

CformsHibernateAndFlow

Important

- The code attached to this page is not working (as at 14 Jun 2004). Help is needed!
- Necessary amendments to code (tested at 30 sep 2004 ...)
- `binding_example.js` line 64 change `var bean = new hib.Form2Bean();` to `var bean = new Packages.hib.Form2Bean();`
- `form2_model.xml` line 100 change `<fd:enum>org.apache.cocoon.forms.samples.Sex</fd:enum>` to `<fd:enum>hib.Sex</fd:enum>`
- `form2_model.xml` line 103 change `<fd:selection-list type="enum" class="org.apache.cocoon.forms.samples.Sex"/>` to `<fd:selection-list type="enum" class="hib.Sex"/>`
- NB for upcoming hibernate release 3.x - in the 2 Factory classes and `hibernate.properties`, package `net.sf.hibernate` should be replaced by `org.hibernate`
- With the above changes I have the attached code working - except add contact - with cocoon 2.1.5 and hibernate 3.0alpha and mysql 4.0.13-max-nt - necessary jars to copy to WEB-INF/lib from hibernate were `hibernate3.jar`, `dom4j-1.4.jar`, `cglib-full-2.0.2.jar` and `jta.jar`. Also using jdbc driver `mysql-connector-java-3.0.11-stable-bin.jar`
- This page [CformsHibernateAndFlow](#) was developed from the original at [WoodyHibernateAndFlow](#) by Hugo Burn. Modifications have been made to adapt it to a subset of the CForms sample, shipped with Cocoon 2.1.5, and available in the `samples/block/forms` section.

Setup

1. Read [UsingHibernateToMakeYourJavaBeansPersistent](#) to perform the integration of Hibernate with Cocoon. Run the test sample to see that Hibernate is working properly.
2. Copy the contents of the `hib` path (in the zip file, attached to this page), into the `WEB-INF/classes/hib` folder under your Cocoon directory.
3. Copy the contents of `form2bean2hib` path (in the zip file, attached to this page), into a `form2bean2hib` folder under your Cocoon directory
4. Add the following line into your `cocoon.xconf` (this adds the Avalon component that creates the Hibernate session):

```
<component role="hib.PersistenceFactory" class="hib.HibernateFactory"/>
```

1. Create the table in your mysql database with the `mysql.sql` script (in the zip file, attached to this page).
2. Copy the `hibernate.properties` file (in the zip file, attached to this page) into the `WEB-INF/classes` folder under Cocoon. Edit the file to specify the appropriate login credentials for your database. **Note:** Hibernate has support for many different databases. Change the SQL dialect in `hibernate.properties` and edit the above script if you don't use mysql but something else.
3. Restart Tomcat. Near the end of the debug info in the console output (`stdout.log` in the Tomcat `logs` directory), there should be a line reading `Hibernate initialize called`. If you don't see this line, one of the points above went wrong.
4. Try `http://.../form2bean2hib/` (where `"..."` is the path to your application under Cocoon).

Explanation

- The CForms Java beans `Contact`, `Sex`, and `Form2Bean` are **not** modified.
- The sitemap of the CForms example is **not** modified.
- Two classes are added - the interface of an Avalon component `PersistenceFactory` and its implementation `HibernateFactory`). This component creates the Hibernate session.
- The flowscript `binding_example.js` is changed. The function `form2bean` tries to load the `Form2Bean` object from disk with a hard-coded email address as the key. If it fails, it creates a new instance and saves it to disk.
- The real beef is in the file `WEB-INF/classes/hib/Form2Bean.hbm.xml`. This configuration file tells Hibernate how to do the real work: hide all details about the one-to-many relationship (contacts that are added to the bean) for the developer. When you try the example and look into the database, you can see that contacts are added with an unique key that is generated by Hibernate (the `id` field; don't edit it, as the Woody instructions tell you!) and a foreign key (the email address). In a real-world application, you almost always will need the `lazy` and `inverse` attributes (see Hibernate docs for explanation). In this case, the aim was to stay as close as possible to the original classes of the CForms sample.

[DerekH](#)

Attachment: [form2bean2hib.zip](#)