Extra Lab 2: Mapping a Simple Class

In this lab, you will use the JBoss Developer Studio to map a simple entity to the database.

Objectives:

- To use JBoss Developer Studio to map a class

Part 1: Getting Started

Steps:

_1. If it's not already running, start JBoss Developer Studio using the Windows Start menu and open the same workspace as before.

_2. In this lab, you will add to the ExtraLab1 project you created earlier, but first, to save typing, copy the MusicItem class from the Eclipse project:

   a. In JBoss Developer Studio, right-click on the ExtraLab1 project and choose *New - Package* and create a package named **com.javatunes.persist**.

   b. Back in Eclipse, in the Navigator or Package Explorer, find the MusicItem.java file.

      Select it to highlight it, then press Ctrl+C.

   c. Back in the JBoss Developer Studio, right-click on the *com.javatunes.persist* package and choose *Paste* to copy the file.

_3. In the Developer Studio, right-click on MusicItem.java and choose *New - Other - Hibernate - Hibernate XML Mapping File (hbm.xml)* and press Next to start the wizard.

   a. On the first wizard page, note that the MusicItem is selected, then press Next.

   b. On the next page, note the proposed name of the mapping file, then press Next.

   c. On the next page, briefly note the proposed mapping content, then press Finish. The Studio creates the mapping file.

_4. The Studio gives us a good starting point, but we will need to tweak the generated mapping, so double-click on it in the Package Explorer to open it into an editor. Then update it:

   a. In the editor's left-hand pane, expand the tree until you see the *com.javatunes.persist.MusicItem* entry, then click on it.

      Since the database table name doesn't match the entity class name, we need to adjust the mapping. In the *Table* entry field, change the value to **ITEM**.

   b. In the editor's left-hand pane, expand the tree until you see the *id* field under the *com.javatunes.persist.MusicItem* class, the expand the *id* entry to see the *ID* entry (all uppercase) that represents the property's column mapping.

      Since the property name doesn't match the database column name, we need to adjust the mapping. In the *Name* entry field, enter **item_id**.

   c. In the tree, click on the *assigned* entry. We need to change this to **identity** so that the mapping matches the database table's generation strategy.

      Press Ctrl+S to save your work so far.

Copyright © Descriptor Systems 2013. All Rights Reserved. Materials may not be reproduced in any form without prior written consent of Joel Barnum.

d. Press the *Source* tab at the bottom of the editor and examine the mapping file's XML and ensure you understand it.

_5. Next, update the hibernate.cfg.xml file to reference the new MusicItem mapping file:

a. In the Studio, double-click on hibernate.cfg.xml to open it into the editor, then click the *Session Factory* tab at the bottom of the editor.

b. In the *Mappings* section, click the Add button, then press the bottom Browse button (for *Resource*), and browse to *ExtraLab1* - *src* - *com* - *javatunes* - *persist* and select MusicItem.hbm.xml, then press OK followed by Finish.

c. Click the *Source* tab at the bottom of the editor and ensure you understand what you just did.

Save and close the file.

_6. In the Studio, open the TestHibernate.java class in the *test* package, and update it so it queries the MusicItem with ID of *1L* and displays a property or two. You can copy this code from the Eclipse version of the lab.

_7. Run TestHibernate.java as a Java Application. It should work the same as it did in Eclipse.

Copyright © Descriptor Systems 2013. All Rights Reserved. Materials may not be reproduced in any form without prior written consent of Joel Barnum.