Using RESTfull services and remote SQL

from Apex 18.x

smart4 solutions

Agenda

- What is REST
- Using REST within APEX
    - Web Source Modules
    - Legacy Web Service References
- Build a Restful API for MySQL with NodeJS
- MySQL data into APEX
    - Reporting / Charting
    - Edit MySQL data with Apex
- Web-service-module on an ORDS instance
- Apex 18.1, remote SQL

Who is Richard?

- Smart4solutions partner (2017)
- Smart4apex founding member (2010)
- Oracle since 2002 (Oracle 8i)
- PL/SQL, Apex, HTML(5), CSS(3), JavaScript, XML, XSLT
- Special interest in UI
- Trainer at skillbuilders.com

ORACLE ACE

smart4 apex Everything for Apex

RIMA on Oracle Forums

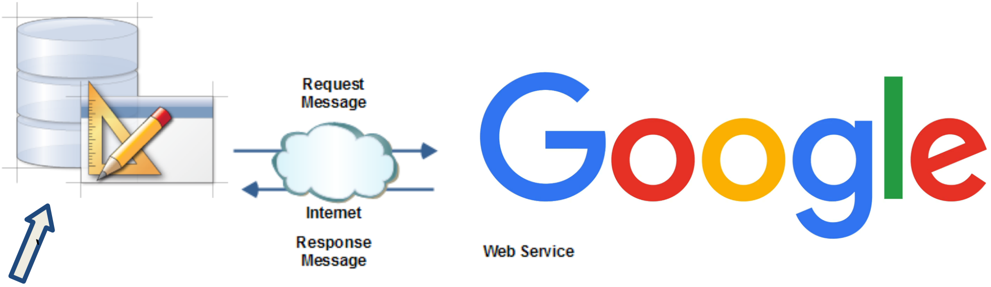
@rhjmartens

richardmartens.blogspot.nl

smart4 solutions

What is REST I - overview

- A web service is just like a regular web-page
- The result of the REST web service is often {JSON} but can have any content (HTML, XML, CSV)
- Benefits
  - Statelessness
  - Performance
  - Cacheability



Our Oracle / Apex server

What is REST II – request types

- 4 main methods / REQUEST types

SQL	HTTP request
select	GET
insert	POST
update	PUT
delete	DELETE

What is REST III – return values

URI	Request	Description	Response {JSON}
htt		sons	A collection of persons
ht		fic person	The person
ht		erson	The person as is stored in the DB
ht			The person as stored in the DB
ht		erson	{"rowsDeleted": 1} or {"rowsDeleted": 0}

```
{
    "empno": 7839,
    "ename": "KING",
    "job": "PRESIDENT",
    "mgr": null,
    "hiredate": "1981-11-17T00:00:00Z",
    "sal": 5000,
    "comm": null,
    "deptno": 10
}
```

demo

Using REST within APEX

- Report on public REST service
  - weather data

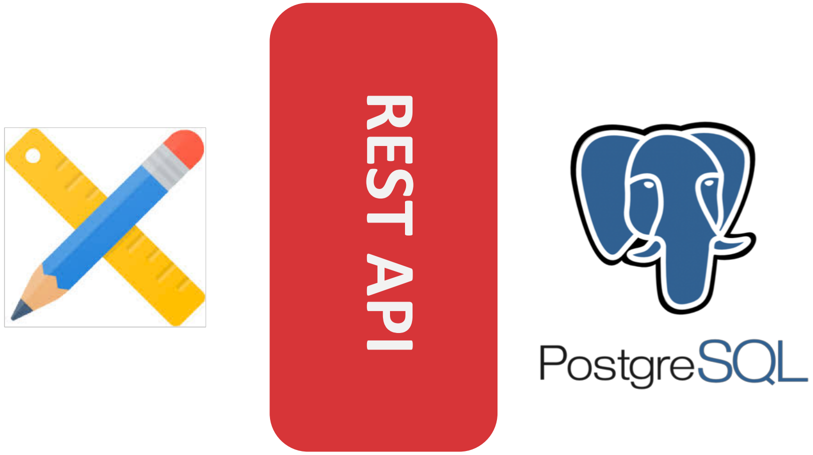
demo

APEX 18.x goes rogue

- Lifts the 11 GB boundary for Oracle Database XE
- We can start developing on top of
  - Jira / Confluence / BitBucket
  - Github
  - Online Accounting Applications
  - Other databases
- We do pay some penalties
  - Performance
  - Declarability

