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

Databees.

DATABEE
Die IT-Architekten

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

Low Code

Low Code

- Defined in 2014 by Forrester Research
- evolved from RAD Products (1990's)
  - Visual Basic, Delphi, Oracle Forms
- easy to learn
  - even by Citizen Devs
- Visual Design, Building Blocks
- Minimal Coding

Examples

- WordPress
- Mendix
- Oracle APEX

Oracle APEX
Low Code App Development

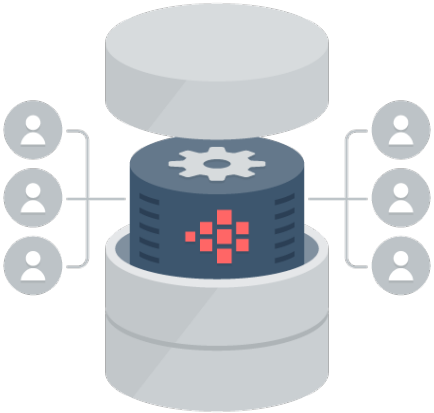
Oracle APEX: LowCode App Development

Oracle Application Express

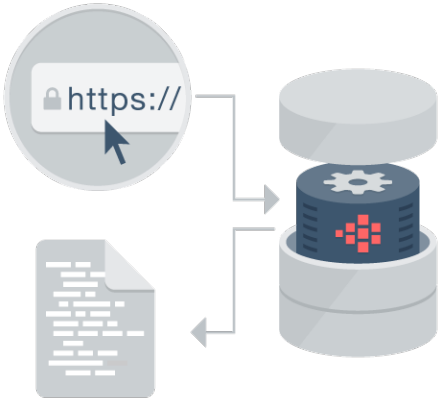
Database-centric web application development framework

