



A person is sleeping in a bed, wearing a white t-shirt and a grey eye mask. The bed is covered with a light blue blanket. In the background, there is a metal plant stand with several potted plants, including a succulent and a fern. A white mug sits on one of the shelves. To the left, there are dark grey curtains. The overall scene is a peaceful bedroom setting.

RESTful Services with ORDS AutoREST

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



AutoREST

REST enable Table or Procedure very easy

Read / Write

Stateless / no transaction

SQLDeveloper Wizards

PL/SQL API

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

```
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



ORDS Base

Table

Parameter

`http://192.168.56.101:8080/ords/rdbl/Tab-StockTicker/?offset=55&limit=100/`
`ORCL,2016-01-08`

Protocol

Host:Port

Schema

PKCol1,PKCol2
or Primary Key

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)



```
1 {
2   "items": [
3     {
4       "symbol": "TDC",
5       "id1": 56,
6       "tstamp": "2017-05-06T23:27:00Z",
7       "price": 20.625,
8       "links": [
9         {
10          "rel": "self",
11          "href": "http://127.0.0.1:8080/ords/rdb1/Tab-StockTicker/56"
12        }
13      ]
14    },
15    {
16      "symbol": "ORCL",
17      "id1": 57,
18      "tstamp": "2017-05-06T23:28:00Z",
19      "price": 42,
```

Table Rows

**Columns &
Values**

```
1 {
2   "items": [↔],
28  "hasMore": true,
29  "limit": 2,
30  "offset": 55,
31  "count": 2,
32  "links": [
33    {↔},
37    {↔},
41    {↔},
45    {↔},
49    {
50      "rel": "next",
51      "href": "http://127.0.0.1:8080/ords/rdbl/Tab-StockTicker/?offset=57&limit=2"
52    },
53    {↔}
57  ]
58 }
```

Hypermedia as the Engine of Application State (HATEOAS)

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

A large, 3D-rendered graphic of the text 'PL/SQL' in red, bold, sans-serif font. The letters are positioned over a background of several interlocking grey gears. The 'P' and 'L' are on the left, the '/' is in the middle, and the 'S', 'Q', and 'L' are on the right. The entire graphic has a slight shadow and a metallic sheen.



