

DOAG

datenbank.doag.org

Low Code Development Oracle APEX & REST

Montag, 30. Mai 14:15



Robert Marz
DATABEE
Die IT-Architekten



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




@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



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



**Oracle
Groundbreakers**

Nominate

yourself or someone you know:

acenomination.oracle.com

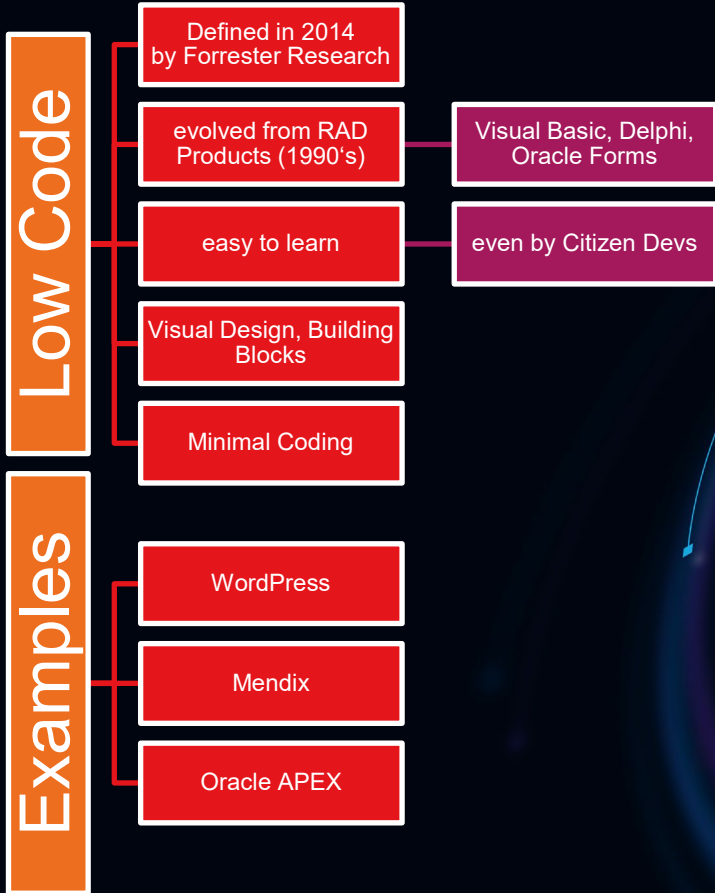
Connect:  oracle-ace_ww@oracle.com

 Facebook.com/oracleaces

 [@oracleace](https://twitter.com/oracleace)



Low Code



ā'pěks

(#orclapex)

Oracle APEX
Low Code App Development



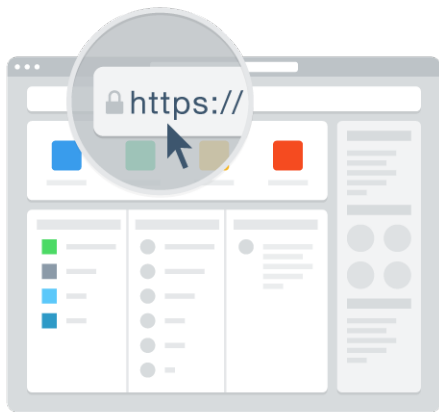
Oracle Application Express

Database-centric web
application
development
framework

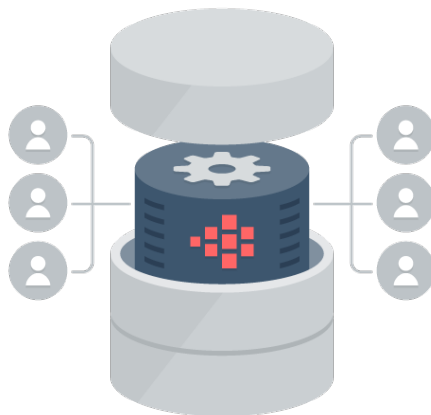




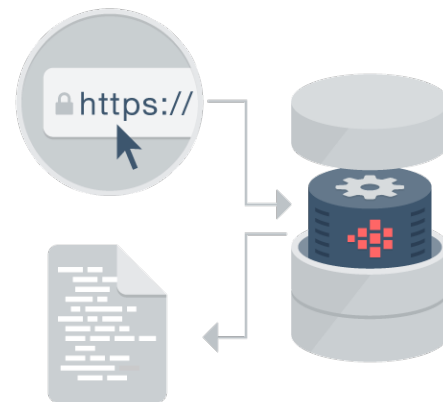
Oracle APEX: Overview



App Development
IDE is a web browser.
**No client software
needed**



App definitions are stored
in the database as meta
data.
**Declarative – No code
generation**



