Web Services Using Standard Oracle EBS

Divya Kantem
Senior Consultant
Apps Associates LLC

July 23, 2015
The content of this document is proprietary, confidential information pertaining to Apps Associates’ products, services, solutions and methods. Information herein is intended for use by the client or prospective client only and must not be shared with any third party without consent from Apps Associates.
1. Introduction to Apps Associates
2. Web Services – Overview
3. Consume Web Services – Methods
4. OANDA
5. Expose Web Services
6. Limitations of PL/SQL Web Services
7. SOA Gateway
8. Oracle Integration Repository
9. Q&A
• Global Reach, Broad Service Profile
  – Founded in 2002, 650+ employees
  – US, Europe, India, Middle East
  – Service Offerings: Applications, CRM, Analytics, EPM, Cloud, Middleware, Application Development, App & Infrastructure Managed Services

• Significant Capabilities
  – Cloud (IaaS, PaaS, SaaS)
  – Business Process & System Integration
  – Analytics & Big Data

• Strategic Partnerships, Certifications, Credentials
  – Oracle Platinum Partner
  – AWS Advanced Consulting Partner
  – Salesforce Cloud Alliance Partner
  – MuleSoft Partner
## Competencies and Capabilities

<table>
<thead>
<tr>
<th>Applications</th>
<th>Business Intelligence/ EPM</th>
<th>Technology Services</th>
<th>Infrastructure Services</th>
<th>Product Development</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle EBS</td>
<td>OBIEE/BI Apps Hyperion/Essbase</td>
<td>Fusion Middleware AIA/SOA ADF</td>
<td>DBA Sys Admin Cloud Hosting</td>
<td>Industry Solutions</td>
</tr>
<tr>
<td>Salesforce.com</td>
<td>Exalytics Exadata</td>
<td>Microsoft Services SharePoint .NET App Development</td>
<td>Middleware Admin Security Solutions High Availability</td>
<td>ShipConsole</td>
</tr>
<tr>
<td>Fusion Apps</td>
<td>BICS, Endeca</td>
<td></td>
<td></td>
<td>Freight Shopping</td>
</tr>
<tr>
<td>Project Accounting</td>
<td></td>
<td></td>
<td></td>
<td>Customs Doc.</td>
</tr>
</tbody>
</table>

### Project Services
- Assessments, Implementations, Upgrades, Migrations

### Support Services
- Managed Services and Staff Augmentation

### Quality Assurance
- PMF, CMMI Certification, ITIL Methodologies, SSAE16/SAS70 Type II
Multi-Pillar Competencies & Partnerships

Centers of Excellence

<table>
<thead>
<tr>
<th>Enterprise Applications</th>
<th>Analytics</th>
<th>Integration &amp; Custom Dev</th>
<th>Product Development</th>
<th>Infrastructure as a Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>• ERP</td>
<td>• BI</td>
<td>• Dev to Client Reqs.</td>
<td>• Our Products</td>
<td>• Cloud Adoption</td>
</tr>
<tr>
<td>• Planning</td>
<td>• Big Data / High Volume Data Sets</td>
<td>• Business Process Integration</td>
<td>– ShipConsole – Photo capture app</td>
<td>• Managed Services</td>
</tr>
</tbody>
</table>

Strategic Partnerships & Certifications

- ORACLE Platinum Partner
- Amazon Web Services Partner Network
- Salesforce Cloud Alliance Partner
- UPS
- MuleSoft

Web Services
Web Services Overview

• Application-to-application interaction
• XML, SOAP, WSDL and UDDI
• Services in middle-tier application servers
Database Web Services

Two directions:

• Accessing database resources (database call-in)
  – PL/SQL packages
  – Java classes

• Consuming external Web services (database call-out)
  – SQL engine
Examples:
- Online store
- Quotes to customers
Examples:

- Inventory information from multiple suppliers
- OANDA
Consume Web Services
Consume Web Services

• Using UTL_HTTP
  – Communicate directly with the WebService and send and retrieve entire SOAP messages

• Develop Java Classes
  – Consume the WebService and load these into the database wrapped with a PL/SQL package interface

• Using UTL_DBWS
  – Allows generic WebService consumption from within PL/SQL
Method 1: UTL_HTTP
• Makes HTTP callouts from SQL and PL/SQL
• Access data on the Internet
• Supports HTTPS

<table>
<thead>
<tr>
<th>Common Procedure/Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEGIN_REQUEST</td>
</tr>
<tr>
<td>WRITE_TEXT</td>
</tr>
<tr>
<td>READ_TEXT</td>
</tr>
<tr>
<td>SET_TRANSFER_TIMEOUT</td>
</tr>
</tbody>
</table>
Sequence of steps

- $\text{l\_http\_request} := \text{UTL\_HTTP.begin\_request(url, method, http\_version, request\_context)}$
- $\text{UTL\_HTTP.set\_header(l\_http\_request, name, value)}$
- $\text{UTL\_HTTP.write\_text(l\_http\_request, l\_param\_list)}$
- $\text{l\_http\_response} := \text{UTL\_HTTP.get\_response(l\_http\_request)}$
- $\text{UTL\_HTTP.read\_text(l\_http\_response, l\_response\_text)}$
- $\text{UTL\_HTTP.end\_response(l\_http\_response)}$
DECLARE

l_parameters VARCHAR2(512);
l_http_request UTL_HTTP.req;
l_http_response UTL_HTTP.resp;
l_response VARCHAR2(32767);
BEGIN
  -- input parameters
  l_parameters := 'FromCurrency=USD'||'&'||'ToCurrency=INR';
  -- preparing request
  l_http_request := UTL_HTTP.begin_request('http://www.webservicex.net/currencyconvertor.asmx/ConversionRate', 'POST', 'HTTP/1.1');
  UTL_HTTP.set_header(l_http_request, 'Content-Type', 'application/x-www-form-urlencoded');
  UTL_HTTP.set_header(l_http_request, 'Content-Length', LENGTH(l_parameters));
  -- set input parameters
  UTL_HTTP.write_text(l_http_request, l_parameters);
  -- get response
  l_http_response := UTL_HTTP.get_response(l_http_request);
  UTL_HTTP.read_text(l_http_response, l_response);
  DBMS_OUTPUT.PUT_LINE('l_response'||l_response);
  -- finalizing
  UTL_HTTP.end_response(l_http_response);
EXCEPTION
  WHEN UTL_HTTP.end_of_body THEN
    UTL_HTTP.end_response(l_http_response);
END;

© Copyright 2015. Apps Associates LLC.
UTL_HTTP Example

Response from Web Service

```xml
<response version="1.0" encoding="utf-8">
  <double xmlns="http://www.webserviceX.NET/">63.3304</double>
</response>
```
OANDA Currency Exchange Rates
OANDA Exchange Rates API

- RESTful API
- Access to OANDA’s daily exchange rate data
- Accessed via HTTP
- JSON, XML and CSV
Method 2: Java Classes
Develop Java Classes

- Web Service Wizards in JDeveloper
- Can be called like any normal PL/SQL procedure call
Using Jdev 10g create Java Client for Web Service (Currency Exchange Rate)

• Using Jdev 10g create Java Client for Web Service (Currency Exchange Rate)
• Create a new Application Workspace and Project
• New -> Business Tier -> Web Services -> Web Service Stub/Skeleton
• Enter the URL for the WSDL in the Wizard: http://www.xmethods.net/sd/2001/CurrencyExchangeService.wsdl
• Finish the Wizard
• Use CurrencyExchangeServiceStub class
public static void main(String[] args) {
    try {
        CurrencyExchangeServiceStub stub = new CurrencyExchangeServiceStub();
        // Add your own code here.
        System.out.println(stub.getRate("us", "uk"));
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}
Deploy Web Service Client as Stored Java Procedure

- Create DB Schema
- Grant required privileges
Java Classes – Deploy Web Service

Create Deployment Profile

Create a profile for deploying Java classes and Java stored procedures to the Oracle database.

1. Open the Application Navigator.
2. Right-click on the project and select ‘Create Deployment Profile’.
3. Choose ‘Java and Java Stored Procedures’.
4. Select the desired files and directories for deployment.
5. Click ‘OK’ to create the profile.

This profile allows you to compile the project and deploy it to the Oracle database, ensuring that all files are correctly configured for deployment.
Create Deployment Profile
Deploy Deployment Profile to the Database Schema
Java Classes – Deploy Web Service

Test the Web Service
Method 3: UTL_DBWS
UTL_DBWS

1. Provides Database Web Services
2. Provides a means of using a Dynamic Invocation Interface

<table>
<thead>
<tr>
<th>Common Procedure/Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>TO_QNAME</td>
</tr>
<tr>
<td>CREATE_SERVICE</td>
</tr>
<tr>
<td>CREATE_CALL</td>
</tr>
<tr>
<td>SET_TARGET_ENDPOINTER_ADDRESS</td>
</tr>
<tr>
<td>SET_PROPERTY</td>
</tr>
<tr>
<td>INVOKE</td>
</tr>
</tbody>
</table>
Sequence of steps

1. service_qname := utl_dbws.to_qname(namespace, name);
2. service := utl_dbws.create_service(service_qname);
3. call := utl_dbws.create_call(service);
4. utl_dbws.set_target_endpoint_address(call, endpoint);
5. utl_dbws.set_property( call, key, value);
6. response := utl_dbws.invoke(call, request);
UTL_DBWS Example

Get Sum of Two Numbers

```
CREATE OR REPLACE FUNCTION add_numbers(  
  p_int_1 IN NUMBER,
  p_int_2 IN NUMBER) RETURN NUMBER
AS
  l_service UTL_DBWS.service;
  l_call UTL_DBWS.call;
  l_wsd1_url VARCHAR2(32767);
  l_namespace VARCHAR2(32767);
  l_service_qname UTL_DBWS.qname;
  l_port_qname UTL_DBWS.qname;
  l_operation_qname UTL_DBWS.qname;
  l_xmltype_in SYS.XMLTYPE;
  l_xmltype_out SYS.XMLTYPE;
  l_return NUMBER;
BEGIN
  l_namespace := 'http://oracle-base.com/webservices/';
  l_service_qname := UTL_DBWS.to_qname(l_namespace, 'Calculator');
  l_port_qname := UTL_DBWS.to_qname(l_namespace, 'CalculatorPort');
  l_operation_qname := UTL_DBWS.to_qname(l_namespace, 'ws_add');
  l_service := UTL_DBWS.create_service (wsdl_document_location => URIFACTORY.getURI(l_wsd1_url), service_name => l_service_qname);
  l_call := UTL_DBWS.create_call (service_handle => l_service, port_name => l_port_qname, operation_name => l_operation_qname);
  <ws_add xmlns="" l_namespace || "">
  <int1> || p_int_1 || '</int1>
  <int2> || p_int_2 || '</int2>
  </ws_add>);
  l_xmltype_in := UTL_DBWS.invoke(call_Handle => l_call, request => l_xmltype_in);
  UTL_DBWS.release_service (service_handle => l_service);
  l_return := l_xmltype_out.extract('/return/text()').getNumberVal();
RETURN l_return;
END;
```
Get Sum of Two Numbers

```
SELECT add_numbers(1, 5) FROM dual;
```

```
ADD_NUMBERS(1,5)
------------------
   6
```
Expose PL/SQL as a Web Service
Expose PL/SQL as a Web Service

Sequence of steps in Jdeveloper
1. Create Application Workspace and Project
2. Create Database connection
3. Install PL/SQL package
4. Publish PL/SQL package
5. Deploy the web service
Expose Web Service Example – Step 1

Create Application Workspace and Project
Expose Web Service Example – Step 2,3,4

Create DB WS_LOCAL and Compile APP_SALES Package and Publish
Expose Web Service Example

Response

```xml
<?xml version='1.0' encoding='UTF-8'?>
    <return xsi:type="xsd:decimal">13667.06</return>
  </ns1:getQuoteResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
Limitations of PL/SQL Web Services
PL/SQL Web services cannot be created in below situations:

1. Overloaded Program Units
2. BFILE Type
3. OUT and IN-OUT parameters
4. PL/SQL Records
5. Stored procedures of the same name which are accessible in more than one schema
6. Ref cursors return types
7. SYS schema
8. PL/SQL nested tables
SOA Gateway
Oracle E-Business Suite Integrated SOA Gateway

1. Exposing and providing out-of-the-box Web services from Oracle E-Business Suite
2. Faster design-to-deploy integration flows
3. Transform public interfaces into standard Web services
4. Provides interoperability advantage for SOA-based Integration
5. Native service invocation framework to invoke all 3rd party external Web services
Oracle Integration Repository
Oracle Integration Repository

1. Ability to access an individual application through a business interface
2. Interfaces can be used from application-to-application (A2A), or from business-to-business (B2B)
3. Supports Interface Types
4. What API would be suitable for a specific business process?
Oracle Integration Repository

1. Like any other Oracle E-Business Suite application
2. User with sufficient privileges
3. Responsibility – Integrated SOA gateway
4. Browse integration interfaces:
   1. Product family
   2. Business Entity/Interface Type/Integration Standard
1. Introduction to Apps Associates
2. Web Services – Overview
3. Consume Web Services – Methods
4. OANDA
5. Expose Web Services
6. Limitations of PL/SQL Web Services
7. SOA Gateway
8. Oracle Integration Repository
Connect with Us

- Web: www.appsassociates.com
- Email: marketing@appsassociates.com
- YouTube: www.youtube.com/user/AppsAssociates
- LinkedIn: www.us.linkedin.com/company/apps-associates
- Twitter: @AppsAssociates
- Facebook: www.facebook.com/AppsAssociatesGlobal