Exposing Oracle 12c with ORDS

Presented by: John Jay King

Download this paper from: http://www.kingtraining.com

Session Objectives

- Understand the capabilities of Oracle Rest Data Services (ORDS)
- Learn how ORDS makes Oracle data available via standard APIs
- Use Oracle data using ORDS calls

Who Am I?

- John King – Partner, King Training Resources
- Oracle Ace Director
- Member Oak Table Network
- Providing training to Oracle and IT community for over 25 years – http://www.kingtraining.com
- "Techie" who knows Oracle, ADF, SQL, Java, and PL/SQL pretty well (along with many other topics)
- Member of AZORA, ODTUG, IOUG, and RMOUG

Arizona, USA

Who Are You?

- Application Developer
- DBA
- Business Analyst
- Other?

The Crown Jewels

- Data; your Crown Jewels
  - Today's Apps Built with Polyglot of Tools
  - Developers don't know SQL or PL/SQL
  - Applications need to display data
    - Data gets dumped to flat files or spreadsheets then made available to apps
    - Aaargh! What just happened to all of my planning and investment in thick database?




Developer Expectations

- Today's Developers juggle many tools in an ever-changing landscape
  - SQL and PL/SQL are not high on their list of desired skills
- Expectations for Data Access
  - Web Services using RESTful APIs
  - Data input and output using JSON

REST

- REST (Representational State Transfer) describes a different architecture than SOA (the term REST originated in a doctoral dissertation about the web written in 2000 by Roy Fielding – Google it for more…)

What?

- REST implements Web Services using a simpler technology stack than SOAP
- The term REST is often used to (loosely) describe transmission of domain-specific data via HTTP without SOAP or any type of session tracking
- REST allows execution of Java Web Services using simple HTTP technology without the need of SOAP and WSDL

How RESTful APIs Work

- RESTful APIs work with resources not procedures
  - Resources are uniquely identified by URI
  - HTTP verbs act on resources in a standard way:
    - GET Query
    - PUT Create (sometimes Update)
    - POST Update (sometimes Create)
    - PATCH Partial Update
    - DELETE Delete

You Already Know REST

- One of the strengths of (what we call) REST is that you already use it everyday
  - You use a browser to access a "resource" http://my.website.com/zebra/index.html
    - URI includes website or IP address
    - "resource" (zebra above)
    - "resource" web page (index.html above)
  - You "GET" the data corresponding to the URI

REST in Action

- URLs use nouns for resources (not verbs for actions)

```
GET /ords/hr/employees/          Preferred
GET /ords/hr/GetAllEmployees/ Not Preferred
```

- Uniform API for operations: GET, POST, PUT, DELETE
- Requests are stateless, state information is contained in URI’s (as Oracle's Jeff Smith says, "it's all about the hyperlinks")

REST Error Code Families

1xx	Informational
2xx	Success
3xx	Redirection
4xx	Client Error
5xx	Server Error

Web Services

- What is a Web Service anyway? Here's a definition from the W3C (World-Wide Web Consortium) website:

  "A Web service is a software application identified by a URI, whose interfaces and binding are capable of being defined, described and discovered by XML artifacts and supports direct interactions with other software applications using XML based messages via internet-based protocols"

Web Service Payloads

- Web Services interactions were originally XML-based but JSON-based interactions are pretty much the standard for Mobile
  - Plain-text of XML and JSON makes them ideal for cross-language/platform use
  - Using XML or JSON with standard interfaces like SOAP and REST makes Web Services potentially programming language and operating system independent

Emergence of JSON

- JSON (JavaScript Object Notation) has become the mechanism of choice for sharing and passing data to-and-from mobile applications
- JSON is an international, open standard governed by the ECMA

http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf

JSON-XML Similarities

- JSON is text only, just like XML and thus is an excellent vehicle for data interchange
- JSON and XML are “human readable” and "self-describing” (sort of…)
- JSON and XML are hierarchical (data sets nested within data sets)
- JSON and XML offer validation capability; XML’s is more mature and capable today

JSON-XML Dissimilarities

- XML is verbose, JSON is less-verbose
- JSON has no end tags, required in XML
- **JSON is quicker to read and write**
- Reading XML documents requires “walking the DOM” – JSON does not
- JSON works more easily and is faster than XML when working with AJAX
- **XML documents must be tested for “well-formed”-ness before processing**

XML File

```
<?xml version="1.0"?>
<myBooks>
  <book>
    <name>Learning XML</name>
    <author>Eric T. Ray</author>
    <publisher>O'Reilly</publisher>
  </book>
  <book>
    <name>XML Bible</name>
    <author>Elliotte Rusty Harold</author>
    <publisher>IDG Books</publisher>
  </book>
  <book>
    <name>XML by Example</name>
    <author>Sean McGrath</author>
  </book>
</myBooks>
```

JSON File

```
{"myBooks":
  [ {"book": {
            "name":"Learning XML",
            "author":"Eric T. Ray",
            "publisher":"O'Reilly" }
     },
     {"book": {
            "name":"XML Bible",
            "author":"Elliotte Rusty Harold",
            "publisher":"IDG Books" }
     },
     {"book": {
            "name":"XML by Example",
            "author":"Sean McGrath",
            "publisher":"Prentice-Hall" }
     }
  ]}
```

Why ORDS?

- What if you could expose your Tables, Views, and stored PL/SQL (thick database) as web services?
- What if you could accept input in JSON and generate JSON easily?
- What if you could make data available to today's developers safely in a form they're familiar with and want to use?