Oracle APEX: Overview

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

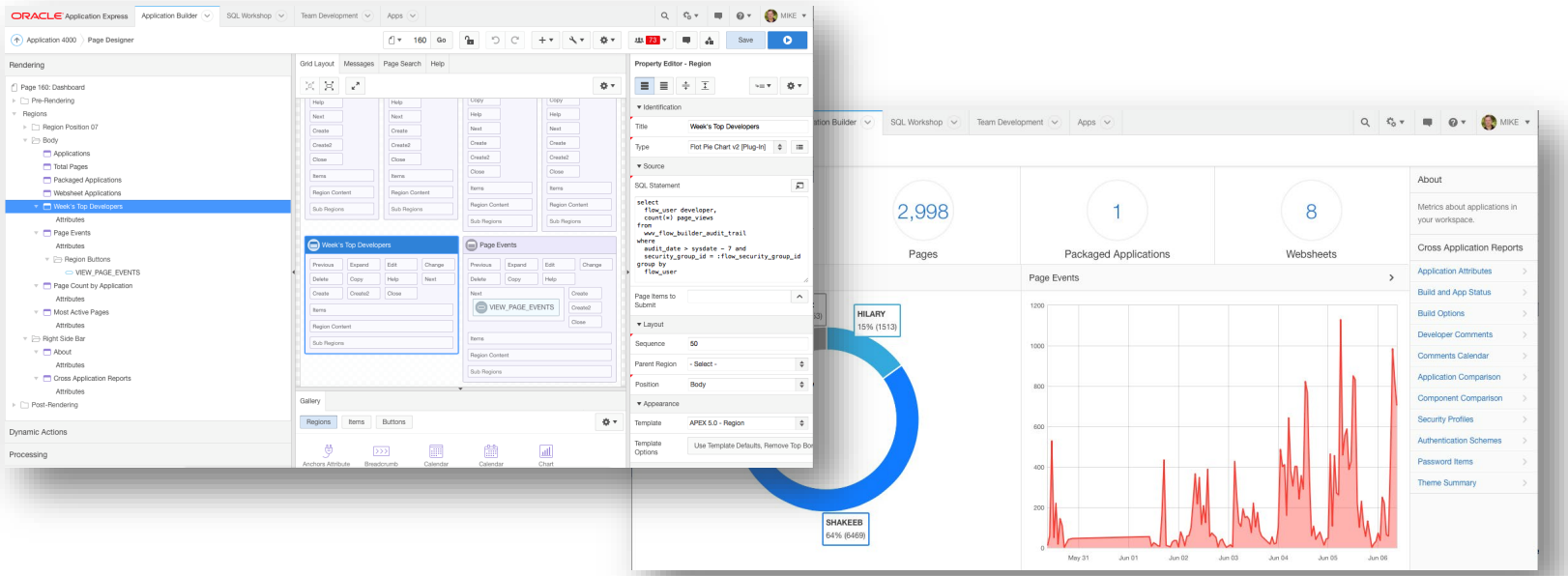


App definitions are stored in the database as meta data.
Declarative – No code generation