REST API

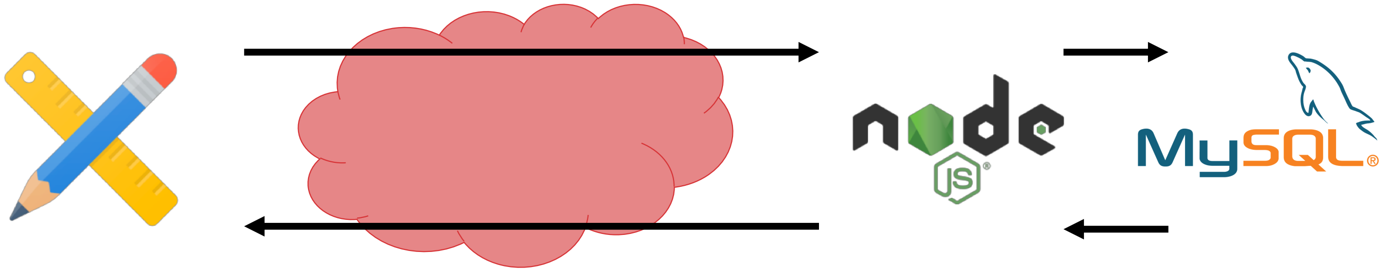
APEX 18.x goes rogue



REST API

Build a Restful API for MySQL with NodeJS

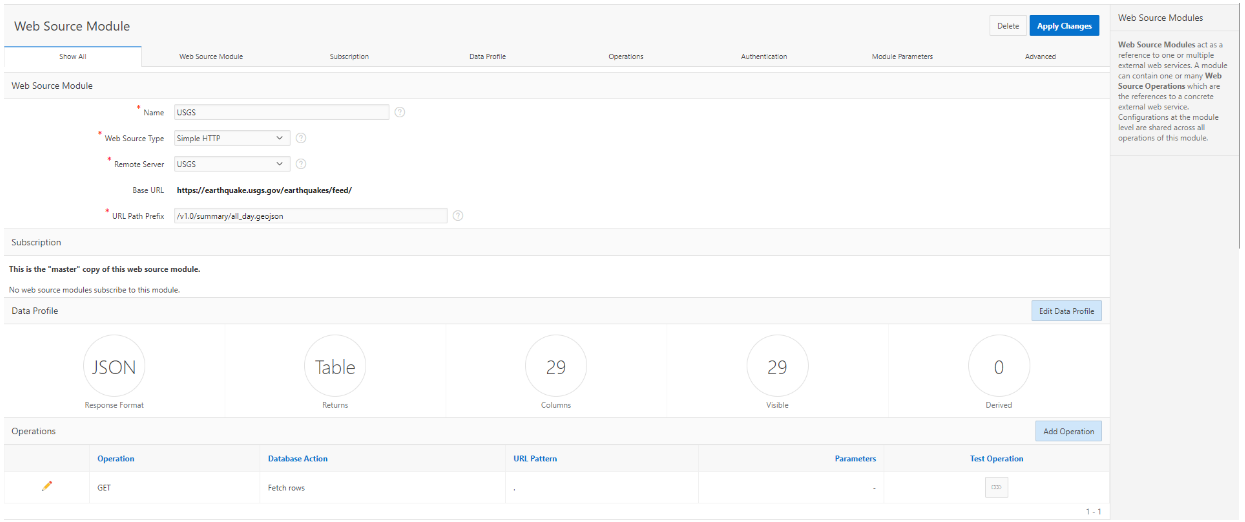
- Node.js API on a MySQL database providing REST full API
- Only 80 lines of code (JavaScript)



demo

Display MySQL data into APEX

- Define the MySQL restful service in apex web-source-module
- Parameters
  - Interactive Report
  - Classic Report
  - Charting



Web Source Module

* Name	USGS
* Web Source Type	Simple HTTP
* Remote Server	USGS
Base URL	**https://earthquake.usgs.gov/earthquakes/feed/**
* URL Path Prefix	/v1.0/summary/all_day.geojson

Subscription

This is the "master" copy of this web source module.