Page generation is
efficient with only one
request and one response.
**Data processing done in
the Database**



Oracle APEX: Application Builder

The screenshot displays the Oracle APEX Application Builder interface. The top navigation bar includes 'ORACLE Application Express', 'Application Builder', 'SQL Workshop', 'Team Development', and 'Apps'. The main workspace is divided into several sections:

- Rendering:** A sidebar on the left with a tree view showing the application structure, including 'Week's Top Developers' and 'Page Events'.
- Grid Layout:** A central area showing a preview of the application page with various components like buttons and text.
- Property Editor - Region:** A panel on the right for configuring the selected region, including 'Identification', 'Source' (with a SQL statement), 'Layout', and 'Appearance'.
- Dashboard:** A view showing a 'Week's Top Developers' chart with a blue donut chart and a 'Page Events' line chart. The donut chart shows 'HILARY' at 15% (1513) and 'SHAKEEB' at 64% (6489). The line chart shows page events from May 31 to June 06.
- Summary Cards:** Three cards at the top right showing '2,998 Pages', '1 Packaged Applications', and '8 Websheets'.
- Navigation:** A bottom navigation bar with icons for 'Regions', 'Items', and 'Buttons'.

Integrated Development Environment (IDE) with Application Development Graphical “Page Designer”
Features tight SQL and PL/SQL integration



