ADF Code Corner
74. Hands-on: How to use the ADF URL Data Control for parametrized queries

Abstract:

This hands-on tutorial shows how to use the ADF URL Data Control to access the Oracle JDeveloper twitter account to display recent messages in an ADF Faces table. It also shows how to change the ADF URL Data Control configuration to show messages of other twitter users.

Duration: 60 Minutes

Author: Frank Nimphius, Oracle Corporation
twitter.com/fnimphiu
21-FEB-2011
How to use the ADF URL Data Control for parametrized queries

Oracle ADF Code Corner is a loose blog-style series of how-to documents that provide solutions to real world coding problems.

Disclaimer: All samples are provided as is with no guarantee for future upgrades or error correction. No support can be given through Oracle customer support.

Please post questions or report problems related to the samples in this series on the OTN forum for Oracle JDeveloper: http://forums.oracle.com/forums/forum.jspa?forumID=83

Introduction

This Oracle JDeveloper hands-on steps the user through the basic steps required to work with the Oracle ADF URL Data Control. It also explains the data control behavior beyond of this exercise.

At runtime, a table is displayed that contains most recent Oracle JDeveloper tweets.

Note: This hands-on focuses on the functional part and not how-to create a good looking page. All the steps to follow are the minimum it takes to build a little sample that accesses Twitter from the URL Data Control and show the query results.

Note: If you experience problems to connect to the Twitter website during the hands-on, try setting or unseting the Oracle JDeveloper proxy configuration and re-start JDeveloper. The browser proxy settings can be changed using the Tools | Preferences | Web Browser and Proxy.
To complete this Oracle JDeveloper hands-on requires a recent version of Oracle JDeveloper 11g and the twitter.xsd schema file.

Building the URL Data Control Model

1. Open Oracle JDeveloper and start with a new application. You do this by selecting "New Application" in the Oracle JDeveloper Application Navigator or choosing File | New | Applications from the menu.

2. Select the "Generic Application" option, which allows us to define and configure projects as we need them.

3. Provide "TwitterUrlDC" as the Application name and "adf.sample.twitter" as the package prefix.

4. Press Next.
5. Enter "TwitterUserTweetUrlDC" as the project name. Note that the project name also determines the project folder the Java classes and metadata files are saved in.

4. The URL Data Control is part of the ADF Web Service binding. Select "Web Services" in the "Available" list and move it to the "Selected" list. The "Java" technology scope is automatically added.

5. Press "Finish" to complete the project creation. The new project is configured with the required Java EE libraries for Web Services and Java development.
6. With the project selected, choose "New" from the right mouse button context menu.

7. Select the Business Tier | Web Services category and choose the "URL Service Data Control" entry.

8. Name the Data Control "TwitterUrlDataControl" and press the "Next" button.
9. Provide a name for the endpoint configuration and (optional) authentication. This information is stored in a connection registry and is identified by the name you provide. As the connection name, enter "TwitterQuery".

10. As the "URL Endpoint", type http://twitter.com/statuses/user_timeline.xml (without the quotes). Twitter provides this API to access public tweets.

11. Add "?screen_name=jdeveloper" (without the quote) into the "Source" field. This specifies the query parameters for the tweet to lookup.
12. Press "Next"

Note: If you can't connect to the URL, check your Oracle JDeveloper proxy setting in Tools | Preferences | Web Browser and Proxy. You may also test the Twitter API by typing

http://twitter.com/statuses/user_timeline.xml?screen_name=jdeveloper

into the browser URL field

13. Choose XML as the Data Format of the expected result set. As a result of selecting XML as the Data Format, the editor layout changes as shown in the next image.

14. In the XSD URL field, type the location you copied the twitter.xsd file to. Use the following entry, assuming twitter.xsd is located in the c:\temp directory

File:///c:/temp/twitter.xsd

The XS file is only used at design time for the Data Control to define the data structure. At runtime it is no longer needed so that this file does not need to be saved or deployed with the application.

