Seam Tools Tutorial

Version: 4.2.0.Final-SNAPSHOT

1. Create a Seam Application 1
1.1. Start Development Database 1
1.2. Create and deploy Seam Web Project 3
1.3. Start JBoss Application Server 19
1.4. Workshop Project Code Overview 25
2. Seam Action Development
2.1. Create a New Seam Action 27
2.2. Test Seam Action
2.3. Modify Seam Action User Interface 31
3. Declarative Security
3.1. Edit Login Authentication Logic 35
3.2. Secure Seam Page Component 35
4. Browsing Workshop Database
4.1. Database Connectivity Setup 39
4.2. Browse Workshop Database 40
5. Database Programming
5.1. Reverse Engineer CRUD from a Running Database
5.2. Use Hibernate Tools to Query Data via JPA 47
5.3. Use Hibernate Tools to visualize the Data Model 54
6. Rich Components
6.1. Add a Richfaces component to the CRUD Application 57

Create a Seam Application

In this section you will learn how to create a Seam project in JBoss Developer Studio, how to start the server and what structure your project has after it is created.

1.1. Start Development Database

Before opening the JBoss Developer studio you need to download and start the *Workshop Database* [http://docs.jboss.org/tools/resources/GSG_database.zip].

To start the database just run ./runDBServer.sh or runDBServer.bat from the database directory.

The end result should be a console window that looks like:

Terminal - matthew@matthew-desktop: /redhat/p
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>T</u> erminal <u>G</u> o <u>H</u> elp
<pre>matthew@matthew-desktop:/redhat/projects/database\$</pre>
<pre>matthew@matthew-desktop:/redhat/projects/database\$</pre>
Starting database engine
<pre>[Server@19b49e6]: [Thread[main,5,main]]: checkRunni</pre>
<pre>[Server@19b49e6]: [Thread[main,5,main]]: checkRunni</pre>
[Server@19b49e6]: Startup sequence initiated from n
[Server@19b49e6]: Loaded properties from [/redhat/p
ver.properties]
[Server@19b49e6]: Initiating startup sequence
[Server@19b49e6]: Server socket opened successfully
<pre>[Server@19b49e6]: Database [index=0, id=0, db=file:</pre>
ssfully in 503 ms.
[Server@19b49e6]: Startup sequence completed in 539
[Server@19b49e6]: 2011-01-14 08:36:09.262 HSQLDB s€
[Server@19b49e6]: To close normally, connect and e>
<pre>[Server@19b49e6]: From command line, use [Ctrl]+[C]</pre>

Figure 1.1. Starting the Database



1.2. Create and deploy Seam Web Project

Minimize the terminal window and run JBoss Developer Studio from Applications Menu or from the desktop icon.

First you will see the Workspace Launcher. Change the default workspace location if it's needed. Click the **OK** button.

@	Workspace Launcher				
Select a wo	Select a workspace				
JBoss Developer Studio stores your projects in a folder called a workspace Choose a workspace folder to use for this session.					
<u>W</u> orkspace:	/redhat/workspaces/workspace-jbds4/				
Use this a	is the default and do not ask again Cancel				

Figure 1.2. Workspace Launcher Dialog

After startup, you see the welcome page. Select **Create New...** icon and then press on **Create Seam Project** link.

The New Seam Project wizard is started. You need to enter a name (e.g., "workshop") and a location for your new project. The wizard has an option for selecting the actual Server (and not just WTP runtime) that will be used for the project. This allows the wizard to correctly identify where the destination folder for the required datasource and driver libraries.

 \sim

🛞 New Seam Project	
Seam Web Project	
Create standalone Seam Web Project	
Project na <u>m</u> e: workshop	
Project location	
✓ Use <u>d</u> efault location	
Location: //redhat/workspaces/workspace-jbds4/workshop	
Target r <u>u</u> ntime	
jboss-eap Runtime	Nev
Dynamic web module <u>v</u> ersion	
2.5	
Target Server	
jboss-eap	
<u>C</u> onfiguration	
Dynamic Web Project with Seam 2.2	 \$
Configures a Dynamic Web application to use Seam v2.2	
Figure 1.3. New Seam Project Wizard	

All settings are already specified here, you can just modify the Configuration. Click on the **Modify...** button to configure your custom facet options:

@	Project Facets		
Project Facets			
Select the facets that should be enabled for this proj	ect.		
<u>Configuration</u> : Dynamic Web Project with Seam 2.2			
Project Facet	Version		
Axis2 Web Services			
CDI (Contexts and Dependency Injection)	1.0		
CXF 2.x Web Services	1.0		
🗹 🚋 Dynamic Web Module	2.5		
🗹 🛃 Java	1.5		
🗌 🗎 JavaScript	1.0		
🗹 📄 JavaServer Faces	1.2		
JAX-RS (REST Web Services)	1.1		
JBoss Portlets			
🗌 🏄 JBoss Web Services Core	3.0		
□ ↔ JPA	2.0		
🗹 🗎 Seam	2.2		
🗌 🗎 WebDoclet (XDoclet)	1.2.3		
(?)			
Figure 1.4. Project Facets Specifying			

On the whole the dialog allows to select the "features" you want to use in your project. JBoss Developer Studio will then setup the appropriate tooling for your project. Since JBoss Seam integrates all popular Java EE frameworks, you can select any combination of technologies from the list. Here, for the default configuration, Dynamic Web Module, Java, JavaServer Faces (JSF), and Seam Facet are already selected for a typical database-driven web application. The default project facets should suffice.

In the Project Facets form you can also bring up server runtimes panel by clicking Runtimes tab on the right corner. This panel shows available server runtimes.

	Project Facets		
Project Facets			
Select the facets that should be enabled for this proj	ect.		
. ,			
Configuration: Dynamic Web Project with Seam 2.2			
Project Facet	Version		
🕨 🔲 🗎 Axis2 Web Services			
CDI (Contexts and Dependency Injection)	1.0		
CXF 2.x Web Services	1.0		
🗹 🚋 Dynamic Web Module	2.5		
🗹 🔬 Java	1.5		
🗌 🗎 JavaScript	1.0		
🗹 🗎 JavaServer Faces	1.2		
JAX-RS (REST Web Services)	1.1		
Image: Boss Portlets			
🗌 🏄 JBoss Web Services Core	3.0		
□ � JPA	2.0		
🗹 📄 Seam	2.2		
🗌 🗎 WebDoclet (XDoclet)	1.2.3		
(?)			
Figure 1.5. Runtimes Selecting			

Click the **OK** and then the **Next** button to proceed to the next step.

A dynamic web application contains both web pages and Java code. The next wizard will ask you where you want to store Java files.

New Seam Project					
Java					
Configure project f	Configure project for building a Java application.				
<u>S</u> ource folders on b	uild path:				
[₽] src					
<u>D</u> efault output fold	er:				
build/classes					
?	< <u>B</u> ack	<u>N</u> ext >	Car	ncel	

Figure 1.6. Java Build Path

Following page provides you Web Module Settings .You can just leave the default values or choose another folder.

e	New Seam Project			
Web Module				
Configure web mo	Configure web module settings.			
Context <u>r</u> oot:	workshop			
Content <u>d</u> irectory:	WebContent			
☑ <u>G</u> enerate web.x	ml deployment descriptor			
?	< <u>B</u> ack <u>N</u> ext > Cancel			

Figure 1.7. Web Module Settings

On the next form, you will be able to select where those library JARs come from. The easiest is just to select the JARs provided by the JBoss AS runtime associated with this project. That is why it is important to chose the right JBoss AS 4.2 runtime in the project setup window.