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

Oracle APEX: Application Builder



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

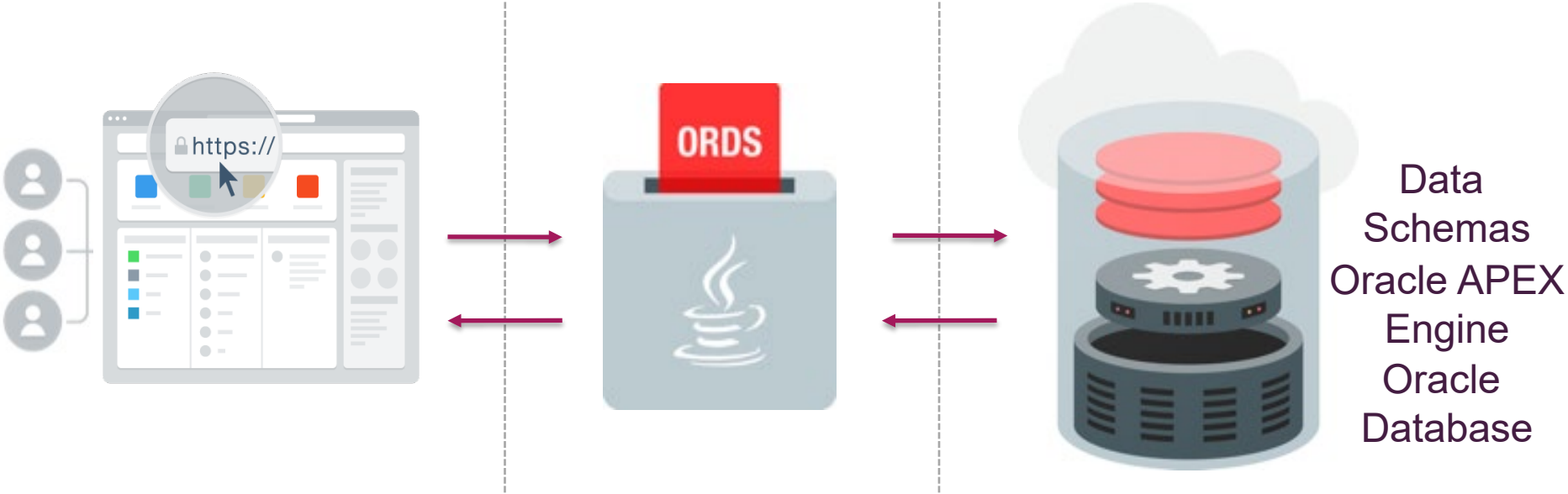
Consumes REST Services

e.g. samples REST API

Build Interactive Grids
based on REST Services

Oracle RDBMS RESTful Apps: 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

ords.war

Java Web Archive

Deploy in Application Server

- Tomcat
- Glassfish (deprecated)
- WebLogic

Standalone mode

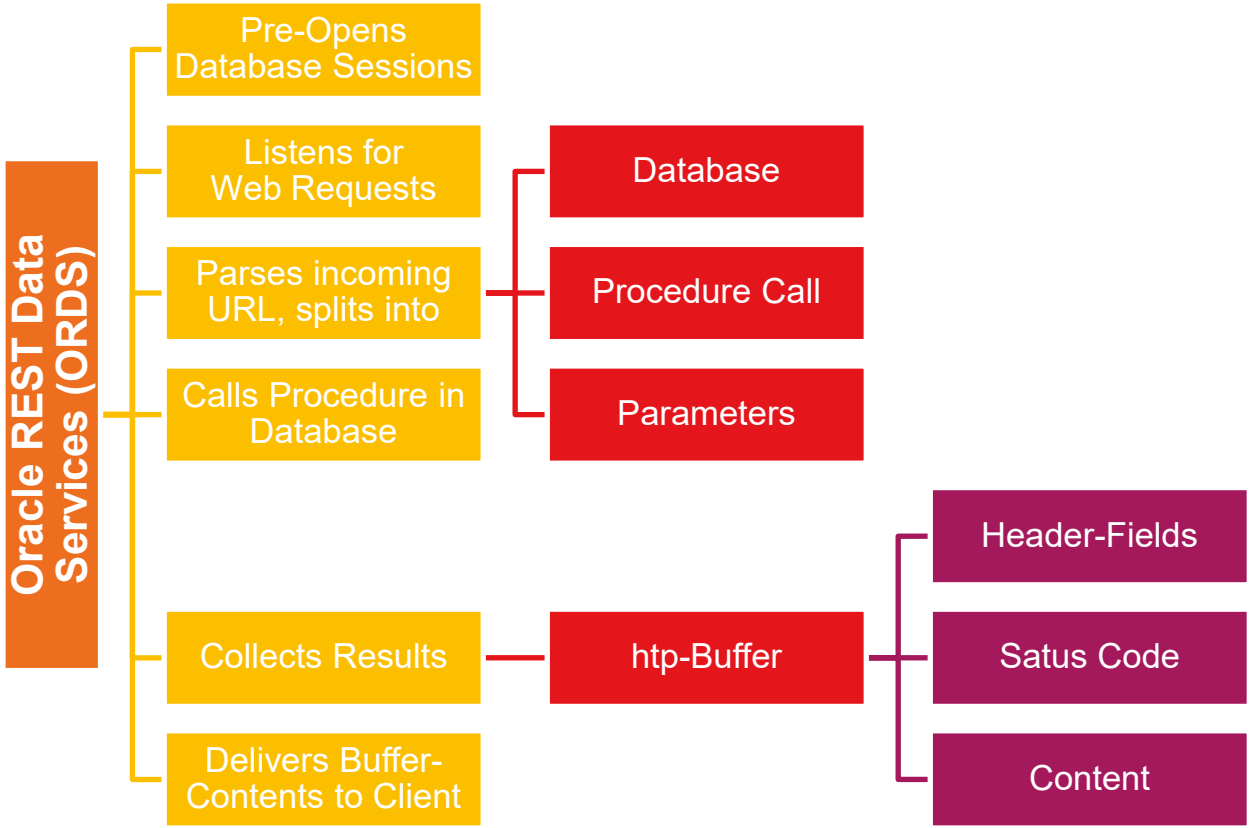
Brings own http-server
