

SOA MEANS SERVICE-ORIENTED APEX RIGHT?

Consuming and Providing Web Services with APEX

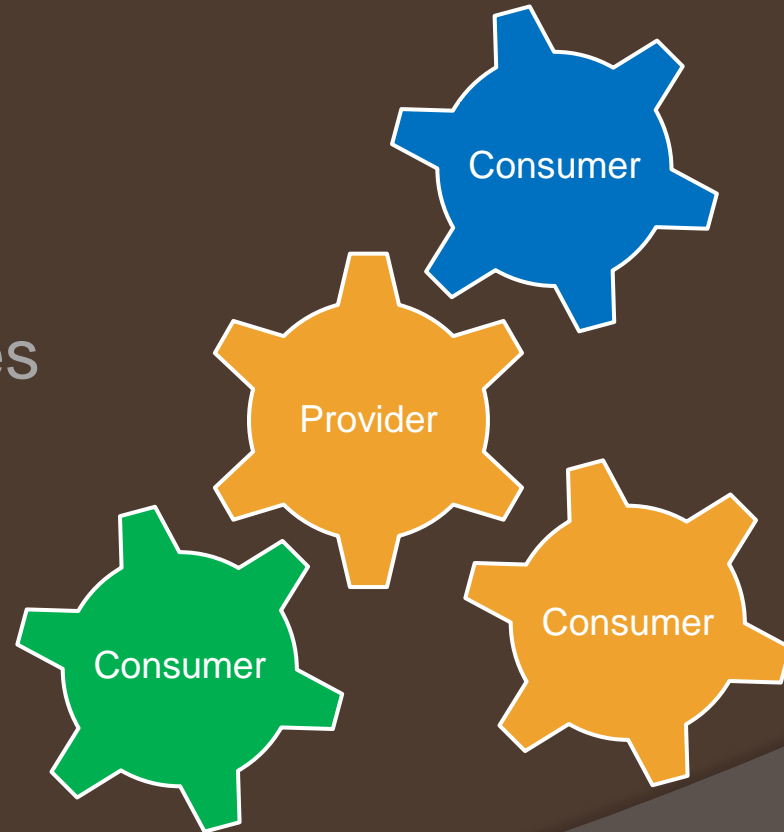
Jonathon Taylor
UTOUG Fall Symposium

Overview

- ⦿ Service Oriented Architecture
- ⦿ Web Services Overview
- ⦿ Consuming Web Services
 - Tools for Testing
- ⦿ APEX Web Services
 - Providing RESTful Access
 - Consuming other Web Services
 - Strengths
 - Weaknesses
- ⦿ Questions

