

APEX 4.0 Web Services Evaluation Guide

Application Express 4.0 will offer a number of improved and new Web services features including support for RESTful style Web services both from the consumer and provider perspective. This guide will explain these features and give you step by step instructions on how to use them.

Note: The instructions below are written assuming that you are using the classic version of page editing and not the Tree view. Also, some exercises will not work properly with Internet Explorer. You should use a browser that is supported for the evaluation such as Firefox.

Consuming RESTful Style Web Services

REST (REpresentational State Transfer) is a simpler architecture for accessing Web services than traditional SOAP style Web services. Parameters to a RESTful style Web service are typically passed as part of the URI and the operation is usually described by the HTTP method used: GET, HEAD, POST, PUT, DELETE). Responses are typically in XML or JSON. Application Express 4.0 provides a declarative wizard based method for creating references to RESTful style Web services. Once a reference is created, there are facilities to test calling the Web service as well as wizards to create forms to call the Web service within Application Express and reports to report on the result.

Consuming a RESTful Style Web Service with an XML Response

In this exercise you create a reference to the Yahoo Maps RESTful Web service and then create a form and report to interact with the service. The response returned is XML. You will need to sign up for a Yahoo developer key [here](#).

1. Navigate to the Application Builder home page and click on an application or create a new one.
2. Click Shared Components.
3. Click Web Service References.
4. Click Create >.
5. Choose REST and click Next >.
6. Open <https://developer.yahoo.com/maps/rest/V1/> in a new tab or new browser session. You will continue to create the reference to this Web service by reviewing its documentation.
7. Return to Application Express and enter **Yahoo Map** in the Name field.
8. Enter **http://local.yahooapis.com/MapsService/V1/mapImage** in the URL field.
9. Click Next >.
10. Enter **appid** in the Name field and click Add Parameter.
11. Enter **location** in the Name field and click Next >.
12. Leave Output Format as XML.

13. Enter **/Result** in the XPath field.
14. Enter **url** in the Name field.
15. Enter **/text()** in the Path field.
16. Click Create.
17. Click Create Form and Report on Web Service.
18. Choose Yahoo Map from Web Service Reference list.
19. Choose doREST from the Operation list and click Next >.
20. Enter **Address** in the Form Region Title field.
21. Enter **Map** in the Report Region Title and click Next >.
22. Click Next >.
23. Click the checkbox next to url and click Finish.
24. Click the Run Page icon.
25. Enter your Yahoo App ID key in the AppId field.
26. Enter **747 Howard Street, San Francisco, CA** in the location field.
27. Click Submit.

You should get a report with a long URL in the url column. To display the map at the URL, do the following:

1. Click Edit Page in developer toolbar.
2. Click Report link next to Map region.
3. Click edit icon next to the url column.
4. Enter **** in the HTML Expression text area.
5. Click Apply Changes.
6. Click Run Page icon.

You should now see the map. It is also very easy to make changes to RESTful Web references, for example adding additional input parameters. The Yahoo maps API has optional parameters `image_height` and `image_width`. We can use those parameters to make our map larger.

1. Navigate to Shared Components.
2. Click Web Service References.
3. Click on Yahoo Map.

4. Click Add Input Parameter under REST Input Parameters.
5. Enter **image_height** in the Name text field and click Add Input Parameter.
6. Enter **image_width** in the Name text field and click Apply Changes.
7. Navigate to the page definition of the page created in the previous exercises.
8. Click on the Process Web Service Request in the Processes area of the Page Processing section.
9. Enter **1024** in the image_height value field.
10. REST input parameters can also be populated from the result of a function. Enter **return '1024';** in the image_width value field and choose **Function** from the Source select list and click Apply Changes.
11. Click Run Page icon.
12. Click Submit.

The map now appears 1024 by 1024.

Consuming a RESTful Style Web Service with a Text Response

In this exercise you create a Web service reference to Google's geo-coding API and then create a form on the inputs and a report on the response using the wizard. The response is returned comma separated. You will need to sign up for a Google developer key [here](#).