Oracle Application Express & REST

REST Support

continously improved

Provides REST SERVICES

similar to ORDS REST Services

Consumes REST Services

e.g. samples REST API
Build Interactive Grids
based on REST Services





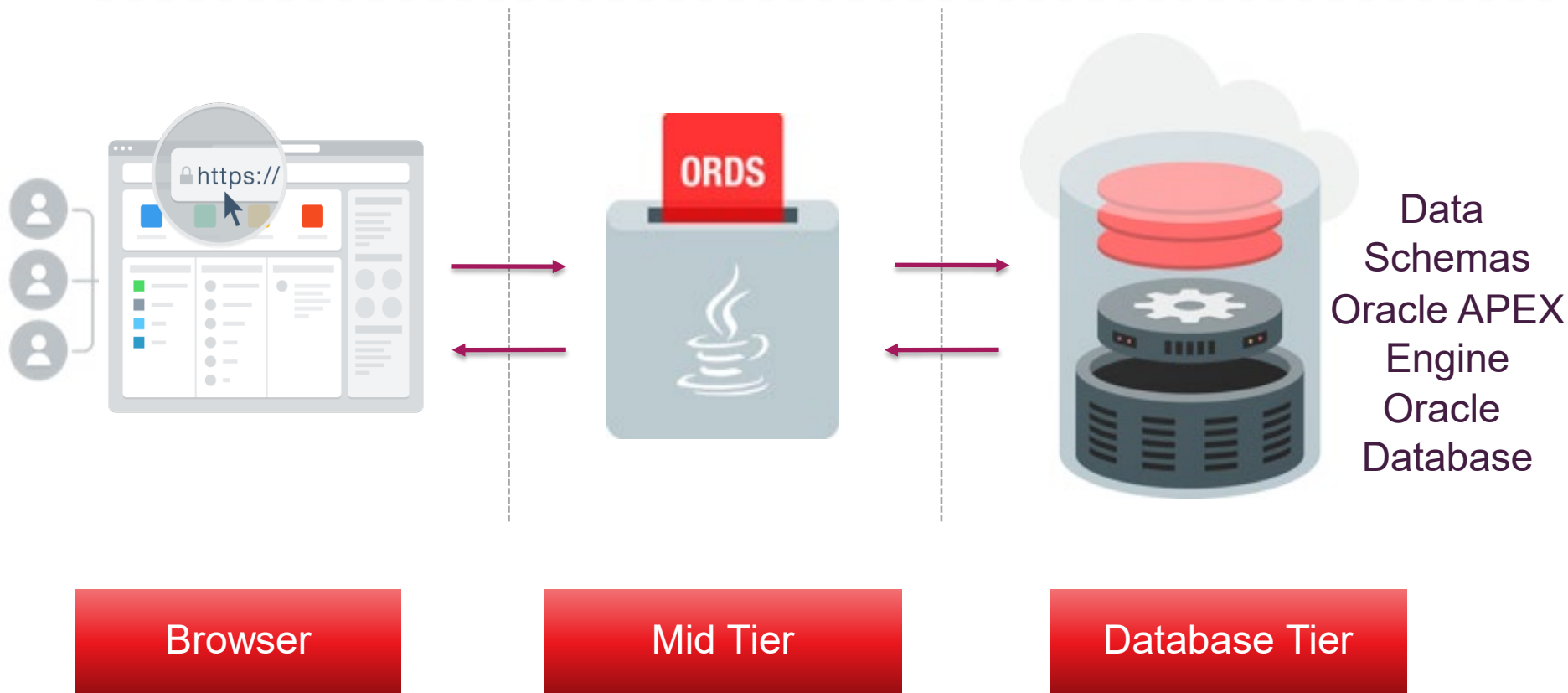
WATCH
LIVE



Oracle RDBMS RESTful Apps: An Overview



Oracle Application Express 3-Tier Architecture



ORDS = Oracle REST Data Services



Oracle REST Data Services (ORDS)