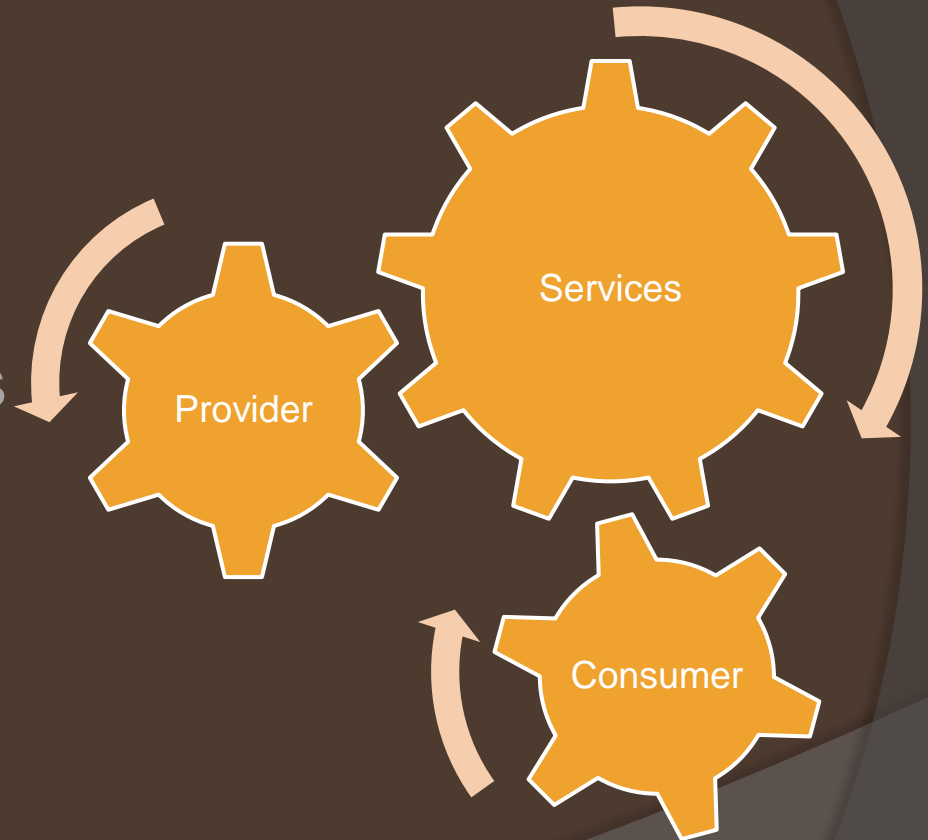
Service-Oriented Architecture

- S - Web Services
- O – Aligned
- A – Framework
 - Business Processes
- Providers
- Consumers



Service-Oriented Architecture

- S - Web Services
- O – Aligned
- A – Framework
 - Business Processes
- Providers
- Consumers
- Services
 - Define Interaction



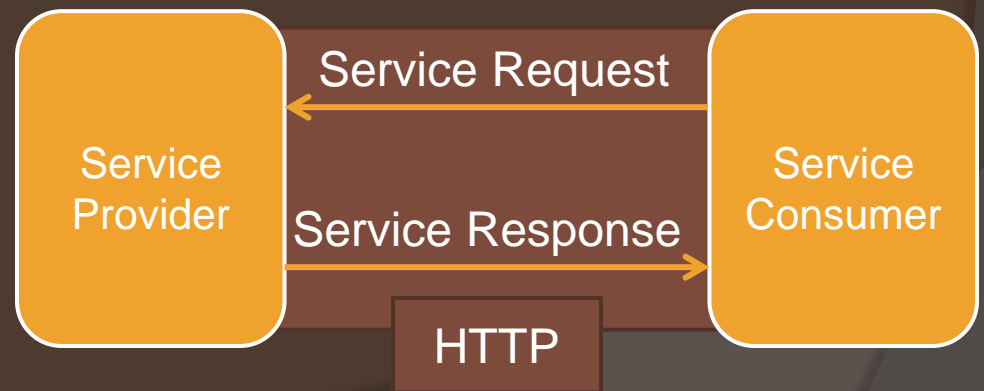
Web Services Overview

- An Implementation of SOA
- Define Service Interface
- Uses HTTP
- Can use XML (JSON)



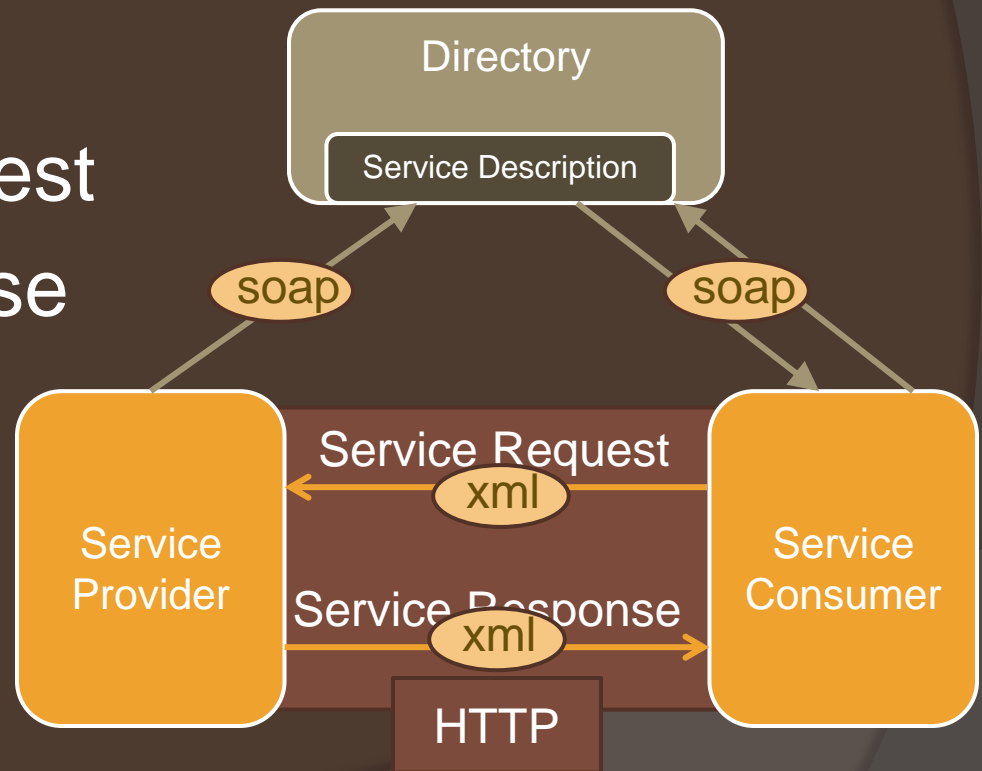
how do web services work?

- REST – Representational State Transfer
 - Architecture Style
 - GET, POST, PUT, DELETE
 - RESTful Web Service over HTTP
- SOAP – Simple Object Access Protocol
 - Uses XML
 - Over HTTP



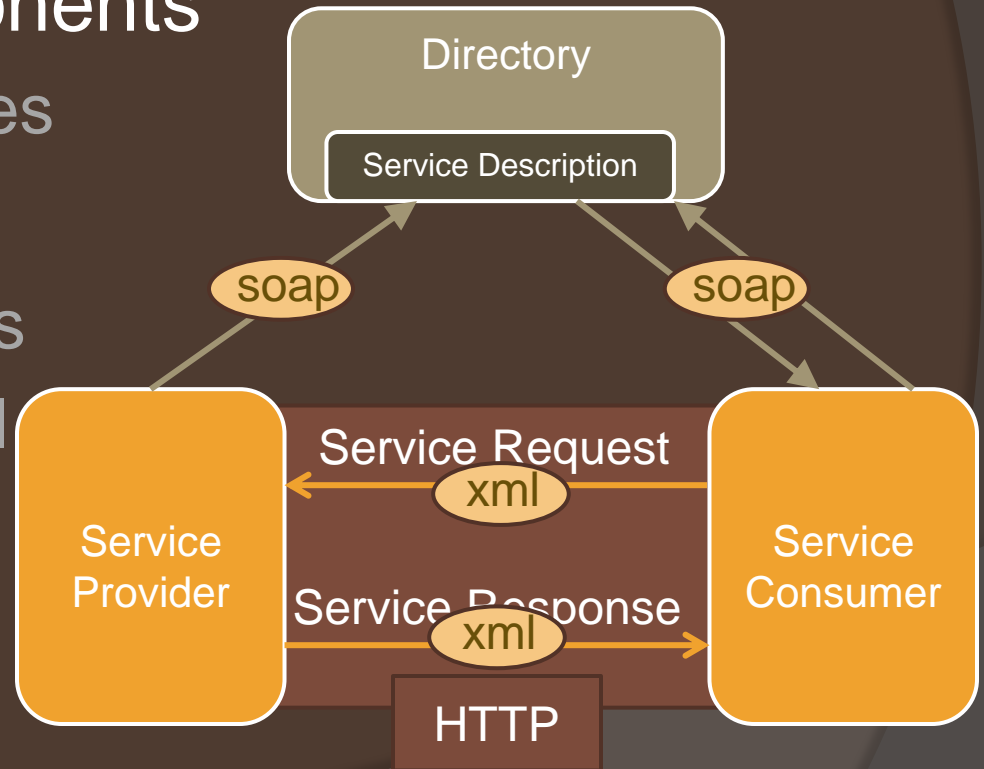
Consuming Web Services

- ◎ WSDL – Web Services Description Language
 - Where to request
 - How to request
- ◎ Format the Request
- ◎ Interpret Response



what are the use cases of web services?

- Layer of Abstraction
- Reusable Components
 - Enterprise Services
- Interoperability
 - Different Platforms
 - Standard Protocol