1. Navigate to Shared Components.
2. Click Web Service References.
3. Click Create >.
4. Choose REST and click Next >.
5. Open <http://code.google.com/apis/maps/documentation/geocoding/index.html> in a new tab or new browser session. You will continue to create the reference to this Web service by reviewing its documentation.
6. Return to Application Express and enter **Google Geocode** in the Name field.
7. Enter **http://maps.google.com/maps/geo** in the URL field.
8. Click Next >.
9. Enter **q** in the Name field and click Add Parameter.
10. Enter **key** in the Name field and click Add Parameter.
11. Enter **sensor** in the Name field and Click Add Parameter.
12. Enter **output** in the Name field and click Next >.

13. Choose Text for the Output Format.
14. Leave defaults for parameter delimiter and New Record Delimiter. They should be filled with a comma (,) and a new line (\n) respectively.
15. Enter **latitude** in the Name field.
16. Enter **3** in the Path field (replacing the 1 that will appear) and click Add Parameter. Paths for text responses are determined by position. So this indicates that latitude will be found in the third position of the response.
17. Enter **longitude** in the Name field.
18. Enter **4** (replacing the 2 that will appear) in the Path field and click Create.
19. Click Create Form on Web Service.
20. Choose Google Geocode from Web Service Reference list.
21. Choose doREST from the Operation list and click Next >.
22. Enter **Geocode** in the Region title field and click Next >.
23. Change the Item Label Q to **Address** and click Next >.
24. Click Finish.
25. Click the Run Page icon.
26. Enter **747 Howard Street, San Francisco, CA** in the Address field.
27. Enter your Google API Key in the Key field.
28. Enter **false** in the Sensor field.
29. Enter **csv** in the Output field.
30. Click Submit.

The Latitude and Longitude fields should be filled with values.

Improved Manual Web Reference Support

Manual Web references are created by visually inspecting the WSDL document as well as using a tool to determine the SOAP envelope for the Web service request. Prior to Application Express 4.0, the developer would have to manually create pages, items, regions and processes to interact with Manual Web references. Now you can simply run one either the Create Form or Create Form and Report on Web reference wizards.

Create a Web Service Reference Manually

To create a Web reference manually, you will copy code from the WSDL for a service called MovieInformation.

To create a manual Web reference:

1. On the Application home page, click **Shared Components**.
2. Under Logic, click **Web Service References**.
3. Click **Create**.
4. Choose **Manual** for the Web reference type and click **Next**.

The Create/Edit Web Service page appears.

5. In the Name field, enter **Movie Info**.
6. Locate the endpoint of the MovieInformation service:
 - a. Open the WSDL by going to:
<http://www.ignyte.com/webservices/ignyte.whatsshowing.webservice/moviefunctions.asmx?wsdl>
 - b. In the WSDL, find the location attribute of the soap:address element, which is a child of the port element. You can search for the following term within the code: soap:address location.
7. In the URL field on the Create/Edit Web Service page, enter the endpoint of the MovieInformation service you located.
http://www.ignyte.com/webservices/ignyte.whatsshowing.webservice/moviefunctions.asmx
8. Locate the SOAP action for the GetTheatersAndMovies operation:
 - a. If necessary, open the WSDL again. See Step 6a.
 - b. In the WSDL, find the soapAction attribute of the soap:operation element, which is a child of the operation element that has a name attribute of GetTheatersAndMovies. You can search for the following term within the code: soap:operation soapAction.
9. In the Action field on the Create/Edit Web Service page, enter the SOAP action you located.
http://www.ignyte.com/whatsshowing/GetTheatersAndMovies

10. In the SOAP Envelope field on the Create/Edit Web Reference page, enter the xml code representing the SOAP Request message:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tns="http://www.ignyte.com/whatsshowing"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <tns:GetTheatersAndMovies>
```