Supported for Production

```
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?



Demo – Build a RESTfull Service in 100 seconds

Prerequisites

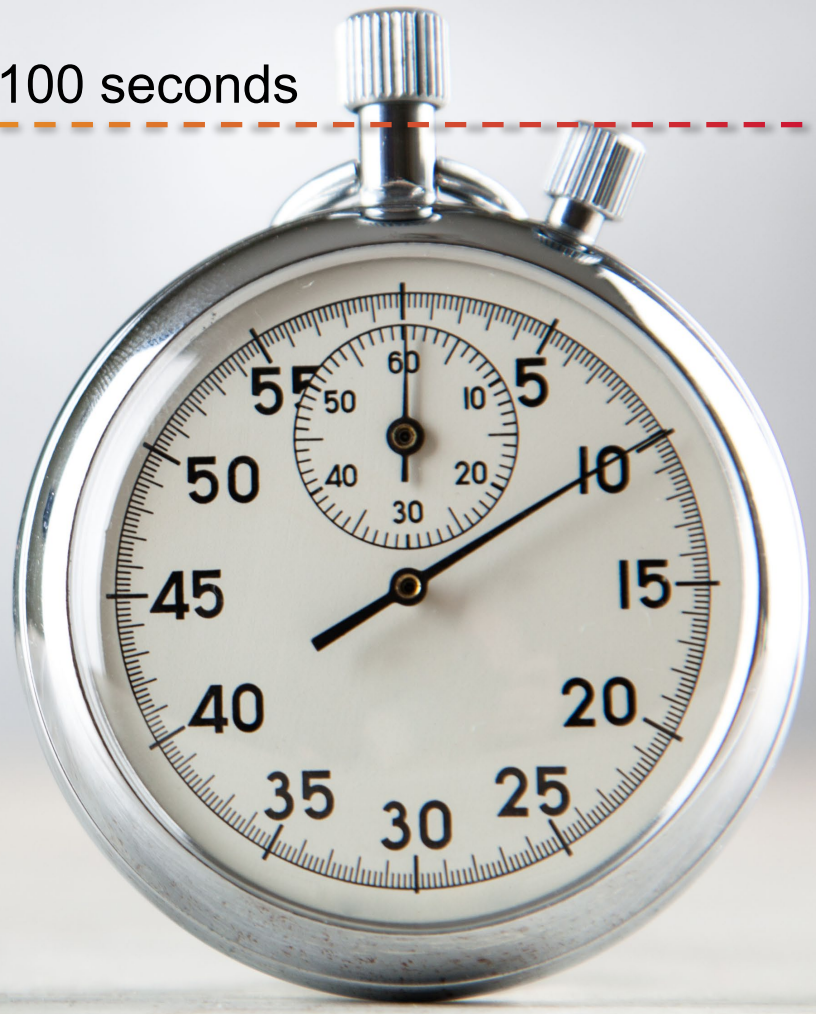
- Running Oracle DB 11gR2 or higher
  - Schema with Table & Data
  - DBA Access
- Installed & Configured ORDS

Tools

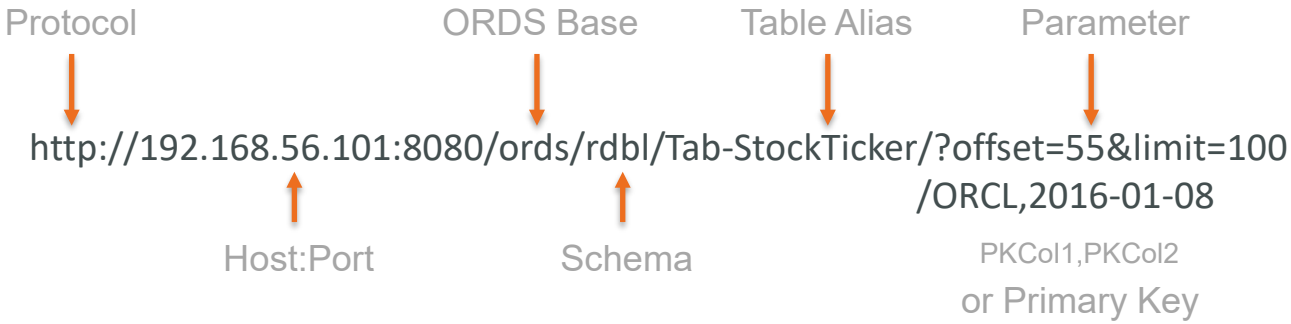
- Insomnia REST Client
- SQL Developer
- Oracle Visual Builder Add-in for Excel

Demo Sequence

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



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)