No web source modules subscribe to this module.

Data Profile

JSON	Table	29	29	0
Response Format	Returns	Columns	Visible	Derived

Operations

	Operation	Database Action	URL Pattern	Parameters	Test Operation
	GET	Fetch rows	.	-	

1 - 1

Web Source Modules

Web Source Modules act as a reference to one or multiple external web services. A module can contain one or many **Web Source Operations** which are the references to a concrete external web service. Configurations at the module level are shared across all operations of this module.

demo

smart4 solutions®

Edit MySQL data with Apex

Editing data over REST consists of:

1. Create Web-source-modules (GET, PUT, POST, DELETE)
2. Create report on web-source-module
3. Create form
4. Create the GET, PUT, POST, DELETE page-process

demo

Web-service-module on an ORDS instance

1. ORDS has some “extensions” on top of a normal REST implementation
2. Allows APEX to retrieve more information (metadata) during the “create web-service-module wizard”
3. We still need to add parameters, bodies etc. to the methods

For the rest these will act like any other REST web-service-module

demo

Web-service-module on an ORDS instance

- APEX_EXEC correctly renders NULL values in the {JSON}

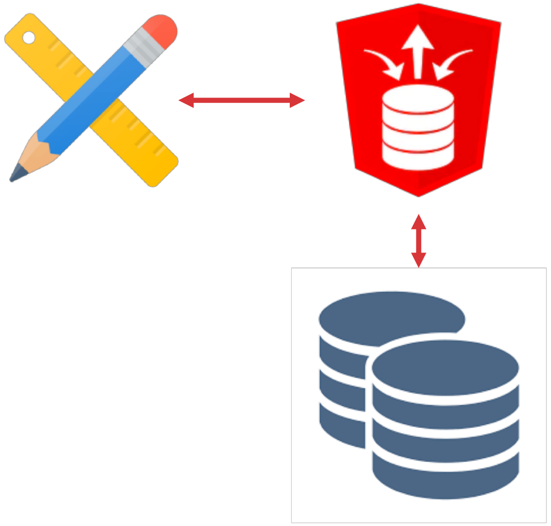
```
{ "empno": 7839,
  "ename": "KING",
  "job": "PRESIDENT",
  "mgr": null,
  "hiredate": "1981-11-17T00:00:00Z",
  "sal": 5000,
  "comm": null,
  "deptno": 10 }
```

- However when POSTing or PUTting, ORDS generates a “HTTP 400: Bad request”

demo

Apex 18.2 remote SQL feature

- Allows querying a remote database / use as a “remote datasource”
  - Requires ORDS 17.3 + on the remote site
- Remote SQL supports
  - SQL (DML and DDL)
  - PL/SQL (packages, procedures etc.)
- APEX Out of the box
  - Classic report
  - Interactive report
  - CSS Calendar
  - (in essence SELECT-stuff)
- APEX (Not yet) out of the box
  - Components that do updates
  - We can use the new apex_exec package for this