Evolved

from APEX Listener

ords.war

Java Web Archive

Deploy in Application Server

- Tomcat
- Glassfish (deprecated)
- WebLogic

Standalone mode

Brings own http-server
Supported for Production

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



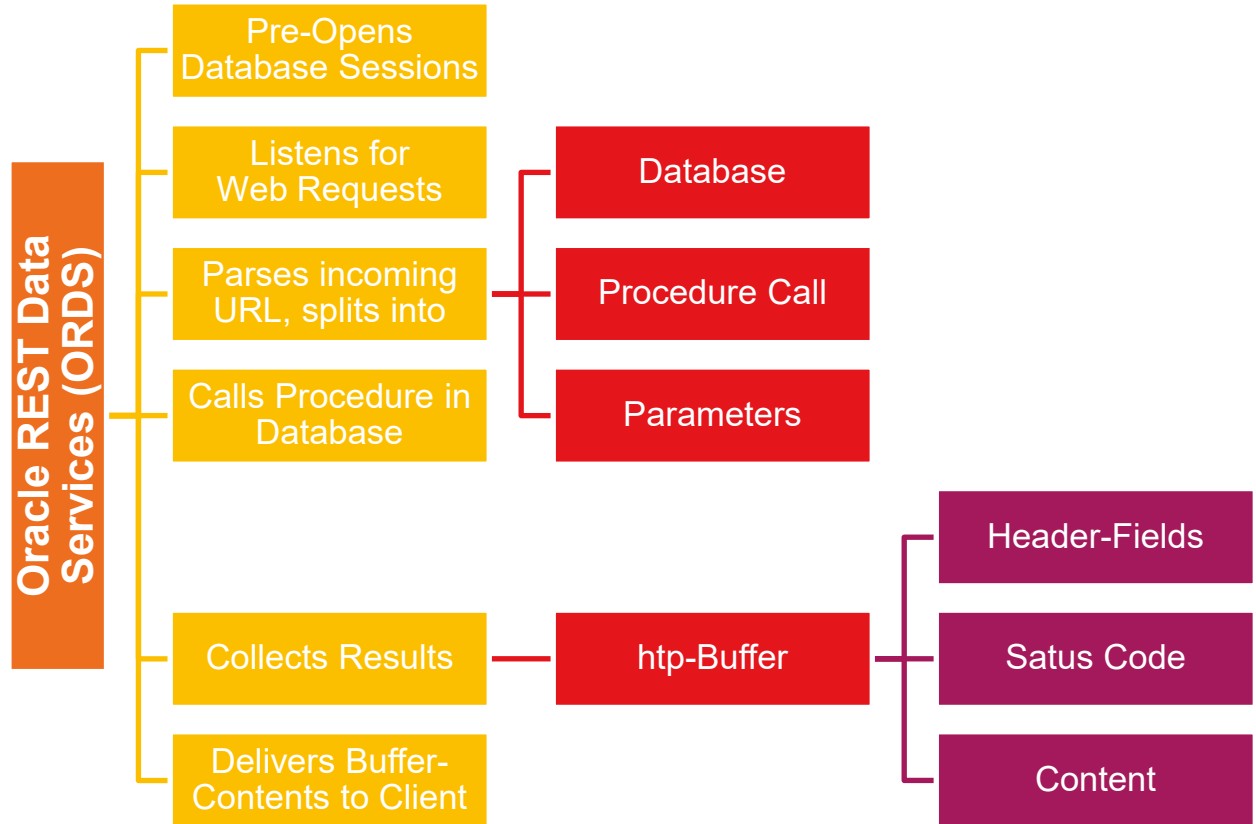


What is REST

REST	RE presentational S tate T ransfer	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 photograph of a person 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 black metal plant stand with several potted plants and a white cup of coffee on a small table. The room has dark curtains on the left and a plain white wall.

RESTful Services with ORDS AutoREST



Demo – Build a RESTfull Service in 100 seconds

Prerequisites

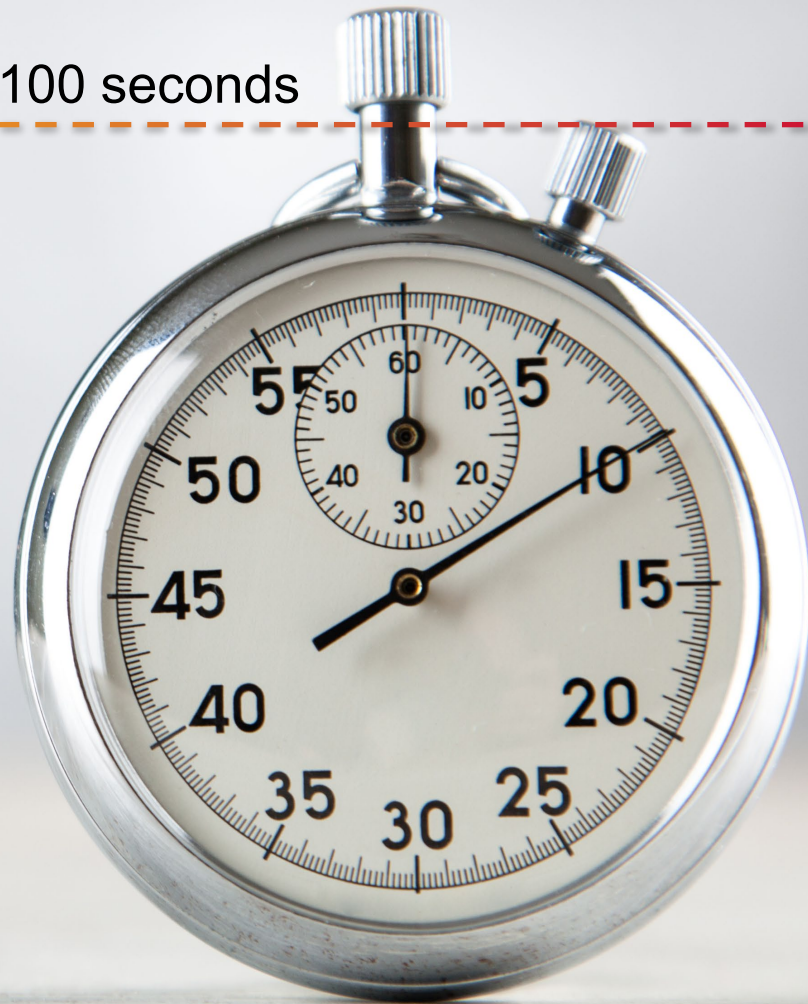
- Running Oracle DB 11gR2 or higher
- Schema with Table & Data
- DBA Access
- Installed & Configured ORDS

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





WATCH
LIVE



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/rdbl/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