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

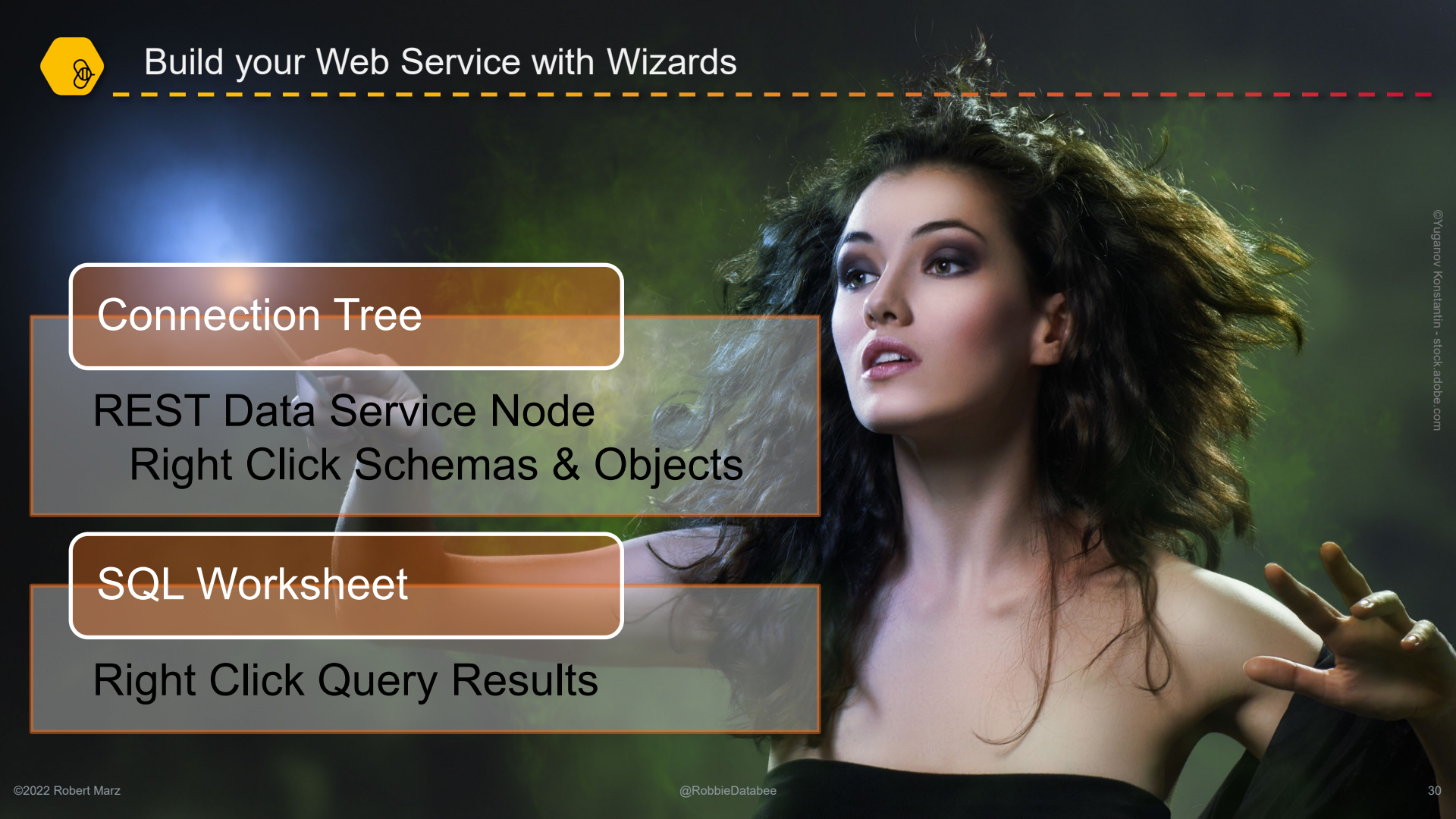
1xx Informational „Hold on“	100 Continue
	101 Switching Protocols
	102 Processing
2xx Success „Here you go“	200 OK
	201 Created
	208 Already Reported
3xx Redirection „Go away“	301 Moved Permanently
	304 Not modified
	307 Temporary Redirect
4xx Client Error „You fucked up“	400 Bad Request
	401 Unauthorized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

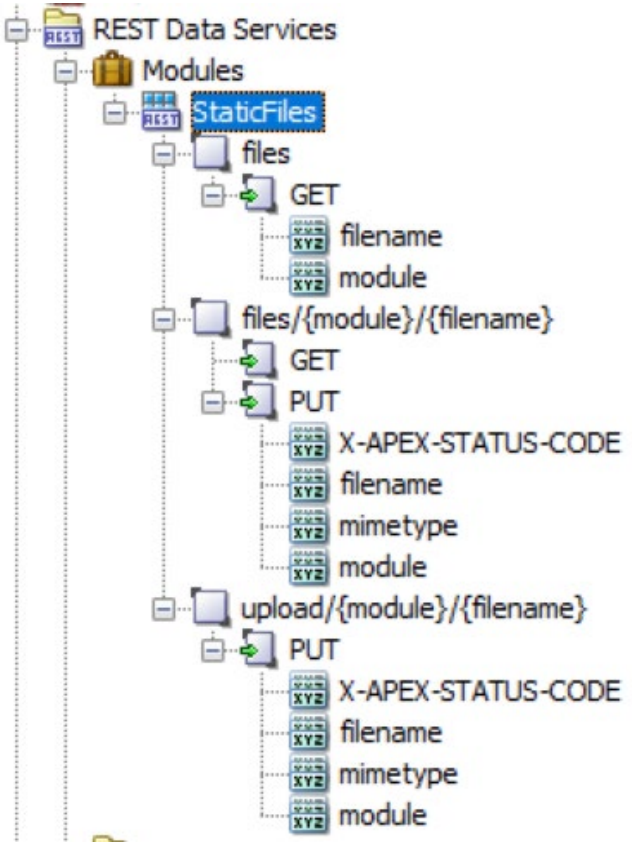
SQL Worksheet