• Select *Library Provided by Target Runtime* as Type of JSF Implementation Library. We will use the JSF implementation that comes with JBoss server.

• Click the Next button

9	New Seam Project				
JSF Capabilities	JSF Capabilities				
Add JSF capabilities to t	Add JSF capabilities to this Web Project				
JSF Implementation Lib	rary				
Type: Library Provided	Type: Library Provided by Target Runtime				
-	The targeted runtime is able to provide the library required by this facet. S this option will configure the project to use that library.				
JSF <u>Configuration File</u> :	/WEB-INF/faces-config.xml				
JSF <u>S</u> ervlet Name:	Faces Servlet				
JSF Servlet Class Name:	javax.faces.webapp.FacesServlet				
URL Mapping Patterns:	*.seam				
?	< <u>B</u> ack <u>N</u> ext > Cancel				

Figure 1.8. JSF Capabilities Adding

Next wizard step needs more settings that previous. Let's start with General section.

Leave the default Seam runtime and check a WAR deployment.

@	New Seam Project			
Seam Facet	Seam Facet			
Configure Seam Facet Settings				
General				
Seam Runti	me:	Seam 2.2.EAP5		
Deploy as:		● WAR ○ EAR		
EJB project	: name:	workshop-ejb		
EAR project	t name:	workshop-ear		

Figure 1.9. Seam Facet Setting

The Database section is a little tricky. The Connection Profile needs to be edited so that the new project works properly with the external HSQLDB server. By default the project wizard tries to use the JBoss embedded HSQLDB, but the tutorial uses an external database to replicate a more real world development scenario. Click on the **Edit** button to modify the Connection Profile.

Database		
Database Type:	HSQL	
Connection profile:	DefaultDS 😂 Ed	i
Database Schema Name:		
Database Catalog Name:		
DB Tables already exists in database:		
Recreate database tables and data on deploy:		

Figure 1.10. DataBase Setting

Select HSQLDB Profile Properties. Make sure the Database location is set to hsql://localhost:1701

e	Properties for
type filter text 🔏	HSQLDB Profile Properties
Common	
Default Schema Filter	Drivers: Hypersonic DB
Default Stored Procedure Filter	Properties
Default Table Filter HSQLDB Profile Properties	General Optional
Version	Database: Default
	Database location: hsql://lo
	User name: sa
	Password:
	Save Password
	URL: jdbc:hsq
?	

Figure 1.11. JDBC Connection Properties

Click the **Test Connection** button. At this point it probably won't work. This happens if the HSQL JDBC driver is not exactly the same. This can be solved by modifying the HSQLDB database driver settings. To modify the settings, click the **Edit Driver Definition Driver** button.

9	Properties for
type filter text 🧳	HSQLDB Profile Properties
Common	
Default Schema Filter	Drivers: Hypersonic DB
Default Stored Procedure Filter	Properties
Default Table Filter	General Optional
HSQLDB Profile Properties Version	Database: Default
	Database location: hsql://lo
	User name: sa
	Password:
	Save Password
	URL: jdbc:hsq
?	

Figure 1.12. Driver Details

The proper Driver JAR File should be listed under Jar List. Select the hsqldb.jar file found in the jbdevstudio/jboss-eap/jboss-as/common/lib/ directory and click the **OK** button.

1)			Edit Driver Definition		
F	Provide Driver Details					
				o provide a unique name, a list of requir operty values.		
ſ	Name/Type	JAR List	Properties			
	Driver <u>fi</u> les:		•			
	/redhat/app	os/jbdevst	udio4/jboss-	eap/jboss-as/common/lib/hsqldb.jar		
	(?)			Cancel		
	•			Cancer		

Figure 1.13. Driver Details

Now, the **Test Connection** should succeed. After testing the connection, click the **OK** button.

۲		Success	
	Ping succeeded!		
No.	-		
			ſ

Figure 1.14. Connection Testing

You can leave the Code Generation section as is. It refers to Java packages in which the generated code will be placed.

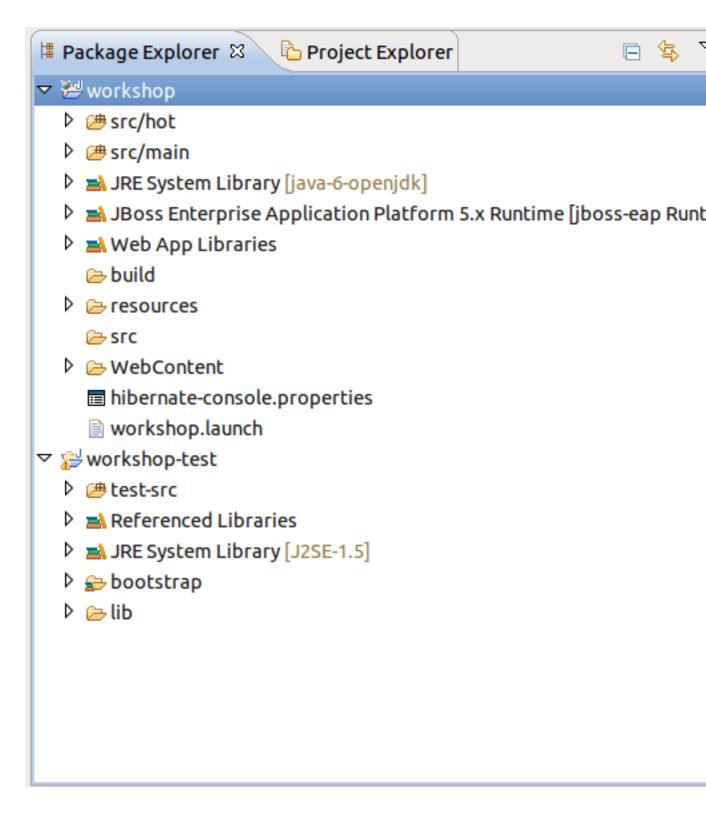
Code Generation	
Session Bean Package Name:	org.domain.workshop.session
Entity Bean Package Name:	org.domain.workshop.entity
Create Test Project:	
Test project name:	workshop-test
Test Package Name:	org.domain.workshop.test

Figure 1.15. Code Generation Setting

Ш



Click on **Finish** button. Now, there should be a new Seam project called "workshop" listed in the Package Explorer view.





1.3. Start JBoss Application Server

Start the server by clicking on the Start the server icon (

) in the Servers view.

Then run the project by selecting the project then selecting Run As... \rightarrow Run on Server.