15. Press "Next"
What you should know: Twitter does not provide XML schema for their metadata service. For this how-to, the XSD is generated using "Trang" (http://www.thaiopensource.com/relaxng/trang.html). You may find other more up to date tools to generate XSD from XML content.

16. Click "Test URL Connection" to verify your configuration
Creating the ADF Faces View Layer

With the URL Data Control in place, it is time to build the user interface to display the Oracle JDeveloper tweets in a table.
1. Choose File | New | General | Projects from the Oracle JDeveloper menu to create a new "Generic Project"

2. Name the new project "ViewController" and select the "ADF Faces" and "ADF Page Flow" technologies. Move the selected technologies to the "Selected". The "JSF, JSP and Servlet and Java entries are automatically added.
3. Define a default package name, like "adf.sample.twitter"

4. Press "Finish" to create the new project
How to use the ADF URL Data Control for parametrized queries

5. Select the ViewController project and choose "New" from the right mouse context menu

6. Select Web Tier | JSF from the New Gallery and choose "JSF Page"

7. Press "Ok" to open the Create JSF dialog
8. Provide "TwitterViewer.jspx" as the page name and choose a one column Quick Layout. This will help you layout the page by automatically stretching the table that you will put on this page later in the tutorial.

9. Expand the DataControl panel and expand the "TwitterUrlDataControl" entry. If you don't see the Data Control entry, press the blue refresh icon.

10. Drag the "status" collection from the Data Controls panel and drop it on the JSF page.
11. In the opened context menu, choose Table | ADF Read-only Table
12. In the table editor, select all attributes except "created_at", "id","in_reply_to_screen_name", "retweeted" and "text".

**Note:** the UI of the dialog has changed between Oracle JDeveloper 11.1.1.3 and Oracle JDeveloper 11.1.1.4. So if the dialog you see differs from the screen shot, there is no need to worry

13. Delete the selected attributes

14. Reorder the attributes as shown in the image below
15. Press "Ok"

16. Select the table in the visual editor or the Structure Window and open the Property Inspector (ctrl+shift+I)

17. Search for the "ColumnStretching" attribute and select the "text" column as the column to stretch. In the screenshot shown above, this was the 3rd column for us when creating the hands-on. This may be a different column index for you (you may have to try it out)
18. Click onto the JSF page in the visual editor and choose "Run" from the context menu.

The browser should show the Twitter table as shown below.

Note: If you wanted to refresh the table view to see recent Tweets posted by the account, drag and drop the loadData() method to the JSF page and add it as a command button. Pressing the button then re-executes the Data Control query. However, a similar use case is handled in a later section in this hands-on.
Changing the URL DC endpoint query

In this section, you will change the tweets displayed in the table to be read from the "adfcodecorner" Twitter account, or you may try your own Twitter account instead.

1. Select the TwitterViewer.jspx entry in the Oracle Application Navigator
2. Expand the Data Controls panel and select the "TwitterUrlDataControl" entry.
3. Choose "Edit Definition" from the context menu

4. Change the "Source" field to read

?screen_name=adfcodecorner

5. In the Application Navigator, select the DataControls.dcx entry and open the Structure Window
6. Select the "TwitterViewer.jspx" page entry and choose "Run" from the context menu to see the changed data.
Changing the Twitter account dynamically

What we have not covered in this tutorial so far is the ability to specify variables in the query string you add as the "Source" information when configuring the URL Data Control. For example, instead of "hard coding" the twitter account to "jdeveloper" shown below

?screen_name=jdeveloper

you could use a variable ##screenName## instead

?screen_name=##screenName##

Variables in the ADF URL DataControl are in the form of ##variable_name##. You could have defined a variable when building the ADF URL Data Control.