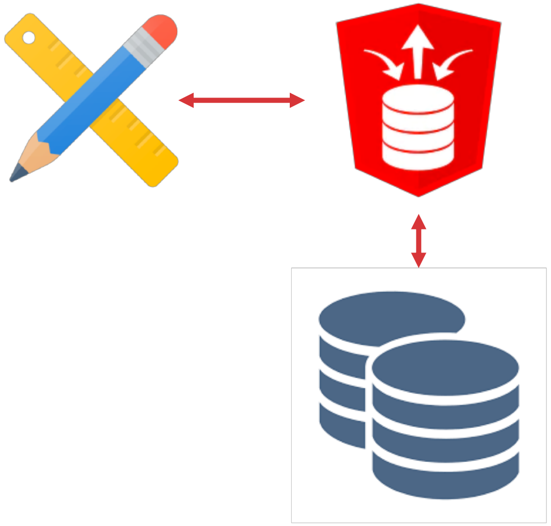


ORDS rSQL configuration steps

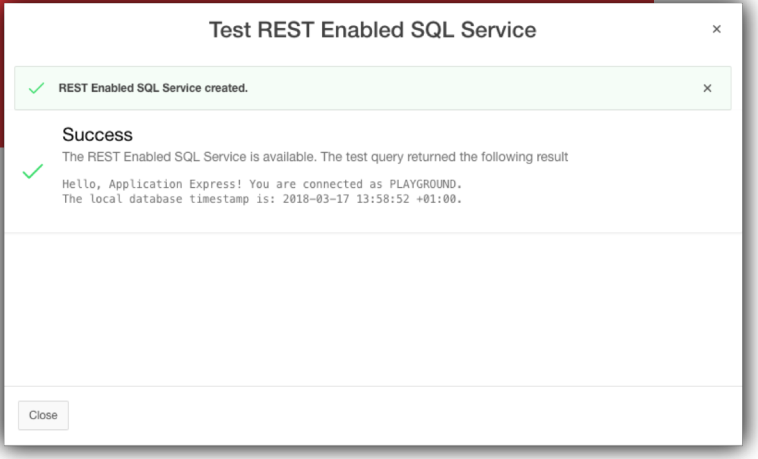
- Config of ORDS in defaults.xml
  - `<entry key="restEnabledSql.active">true</entry>`
  - `<entry key="jdbc.maxRows">1000</entry>`
- Enable schema for ORDS
  - ORDS.enable_schema(

```
  ORDS.enable_schema(
    p_enabled              => true
  , p_schema               => schema_name
  , p_url_mapping_type     => 'BASE_PATH'
  , p_url_mapping_pattern  => 'schema_name'
  , p_auto_test_auth       => false);
```

- We need at least ORDS 17.3
- My reverse proxy (NGINX) doesn't work yet ☹



APEX and ORDS rSQL



Test REST Enabled SQL Service

✓ REST Enabled SQL Service created.

✓ Success

The REST Enabled SQL Service is available. The test query returned the following result

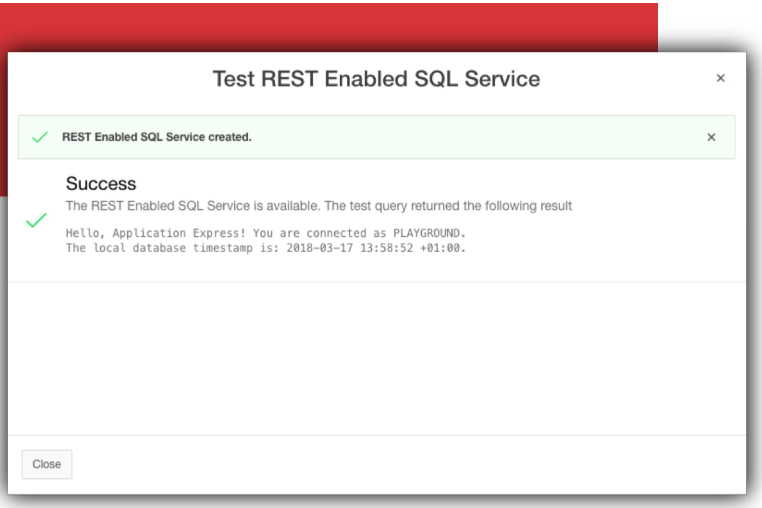
```
Hello, Application Express! You are connected as PLAYGROUND.
The local database timestamp is: 2018-03-17 13:58:52 +01:00.
```

- Shared components - REST Enabled SQL
- Provide name and endpoint URL
  - Ommit "/_/sql/"
- Create Service using the wizard
  - Basic Auth uses
    - schema username/password or
    - ORDS username/password ("SQL Developer")
  - Oauth2 uses credential flow using ClientID and Client Secret
    - `oauth.create_client`
    - `oauth.grant_client_role`

demo

APEX and ORDS rSQL

- Remote database does suport the “validate query” feature in the SQL editor



Test REST Enabled SQL Service

✓ REST Enabled SQL Service created.

Success

The REST Enabled SQL Service is available. The test query returned the following result

```
Hello, Application Express! You are connected as PLAYGROUND.
The local database timestamp is: 2018-03-17 13:58:52 +01:00.
```

demo

Questions ?

smart4 solutions®