API

Designing a REST API

“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.



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

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

- 
- Try it, Test it
 - Be redundant
 - Use nouns, but no verbs, Nouns are plural
 - GET method should never alter states
 - Use HTTP headers
 - Use **Hypermedia as the Engine of Application State (HATEOAS)**
 - Provide Filtering, Sorting, Field Selection & Paging

Building ORDS RESTful Services



Connection Tree

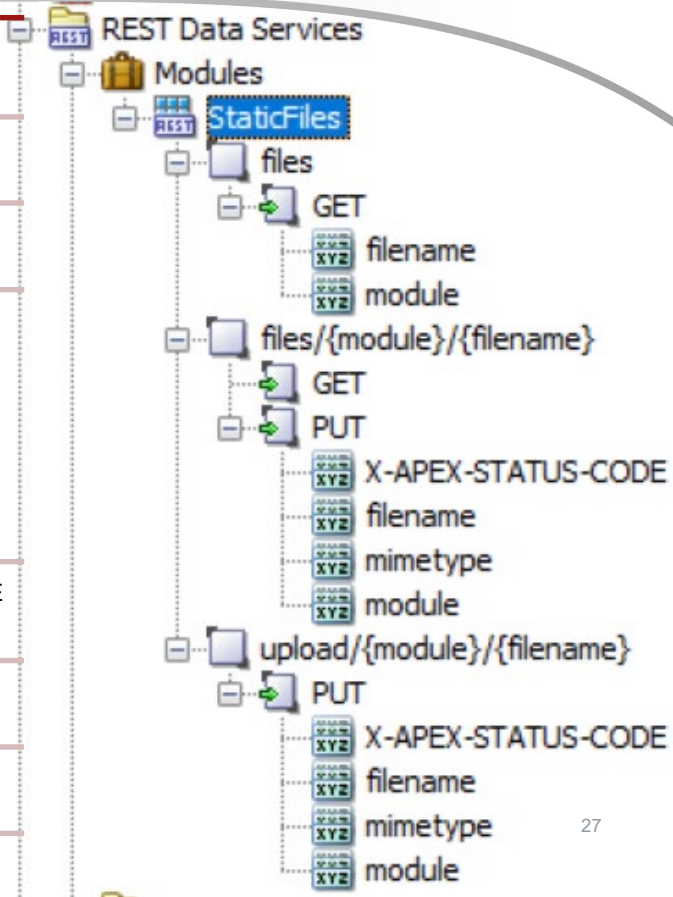
Right Click
Schemas & Objects
REST Data Service Node

SQL Worksheet

Right Click Query Results

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 one per Template
Handler	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

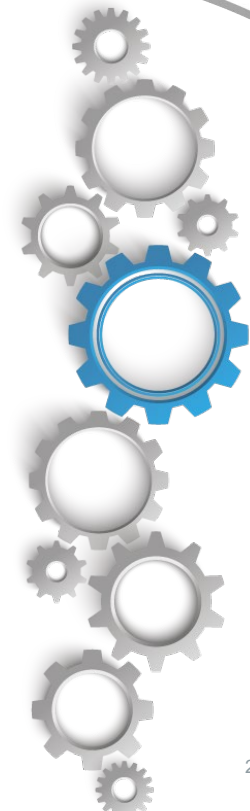
Out, In / Out

passed as
Bind Variables

Source Type

http-Header

URI



A person in a dark suit and blue tie is shown from the chest up, interacting with a futuristic digital interface. The interface consists of two large, circular, glowing blue panels. The left panel features a fingerprint scanner icon, and the right panel features a white padlock icon. A bright yellow light beam connects the two panels. The background is dark with some light flares.

Secure your Webservice

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

Client receives a "bearer" token

Token authorizes your app/API

Redirect Client to external service

Let them do the authentication



Define

Roles
Privileges

Assign Privileges

by Module
by Resource (URI Pattern)

OAuth Token

Metadata in ORDS-Tables



API Documentation

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/static/
```

```
{"swagger":"2.0","info":{"title":"ORDS generated API for StaticFiles","version":"1.0.0"},"host":"localhost:8080","basePath":"/ords/util/static","schemes":["http"],"produces":["application/json"],"paths":{"/":{"put":{"description":null,"responses":{"200":{"description":"output of the endpoint","schema":{"type":"object","properties":{}}}}},"consumes":["application/octet-stream"],"parameters":
```

[...]

The screenshot displays a Swagger UI interface for a PUT endpoint. The endpoint path is `/upload/{module}/{filename}`. The response body is a JSON object with the following structure:

```
{  "mimetype": "string",  "module": "string",  "filename": "string",  "body": "string"}
```

The interface includes a "Parameters" section with a "Try it out" button. The "Responses" section shows a 200 response with a description of "output of the endpoint".

Name	Description
mimetype * required	string (header)
payload * required	(body)

Example Value | Model

```
{  "mimetype": "string",  "module": "string",  "filename": "string",  "body": "string"}
```

Parameter content type: application/octet-stream

Code	Description	35
200	output of the endpoint	



Backup & Version Control



ORDS Schemas

ORDS_PUBLIC_USER

Used by ORDS

connect to DB when

executing RESTful
Services

ORDS_METADATA

Contains All ORDS
Objects

RESTful Services

Definitions

Parameters

Authentication



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



Oracle REST Data Services:

[ORDS Download](#)

[ORDS Documentation](#)

[Oracle Visual Builder Add-in for Excel](#)

Miscellaneous:

[Swagger](#)

[Insomnia REST Client](#)

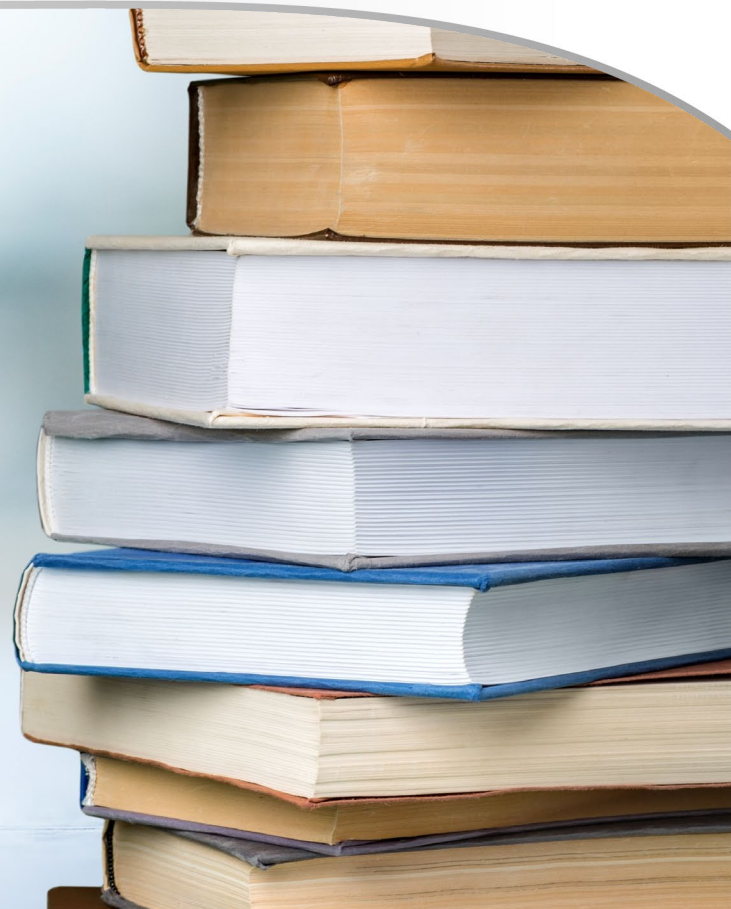
[Media Type application/vnd.oracle.resource+json](#)

[Oracle DB Tools GitHub ORDS examples](#)

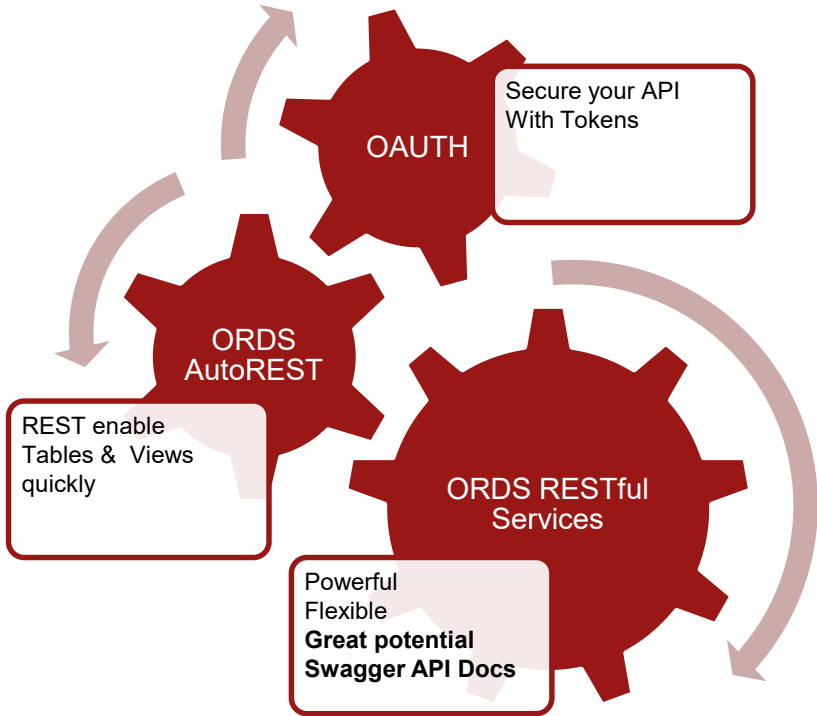
Blogs:

[Jeff Smith \(Oracle Productmanager\)](#)

[Kris Rice \(Oracle VP Development\)](#)



ORDS RESTful Services – Powerful, Flexible & Simple





Vielen Dank für Ihre
Aufmerksamkeit.