XMLFormJXFormHibernateAndFlowscript

History

20030501: Created.

20031111: Updated for latest version of Cocoon (2.1.3 CVS). This should work for the released versions 2.1. Important: the XMLForm sample is not updated. Only the JXForm is. The map xmltest is renamed into xmltest_deprecated. When you start adding persistence to your xmlform project (which is a serious step), consider upgrading to jxforms first.

Introduction:

This is a rewrite of the flowscript prefs sample (http://localhost:8080/cocoon/samples/flow/prefs/) It has already been adapted to XMLForm by someone else (but I cannot find it anymore in the current samples).

This howto adds persistence by using Hibernate.

And I ported it to JXForm.

If you are new to Hibernate and/or run into trouble with the example below, try the less complicated example at the Wiki page: UsingHibernateToMakeYourJavaBeansPersistent. This page also explains howto include the Hibernate jars in Cocoon.

All the files mentioned below are in the attachment.

Steps:

- Make sure the Hibernate 2.0 jars are in your WEB-INF/lib.
- Create the table "user" in your database. See mysql.sql for a mysql script.
- To WEB-INF/classes: add the file hibernate.properties. This file should contain the login credentials for your database
- To WEB-INF/classes/htest: add the file User.hbm.xml. This file contains the Hibernate mapping between java class en database table
- To WEB-INF/classes/htest: add the five compiled classes: HibernateFactory.class, PersistenceFactory.class, PersistenceFactory$1.class, UserRegistry.class, User.class
- Add to the file WEB-INF/cocoon.xconf the line below at the first level (as child of <cocoon>): <component role="htest.PersistenceFactory" class="htest.HibernateFactory"/>
- Add the content dirs "xmltest" and "jxtest" to your cocoon dir (at the same level as samples). These maps contains xml content, stylesheets and the flow.js flowscript. jxtest is de JXForm implementation and xmltest is de XMLForm implementation. (note: xmltest is no longer supported).
- Restart tomcat and try to login: http://localhost:8080/cocoon/jxtest/ (don't skip that trailing slash.)

About the implementation:

The main parts are:

- Some HTML files, XML files, and stylesheets. Straightforward.
- A flowscript .js file. This file uses the flowscript getComponent to lookup a Hibernate factory. I dived into Avalon and Excalibur because I did not want the Hibernate database code in a static java class (see the design of the original prefs example). The Hibernate factory creates a UserRegistry. The UserRegistry creates, purges, and tries to find instances of the User class in its register (in this implementation: in the database).
- The java classes
  - The interface PersistenceFactory makes the method CreateUserRegistry public (and possibly many other CreateXXXRegistry methods).
  - The class HibernateFactory implements the PersistenceFactory interface. It maintains the connection with the database and produces registry classes. (In Hibernate speak: The HibernateFactory manages the Hibernate SessionFactory)
  - The class UserRegistry. This is one of the classes produced by HibernateFactory. The UserRegistry creates, purges, and tries to find instances of the User class in its register (in this implementation: in the database). In Hibernate speak: the UserRegistry manages the Hibernate Sesion)
  - The User class. This is the bean from the original sample.

Attachment: Flow_Hibernate_Xmlform.zip