Tools for Testing

- WSDL description

- Tools

- Create Requests
- Test Services
- Load Testing

- Open Examples

- soapUI
- WebInject

```
<?xml version="1.0" encoding="UTF-8"?>
<wscdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://www.webserviceX.NET/"
  xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:wscdl="http://schemas.xmlsoap.org/wsdl/"
  targetNamespace="http://www.webserviceX.NET/">
  <wscdl:types>
    <?xml version="1.0" encoding="UTF-8"?>
    <s:schema elementFormDefault="qualified" targetNamespace="http://www.webserviceX.NET/">
      <s:element name="ConversionRate">
        <s:complexType base="baseType">
          <s:sequence base="baseSequence">
            <s:element minOccurs="1" maxOccurs="1" name="FromCurrency" type="tns:Currency"/>
            <s:element minOccurs="1" maxOccurs="1" name="ToCurrency" type="tns:Currency"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:simpleType name="Currency">
        <s:restriction base="string"/>
      </s:simpleType>
      <s:element name="ConversionRateResponse">
        <s:complexType base="baseType">
          <s:sequence base="baseSequence">
            <s:element minOccurs="1" maxOccurs="1" name="ConversionRateResult" type="s:double"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="double" type="s:double"/>
    </s:schema>
  </wscdl:types>
  <wscdl:message name="ConversionRateSoapIn">
    <wscdl:part name="parameters" element="tns:ConversionRate"/>
  </wscdl:message>
  <wscdl:message name="ConversionRateSoapOut">
    <wscdl:part name="parameters" element="tns:ConversionRateResponse"/>
  </wscdl:message>
  <wscdl:message name="ConversionRateHttpGetIn">
    <wscdl:part name="FromCurrency" type="s:string"/>
    <wscdl:part name="ToCurrency" type="s:string"/>
  </wscdl:message>
  <wscdl:message name="ConversionRateHttpGetOut">
    <wscdl:part name="Body" element="tns:double"/>
  </wscdl:message>
  <wscdl:message name="ConversionRateHttpPostIn">
    <wscdl:part name="FromCurrency" type="s:string"/>
    <wscdl:part name="ToCurrency" type="s:string"/>
  </wscdl:message>
  <wscdl:message name="ConversionRateHttpPostOut">
    <wscdl:part name="Body" element="tns:double"/>
  </wscdl:message>
  <wscdl:portType name="CurrencyConverterSoap">
    <wscdl:operation name="ConversionRate">
      <wscdl:documentation xmlns:wscdl="http://schemas.xmlsoap.org/wsdl/">
        <br><b>Get conversion rate from one currency to another currency<br><br><p><b><font color='#000080'>

```

Too Complex!

Example: Testing Using soapUI

The image displays two overlapping windows of the soapUI 4.0.1 application. The background window shows a project tree with folders like '2051', 'MORTGAGE', 'STOCKS', 'WHOIS', 'currency', and 'testremedy'. A 'Request 1' is selected under the 'ConversionRate' folder. The foreground window is a 'New soapUI Project' dialog box. It contains the following fields and options:

- Project Name: CURRENCY
- Initial WSDL/WADL: <http://www.websvicex.net/Currency>
- Create Requests: Create sample requests for all opera...
- Create TestSuite: Creates a TestSuite for the imported...
- Create MockService: Creates a Web Service Simulation o...
- Add REST Service: Opens dialog to create REST Service...
- Relative Paths: Stores all file paths in project relativ...
- Create Web TestCase: Creates a TestCase with a Web Recd...

The 'Request Properties' table in the foreground window is as follows:

Property	Value
Name	Request 1
Description	
Message Size	333
Encoding	UTF-8
Endpoint	http://www.webse...
Timeout	
Bind Address	
Follow Redirects	true
Username	

The 'Request 1' window shows the raw XML request and response. The request XML is:

```
<?xml version='1.0' encoding='UTF-8'>
<soap:Envelope xmlns:soap='http://www.w...
  <soap:Header/>
  <soap:Body>
    <web:ConversionRate>
      <web:FromCurrency>USD</web:From...
      <web:ToCurrency>EUR</web:ToCur...
    </web:ConversionRate>
  </soap:Body>
</soap:Envelope>
```

The response XML is:

```
<?xml version='1.0' encoding='UTF-8'>
<soap:Envelope xmlns:soap='http://www.w...
  <soap:Header/>
  <soap:Body>
    <web:ConversionRateResponse xmlns='ht...
      <web:ConversionRateResult>0.794</C...
    </web:ConversionRateResponse>
  </soap:Body>
</soap:Envelope>
```

At the bottom of the request window, it shows 'response time: 281ms (379 bytes)' and '1:1'.

APEX Web Services

⦿ Provide

- RESTful Access to APEX report
- Modify/Create Records?