ORDS to the Rescue!

- Oracle's Object REST Data Services (ORDS) provides a solution to:
  - Expose Tables/Views via REST APIs
  - Expose PL/SQL via REST APIs
  - Transfer data using JSON or other datatypes

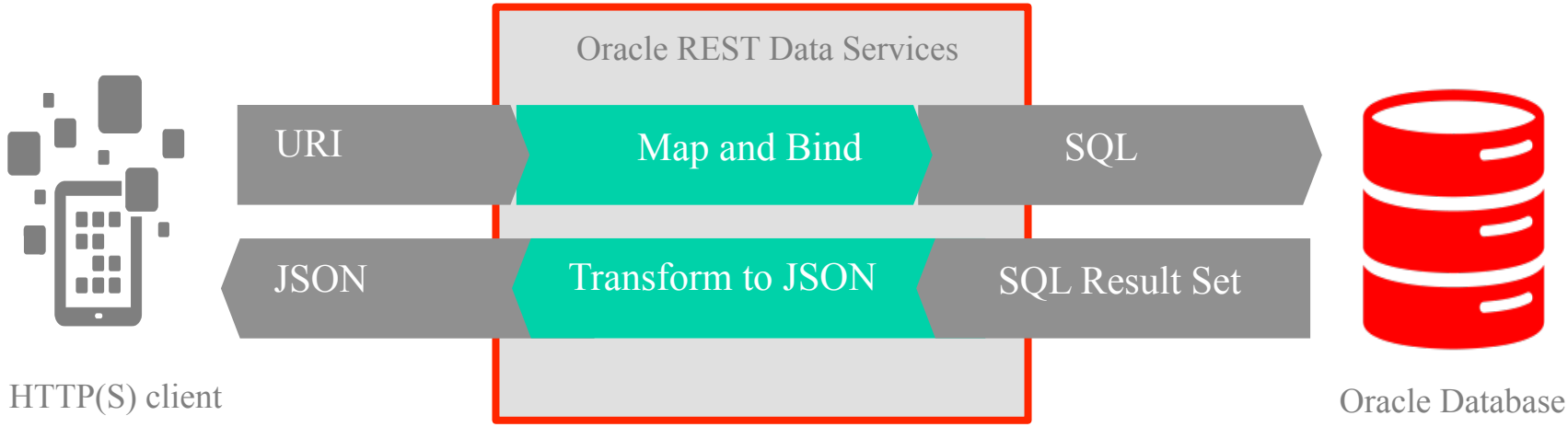
Introduction to ORDS

- Provides RESTful access that modern tools expect
- Maps SQL to REST style HTTP: GET, POST, PUT, DELETE
- May return results in JSON (or other data types)

What is ORDS?

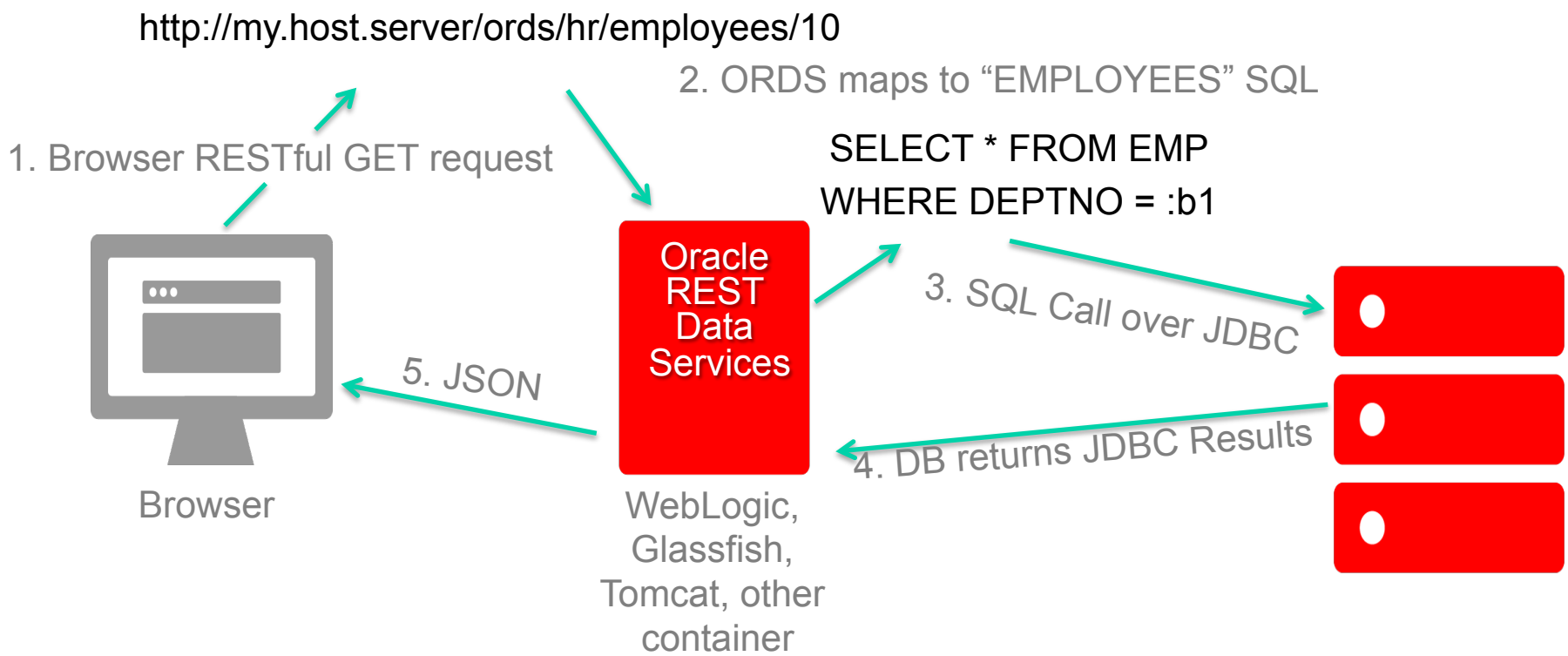
- ORDS is included in Oracle DBaaS instances
- ORDS may be installed easily in any instance
- ORDS runs in any Java EE container (Weblogic, Glassfish, Tomcat, etc.)
- Developers may create standalone version
- ORDS is an enhanced version of Oracle's Java mod_plsql Apache module

Oracle REST Data Services



- Data stored in Oracle (relational tables and columns)
- ORDS defines URI-to-SQL mapping with SQL results mapped to JSON (or other data types)
- Applications use URIs via HTTP(S) to GET and POST data

Oracle REST Data Services 3.0



- ORDS is an improved version of the original mod_plsql
- ORDS maps requests to SQL and transforms results to JSON (or CSV, Text, Binary, Excel, more…)
- Mappings are automatic or created using SQL Developer

ORDS REST APIs

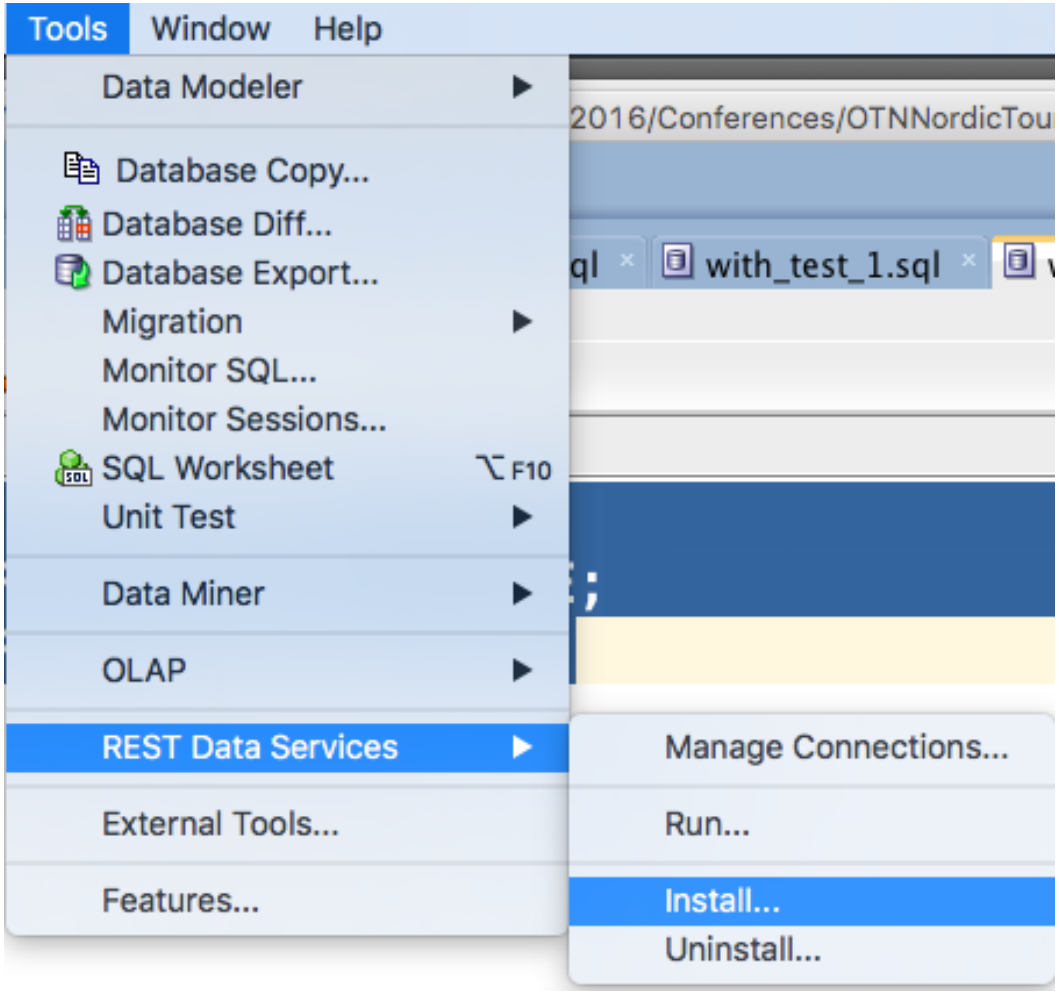
GET	'Read' a resource
POST	'Create' a resource
PUT	'Update or Create'
DELETE	'Delete' a resource
HEAD	'Read' resource headers

Typical REST “Payload”

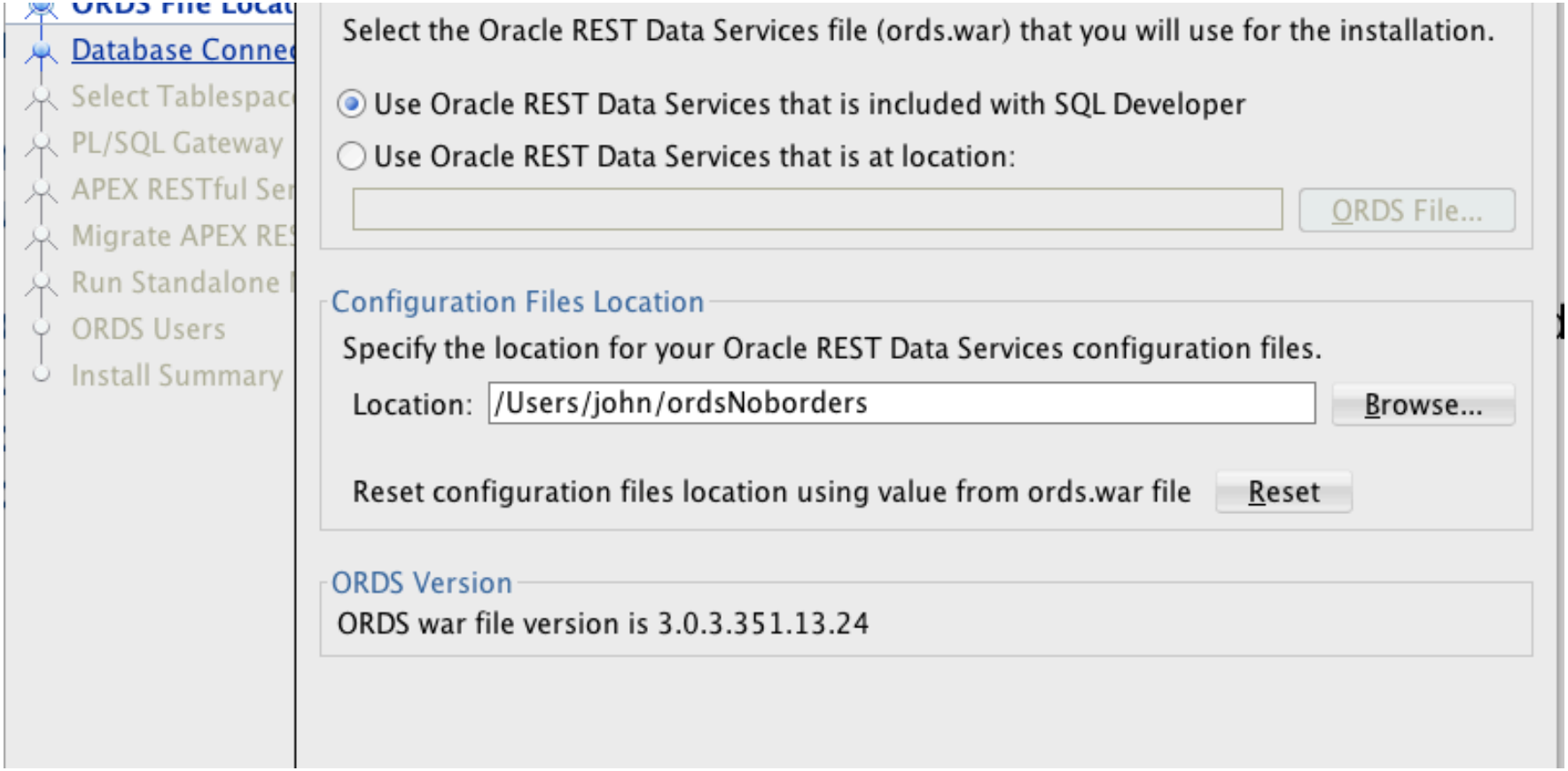

```
{  "items": [
     {
       "employee_id": 100,
       "first_name": "Steven",
       "last_name": "King",
       "email": "SKING",
       "phone_number": "515.123.4567",
       "hire_date": "1987-06-17T04:00:00Z",
       "job_id": "AD_PRES",
       "salary": 24000,
       "commission_pct": null,
       "manager_id": null,
       "department_id": 90,
       "links": [
        .... More Here ....
     },
```

ORDS Installation

- Installation is easy with SQL Developer

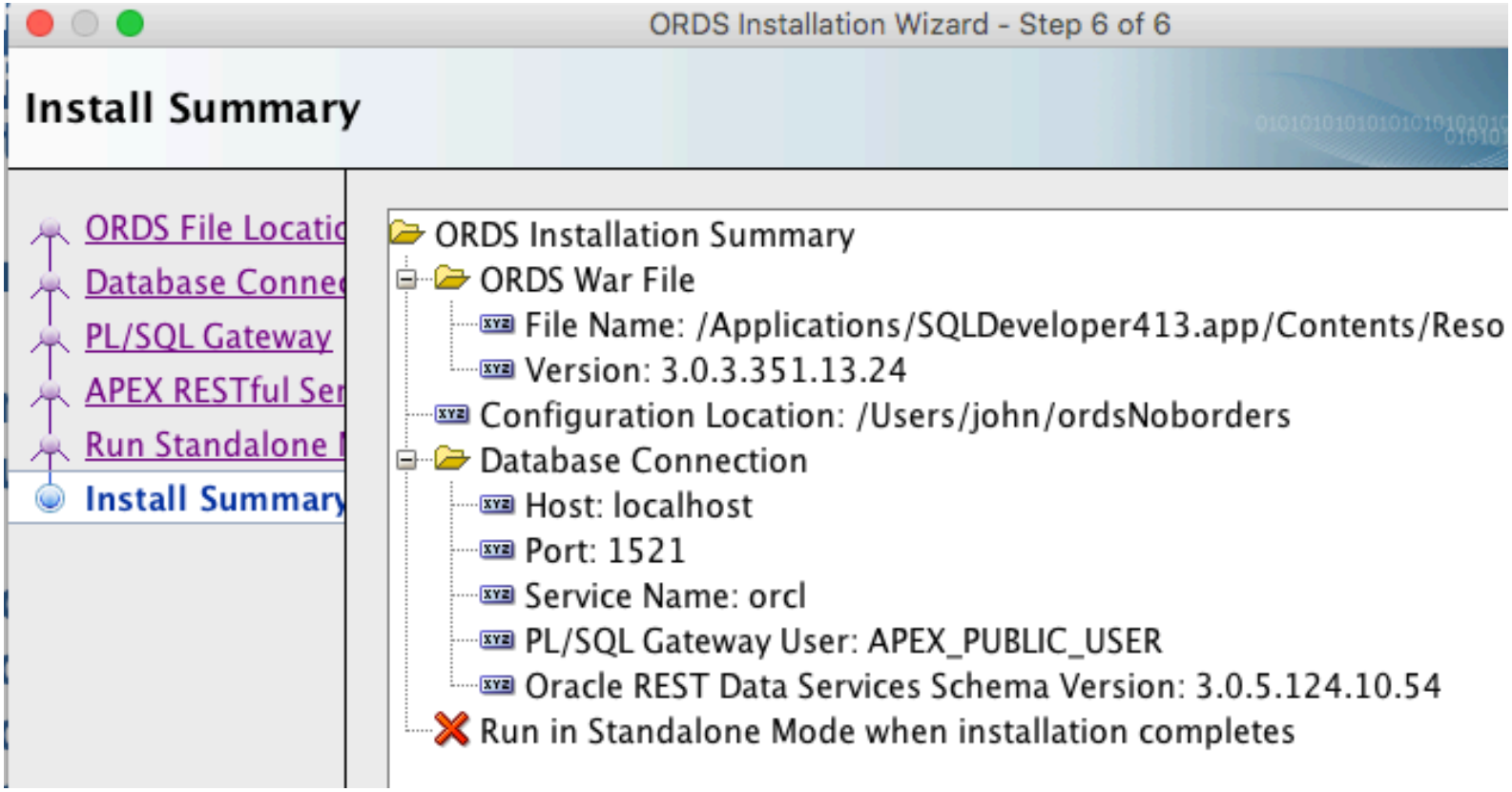


ORDS Setup Steps



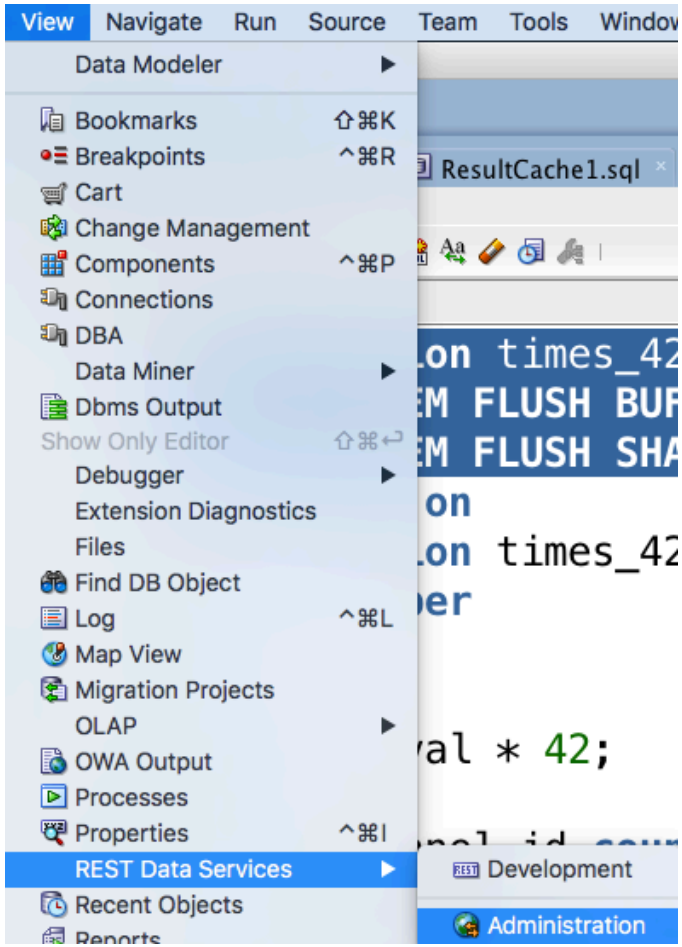
ORDS Installed

- After several wizard-based steps installation is complete



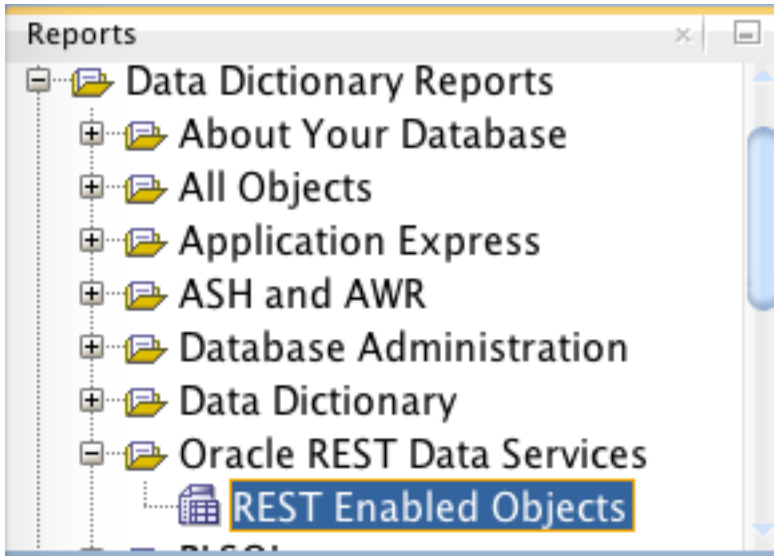
ORDS Administration

- SQL Developer provides an Administration capability for ORDS



ORDS Reporting

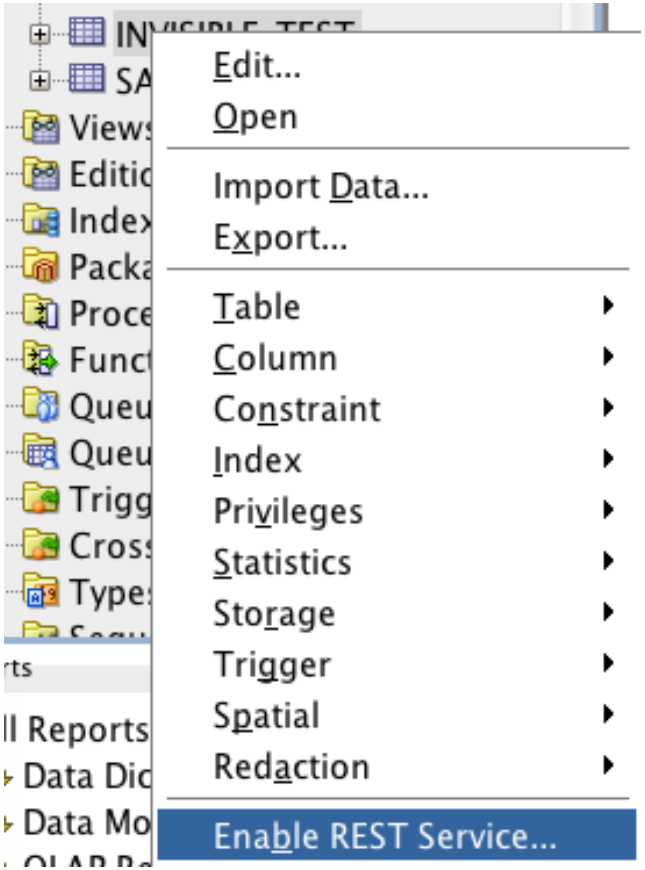
- SQL Developer has reporting built-in

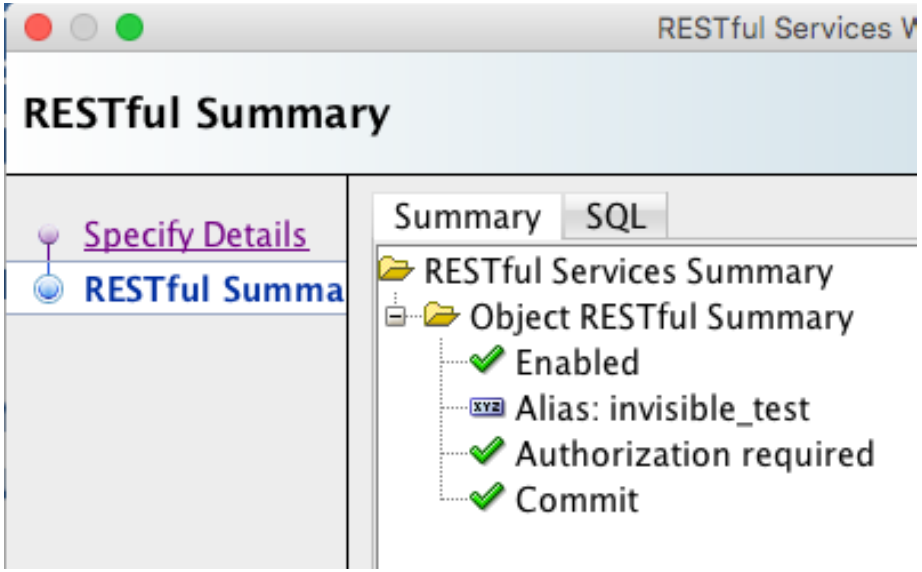


	Schema	Object	Alias	Authorization	Type
1	JOHN	EMP	emp	ENABLED	TABLE
2	JOHN	INVISIBLE TEST	invisible test	DISABLED	TABLE

