Xap XAPStruts

Introduction

This page describes how to integrate Apache XAP with Struts2. I use the Person Manager struts sample to illustrate how to upgrade an html application to use XAP. Overall the integration is quite simple. I was able to repurpose all of the Java code (Data Model, ActionHandlers) from the sample. All that needed to be done was to re-write the *.ftl files to output a combination of *xModify* and *XAL* instead of html.

I have uploaded a zip file of the converted Person Manager application to my people directory. Download the XAP Person Manager Code To run the code unzip the archive ontop of an exploded version of the struts2-showcase-2.0.6.war. You can then run the sample by going to http://<server-address>:<port> /struts2-showcase-2.0.6/xap/ajax-index.html

The reason why integration with web frameworks is so easy, is that the XAP client retrieves the markup used to describe the application from the server and then renders it entirely on the client. This is the opposite of other Ajax toolkits where:

*You either are responsible for moving data from the server to the client using XHR and updating the UI with JavaScript code or

*The markup is down convert into HTML and JavaScript on the server as in some JSF implementation.

By rendering the markup on the client, the developer can reuse alot of existing code they already have in their application. Most of the time the Model and Controller code is reused. This leave developers to only change the view layer of their application. This is exactly what I did in the conversion of the Person Manager Sample.

Another reason that integration is fairly seamless with web frameworks is that XAP communicates with the server in the same manner that a HTML application does, which is through the use of HTTP get and posts.

Sample Files

In the example there are the following files

- *struts.xml* - Added an include to the struts-xap.xml file
- *struts-xap.xml* - Created action handlers for my XAP application. I used the existing ActionHandler and beans from the person manager example.
- *decorators.xml* - Add excludes for the /xal-actions/* and *.xal files, otherwise they would get templated.
- *xap/ajax-index.html* - This file includes the code to bootstrap XAP and is the file you open to run the application.
- *xap/ajax-index.xal* - Initial UI layout and Managed Client Object definition.

The "strutsMco" object is defined in the ajax-index.xal file.
<mco xmlns="http://openxal.org/core/mco" id="strutsMco" class="DisplayUIDocument"/>
Managed Client Objects or "mcos" are XAP way of complete seperating the business logic from the UI definition. This is important when building enterprise applications because, if makes the application much easier to maintain.

- *xap/xal-actions/listPeople.ftl* - Contains the xModify and XAL markup used to a display table of person objects. A xModify **replace-children** statement is used to update the **contentPane** with a table that display all of the "Person" objects.

- *xap/xal-actions/newPeople.ftl* - Contains the xModify and XAL markup used to cisplay a form to create a new person. A xModify **replace-children** statement is used to update the **contentPane** with UI used to create .
- *xap/src-js/DisplayUIDocument.js* - Contains the code for the Managed Client Object. There are two functions contained in the file.

** *displayAlert* - Show what is in XAP's UI Document. The UI document is what you make changes to in order to update the screen.

** *newPersonAction* - Called from the new person form.

Using the request service, ui information is sent to the struts action. Once sent, the struts action is executed in the same way a HTML for is handled.

- *All other documents* are used in the XAP runtime.