**Note:** Oracle JDeveloper attempts to verify the URL, which in the case of Twitter does not work well because Twitter does not accept requests to accounts that don't exist or non-public accounts. The empty variable string thus is bounced as unauthorized by Twitter. However, in this tutorial you will learn how to add variables after URL Data Control creation

1. In the Application Navigator, select the DataControls.dcx file
2. Expand the Data Controls panel and select the "TwitterUrlDataControl" entry. Choose "Edit Definition" from the context menu
3. In the opened dialog, change the "Source" string and replace the screen_name value from a string to the ##screenname## variable.

4. Finish the dialog
5. With the DataControls.dcx file selected in the JDeveloper Application Navigator, select the "TwitterUrlDataControl" entry in the Structure Window. Choose "Update" from the context menu to refresh the data control definition.

6. Save the Data Control project

7. Expand the Data Controls panel, which to this time shows the "loadData()" entry as it is not yet aware of the Data Control signature change.

8. Press the refresh icon on the Data Control panel
9. The "loadData" method should change after the refresh showing a String argument. If this is not the case ensure you saved the URL Data Control project after adding the variable to the "Source" configuration.

**Note**: You may have to refresh the Data Control panel more than once. Sometime JDeveloper doesn't seem to recognize the change at the first refresh.

**Note**: the previously TwitterViewer page no longer works after this change because it does not provide an input argument for the loadData() method. You can fix the binding later if you want. For the learning experience in this hands-on you are going to build a new JSF page for the dynamic Twitter access.
10. Select the View Controller project and choose "New" from the right mouse context menu.

11. In the New Gallery, select Web Tier | JSF | JSF Page to create a new ADF Faces page.
12. Name the page "DynamicTwitterViewer.jspx" and choose the one column QuickStart layout

13. Open the "Layout" section of the ADF Faces component list in the JDeveloper component palette (ctrl+shift+P)

14. Select the Panel Splitter component and drag it onto the JSF page
15. Select the Panel Splitter component in the visual editor or the Oracle JDeveloper Structure Window and open the Property Inspector

16. In the Property Inspector search for the “Orientation” property

17. Set the orientation to “vertical”

18. Drag the loadData(String) method to the upper panel of the JSF page
19. Choose ADF Parameter Form from the context menu

20. In the Data Controls panel, select the "status" collection and drag it to the page

21. In the opened context menu, choose the Table | Read Only Table option

22. In the table configuration dialog, select all columns, except "created_at", "id", "in_reply_to_screen_name", "retweeted" and "text"

23. Delete these columns by pressing the delete icon
24. Rearrange the column order and make the "text" column becoming second after the "created_at" column

25. OK the dialog

26. Select the "loadData" button and open the Property Inspector.
27. Find the "PartialSubmit" property and set it to "true". This ensures the JSF page is not fully reloaded when pressing it.

28. Select the table component and find its "PartialTriggers" property in the Property Inspector.  

**Note:** Make sure you selected the table and not a single column. Best is to verify this in the Oracle JDeveloper Structure Window.

29. From the context menu of the "PartialTriggers" property, choose the "Edit" link.

30. In the Edit Property dialog, find the "loadData" command button and move it to the "Selected" list.  

**Note:** The load data button is not shown with its label in the "Edit Property" dialog, but its Id.
To avoid errors, you don't want to query Twitter without an account name. In the visual JSF page editor, click the "Bindings" tab at the bottom of it.

Press the alt+Home button (or alt+Pos on non US keyboards) to find the PageDef file in the Application Navigator.
33. Select the PageDef file in the Application Navigator and open the Structure Window to see the binding details.

34. Expand the executables | variables node and select the loadData_screenname entry.

35. Open the Property Inspector and set the DefaultValue property to "jdeveloper" (without the quotes).

36. Save the project.

37. Click onto the page in the visual editor or the Application Navigator and choose the "Run" option from the right mouse context menu.

38. The screen renders with the table showing tweets of the "jdeveloper" account. Add "adfcodecorner" or your Twitter account name into the input field and press the loadData button.
How to use the ADF URL Data Control for parametrized queries

Download

A zip file containing additional instructions and a completed solution can be downloaded as sample #74 from the ADF Code Corner website at: