At Your Service: Web Services in APEX

Scott Spendolini
Vice President, APEX+ Practice

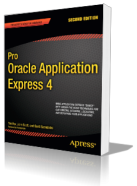
VISCOSITY
NORTH AMERICA

Welcome

About Me

scott.spendolini@viscosityna.com

@sspendol



ORACLE

Sumner TECHNOLOGIES

VISCOSITY NORTH AMERICA

accenture

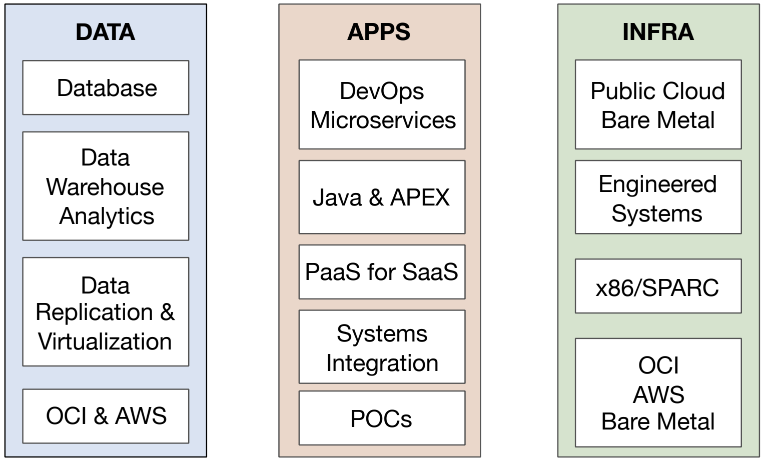
sumnēva Application Success

enkitec

SUMNER technologies

ORACLE ACE Director

About Viscosity: Services



DATA	APPS	INFRA
Database	DevOps Microservices	Public Cloud Bare Metal
Data Warehouse Analytics	Java & APEX	Engineered Systems
Data Replication & Virtualization	PaaS for SaaS	x86/SPARC
OCI & AWS	Systems Integration	OCI AWS Bare Metal
	POCs	

About Viscosity: APEX

- Offers a wide range of APEX related Services:
  - Architecture Design & Reviews
  - Security Reviews
  - Health Checks
  - Education
    - On-site, On-line, On-Demand
    - Custom & Mentoring
  - DevOps
  - Curators of APEX-SERT & sumnerAMP

ORACLE Platinum Partner

Agenda

- Overview
- ACL & Oracle Wallet
- Web Source Modules in APEX 18.x
- Web Source Modules in APEX 19.1
- Summary

Viscosity @ RMOUG

- **Tuesday, February 19**
- *9:00 AM – 12:00 PM | Standley 1 | Charles Kim, Jerry Ward, Scott Spendolini | APEX for the DBA, Pre-Conference Workshop*
- ***1:00 PM – 2:00 PM | Meadowbrook 1 | Scott Spendolini | At Your Service: Web Services & APEX***
- 3:45 PM – 4:45 PM | Standley 1 | Nitin Vengulekar | Oracle Autonomous Data Warehouse Cloud: Testing, Experiences, Results
- **Wednesday, February 20**
- 8:30 AM – 9:30 AM | Windsor | Nitin Vengurlekar | Oracle cloud for EBS/Exadata Cloud Service: From Planning to Provisioning
- 8:30 AM – 9:30 AM | Standley 1 | Rich Niemiec | The Oracle 18c Best New Features & a Few 12cr2 Tips
- 1:30 PM – 2:30 PM | Standley 1 | Charles Kim | Get Ready for Brain Overload with Oracle Database 12.2 & 18c Features
- *1:30 PM – 2:00 PM | Meadowbrook 2 | Scott Spendolini | APEX Security Checklist*
- 4:15 PM – 5:15 PM | Cotton Creek 1 | Charles Kim | Bulletproof Your Data Guard with Best Practices
- 6:30 PM – 9:30 PM | Westin Westminster | Happy Hour at Kachina Southwest Grill
- **Thursday, February 21**
- 11:15 AM – 12:15 PM | Standley 1 | Rich Niemiec | Innovation, the Oracle Cloud, Big Data, & The Internet of Things

Overview

Integration

- Exchanging data between two Oracle database has traditionally been easy
  - Create a database link
  - And you're done…

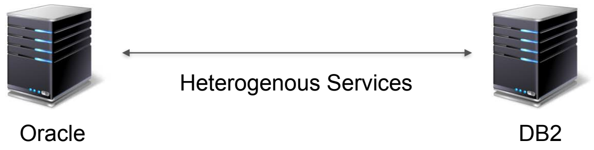
Database Link

Oracle

Oracle

Integration

- Integrating Oracle to other popular databases is also not that difficult (but not cheap)
  - Use Oracle Heterogenous Services
  - And you're done…

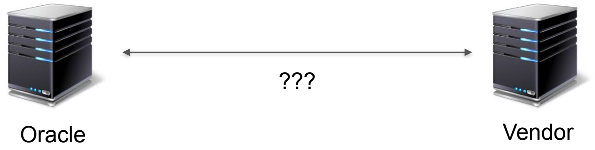

Heterogenous Services

Oracle

DB2

Integration

- Integrating Oracle to an external data source has been difficult
  - No idea what flavor the database is
  - Definitely not going to get DB Link or similar access


???

Oracle

Vendor

Integration

- Thus, the industry has adopted web services as a way to get two servers to communicate
  - Early days used SOAP & XML
  - Difficult to configure and use
  - Payload always outweighed the message

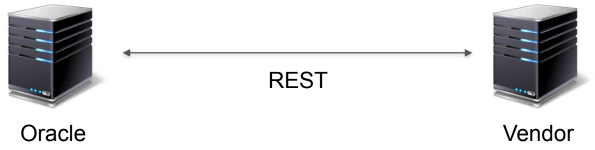
SOAP

Oracle

Vendor

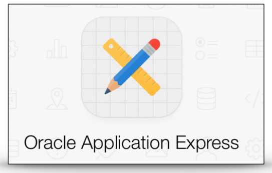
Integration

- Because of SOAP’s difficulties, the industry turned to REST-based web services
  - Based on JSON
  - Simple to use & consume
  - Lightweight payload for more efficient transfers



Evolution of Web Service Support

- APEX has supported some sort of web service integration since APEX 3.0
  - Initially **SOAP & XML**; eventually **REST** via the **Web Service Reference** shared component
  - Extended support for both SOAP & REST via the **APEX_WEB_SERVICE** API
- Starting with APEX 18.1, a new component - **web source modules** - makes consuming RESTful web services as declarative as the rest of APEX



Web Source Modules

- **Web Source Modules** are a new **shared component** in APEX 18.1 that allows **developers to declaratively define and refer to a RESTful web service**
  - As simple as pasting the URL and following the wizard
- Web Source Modules can be used as the source of:
  - Interactive Reports
  - Charts
  - Calendars
  - Classic Reports

ACL & Oracle Wallet

ACL & Oracle Wallet

- When making a web service call, there's **two steps** that need to be considered
  - **Create an ACL entry for the site**
  - **If using HTTPS, import a certificate into the Oracle Wallet**
- If these steps are not performed, then there is no way to call any web service

ACL

- The ACL - or **Access Control List** - is a way to restrict which servers the database can communicate with
- By default, there are no lists
  - Thus, the database cannot communicate with any external server - even those in the same network
- **Any site** - regardless of whether SSL is used or not - **will need a corresponding entry in the ACL** for web services to be used
- Created and managed by SYS/SYSTEM

Creating the ACL

```
BEGIN

DBMS_NETWORK_ACL_ADMIN.CREATE_ACL(
  acl         => 'scratch.xml',
  description => 'APEX Web Services Demo',
  principal   => 'APEX_180200',
  is_grant    => TRUE,
  privilege   => 'connect');

DBMS_NETWORK_ACL_ADMIN.ASSIGN_ACL (
     acl         => '/sys/acls/scratch.xml',
     host        => 'api.openweathermap.org',
     lower_port  => 80,
     upper_port  => 80
     );
DBMS_NETWORK_ACL_ADMIN.ASSIGN_ACL (
     acl         => '/sys/acls/scratch.xml',
     host        => 'apex.oracle.com',
     lower_port  => 443,
     upper_port  => 443
     );

COMMIT;

END;
/
```

Creates an ACL called "scratch.xml"; principal should be APEX_180200 if calling from APEX

Adds the ability to contact api.openweathermap.org on port 80

Adds the ability to contact apex.oracle.com on port 443

Verifying the ACL

- You can query the report **DBA_NETWORK_ACLS** to see what host is mapped to which ACL

```
SELECT
  host,
  lower_port,
  upper_port,
  acl
FROM
  dba_network_acls
WHERE
  acl LIKE '%scratch.xml'

HOST                      LOWER_PORT UPPER_PORT  ACL
------------------------- ---------- ---------- -------------------------
api.openweathermap.org            80         80  /sys/acls/scratch.xml
apex.oracle.com                  443        443  /sys/acls/scratch.xml
```

- More details re: ACL can be found here:
  - https://oracle-base.com/articles/11g/fine-grained-access-to-network-services-11gr1

Oracle Wallet

- Most web services will be served over **HTTPS**
  - And you won’t have any way to change this
- If that is the case, then you’ll need to install their certificate into your **Oracle Wallet** on your database server where APEX is running
  - Which you must manually obtain and install

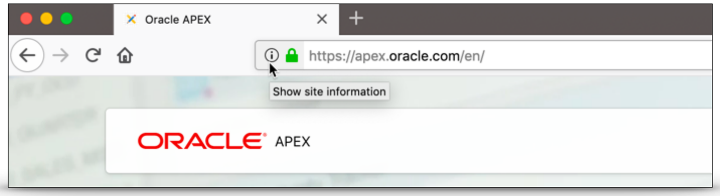
Obtaining Site Certificates

Obtaining a Site Certificate

- **You must perform these steps with Firefox**
  - Doesn’t matter what browser you prefer or use
  - Firefox has the most reliable way to download site certificates
- More details can be found here:
  - https://blogs.oracle.com/apex/apex-https-certificates-and-the-oracle-wallet
  - https://apex.oracle.com/pls/apex/germancommunities/apexcommunity/tipp/6121/index-en.html

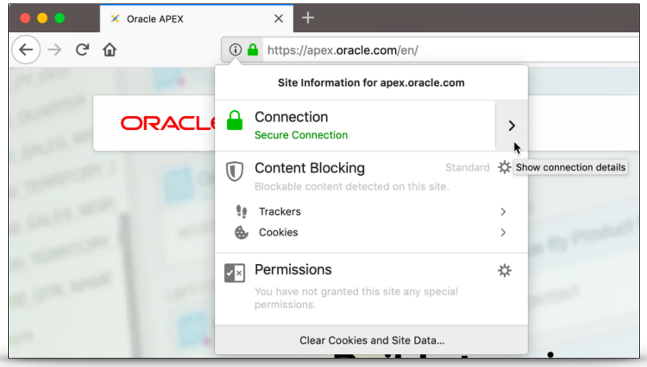
Step 1

- Click the **Info** icon next to the lock in the URL bar



Step 2

- Click the “>” icon next to the Connection portion of the window



Site Information for apex.oracle.com

Connection
Secure Connection

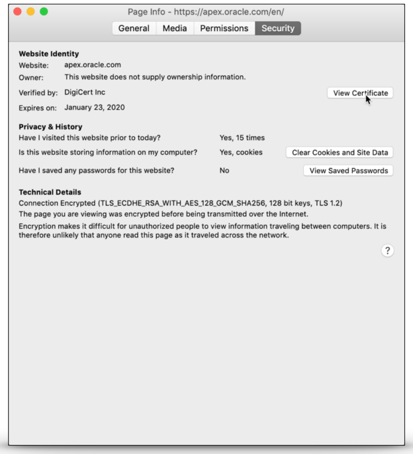
Content Blocking — Standard
Blockable content detected on this site.

Trackers
Cookies

Permissions
You have not granted this site any special permissions.

Step 3

- When the dialog window opens, select **View Certificate**



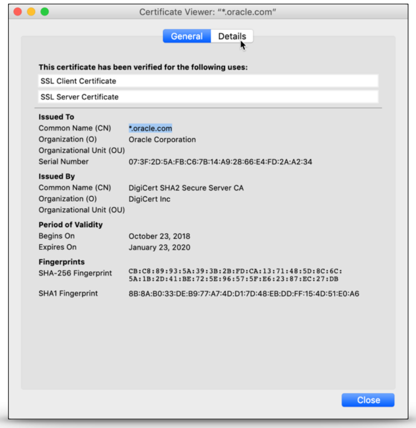
Website Identity
Website: apex.oracle.com
Owner: This website does not supply ownership information.
Verified by: DigiCert Inc
Expires on: January 23, 2020

Privacy & History
Have I visited this website prior to today? Yes, 15 times
Is this website storing information on my computer? Yes, cookies
Have I saved any passwords for this website? No

Technical Details
Connection Encrypted (TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, 128 bit keys, TLS 1.2)
The page you are viewing was encrypted before being transmitted over the Internet.
Encryption makes it difficult for unauthorized people to view information traveling between computers. It is therefore unlikely that anyone read this page as it traveled across the network.

Step 4

- Click on the **Details** tab



This certificate has been verified for the following uses:
SSL Client Certificate
SSL Server Certificate

Issued To
Common Name (CN) *.oracle.com
Organization (O) Oracle Corporation
Organizational Unit (OU)
Serial Number 07:3F:2D:5A:FB:C6:7B:14:A9:28:66:E4:FD:2A:A2:34

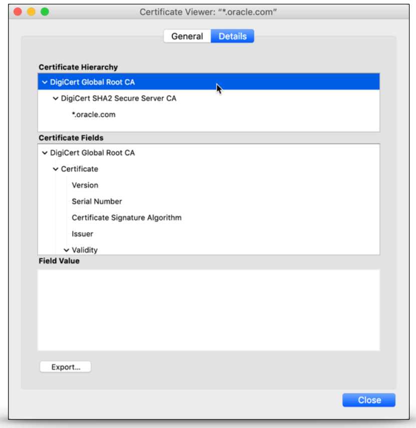
Issued By
Common Name (CN) DigiCert SHA2 Secure Server CA
Organization (O) DigiCert Inc
Organizational Unit (OU)

Period of Validity
Begins On October 23, 2018
Expires On January 23, 2020

Fingerprints
SHA-256 Fingerprint CB:C8:89:93:5A:39:3B:2B:FD:CA:13:71:48:5D:8C:6C:5A:1B:2D:41:BE:72:5E:96:57:5F:E6:23:87:EC:27:DB
SHA1 Fingerprint 8B:8A:B0:33:DE:B9:77:A7:4D:D1:7D:48:EB:DD:FF:15:4D:51:E0:A6

Step 5

- Select the **top node** in the **Certificate Hierarchy** window
  - The names will differ depending on which site you’re visiting



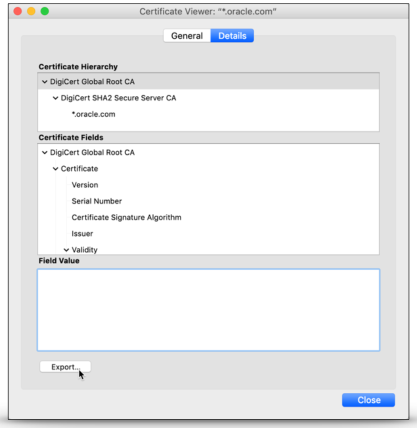
Certificate Hierarchy
DigiCert Global Root CA
DigiCert SHA2 Secure Server CA
*.oracle.com

Certificate Fields
DigiCert Global Root CA
Certificate
Version
Serial Number
Certificate Signature Algorithm
Issuer
Validity

Field Value

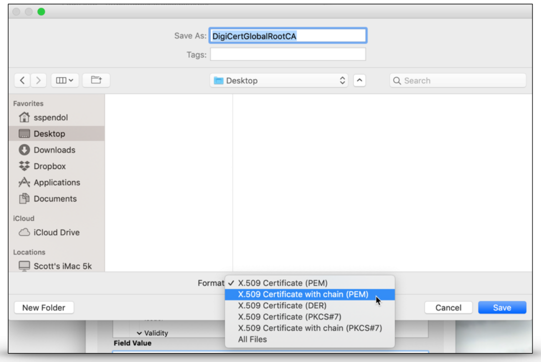
Step 6

- Click **Export…**



Step 7

- Set the **Format** to **X.509 Certificate with chain (PEM)** and click **Save**



Save As: DigiCertGlobalRootCA

Format: X.509 Certificate with chain (PEM)

Creating the Wallet

- Create a wallet
  - `orapki wallet create -wallet /ORCL/u01/app/oracle/wallet/ -pwd oracle123 -auto_login`
- Add a certificate to the wallet

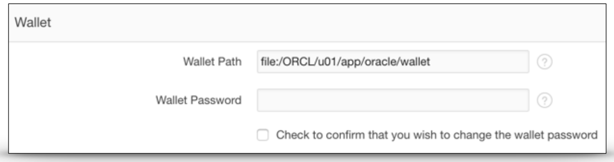
  -
    ```
    orapki wallet add -wallet /ORCL/u01/app/oracle/wallet/
      -trusted_cert -cert
      "/ORCL/u01/app/oracle/ssl/DigiCertGlobalRootCA.crt"
      -pwd oracle123
    ```

Your certificate path & name here

- List the wallet contents to verify it
  - `orapki wallet display -wallet /ORCL/u01/app/oracle/wallet/`

Associating the Wallet with APEX

- **Location & Password** for the **Oracle Wallet** must be defined via the **APEX Instance Admin** console
  - **Manage Instance > Instance Settings**
  - Be sure to prefix the **Wallet Path** with **`file:`**



Wallet

Wallet Path: file:/ORCL/u01/app/oracle/wallet

Wallet Password:

□ Check to confirm that you wish to change the wallet password

Troubleshooting

- At this point, your **HTTPS-based web service should be able to be called from APEX**
- If not, **check on the following**
  - Wallet password is correct and in correct location
  - Wallet Location is prefixed with file:
  - Re-download & import certificate using different formats
  - Download & import intermediate certificate and/or site certificate as well
  - ACL lists all expected entries
  - Principal Name in ACL entry is ALL CAPS
  - Port Numbers in ACL are correct

Troubleshooting

- Use **APEX_WEB_SERVICE** via APEX's SQL Workshop to test the web service
  - Won't work in SQL Developer/SQLcl because the principal in the ACL is **APEX_180200**, not the parse-as schema