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

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

Generate Swagger Doc

URL:

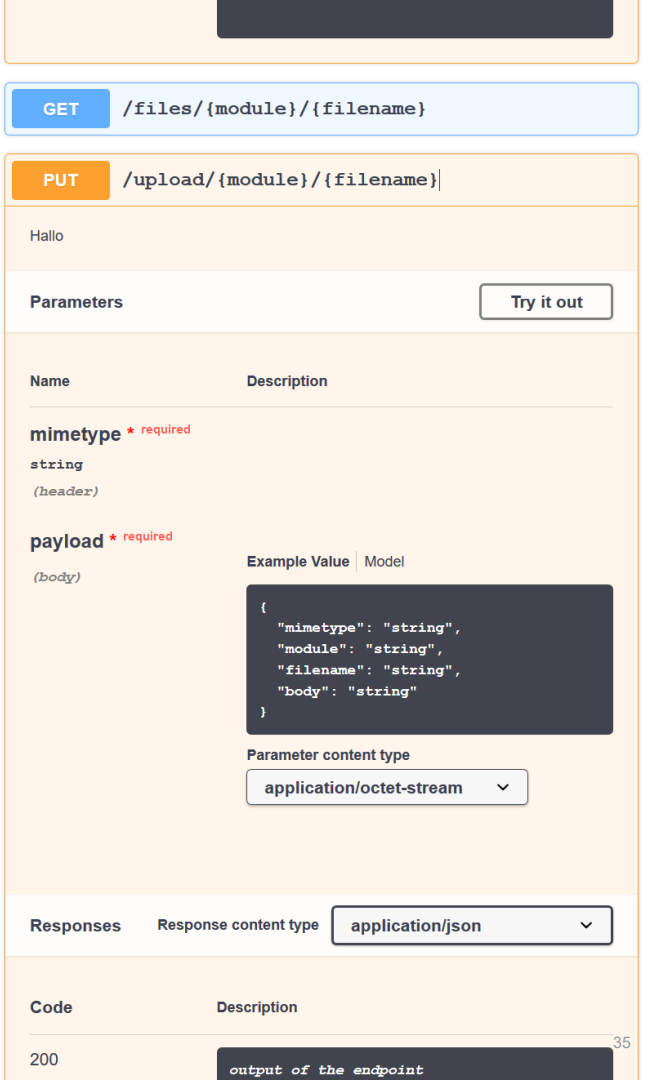
<ords-base>/<schema-alias>/open-api-catalog/<module>/

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":

[...]
```



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 Dresing: 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 quckly with AutoREST

Oracle APEX

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

Low Code

ORDS & Apex
Foundation for Oracle Based modern Apps