🗏 Package Ex	xplorer 🛛 🖒 Project Explorer	•	
 ✓ Worksho ▷ # src/h ▷ # src/n ▷ A JRE S ▷ A JBoss ▷ A Web ▷ A Web ▷ Duild ▷ A Web ▷ Duild ▷ A Web ▷ Build ▷ A Web ▷ A	Go Into Open in New Window Open Type Hierarchy Show In Copy Qualified Name Paste Delete Build Path Source Refactor Import Export Export Refresh Close Project Assign Working Sets	F4 Shift+Alt+W > Ctrl+C Ctrl+V Delete Shift+Alt+S > Shift+Alt+T > F5	oss-eap Runt
	<u>R</u> un As	\rightarrow	<u>1</u> Run on 1
	<u>D</u> ebug As	>	<u>2</u> Java Ap
	Profile As	>	<u>3</u> Java Ap
	Mark as Deployable		<u>4</u> TestNG
Properties			Ru <u>n</u> Conf
	T <u>e</u> am	>	

Figure 1.17. "workshop" Run As

Select the server you want to run the project on, and click the **Finish** button.

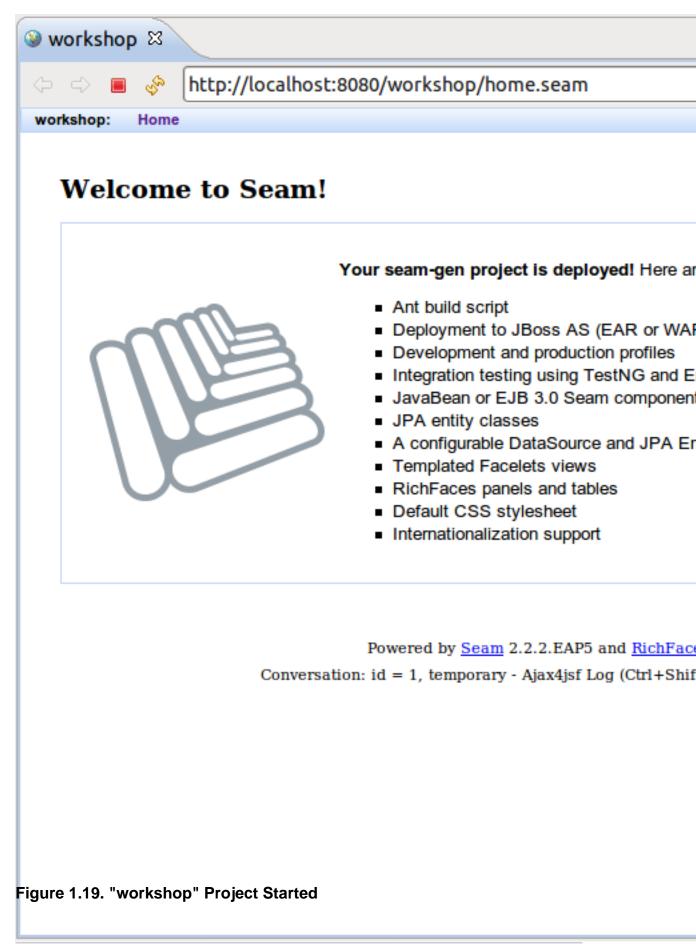
🛞 Run On Server	
Run On Server	
Select which server to use	
How do you want to select the server?	
Choose an existing server	
Manually define a new server	
Select the server that you want to use:	
type filter text	
▽ 🗁 localhost	
🕻 jboss-eap 🚯 Started	
IBoss Enterprise Application Platform (EAP) 5 x	
JBoss Enterprise Application Platform (EAP) 5.x	
Always use this server when running this project	
Figure 1.18. "workshop" Run On Server <u>N</u> ext > Cancel	



Note:

If the project does not show up, then you can use a normal browser and use *http://localhost:8080/workshop/home.seam* as the URL.

Your project looks like this:



1.4. Workshop Project Code Overview

Now let's examine the project and its structure. Go back to the Package Explorer view in JBoss Developer Studio.

It seems like it's not much for a project but this shell application contains a login screen with default login logic, a menu template that can be further modified, and other layout templates.

It's important to note that the business logic will reside in the src/hot folder, by default. And, the package naming conventions that were used in New Seam project wizard could have been changed to something different from *org.domain.workshop.session*. Also, notice that there is a default Authenticator.java file. This is where custom security logic can be added. Seam has a nice declarative security model that we will explore in more detail later on. The src/main folder is a model directory. It stores the project's JPA entity beans.

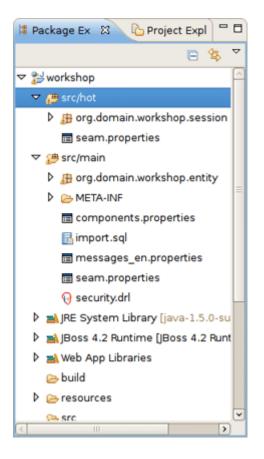


Figure 1.20. Project Structure

The view tier of the application is also important. Seam uses facelets and there is a built-in facelets GUI editor that includes nice WYSIWYG and component drag/drop functionality. Try this out by opening home.xhtml from WebContent folder.

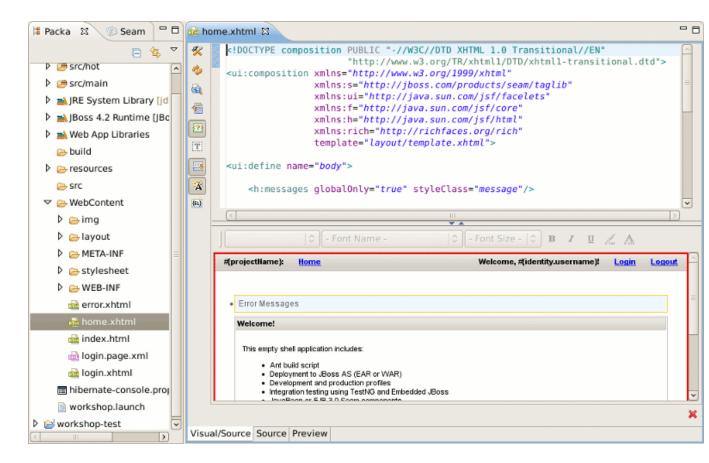


Figure 1.21. Facelets GUI Editor

Notice that the templates reside in the WebContent/layout folder. There is a stylesheet in the WebContent/stylesheet folder. There is also a login and default error page. The Facelet editor will be explored in more detail later in the lab.

The project already has a datasource that was created via the Seam project wizard database settings. All of the Seam specific configuration files and JAR dependencies are included and located in their proper locations. On last noteworthy line item is related to the build script. There isn't a build script because the Eclipse WTP (Web Tools Project) plugin is used to publish web application changes. As you can see, JBoss Developer Studio is removing a great deal of complexity from the enterprise Java project setup and deployment process. The end result is the developer is writing code, not spending time trying to figure out how to get a decent development environment and project build process.

Seam Action Development

Now it's time to write some code. The good news is that JBoss Developer Studio can also help out in this respect. In this section, we will create a new Seam Action POJO and facelet with some custom business logic and some GUI changes.

2.1. Create a New Seam Action

Go to main menu bar and click on $\textbf{File} \rightarrow \textbf{New} \rightarrow \textbf{New}$ Seam Action to start the New Seam Action wizard.

Specify a Seam component name (e.g., "myAction"). The other properties will be auto-completed for you so there is no need to change them. Click on the **Finish** button.