Enable ORDS for Table

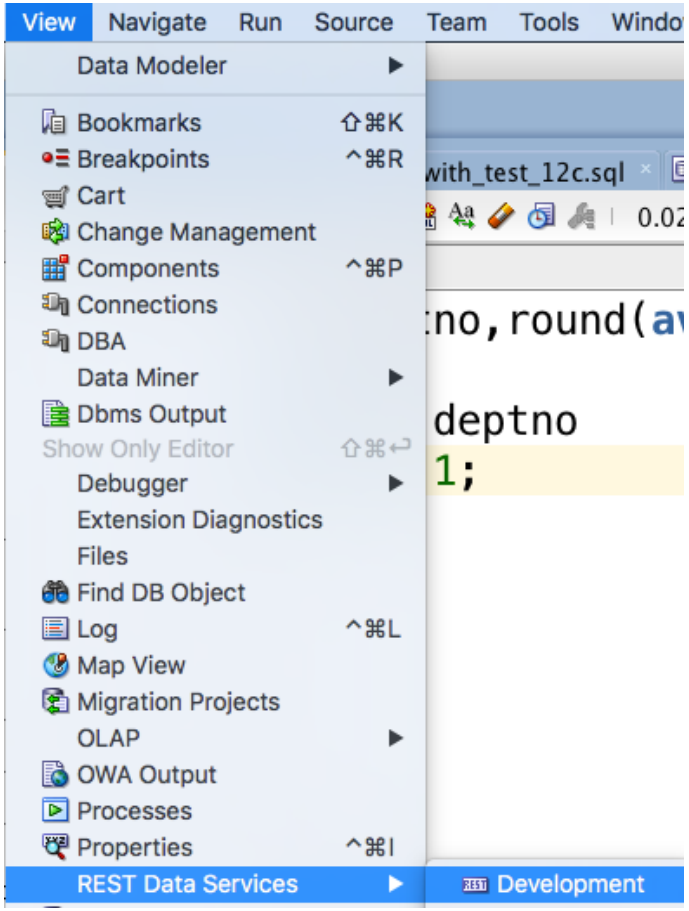
- Right-click on a table to enable ORDS using a wizard-based process

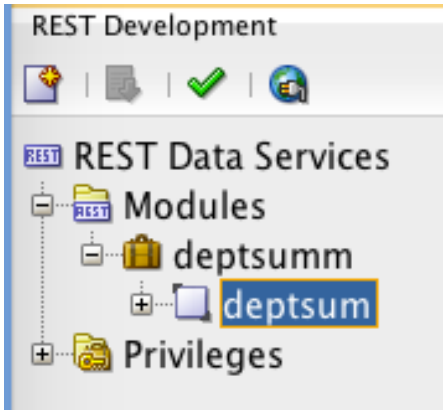


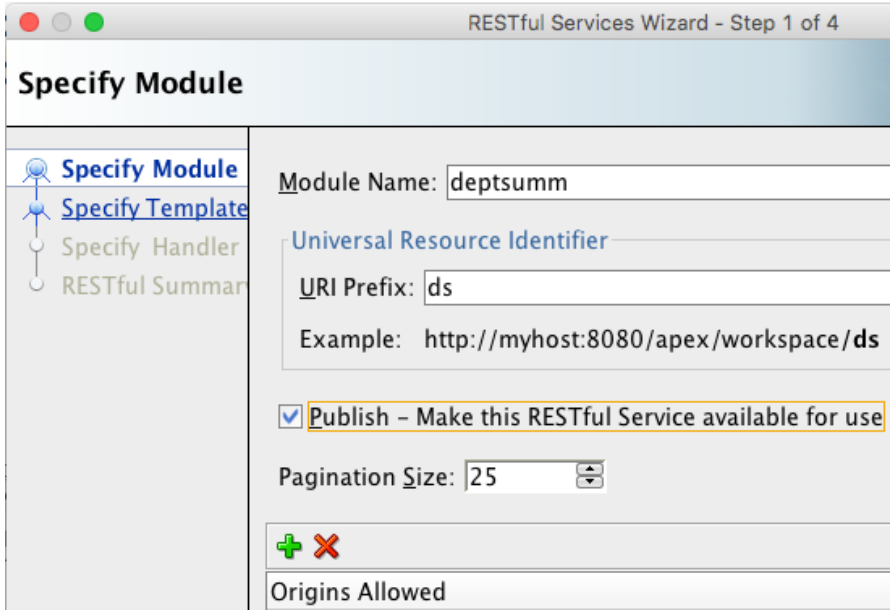


Manual ORDS

- Use SQL Developer wizards to map ORDS for SQL and PL/SQL







PL/SQL ORDS APIs

- Oracle provides a PL/SQL package that allows creation and manipulation of ORDS in code
- Complete PL/SQL API for defining and configuring RESTful services
  - May be put into scripts
  - May be repeated

https://docs.oracle.com/cd/E56351_01/doc.30/e56293/ords_ref.htm#AELIG90180

PL/SQL API in Use

```
begin
  ords.create_service(
    p_module_name    => 'samples.employees',
    p_base_path
          => '/samples/employees',
    p_pattern        => '.',
    p_items_per_page => 5,
    p_source
          => 'select * from hr.employees
               order by employee_id');
    commit;
  end;
```

Security

- Security is an important part of todays applications
- ORDS provides a standardized mechanism for URIs
  - May tie into Oracle Identity Management via WebGate to access Single Sign On (SSO)
  - May use OAuth2 (built-in)

View Table Contents

- Use URL to view all table rows

localhost:8080/ords/hrrest/employees/


```
// 20161011124847
// http://localhost:8080/ords/hrrest/employees/

{
  "items": [
    {
      "employee_id": 100,
      "first_name": "Steven",
      "last_name": "King",
      "email": "SKING",
      "phone_number": "515.123.4567",
      "hire_date": "1987-06-17T04:00:00Z",
      "job_id": "AD_PRES",
```