⦿ Consume

- Web Service References
- Reports on Web Services
- Front-End Forms on Web Services

Example: Providing RESTful Access

- 1. Configure Instance Settings
 - Login to Internal Administrator Workspace
 - “Manage Instance” – “Security”*
 - “Allow Restful Access” - Yes*

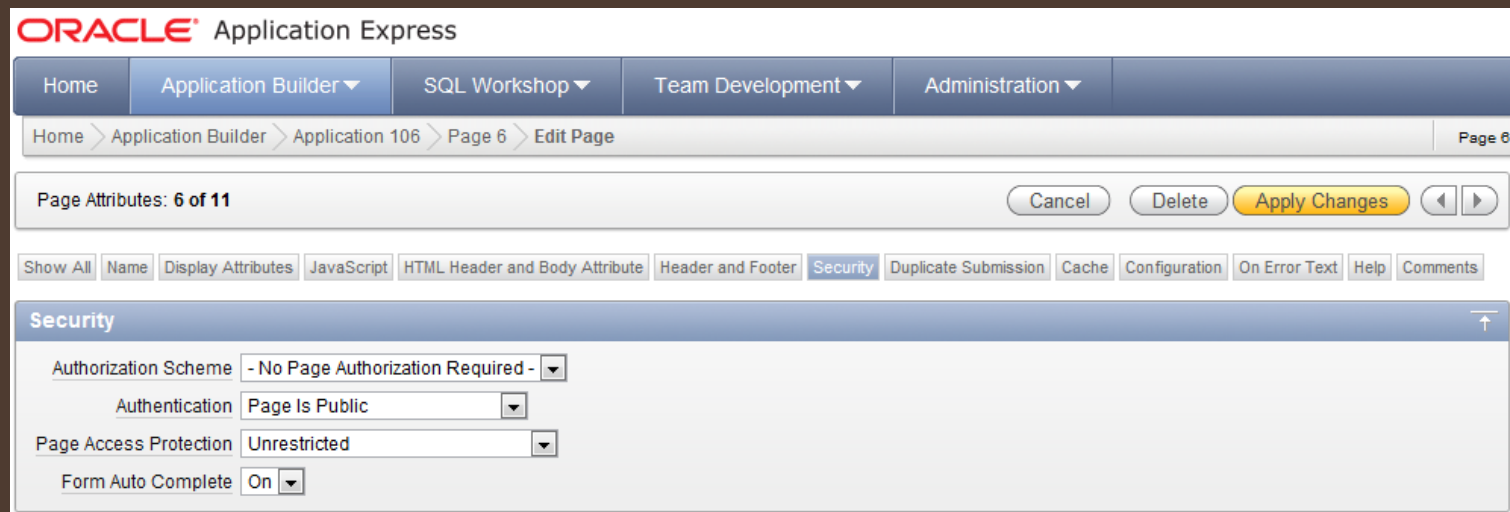
The screenshot shows the Oracle Application Express interface. At the top, there's a navigation bar with tabs: Home, Manage Requests, Manage Instance, Manage Workspaces, and Monitor Activity. Below this is a breadcrumb trail: Home > Manage Instance > Security. There are two tabs: Security Settings (selected) and Authorized URLs. A Cancel button and an Apply Changes button are visible. Below the tabs is a row of filter buttons: Show All, Security, HTTPS, REST (selected), Session Timeout, Excluded Domains, Account Login Control, Workspace Password Policy, and Service Administrator Password Policy. The main content area is titled RESTful Access and contains a single configuration item: Allow RESTful Access, which is set to Yes via a dropdown menu.

Configuration Documentation –

http://docs.oracle.com/cd/E23903_01/doc/doc.41/e21678/adm_mg_service_set.htm#CHDFDBHF

Example: Providing RESTful Access

- 2. Create a Public Page for the Report
 - Page Must be Public!



The screenshot shows the Oracle Application Express interface for editing a page. The breadcrumb trail is: Home > Application Builder > Application 106 > Page 6 > Edit Page. The page attributes section shows 6 of 11 attributes, with the Security tab selected. The Security settings are:

- Authorization Scheme: - No Page Authorization Required -
- Authentication: Page Is Public
- Page Access Protection: Unrestricted
- Form Auto Complete: On

Documentation –

http://docs.oracle.com/cd/E23903_01/doc/doc.41/e21674/advnc_web_services.htm#CHDDBGAI

Example: Providing RESTful Access

- 3. Enable RESTful Access to the Report
 - Enter a value for the Static ID Field
 - "Enable RESTful Access" - Yes

Attributes

Static ID

Region Attributes

Region Display Selector

Region Image

Image tag attributes

Region HTML table cell attributes

Enable RESTful Access

REST Access URL
`http://apexdev.ldschurch.org/pls/apex/apex_rest.getReport?app=106&page=6&reportid=REST_EXAMPLE`

Enabling RESTful Access Documentation –

http://docs.oracle.com/cd/E23903_01/doc/doc.41/e21674/advnc_web_services.htm#CHDFIJDJ