9	New Seam Action	
Seam Action		
Create a new Seam action	n	
Seam Project:	workshop	Browse
Seam component name:	myAction	
POJO class name:	MyAction	
Bean name:	MyActionBean	
Method name:	myAction	
Page name:	myAction	
Package name:	org.domain.workshop.session	
?		Cancel

Figure 2.1. New Seam Action Wizard

Now, open the ${\tt MyAction.java}$ file and replace the "myAction" method with this logic:

```
public void myAction() {
Calendar cal = Calendar.getInstance();
log.info("myAction.myAction() action called");
statusMessages.add("MyAction Executed on:" + cal.getTime());
}
```

You also need to import the java.util.Calendar class by clicking CTRL+Shift+O.

2.2. Test Seam Action

The new action can be tested by browsing the workshop-test project. JBoss Developer Studio has already created a TestNG test case for you.

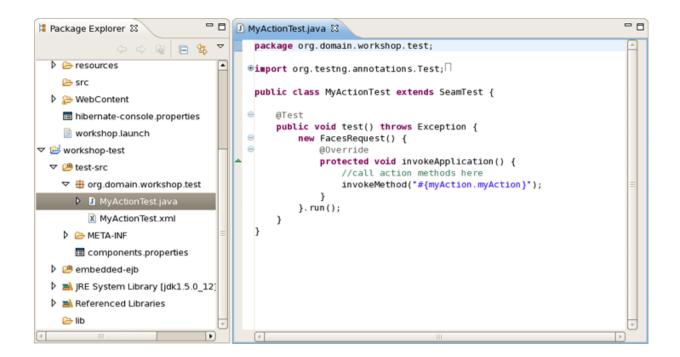


Figure 2.2. "workshop-test" Project



The test case simulates a Seam method execution for the MyAction.myAction() logic.

To run the test case, right click on MyActionTest.xml and click Run As \rightarrow TestNG Suite or use the Run As... toolbar shortcut as shown below.

□ • □ ↓	28 ₩ @•] .4] @] 8] 9 • ** ● � *5] 9 • *8
Package Explorer 8 Bun As	🕨 📙 1 Run on Server 🛛 Shift+Alt+X R
Copen Run Dialo Organize Favor	
 ✓	<pre>public class MyActionTest extends SeamTest { @Test public void test() throws Exception { new FacesRequest() { </pre>
MyActionTest.xml META-INF Gomponents.properties embedded-ejb Do Guntern Library fidika 5.0.12	<pre>@ @Override protected void invokeApplication() { //call action methods here invokeMethod("#{MyAction.myAction}"); } }.run(); </pre>
 B JRE System Library [jdk1.5.0_12 B Referenced Libraries lib 	}

Figure 2.3. TestNG Running

With any luck, the test case will pass. Look at the TestNG view.

🖳 Console 👫 R
Tests: 1/1 Me

Figure 2.4. TestNG Results

Now, it's safe to test the new Seam Action in a web browser. The fastest way to do that is to right click on myAction.xhtml and use **Run As...** \rightarrow **Run On Server** which will show the appropriate URL in the browser. Alternatively you can manually enter *http://localhost:8080/workshop/myAction.seam* into a browser.

workshop	📾 home.xhtml	MyAction.java	🥯 workshop 🖾
🗢 🗢 🔳 🖑	http://localhost:	8080/workshop/myAc	tion.seam
workshop: Hom	e		
MyAction	Executed on:Fri Jan	14 09:25:35 EST 2011	
myAction			
myAction!)		
	Conversat		.2.EAP5 and <u>RichFaces</u> . Gener jax4jsf Log (Ctrl+Shift+D) - <u>De</u>

Figure 2.5. Seam Action in a Web Browser

2.3. Modify Seam Action User Interface

Browse to *http://localhost:8080/workshop/myAction.seam* and click on the **myAction** button. This executes the "myAction" method. This looks pretty good, but we could make this page look a little better.

Open WebContent/myAction.xhtml in JBoss Developer Studio to use the nice facelets editor.



📾 my	Action.xhtml 🕴	- 8		
<pre> <!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional</th> </pre>				
	<ui:define name="body"></ui:define>			
(er)	<pre></pre>			
-				
	○ - Font Name - ○ - Font Size - ○ B I			
	#(projectName): <u>Home</u> Welcome, #(identity.username): <u>Login Logo</u>	1 _		
	Error Messages myAction myAction Powered by Seam. Generated by seam-gen.			
Visua	Source Source Preview	^		
1300	and a second sec			

Figure 2.6. Open Seam Action with Editor

Right click on the "myAction!" button in the visual part of editor and select <h:commandButton> Attributes.

recutation	ICE NOTIC UNICES LISE L	AISTOTHET'S LIST	EMDIOVECS LISE	- C B Z	Or der decails List	UTUCI S LISE
Error	Messages					
ryAction	-					
-						
myA	din la					
0	<h:commandbutton> Attr</h:commandbutton>					
<	Parent Tag (h:form)	2	>			
	Insert around	,	ommandButto			
	Insert before		ommandbutto	<u> </u>		
	Insert after					
	Insert into					
	Replace with					
			·			
	Strip Tag		-			
	Zoom	;	>			
	Cut	Ctrl+X				
	Сору	Ctrl+C				
	Toble					

Figure 2.7. Seam Action Editing

Change the value of the button to something different. If desired, you can change any other text on the page. Then, type **CTRL+S** to save the facelet.

)	Attributes	
h:commandB	utton>	8
name	value	
rendered		
style		
styleClass		
tabindex		
title		
type		

Figure 2.8. Attributes Dialog

Refresh http://localhost:8080/workshop/myAction.seam and now you should see your changes.

Notice that you did not have to publish the application. JBoss Developer Studio auto-published it for you.

i myAction.xhtml ⊠		- 0
#{projectName}: <u>Home</u>	Welcome, #{identity.username} Login	Logout
Error Messages		
myAction		
Click on My Action Button		
	Powered by Seam. Generated by seam-gen.	
Visual/Source Preview		

Figure 2.9. Seam Action Is Modified

Declarative Security

In this section you will see how easy it is to secure the facelets and facelet components in Seam. Let's go ahead and secure the action button, then we will secure the entire page.

3.1. Edit Login Authentication Logic

There is a class called Authenticator.java. The login page will execute the Authenticator.authenticate() method by default, so we'll start by viewing the authentication logic.

Open Authenticator.java in JBoss Developer Studio and you will see that it contains the authenticate() method with this code:

```
public boolean authenticate()
{
    log.info("authenticating {0}", credentials.getUsername());
    //write your authentication logic here,
    //return true if the authentication was
    //successful, false otherwise
    if ("admin".equals(credentials.getUsername()))
    {
        identity.addRole("admin");
        return true;
    }
    return false;
}
```

3.2. Secure Seam Page Component

Open myAction.xhtml and add a new secured command button:

```
<h:commandButton id="myActionSecured"
value="Secured Action Button"
action="#{myAction.myAction}"
rendered="#{s:hasRole('admin')}"/>
```

Refresh *http://localhost:8080/workshop/myAction.seam* If you are not logged in you will only see one button. If you are logged in, there will be two buttons.

🥹 workshop 🛱
🗇 🔿 🔳 🦑 http://localhost:8080/workshop/myAction.seam
workshop: Home
myAction
inyAction
Click on My Action Button
Dewered by Seem 2.2.2 EAD5 and PickEe
Powered by <u>Seam</u> 2.2.2.EAP5 and <u>RichFa</u>
Conversation: id = 8, temporary - Ajax4jsf Log (Ctrl+Sh

Figure 3.1. One Button on a Page

The secured button is not visible because the user isn't logged in as "admin".

🥹 workshop 😫
🗢 🗢 🔳 🤣 http://localhost:8080/workshop/myAction.seam
workshop: Home
myAction
Click on My Action Button Secured Action Button
Powered by <u>Seam</u> 2.2.2.EAP5 and <u>RichFac</u> Conversation: id = 11, temporary - Ajax4jsf Log (Ctrl+Shi

Figure 3.2. Secured Button is Visible

The user is logged in as "admin". Securing components is easy but securing pages is pretty simple as well.

Open *WebContent/WEB-INF/pages.xml*. Then add this markup directly underneath the <pages> element:

```
<page view-id="/myAction.xhtml" login-required="true"/>
```

Refresh *http://localhost:8080/workshop/myAction.seam* If you are not logged in you will get bounced back to the login page.

workshop [™]
🗢 🗢 🔳 🦑 http://localhost:8080/workshop/login.seam
workshop: Home
Login
Please login here
Username
Password
Remember me
Note - You may login with the username 'admin' and a blank password.
Login
Powered by Seam 2.2.2.EAP5 and RichFac
Conversation: id = 13, temporary - Ajax4jsf Log (Ctrl+Shi

Figure 3.3. Login Page

Thus, if you enter login credentials for the "admin" user, you will be re-directed to the secured page and secured component. If you enter different login credentials, page access will be granted, but the secured component will not be displayed.

Congratulations! You have secured your new action both at the facelet component and page level. You also added custom authentication logic to the login action.

Browsing Workshop Database

In this section you get to know how to use the workshop database that was started at the beginning of the lab.

4.1. Database Connectivity Setup

The workshop data can be browsed inside of JBoss Developer Studio.

To open the Data Source Explorer, click on Window \rightarrow Open Perspective \rightarrow Other \rightarrow Database Development.

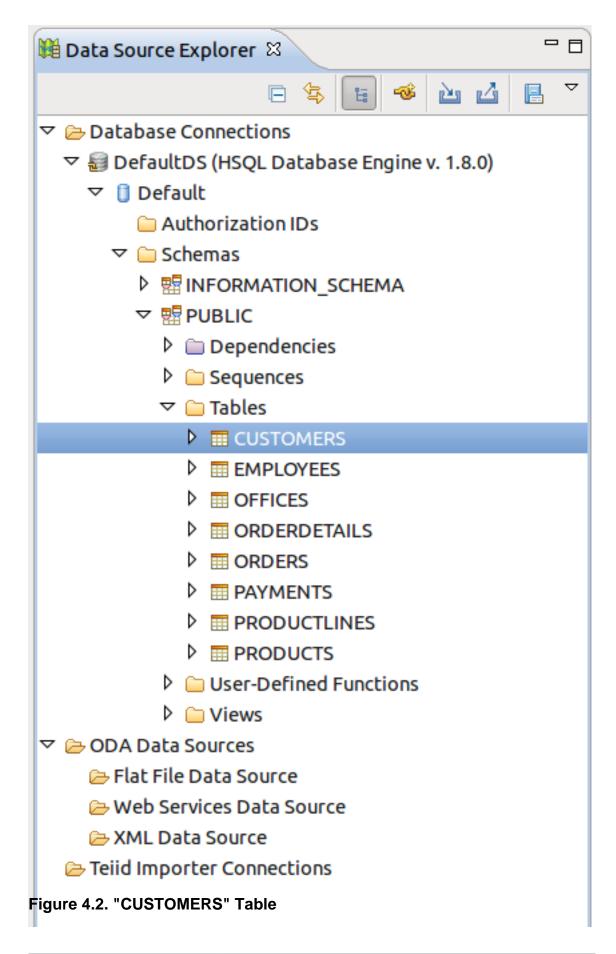
In the Data Source Explorer, expand the Databases node and select the Default database. Right click on it, select **Connect** from the context menu.

🛍 Data Source Explorer	X	- 8
	🖻 🔄 🖶 🛍 🖬	4 🖪 🍸
🗢 🗁 Database Connecti	ons	
🦉 DefaultDS		
🗢 🗁 ODA Data Source	<u>C</u> onnect	
🗁 Flat File Data S	<u>D</u> isconnect	
B Web Services D	<u>P</u> ing	
AML Data Sour	Work Offline	
E Teiid Importer Co	Save Offline	
	Re <u>n</u> ame	F2
	D <u>e</u> lete	Delete
	Duplicate	
	Ref <u>r</u> esh	F5
	Open SQL Scrapboo <u>k</u>	
	<u>P</u> roperties	Alt+Enter



4.2. Browse Workshop Database

Then in the current view, drill down to the CUSTOMERS table.



Right click on CUSTOMERS, select **Data** \rightarrow **Sample Contents** to view the data in the table.

There should be a SQL Results view on the workbench, but it could be hidden. Click on the "Result1" tab in the right side and you should see the data in the CUSTOMERS table.

🗖 SQL Results 🕴 🖳 Console			2				- X	× 🐐 [È 🗎 🏦 🎽	- 8
Type query expression here			Statu	us Result1						
Status	Operation	Date		CUSTOMER	CUSTOMERNAME	CONTACTLAST	CONTACT	PHONE	ADDRESSLINE1	A
🗸 Succe		12/30/0	1	237	ANG Resellers	Camino	Alejandra	(91)74	Gran V\u005cu(
			2	242	Alpha Cognac	Roulet	Annette	61.77.6	1 rue Alsace-Lo	
			3	206	Asian Shopping N	Walker	Brydey	+612 9	Suntec Tower TI	8
			4	103	Atelier graphique	Schmitt	Carine	40.32.2	54, rue Royale	
			5	144	Volvo Model Repli	Berglund	Christina	0921-1	Berguvsv\u005c	
			6	189	Clover Collections	Cassidy	Dean	+353 1;	25 Maiden Lane	F
			7	141	Euro+ Shopping C	Freyre	Diego	(91) 55	C/ Moralzarzal,	
			8	216	Enaco Distributors	Saavedra	Eduardo	(93) 20	Rambla de Cata	
			9	201	UK Collectables, L	Devon	Elizabeth	(171) 5?	12, Berkeley Ga	
			10	148	Dragon Souvenie	Natividad	Eric	+65 22	Bronz Sok.	в
			11	209	Mini Caravy	Citeaux	Fr\u005cu	88.60.1	24, place Kl\u00	
			12	240	giftsbymail.co.uk	Bennett	Helen	(198) 53	Garden House	c
			13	223	Nat\u005cu005cu	Kloss	Horst	0372-5	Taucherstra\u00	
			14	169	Porto Imports Co.	de Castro	Isabel	(1) 356	Estrada da sa\u	
			15	119	La Rochelle Gifts	Labrune	Janine	40.67.8	67, rue des Cint	
			16	112	Signal Gift Stores	King	Jean	702555	8489 Strong St.	
			4							
4		Þ	Total	50 records s	hown					

Figure 4.3. SQL Results View

i Note:

If you can't find the SQL Results view tab, click on Window \rightarrow Show View \rightarrow Other \rightarrow SQL Development \rightarrow SQL Results.

Congratulations! You just connected to the workshop database and queried the content using Database Explorer tools.

Database Programming

Now, it's time to reverse engineer the workshop database into a fully functioning Seam CRUD (Create Read Update Delete) application.

5.1. Reverse Engineer CRUD from a Running Database

In JBoss Developer Studio, switch to the Seam perspective, and then right-click the project and select New \rightarrow Seam Generate Entities.

The "workshop" project in the Seam Generate Entities wizard will be selected automatically. There is no need to change something more, click the **Next** button to proceed to the next step.

8	Generate Seam Entities	
Generate Seam Entities		
Seam Project:	workshop	Browse
Hibernate Console Configural	tion: workshop 😂	
Generation Mode O Use existing entities		
 Reverse engineer from of 	database	
Use existing reveng		
reveng. <u>x</u> ml:		BI
< <u>B</u> a	ck <u>N</u> ext > Can	:el

Figure 5.1. Generate Seam Entities

On the next page use the **Refresh** button to display the database, then click the **Include** button to include all the tables from the database, and finally click the **Finish** button.

Sector 2010	Generate S	Seam Entities
Generate Seam Entities		
Select Tables		
Database schema: Coloradia catalog> PUBLIC CUSTOMERS CUSTOMERS CUSTOMERS OFFICES ORDERDETAILS ORDERS PAYMENTS PRODUCTLINES PRODUCTS	Include Exclude Up Down Remove Remove All	Table filters: ! Catalog .* .* .* .* .* .* .* .* .* .* .* .* .* .* .* .* .* .* .* .* .* .* .* .* .* .*
?	< <u>B</u> a	ack <u>N</u> ext >

Figure 5.2. Selecting Tables

After running the Generate Entities action, you will see new *org.domain.workshop.entity* classes. These classes represent insert/update/delete/query logic.

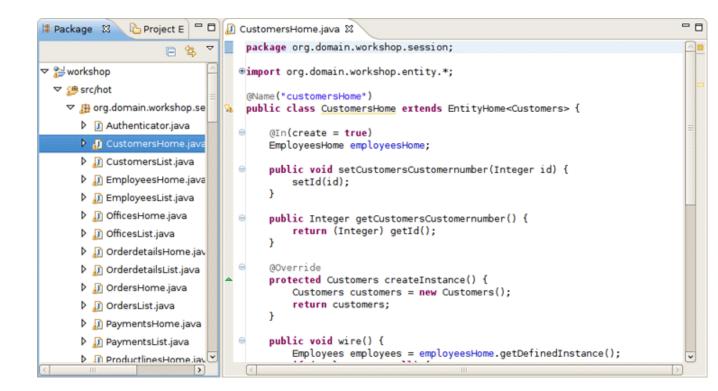


Figure 5.3. org.domain.workshop.entity Classes

There is also the *org.domain.workshop.entity* package that contains the JPA classes. These are the entity beans that are mapped to database tables. Note that you can use Seam refactoring tools with Seam components. Read more about it in *Seam refactoring tools chapter* [http://download.jboss.org/jbosstools/nightly-docs/en/seam/html_single/ index.html#seam_refactoring] of Seam Dev Tools Reference Guide.

Last, but not least, there are facelets for all of the CRUD screens. The best way to get a feel for the generated code is to open a browser and play around with the application. Go to *http://localhost:8080/workshop* and insert/update/delete/query a few records. There is quite a bit of AJAX in this application, but we will explore that later on in the lab. For now, take note of the page tabs, required field logic and data table sorting in the list pages.



Tip

If you see the error java.lang.ClassNotFoundException: org.jboss.seam.servlet.SeamListener in the console output from the Application Server, you may need to copy the jboss-seam.jar file from the lib subdirectory in the Seam library (which can be downloaded from here [http://seamframework.org/Seam2/Seam2DistributionDownloads]) into the /server/default/deploy/workshop.war/WEB-INF/lib/ subdirectory in your Application Server (where "default" refers to the server profile that you are using).



Tip

If you see the error Could not instantiate Seam component: org.jboss.seam.security.ruleBasedPermissionResolver, COPY the mvel2.jar file from the Seam library to the same destination directory mentioned in the tip above.

{projectName}: <u>Home</u>	Offices List	Customers List	Employees List	Products List	Orderde
Error Messages					
Customers search para	ameters				ĸ
customername	#{customersList.c	ustomer			
contactlastname	#{customersList.c	ustomer			
contactfirstname	#{customersList.c	ustomer			
phone	#{customersList.c	ustomer			
addressline1	#{customersList.c	ustomer			
addressline2	#{customersList.c	ustomer			
city	#{customersList.c	ustomer			
state	#{customersList.c	ustomer			
postalcode	#{customersList.c	ustomer			
country	#{customersList.c	ustomer			
Search					
Customers search resu	ults				
No customers exists					

Figure 5.4. CustomersList.xhtml in the Editor

Congratulations! You now have a fully functioning CRUD application that is AJAX enabled.

5.2. Use Hibernate Tools to Query Data via JPA

Now, it's time to write some JPA queries using the Hibernate perspective in JBoss Developer Studio.

In the upper right corner of the workbench there is a small icon (see the figure below), click on it and select **Hibernate**.

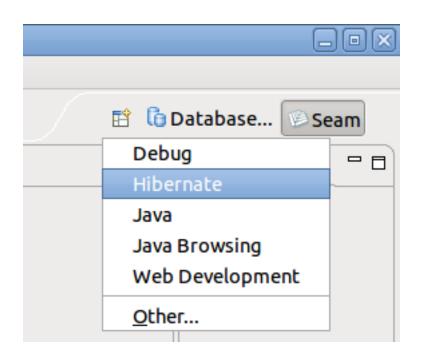


Figure 5.5. Hibernate Perspective

Look at the Hibernate Configurations view. In the "workshop" project, drill down on the Session Factory and notice that the JPA entities/attributes are listed in a nice tree view.

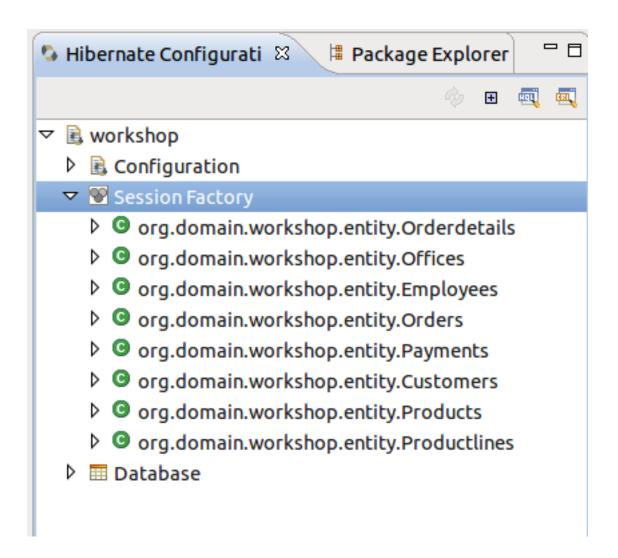


Figure 5.6. Hibernate Configurations View

Right click on the Session Factory and select **HQL Editor**. This will open a JPA query scratch pad window.