Interpreting the ORDS AutoREST Responses (2)

```
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)**



API

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



©charles taylor - stock.adobe.com



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 **Hypermedia as the Engine of Application State (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

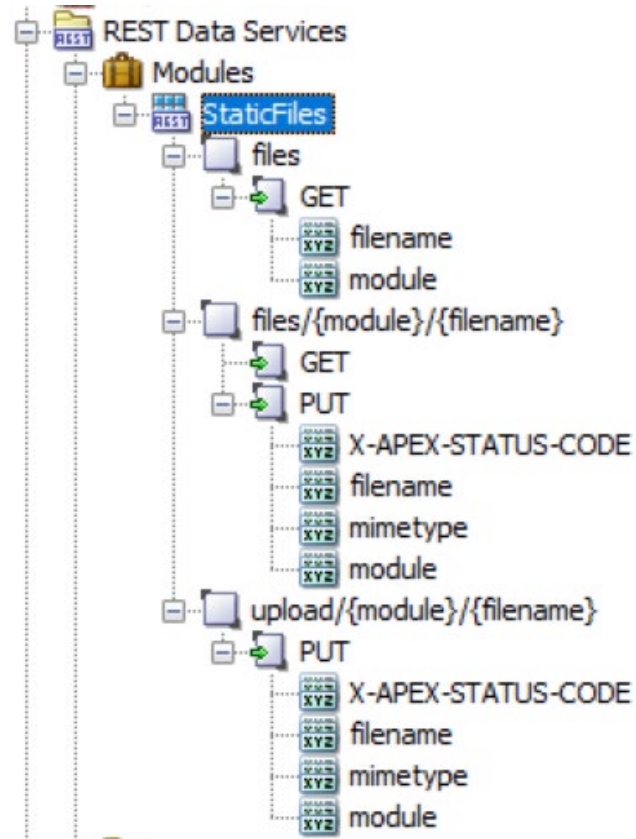
Right Click Query Results



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 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



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/outil/open-api-catalog/stati
```

```
{
  "swagger": "2.0",
  "info": {
    "title": "ORDS generated API for StaticFiles",
    "version": "1.0.0",
    "host": "localhost:8080",
    "basePath": "/",
    "schemes": ["http"],
    "produces": ["application/json"],
    "paths": {
      "/staticfiles/{module}/{filename}": {
        "get": {
          "description": "output of the endpoint",
          "schema": {
            "type": "object",
            "properties": {}
          }
        }
      }
    },
    "consumes": ["application/json"],
    "parameter": {
      "name": "module",
      "in": "path",
      "required": true,
      "type": "string"
    },
    "parameter": {
      "name": "filename",
      "in": "path",
      "required": true,
      "type": "string"
    }
  }
}
```

[...]

The screenshot shows a Swagger UI interface. At the top, there is a light blue bar for a GET endpoint: `/files/{module}/{filename}`. Below it, there is an orange bar for a PUT endpoint: `/upload/{module}/{filename}`. The main content area is light orange and contains the text "Hallo". Below this is a "Parameters" section with a "Try it out" button. The parameters table has two columns: "Name" and "Description".

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

Below the parameters table, there is an "Example Value" field with a "Model" button. The example value is a JSON object:

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

Below the example value, there is a "Parameter content type" dropdown menu with the value "application/octet-stream".

At the bottom, there is a "Responses" section with a "Response content type" dropdown menu set to "application/json". Below this is a table with two columns: "Code" and "Description".

Code	Description
200	output of the endpoint



WATCH
LIVE

Conclusion





Wanna Try? – Oracle Cloud Free Tier

Always Free cloud services

Services you can use for an unlimited time.

Databases

Two Oracle Autonomous Databases 20GB Data
Oracle APEX, ORDS and SQL Developer

Compute

Two AMD Compute VMs
Up to 4 instances of ARM Ampere A1 Compute with
3,000 OCPU hours and 18,000 GB hours per month

A lot More

Block, Object, and Archive Storage
Load Balancer and data egress
Monitoring and Notifications



Dedicated Talk

Holger Dressing: Oracle Cloud Free Tier
Montag 30.05.2022 | 15:15 - 16:00 | Amsterdam 1



Low Code – Apex & REST: Achieve Great Results with minimal Effort

ORDS RESTful Services

Powerful & Flexible
Swagger API Docs
REST enable Tables & Views quickly with
AutoREST

Oracle APEX

Build powerful Data-Driven Apps
Quick Results, even by Citizen Developers
Endless possibilities for experienced Devs

Low Code

ORDS & Apex
Foundation for Oracle Based modern Apps

PLEASE

**DO
TRY THIS
AT HOME**