```
<tns:zipCode>#ZIP#</tns:zipCode>
<tns:radius>#RADIUS#</tns:radius>
</tns:GetTheatersAndMovies>
</soap:Body>
</soap:Envelope>
```

You can use a SOAP message generating tool, such as MindReef, to construct a valid SOAP Request for a given Web service.

11. In the Store Response in Collection field, enter **MOVIE_RESULTS**. This is where the response from the Web service will be stored.
12. Click **Create**.

Create a Form and Report

Next, you need to create a page that contains a form and report to use with your Manual Web Service Reference.

To create a form and report on a Web Service Reference:

1. On the Application Builder home page, click **Create Page**.
2. Choose Form and click **Next**.
3. Choose Form and Report on Web Service and click **Next**.
4. For Choose Service and Operation:
 - a. Web Service Reference - Select **Movie Info**.
 - b. Click **Next**.
5. For Page and Region Attributes:
 - a. Form Region Title - Change to **Theater Information**.
 - b. Accept the other defaults and click **Next**.
6. For Input Items:
 - a. For ZIP and RADIUS, accept the default, **Yes**, in the Create column.
 - b. For ZIP, change the Item Label default to ZIP Code.
 - c. Click **Next**.
7. For Web Service Results:
 - a. SOAP Style - Select **Document**.

- b. Message Format - Select **Literal**.

Note that these two attributes can be determined by manually inspecting the WSDL document for the service.

- c. Result Node Path - Enter:

/GetTheatersAndMoviesResponse/GetTheatersAndMoviesResult/Theater/Movies/Movie

- d. Message Namespace - Enter:

http://www.ignyte.com/whatsshowing

Note that you reviewed both the Result Node Path and Message Namespace when testing the service.

- e. Click **Next**.

8. For Report Parameters, enter Name, Rating, RunningTime, and ShowTimes, and click **Next**.

9. Accept the default for Tab Options and click **Next**.

10. Click **Create Form and Report**.

11. To test the application:

- a. Click **Run Page**. If prompted to log in, enter your workspace user name and password.
- b. In the ZIP Code and Radius fields, enter information and click **Submit**.

The results appear in the Search Results area.

APEX_WEB_SERVICE API

The APEX_WEB_SERVICE API allows you to integrate other systems with Application Express by allowing you to interact with Web services anywhere you can utilize PL/SQL in your application. The API contains procedures and functions to call both SOAP and RESTful style Web services.

In this exercise, you will invoke a Web service and display the response through the use of AJAX. You do this by first creating an Application Process that uses the APEX_WEB_SERVICE API to call the Web service. You will need to sign up for a Yahoo developer key [here](#).

Create Application Process

1. Click Shared Components.
2. Click Application Processes.
3. Click Create >.
4. Enter **callYahooVideo** in the Name field.

5. Choose On Demand as the process point.
6. Click Next >.
7. Enter the following in the Process Text text area replacing **<your dev key here>** with your Yahoo developer key:

```
declare
    l_clob clob;
    l_buffer          varchar2(32767);
    l_amount          number;
    l_offset          number;
begin
    l_clob := apex_web_service.make_rest_request(
        p_url =>
'http://us.music.yahooapis.com/video/v1/list/published/popular',
        p_http_method => 'GET',
        p_parm_name =>
apex_util.string_to_table('appid:format'),
        p_parm_value => apex_util.string_to_table('<your dev key
here>:xml'));

    l_amount := 32000;
    l_offset := 1;
    begin
        loop
            dbms_lob.read( l_clob, l_amount, l_offset, l_buffer );
            http.p(l_buffer);
            l_offset := l_offset + l_amount;
            l_amount := 32000;
        end loop;
    exception
        when no_data_found then
```

```
        null;  
    end;  
end;
```

8. Click Next >.
9. Click Create Process.

Create a Page and Components to Call Application Process

1. Navigate to the Application Builder homepage for this application.
2. Click Create Page >.
3. Choose Blank Page and click Next >.
4. Click Next >.
5. Enter **Yahoo Videos** in the Name field and click Next >.
6. Click Next >.
7. Click Finish.
8. Click Edit Page.
9. Click the create icon under regions.
10. Choose HTML for the region type and click Next >.
11. Choose HTML for the region container and click Next >.
12. Enter **Yahoo Videos** in the Title field and click Create.
13. Click the create icon under the buttons area.
14. Click Next>.
15. Click Next >.
16. Click the [Submit] quick pick.
17. Choose Redirect to URL without submitting page and click Next >.
18. Click Next >.
19. Choose Create for the region position and click Next >.
20. Choose URL from the Target select list.
21. Enter **javascript:callYahooVideo();** in the URL Target text area and click Create Button.
22. Click the create icon under Items area.

23. Choose Text Area from the item type list and click Next >.
24. Enter **RESPONSE** in the item name field and click Next >.
25. Choose label alignment above and click Next >.
26. Choose **Yes** from the Auto-Height select list and click Next >.
27. Click Create Item.

Add Javascript to Page Header that Calls Application Process

1. Click the edit icon in the Page area to edit page attributes.
2. In the Function and Global Variable Declaration text area in the JavaScript section, enter the following javascript code:

```
function callYahooVideo(){  
  
    var get = new htmldb_Get(null,  
$v('pFlowId'), 'APPLICATION_PROCESS=callYahooVideo', $v('pFlowStepId'));  
  
    gReturn = get.get();  
  
    get = null;  
  
    apex.jQuery('#RESPONSE').val(gReturn);  
  
}
```

3. Click Apply Changes.
4. Click Run.
5. Click Submit.

You should see the Response text area filled with an XML document of the popular videos from Yahoo Videos service.

Application Express as a RESTful Web Service Provider

In Application Express 4.0, you can expose report regions in your applications as RESTful Web services that are callable with a URL and return either JSON or XML. In order to expose a report region as a RESTful Web service the following steps must be performed:

1. The instance administrator must enable the REST provider feature in instance settings
2. A developer must enable the report to be callable via REST
3. The page that contains the report must not require authentication

In this exercise you will enable a report to be called via REST and then create a Web reference to it in the same application and use the Web Service reference testing interface to test it.

Create an Application with a SQL Report

1. Navigate to the Application Builder home page.
2. Click Create.
3. Choose Database and click Next >.
4. Click Next >.
5. Enter RESTing in the Name field and click Next >.
6. On Pages step of the Create Application Wizard:
 - a. Choose Report for the page type
 - b. Choose SQL Query as from the page source list
 - c. Enter **Employees** in the Page Name field
 - d. Enter the following query in the Query text area:

```
select *  
  from emp  
  where (job = upper(:P1_JOB) or :P1_JOB is null)  
        and (deptno = :P1_DEPTNO or :P1_DEPTNO is null)
```
 - e. Click Add Page
7. Click Create.
8. Click Create.

Enable Report for REST Access

1. Navigate the page definition of page 1 of the new application.
2. Click the edit icon in the page section to edit the page attributes.
3. Choose Page is Public from the Authentication list under the Security section.
4. Click Apply Changes.
5. Click the link for the EMP region to edit the region attributes.
6. Enter **employees** in the Static ID field under Region Settings
7. Choose Yes from the Enable RESTful Access list
8. Copy the URL that is displayed under the REST Access URL area *up to but not including the question mark (?)* into your buffer.

9. Click Apply Changes.

Create a RESTful Web Reference to the Report

1. Navigate to the Application Builder home page.
2. Click Shared Components.
3. Click Web Service References.
4. Click Create.
5. Choose REST and click Next >.
6. Enter **Employees** in the name field
7. Paste the URL copied in the buffer from step 8 in the previous exercise and click Next >.
8. Enter **app** in the Name field and click Add Parameter.
9. Enter **page** in the Name field and click Add Parameter.
10. Enter **reportid** in the Name field and click Add Parameter.
11. Enter **parmvalues** in the Name field and click Add Parameter.
12. Enter **lang** in the Name field and click Add Parameter.
13. Enter **output** in the Name field and click Next >.
14. Enter **/ROWSET/ROW** in the XPath field.
15. Enter **EMPNO** in the Name field and tab into Path field then click Add Parameter.
16. Enter **ENAME** in the field and tab into Path field then click Create.

Test the Web Reference

1. Click View Web Service References.
2. Change the View select list to Report and click Go.
3. Click the Test icon.
4. Enter the following Input Parameters:
 - a. Enter your Application Express application ID in the app field
 - b. Enter **1** in the page field
 - c. Enter **employees** in the reported field

- d. Enter **xml** in the output field
5. Click Test and observe results.
6. Change output to **json** and click Test.
The results are now returned in JSON.
7. Enter **salesman, 30** in the parmvalues field and click Test.
The results returned only include salesman in department 30.
8. Enter **de** in the lang field and click Test.
The results now show the dates in the appropriate language format.

Conclusion

Application Express 4.0 will make vast improvement in enabling developers to integrate their applications with other systems and applications. Full support for consuming RESTful Web services is provided as well as for the first time, exposing report regions in applications in a RESTful way. The APEX_WEB_SERVICE API will allow developers to interact with Web services virtually anywhere that PL/SQL can be used in Application Express. Finally, developers that have used the Manual Web references feature will find it much easier to create forms and reports based on those references.