Write your query and click on the "Hibernate Dynamic SQL Preview" tab. You should see the SQL that will be executed if this JPA query is run.

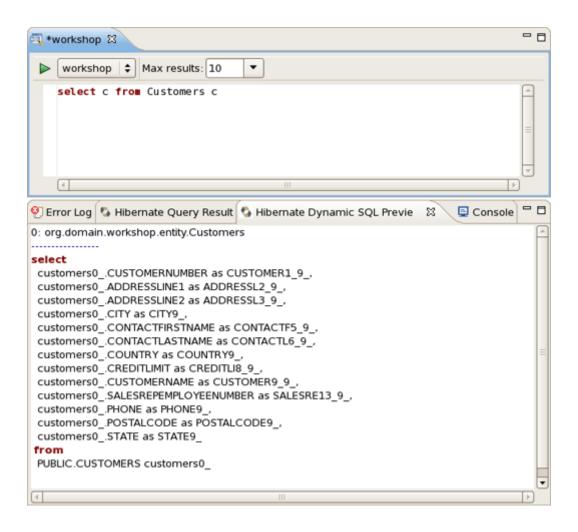


Figure 5.7. JPA Query Editor

Run the query by clicking on the green run icon.

The results are listed in the "Hibernate Query Result" view. There is a "Properties" tab in the workbench that can be used to see a specific JPA result. These results represent the JPA objects because our query did not specify column names.

	Properties 🛿	🖬 🐉 🗔 🔻 🗖	💐 *workshop 🕴	- 0
Pro	operty	Value	workshop 🗘 Max results: 10 💌	
▽	Identifier		select c from Customers c	A
	customernumbe	103		
▽	Properties			=
	addressline1	54, rue Royale		~
	addressline2			
	city	Nantes	🤨 Error Log 😘 Hibernate Query Result 🕱 💊 Hibernate Dynamic SQL Pr 📮 Console	
	contactfirstnam€	Carine		× %
	contactlastname	Schmitt		na 136
	country	France	0	Ê
	creditlimit	21000.0	org.domain.workshop.entity.Customers@c9fc25	
	customername	Atelier graphique	org.domain.workshop.entity.Customers@841423	
	employees		org.domain.workshop.entity.Customers@1463027	
	orderses		org.domain.workshop.entity.Customers@1038048	
	Þ paymentses		org.domain.workshop.entity.Customers@db0d07	
	phone	40.32.2555	org.domain.workshop.entity.Customers@134817a	
	postalcode	44000	org.domain.workshop.entity.Customers@207655	
	state		org.domain.workshop.entity.Customers@138d8fa	
			org.domain.workshop.entity.Customers@12d2db8	
			org.domain.workshop.entity.Customers@11499ff	•
4	111		select c from Custom select c from Custom select c from Custom 🕱 39	

Figure 5.8. Hibernate Query Result View

The query can be refined, and take note that there is nice code completion in the JPA query editor.

در *workshop ۲	
<pre>workshop \$ Max results: 10 * select c.customername, c.employees. from Customers c</pre>	
 * customerses - Employees email - Employees employeenumber - Employees 	
 extension - Employees firstname - Employees jobtitle - Employees jobtitle - Employees lastname - Employees officecode - Employees officecode - Employees reportsto - Employees reportsto - Employees 	
org.domain.workshop.entity.Customers@841423 org.domain.workshop.entity.Customers@1463027 org.domain.workshop.entity.Customers@1038048 org.domain.workshop.entity.Customers@db0d07 select c from Custom_select c from Custom	v

Figure 5.9. Code Completion

A refined query will return results that are more ResultSet oriented. Notice the join logic that JPA supports.

*workshop &		- f
<pre>workshop \$ Max r select c.customern</pre>	ame, c.employees.extension from Customers c	*
•		
🕽 Error Log 🕵 Hibernate G	Query Result 🖾 💊 Hibernate Dynamic SQL Prev 📮 Console 🖻	- 1
/		84
	A	2
0	1	
Atelier graphique	x2028	
Signal Gift Stores	x4065	
Australian Collectors, Co.	×101	
La Rochelle Gifts	x2028	
Baane Mini Imports	x102	
Mini Gifts Distributors Ltd.	x3291	
Blauer See Auto, Co.	×102	
Mini Wheels Co.	x3291	
Land of Toys Inc.	x4102	
Euro+ Shopping Channel	x2028	
		_

Figure 5.10. The Hibernate Query Result

There was no need to specify an Employees table in the from part of the JPA query because JPA supports reference traversal via Java class attribute references. Not only are JPA and HQL queries fully supported, but Criteria based queries can also be written in the Criteria Editor. You should spend some time tinkering with different queries and possibly Criteria based queries, even though the instructions are not provided in this lab.

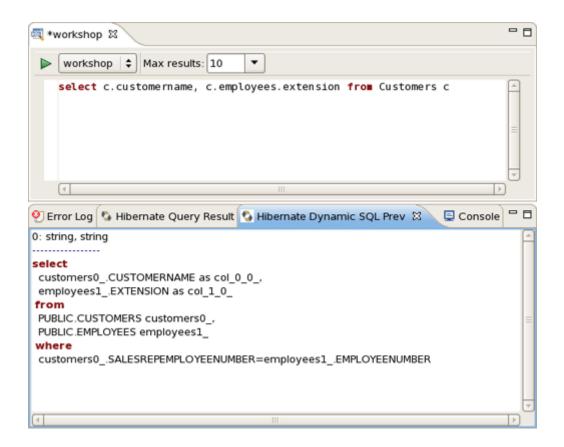


Figure 5.11. Criteria Editor

5.3. Use Hibernate Tools to visualize the Data Model

Now, it's time to view the data model for the workshop database.

In the Hibernate Configurations view, select "workshop" project and expand the Configuration node. Select the Customers entity, right click on it, choose **Mapping Diagram**.

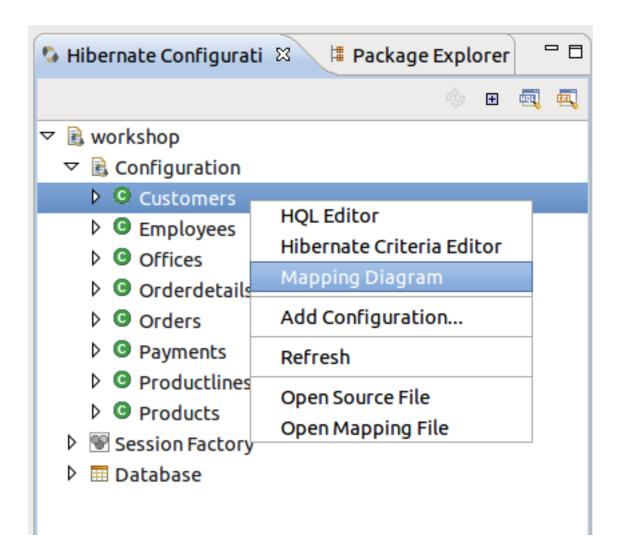
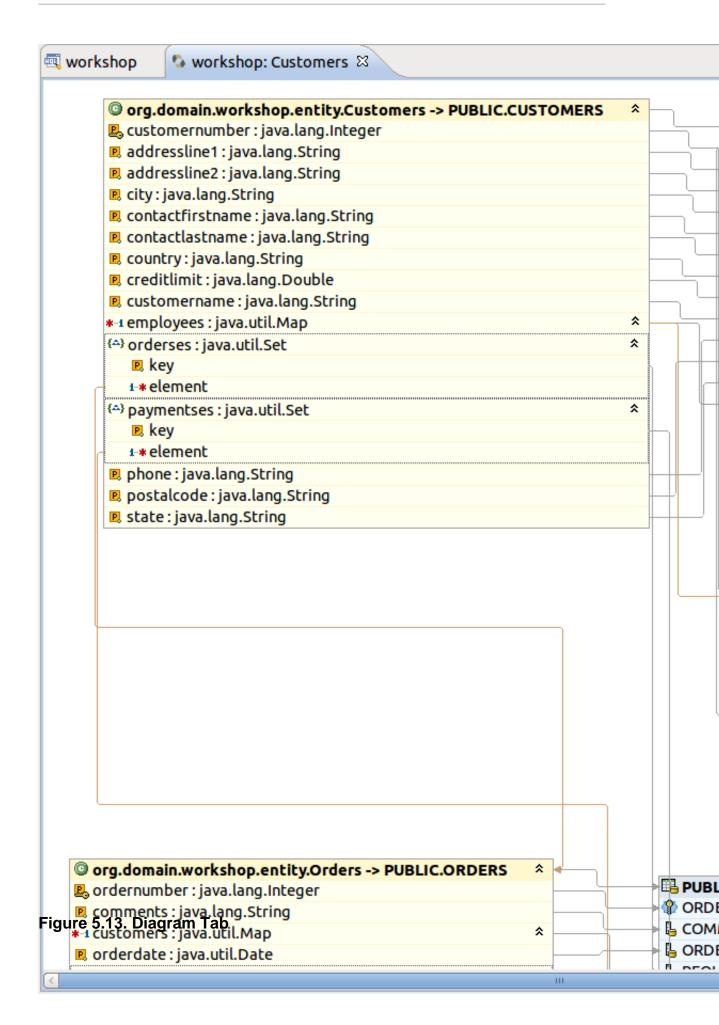


Figure 5.12. Mapping Diagram Opening

You see a Diagram tab for the CUSTOMERS table and any tables that have FK references. This is a handy way to view the data model and JPA mappings. Now, you've got access to something that the Erwin Data Modeler can't do.



Rich Components

This lab will conclude with one last AJAX twist. In this section we add a RichFaces inputNumberSlider to the Order Details edit screen.

6.1. Add a Richfaces component to the CRUD Application

Switch to Seam perspective, and open WebContent/OrderdetailsEdit.xhtml in JBoss Developer Studio.

Change the form field values using the visual editor. Seam has generated the form field names that match the database column names. This is not ideal for business users.

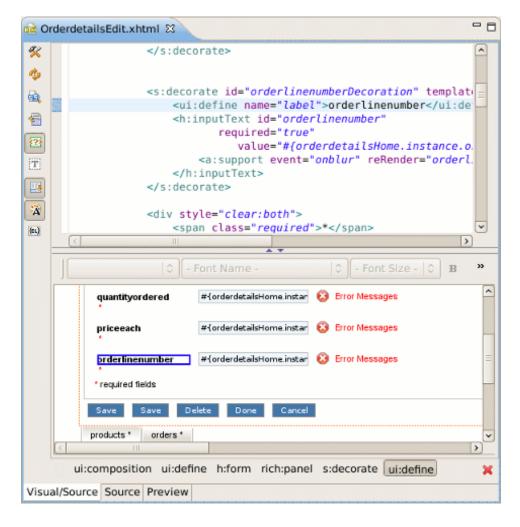


Figure 6.1. Form Fields Editing

Also, replace the QTY Ordered input field with a inputNumberSlider. You can use the JBoss Developer Studio palette or right click on the form and insert the RichFaces component.

]	- Font Name - 🛛 🗘	;][- Fon!	t Size - 0		β <u>Γ</u> <u>Π</u>	la.	<rich:dndparam> <rich:dragindicator></rich:dragindicator></rich:dndparam>	
		Error Mess					<rich:draglistener> <rich:dragsupport> <rich:dropdownmenu></rich:dropdownmenu></rich:dragsupport></rich:draglistener>	
Priceeach #{(Orderlinenumber #{(orderdetailsHome.instan	Error Mess					<rich:droplistener> <rich:dropsupport> <rich:editor> <rich:effect> <rich:extendeddatatable></rich:extendeddatatable></rich:effect></rich:editor></rich:dropsupport></rich:droplistener>	
required fields Save Save Delete Products * Orders *	Insert around Insert before Insert after	>	IBoss	>	Ajax4Jsf	>	<rich:extendeddatatable> <rich:fileupload> <rich:gmap> <rich:graphvalidator></rich:graphvalidator></rich:gmap></rich:fileupload></rich:extendeddatatable>	
There is no products associa Productcode	Replace with	>	JSF name	>	RichFaces Seam	>	<rich:hotkey> <rich:inplaceinput></rich:inplaceinput></rich:hotkey>	lescription
#{_products.productcode} #{	Zoom	> Ctrl+X	oductname}	#(_	products.produc	tscal	<rich:inplaceselect> <rich:inputnumberslider> <rich:inputnumberspinner></rich:inputnumberspinner></rich:inputnumberslider></rich:inplaceselect>	oductdescription
vi:composition ui:de	fi ⊆opy		ui:define				<rich:insert> <rich:jquery> <rich:jsfunction></rich:jsfunction></rich:jquery></rich:insert>	
							<rich:layout> <rich:layoutpanel> <rich:listshuttle></rich:listshuttle></rich:layoutpanel></rich:layout>	

Figure 6.2. Insert RichFaces Component from Context Menu

One the last option is to use the source view and manually copy the inputNumberSlider markup listed below:

```
<rich:inputNumberSlider id="quantityOrdered" required="true" value="#{orderdetailsHome.instance.quantityordered}"/>
```

nd *(OrderdetailsEdit.xhtml 🖾	• 8
* • •	<pre><a:support event="onblur" quantityordered"<br="" rerender="orderlinenumbe
</h:inputText>
</s:decorate>
<rich:inputNumberSlider id=">required="true" value="#{orderdetailsHome.instance.qua </a:support></pre>	
	<pre></pre>	

Figure 6.3. Manually copying Source Code

The end result is an edit page that has better form labels and a new RichFaces control.

🥹 workshop 🖾	
🗇 🔿 🔳 🤣 http:/	//localhost:8080/workshop/OrderdetailsEdit.seam
workshop: Home Bro	wse data
Add Orderdetails	
Quantityordered *	0 100 0
Priceeach *	0.0
Orderlinenumber*	0
* required fields	
Save Cancel	
Orders * Products *	
There is no orders assoc	iated with this orderdetails.
Select orders	
	Powered by Seam 2.2.2.EAP5 and RichFaces
	Conversation: id = 25, long running - Ajax4jsf Log (Ctrl+Shi

Figure 6.4. The Result Page

Congratulations! You have completed the JBoss Developer Studio lab.