View Row Contents

- Use URL to view Row contents

localhost:8080/ords/hrrest/employees/200

Reload this page


```
074740
// http://localhost:8080/ords/hrrest/employees/200

{
  "employee_id": 200,
  "first_name": "Jennifer",
  "last_name": "Whalen",
  "email": "JWHALEN",
  "phone_number": "515.123.4444",
  "hire_date": "1987-09-17T04:00:00Z",
  "job_id": "AD_ASST",
  "salary": 4400,
  "commission_pct": null,
  "manager_id": 101,
  "department_id": 10,
  "links": [
    {
```

ORDS Access from Code

```
// Javascript
var myxhr = new XMLHttpReques
myxhr.onreadystatechange =
function() {
    if (myxhr.readyState ==
                var mydata =
```

```
// Php
<?php
    $url = https://
my.host/ords/hrrest
employees/200';
    $response =
file_get_contents($
```

```
// Python
 import json
 import requests
 url = https://
my.host/ords/hrrest/
employees/200'
 response =
requests.get(url)
 getdata =
response.json()
 print getdata
```

```
// Java
DefaultHttpClient httpClier
new DefaultHttpClient();
HttpGet myGet = new
  HttpGet("my.host/ords/hr
employees/200");
HttpResponse response =
httpClient.execute(myGet);
Bu
Bu
re
))))
```

```
// jquery
$.getJSON("my.host/ords/
employees/200", functio
      processData(getdata
});
```

```
// Android
DefaultHtt
  new Defa
HttpGet my
  new Http
hrrest/emp
HttpRespon
httpClient
BufferedRe
BufferedRe
  new Inpu

response.getEntity().getContent(
))));
```

```
// Ruby
require 'j
require 'n
url = 'my.host/ords/
hrrest/employees/200'
response =
Net::HTTP.get_response(UR
I.parse(url))
getdata =
JSON.parse(response.body)
print getdata
```

Wrapping it all Up

- The Crown Jewels are safe!
- We control access to our data by providing exactly what the developers need and want
  - Web Services available via RESTful APIs
  - Data in and out in JSON as well as other data types
- ORDS helps enable your newest applications

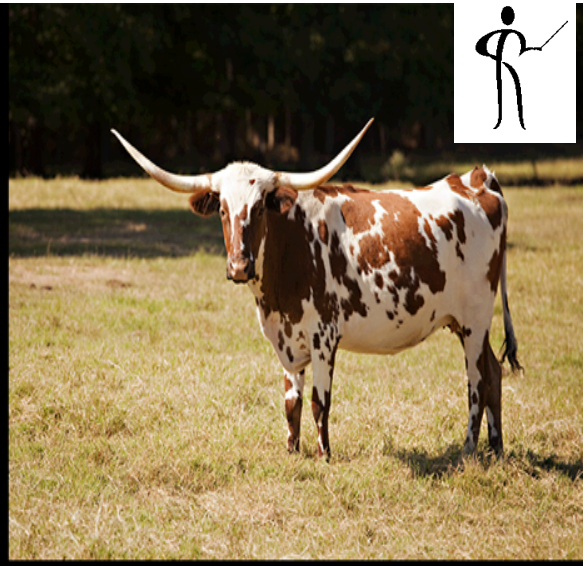
Watch this space !!!

Dates & Venue for RMOUG 2018 coming soon

PHOTO CREDIT: Mike Landrum, SQL Developer and the "Data Tsunami" from i-Behavior


COLLABORATE17
TECHNOLOGY AND APPLICATIONS FORUM
FOR THE ORACLE COMMUNITY
Save the Date for
COLLABORATE 17^
We'll see you again in Vegas
WELCOME
TO The Fabulous
LAS VEGAS
NEVADA
APRIL 2 - 6, 2017 | MANDALAY BAY RESORT & CASINO

King Training Resources



Please Complete Session Evaluations

Exposing Oracle 12c with ORDS

To contact the author:

John King

King Training Resources

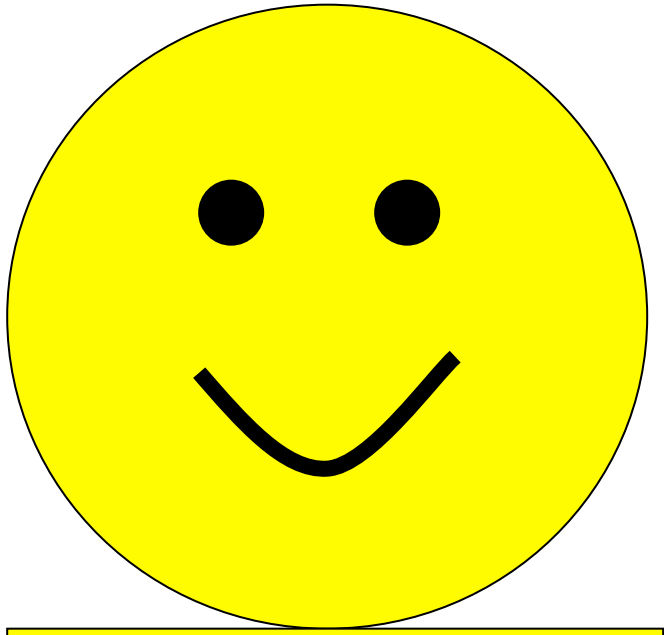
P. O. Box 1780

Scottsdale, AZ 85252 USA

1.800.252.0652 - 1.303.798.5727

Email: john@kingtraining.com

Twitter: royaltwit



Thanks for your attention!

Today's slides and examples are on the web:

http://www.kingtraining.com

- End