Example: Providing RESTful Access

4. Accessing the RESTful Service

- Use the URL provided in the “REST Access URL”
- Add extra options as needed

http://site.com/pls/apex/apex_rest.getReport?app=106&page=6&reportid=REST_EXAMPLE

- | | | |
|--------------|---|---|
| ○ app | – | 106 |
| ○ Page | – | 6 |
| ○ reportid | – | REST_EXAMPLE |
| ○ output | – | {xml,json} (default xml) |
| ○ parmvalues | – | {value(s),parameter(s)} (e.g. 1,2,3,P1,P2,P3) |

Accessing the Web Service Documentation –

http://docs.oracle.com/cd/E23903_01/doc/doc.41/e21674/advnc_web_services.htm#CHDEHFJI

Parameters to APEX RESTful Service –

<https://forums.oracle.com/forums/thread.jspa?threadID=1128851>

Example: Modify/Create Record

- Add Page Item for each attribute of the record
- Add a PL/SQL process before header to insert record
 - Use request = CREATE condition on process

f?p=106:9::CREATE:::P9_DNAME,P9_LOC:DEVELOPMENT,SLC

The screenshot shows the Oracle APEX page editor interface. The top navigation bar includes a 'Page' dropdown set to '9', navigation buttons (Go, Back, Forward), and a 'Run' button. The main workspace is divided into two panes: 'Page Rendering' on the left and 'Page Processing' on the right. The 'Page Rendering' pane shows a tree view of the page structure, including 'Form on DEPT', 'Before Header', 'After Header', 'Before Regions', 'Computations', 'Processes', 'Regions', 'Body (3)', 'Form on DE', 'Items', 'P9_DNAME', and 'P9_LOC'. A tooltip is displayed over the 'create_record' process, showing its configuration: 'Process Type: PL/SQL anonymous block', 'Process: begin insert into dept values (dept_seq.nextval,:P9_DNAME,:P9_LOC); end;', 'Condition Type: Request = Expression 1 ', and 'Expression 1: CREATE'. The 'Page Processing' pane shows a list of processing events: 'After Submit', 'Validating', 'Processing', 'After Processing', and 'AJAX Callbacks'.

- Update by ID
- Change a Status

Web Service References

- ◎ Based on WSDL
 - Extracts the Description and Fields for you!
 - Sets up Requests, can interpret Response
- ◎ REST
- ◎ Manual
 - Specify URL Endpoint and Action
 - Define SOAP Envelope

Understanding Web Service References –

http://docs.oracle.com/cd/E23903_01/doc/doc.41/e21674/advnc_web_services.htm#HTMDB25882

Example: Form/Report on Web Service

- ◎ Create Form and Report on WS
 - Use a Reference from WSDL
 - Easy, already knows the description
 - Use a Manual Reference
 - Difficult, you have to define it
- ◎ Much easier to let a WSDL define it!

Strengths

- ⦿ Quickly expose a RESTful report
- ⦿ Quick custom forms on Web Services
- ⦿ Creates the requests for you!
- ⦿ Quick and easy to use with WSDLs
- ⦿ Allows manual setup

Weaknesses

- ⦿ Providing Web Services
- ⦿ Cannot secure RESTful report access
- ⦿ Restricted to URL length for REST
- ⦿ Documentation and Examples
- ⦿ Namespaces can mess APEX up
 - xmlns:ns0 instead of default xmlns
- ⦿ Wallets, Proxy Server, Instance Settings

Conclusions

- ⦿ Yesterday's web apps don't cut it!
- ⦿ Learn to integrate
 - Loosely coupled systems play nice
- ⦿ Learn how to use the tools for testing
- ⦿ APEX is getting there with Web Services

Questions?

- **Jonathon Taylor**
- **jonathon.taylor@ldschurch.org**

Sources & Links

- APEX 4.1 Documentation – Implementing Web Services
 - http://docs.oracle.com/cd/E23903_01/doc/doc.41/e21674/advnc_web_services.htm#HTMDB28583
- Parameters to APEX RESTful Service
 - <https://forums.oracle.com/forums/thread.jspa?threadID=1128851>
- Creating a Manual Web Service
 - <https://forums.oracle.com/forums/message.jspa?messageID=10213570#10213570>
- soapUI Website:
 - <http://www.soapui.org/>
- WebInject Website:
 - <http://www.webinject.org/>
- Web Service Examples
 - <http://www.webservicex.net/>