```
SELECT
  apex_web_service.make_rest_request
  (
  p_url         => 'https://servername/path_to_webservice',
  p_http_method => 'GET',
  p_wallet_path => 'file:/ORCL/u01/app/oracle/wallet',
  p_wallet_pwd => 'oracle123'
  )
FROM
  dual
```

Web Source Modules in APEX 18.x

OpenWeatherMap

- Free web service offering **various weather forecasts**
  - https://openweathermap.org
  - Need to register for an account, but they are free as long as you don't make more than 60 calls per minute
  - Paid plans available if you need more than that
- Ideal to use something like this to **learn about web services**
  - Data is fairy easy to understand
  - APIs are well documented

Current Weather

Current Weather

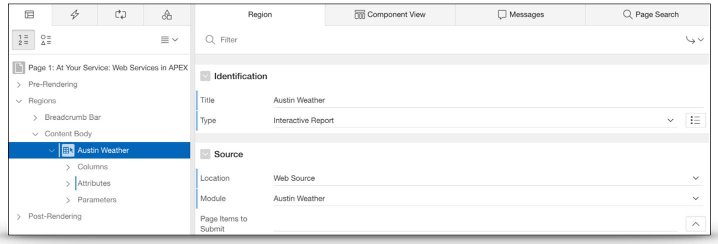
- Let’s start with a simple call - current weather
  - We’ll pass in the ZIP code for **Austin, TX**
  - **http://api.openweathermap.org/data/2.5/weather?zip=78738,us&appid={API Key}&units=imperial**

Web Source Module

- To start, create the **Web Source Module** in the Shared Components
  - **Shared Components > Data Sources > Web Source Modules**
  - Take all of the defaults for now

Report on Web Source Module

- Next, create a new **Interactive Report**
  - Set the Location to **Web Source**
  - Select **Austin Weather** for the Module
  - Save & Run



Identification

Title: Austin Weather

Type: Interactive Report

Source

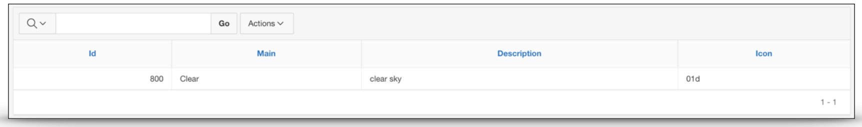
Location: Web Source

Module: Austin Weather

Page Items to Submit

Results

- Results are a single row of data
  - And not very helpful



Id	Main	Description	Icon
800	Clear	clear sky	01d

1 - 1

Web Source Module

Web Source Module

- A **Web Source Module** will have several components that can be configured
  - **Description**
  - **Data Profile**
  - **Operations**
  - **Authentication**
  - **Module Parameters**
  - **Advanced**

Description

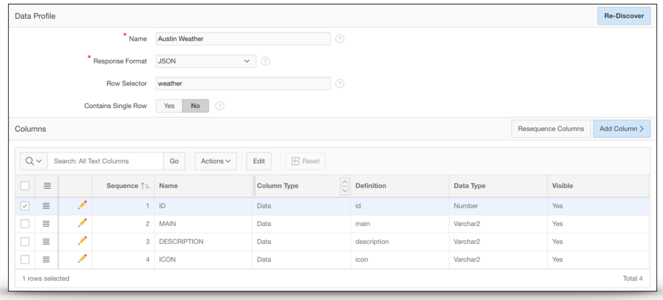
- Web Source Type
  - **Simple HTTP**
    - Used for any REST web service; most components need to be manually set
  - **Oracle REST Data Services**
    - User with an ORDS-based Web Service; server-side filtering & ordering supported
  - **Oracle SaaS**
    - Used for Oracle SaaS; pagination supported

Data Profile

- The Data Profile is how APEX **maps the content of the web service to Oracle datatypes**
  - Data can also be transformed here
- Source can be either **JSON** or **XML**
  - Data can be a single row or multiple rows

Data Profile

- Looking at our weather forecast web service, we see that there are **four rows**
  - Each with their own data type & other options
- **Row Selector** is **weather**



Data Profile

* Name	Austin Weather
* Response Format	JSON
Row Selector	weather
Contains Single Row	Yes [✓ No]

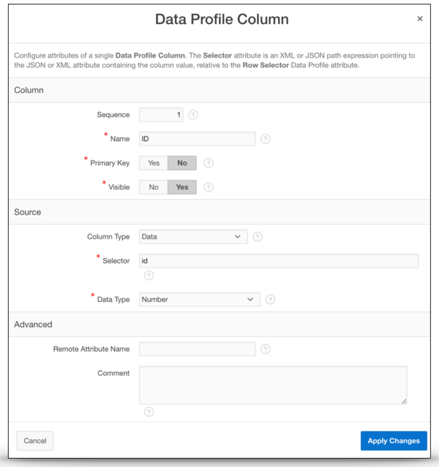
Columns

☐			Sequence	Name	Column Type	Definition	Data Type	Visible
☑			1	ID	Data	id	Number	Yes
☐			2	MAIN	Data	main	Varchar2	Yes
☐			3	DESCRIPTION	Data	description	Varchar2	Yes
☐			4	ICON	Data	icon	Varchar2	Yes

1 rows selected Total 4

Data Profile

- Each column can be **modified/deleted**
  - Or new columns can be **created**
- Columns can also be **derived** via a SQL expression
  - Useful for data format masks or other formatting issues



Data Profile Column

Configure attributes of a single **Data Profile Column**. The **Selector** attribute is an XML or JSON path expression pointing to the JSON or XML attribute containing the column value, relative to the **Row Selector** Data Profile attribute.

Column

Sequence	1
* Name	ID
* Primary Key	Yes [✓ No]
* Visible	No [✓ Yes]

Source

Column Type	Data
* Selector	id
* Data Type	Number

Advanced

Remote Attribute Name	
Comment	

Operations

- Determines the type of web service operation
- Supported operations:
  - **GET**
  - **POST**
  - **PUT**
  - **DELETE**
  - **PATCH**

Operations

- **Database Operation**
  - Fetch Rows
  - Fetch Single Row
  - Insert Row
  - Update Row
  - Delete Row
- Select a database operation that APEX can use when mapping web service modules to APEX forms

Operations

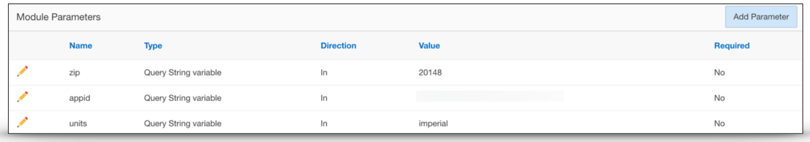
- **Caching**
  - Web source modules can **cache their results**
- Cache can be:
  - **Disabled**
  - **For All Users**
  - **By User**
  - **By Session**
- Invalidate When
  - Determines how long the cache is good for
  - Can use DBMS_SCHEDULER syntax for recurring refreshes

Authentication

- Allows a set of **credentials** to be associated with the web source module
  - **Credentials** are defined as a **Shared Component** in APEX 18.1+
  - Uses **Basic Authentication** or **OAuth2**
  - Can also be used with Social Sign On or other services

Module Parameters

- **Parameters** that can be passed to the web service
  - In our Weather Conditions web service, APEX automatically parsed the URL and created three parameters:
    - zip
    - appid
    - units



Module Parameters

	Name	Type	Direction	Value	Required
✎	zip	Query String variable	In	20148	No
✎	appid	Query String variable	In		No
✎	units	Query String variable	In	imperial	No

Module Parameters

- **Type**
  - HTTP Header
  - HTTP Cookie
  - URL Pattern Variable
    - What's to the left of the "?"
  - Query String Variable
    - What's to the right of the "?"
  - Request/Response Body

Module Parameters

- **Name**
- **Value**
- **Direction**
  - In
  - Out
  - In/Out
- **Static**
  - Determines if parameter can be overridden by APEX components

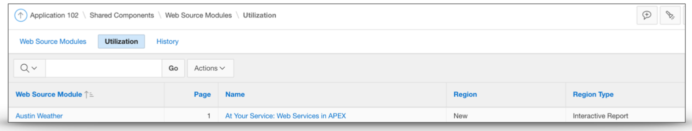
Advanced

- **Static ID**
  - Used to reference web source module in API calls
- **Pass ECID**
  - Passes the ECID to web service for end-to-end tracing
- **HTTP Transfer Timeout**
  - Time to wait before giving up

Altering the Data Profile

Notice

- Since we have a report dependent on this web source module, we will **not be able to remove columns** that are referenced
  - Thus, for our purposes, let's delete the report from our application before we proceed
- Check the **Utilization** report to see which web source modules are used on which page

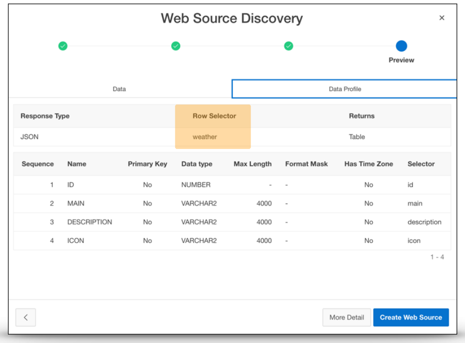


Application 102 \ Shared Components \ Web Source Modules \ Utilization

Web Source Module	Page	Name	Region	Region Type
Austin Weather	1	At Your Service: Web Services in APEX	New	Interactive Report

Editing the Data Profile

- Looking at the data profile, we can see that it chose the data values under the selector "**weather**"
  - This was the only array defined in the web service



Web Source Discovery

Response Type	Row Selector	Returns
JSON	weather	Table

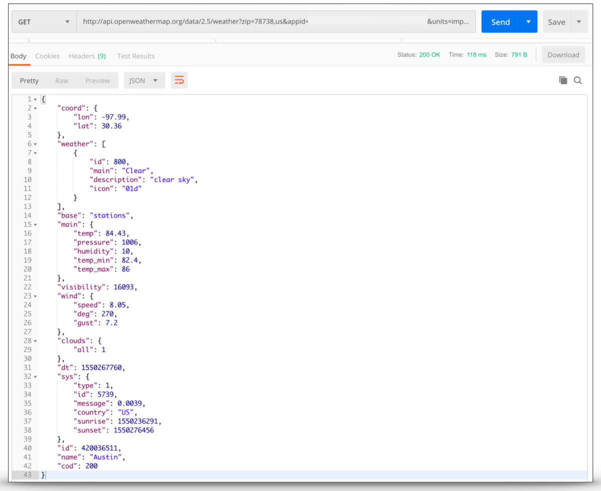
Sequence	Name	Primary Key	Data type	Max Length	Format Mask	Has Time Zone	Selector
1	ID	No	NUMBER	-	-	No	id
2	MAIN	No	VARCHAR2	4000	-	No	main
3	DESCRIPTION	No	VARCHAR2	4000	-	No	description
4	ICON	No	VARCHAR2	4000	-	No	icon

1 - 4

Raw Web Service

- Let's take a look at the raw JSON sent by the web service
- Best approach is to use a 3rd party tool such as **Postman** to view the raw web service results
  - https://www.getpostman.com/
  - Free for most features
  - Offers paid premium features
- Simply cut & paste the URL in a new window in Postman

JSON in Postman



GET http://api.openweathermap.org/data/2.5/weather?zip=78738,us&appid= &units=imp...

Status: 200 OK Time: 118 ms Size: 791 B

```
{
    "coord": {
        "lon": -97.99,
        "lat": 30.36
    },
    "weather": [
        {
            "id": 800,
            "main": "Clear",
            "description": "clear sky",
            "icon": "01d"
        }
    ],
    "base": "stations",
    "main": {
        "temp": 84.43,
        "pressure": 1006,
        "humidity": 10,
        "temp_min": 82.4,
        "temp_max": 86
    },
    "visibility": 16093,
    "wind": {
        "speed": 8.05,
        "deg": 270,
        "gust": 7.2
    },
    "clouds": {
        "all": 1
    },
    "dt": 1550267760,
    "sys": {
        "type": 1,
        "id": 5739,
        "message": 0.0039,
        "country": "US",
        "sunrise": 1550236291,
        "sunset": 1550276456
    },
    "id": 420036511,
    "name": "Austin",
    "cod": 200
}
```

Current Weather: Results

```
{
    "coord": {
        "lon": -97.99,
        "lat": 30.36
    },
    "weather": [
        {
            "id": 800,
            "main": "Clear",
            "description": “clear",
            "icon": "01d"
        }
    ],
    "base": "stations",
    "main": {
        "temp": 61.83,
        "pressure": 1008,
        "humidity": 72,
        "temp_min": 60.8,
        "temp_max": 64.4
    },
    "visibility": 16093,
```

```
    "wind": {
        "speed": 9.64,
        "deg": 238.501
    },
    "clouds": {
        "all": 1
    },
    "dt": 1550243700,
    "sys": {
        "type": 1,
        "id": 3900,
        "message": 0.0037,
        "country": "US",
        "sunrise": 1550236307,
        "sunset": 1550276443
    },
    "id": 420036511,
    "name": "Austin",
    "cod": 200
}
```

Current Weather: Results Distilled

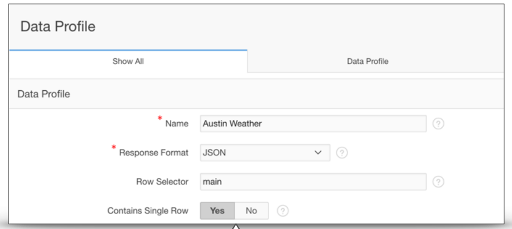
```
coord.lon
coord.lat
weather.id
weather.main
weather.description
weather.icon
base
main.temp
main.pressure
main.humidity
main.temp_min
main.temp_max
visibility
wind.speed
wind.deg
clouds.all
dt
sys.type
sys.id
sys.message
sys.country
sys.sunrise
sys.sunset
id
name
cod
```

APEX chose this selector as the data to display in the web service since it is defined as an array

Current Weather: Results Distilled

```
coord.lon
coord.lat
weather.id
weather.main
weather.description
weather.icon
base
main.temp
main.pressure
main.humidity
main.temp_min
main.temp_max
visibility
wind.speed
wind.deg
clouds.all
dt
sys.type
sys.id
sys.message
sys.country
sys.sunrise
sys.sunset
id
name
cod
```

However, this one looks like it has better forecast data. Let’s change the Web Source Module.



Data Profile

Data Profile

* Name: Austin Weather

* Response Format: JSON

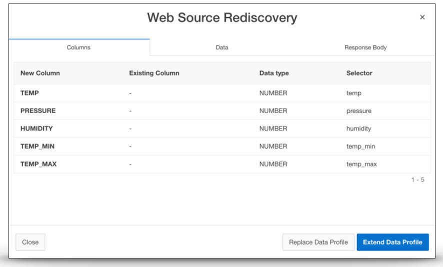
Row Selector: main

Contains Single Row: [✓ Yes] No

We need to change this to “Yes” as well

Web Source Rediscovery

- Edit the **Data Profile**
- Set **Row Selector** to **main** and **Contains Single Row** to **Yes**
- Click **Re-Discover**
- Select **Replace Data Profile** to re-create the web source module with the new columns



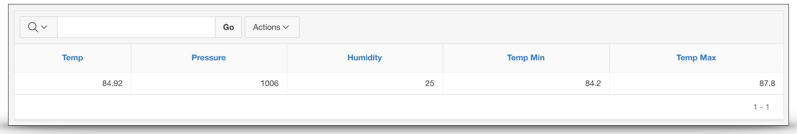
Web Source Rediscovery

New Column	Existing Column	Data type	Selector
TEMP	-	NUMBER	temp
PRESSURE	-	NUMBER	pressure
HUMIDITY	-	NUMBER	humidity
TEMP_MIN	-	NUMBER	temp_min
TEMP_MAX	-	NUMBER	temp_max

1 - 5

APEX IR

- Re-create the IR in your APEX application, referencing the same web source module as before



Temp	Pressure	Humidity	Temp Min	Temp Max
84.92	1006	25	84.2	87.8

1 - 1

Web Source Module w/ Multiple Rows

7-day Forecast

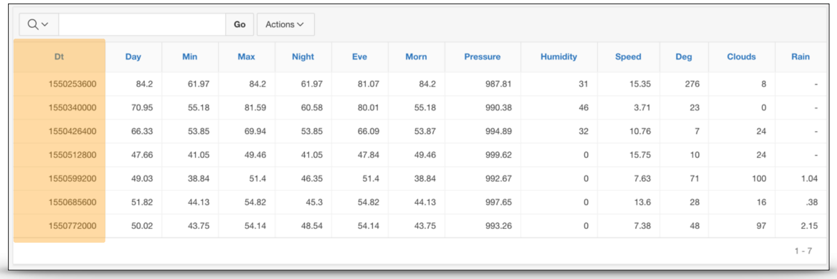
- Next, let's take a look at the 7-day Forecast for Austin
  - **http://api.openweathermap.org/data/2.5/forecast/daily?zip=78738,us&appid={API Key}&units=imperial**

Demo

Multi-Row Web Service

Bad Date

- Create a **new report** based on the **7-day Forecast** web source module
  - Notice that the “**Dt**” column - which should be the Data - is a bit off



Dt	Day	Min	Max	Night	Eve	Morn	Pressure	Humidity	Speed	Deg	Clouds	Rain
1550253600	84.2	61.97	84.2	61.97	81.07	84.2	987.81	31	15.35	276	8	-
1550340000	70.95	55.18	81.59	60.58	80.01	55.18	990.38	46	3.71	23	0	-
1550426400	66.33	53.85	69.94	53.85	66.09	53.87	994.89	32	10.76	7	24	-
1550512800	47.66	41.05	49.46	41.05	47.84	49.46	999.62	0	15.75	10	24	-
1550599200	49.03	38.84	51.4	46.35	51.4	38.84	992.67	0	7.63	71	100	1.04
1550685600	51.82	44.13	54.82	45.3	54.82	44.13	997.65	0	13.6	28	16	.38
1550772000	50.02	43.75	54.14	48.54	54.14	43.75	993.26	0	7.38	48	97	2.15

1 - 7

Bad Date

- That’s because the web service uses “**UNIX epoch time**” to display the date
  - UNIX Epoch time = number of seconds since Jan 1, 1970
- Thus, we can use a derived column to display the date is a more readable format
  - **TO_DATE('19700101','yyyymmdd') + (DT/24/60/60)**

Demo

Derived Web Source Columns

Other Components

- In addition to Interactive Reports, APEX can use Web Source Modules as the source of:
  - **Charts**
  - **Calendars**
  - **Classic Reports**
  - **Interactive Grid (unofficially)**
    - See https://blogs.oracle.com/apex/web-source-modules-and-the-interactive-grid-part-1 for more details

Demo

Charts, Reports & Calendars

Web Source w/ APEX Item Parameters

Passing APEX Items

- Next, let's alter the web source so that we can **pass in the ZIP code** and get forecasts from anywhere in the US
  - Simply **add an item** to the page
  - Set the Web Source report to **submit that item**
  - Change the Web Source parameter to use **&ITEM.** syntax

Demo

Passing APEX Items

ORDS-Based Web Source Modules

ORDS-Based Web Source Modules

- There is a slight advantage if your using ORDS to host your web service and calling it from APEX
  - Ability to **pass in a filter** an apply on the remote server
  - Ability to **sort result set** on the remote server
  - Built-in **pagination**

Filter & Sort

- **Filter & Sort Order** can be passed to the web service using **JSON notation**:
  - Filter
    - `{"column_name" : "search_term"}`
    - `{"column_name" : {"$lt/$gt": 3000}}`
  - Sort Order
    - `{"column_name" : "asc/desc" }`

Local Post-Processing

- If ORDS is not being used, you can **still perform some post-processing activities** on the local database after the web service results are retrieved
  - **Where/Order By Clause**
  - **SQL Query**
  - **PL/SQL Function Returning SQL Query**

Pagination

- With a non-ORDS web service, you're going to get the default number of records from the server per request
  - Let's assume the **result set is 1,000** and the **page size is 100**
- APEX reports will apply their pagination to that result set
  - Let's assume that it's **25**
- Thus, you will see **4 "pages" of 25 rows each**
  - There is no built-in way to get the next 100 records, so you're only seeing 1/10th of the data
  - You can do this manually, but it's a good deal of work

Pagination

- With an ORDS-based web service, this is not the case
- APEX can **automatically get the next result set** and seamlessly integrate it into a report
  - Users have no idea what's going on in the back end
- For that same 1,000 row table, an **APEX report** based on ORDS will allow the user to **paginate through all 1,000 rows** as if the data were local
  - While at the same time, respect the local pagination

Demo

Filter, Sort & Pagination

Web Source Modules in APEX 19.1

“Safe Harbor” Statement

- **NOTE**: this section is based on the **Early Adopter Release** of **APEX 19.1**
  - Things **can and will change** before the production version is released
  - Thus, no major decisions should be made based on the contents of this section

Form Region

- APEX 19.1 introduces a new component - the **Form Region**
  - About time, right?
- Still two processes
  - **Before Header** & **After Submit**
  - Both will reference the Form Regions
- Items will now **map to a Form Region**
  - New attribute defined in the item **Source**
  - Also new - **Data Type**, **Query Only** and **Primary Key**

Form Region

- The Form Region will have a declarative **Data Source Location** attribute
  - Much like reports in APEX 18.1
- Options will include:
  - Local Database
  - Remote
  - Web Source

Form Region & Web Source

- Thus, in APEX 19.1, you will be able to use a **Web Source Module as the source of a Form**
- Provided that:
  - The web source module has the corresponding transaction types
  - Required credentials are included in the calls
  - The database can communicate with the server that hosts the web service

Summary

Summary

- **Web Source Modules** make it **incredibly easy** to consume an external web service via APEX
  - Hard part is ACL + Wallet
  - Takes getting used to getting the right data from the JSON
- If you have **ORDS**, you get **even more benefits**
  - Sorting, Filtering, Pagination
- **APEX 19.1** will introduce **Forms based on Web Source Modules**
  - Making CRUD operations over web services 100% declarative

VISCOSITY

NORTH AMERICA