



Setting up Eclipse Juno, JBOSS 7 AS, RichFaces 4, Hibernate 4, Oracle 11g development environment

Author: Enes Hukic

Date: 6.2.2013

About

Recently I have inherited a Java project that was created using RichFaces 3, JPA Hibernate 3. This project was running on JBOSS 5.1.

The goal was to do the upgrade the project to use JSF 2.1, RichFaces 4 and Hibirnate 4. The database used in this project is Oracle 11g. These instructions can apply to any database. As I have found out, this upgrade was not a straight forward task since a lot of methods in Hibrenate 4 have been deprecated, RichFaces libraries have some dependencies that do not come out of the box and some of the RichFaces tags do not exist anymore.

This project needs at least JRE 1.6.

The goal of this document is to summarize all the steps that are necessary to setup development environment using: Eclipse Juno, JBOSS 7 AS, RichFaces 4, Hibernate 4, Oracle 11g.

Setting up Eclipse

The first step is to download and install (extract) Eclipse Juno. The directory where Eclipse is installed (extracted) will be referred as ECLIPSE_HOME in this document.

To install JBoss tools 4:

1. go to Help -> Eclipse Marketplace
2. in the search box type "jboss tools" and click search icon.
3. Find "JBoss Tools (Juno)" and click install.

I had to settle for the Windows 32 bit version since 64 bit version had some problems (design view did not work, etc).

Setting up JBoss AS 7

JBoss AS 7 can be downloaded from <http://www.jboss.org/jbossas>. Extract JBoss AS zip file into ECLIPSE_HOME directory.

If you have used earlier versions of JBOSS, you will notice that the file structure is different.

If you are using Oracle on the same machine, you might want to change the default port number from 8080. This can be done, by editing standalone.xml file that is located in as ECLIPSE_HOME \jboss-as-7.1.1.Final\standalone\configuration directory.

Database connection

In this example I have created a module for Oracle ojdbc6.jar. The same can be done using ojdbc14.jar.

1. Create directories oracle, ojdbc6 and main inside ECLIPSE_HOME\jboss-as-7.1.1.Final\modules\com. The final path should look like ECLIPSE_HOME \jboss-as-7.1.1.Final\modules\com\oracle\ojdbc6\main
2. Copy ojdbc6.jar into main directory.
3. Create module.xml file (You can copy existing XML file from other modules). The content of module.xml should look like this:

```
<?xml version="1.0" encoding="UTF-8"?>

<module xmlns="urn:jboss:module:1.0" name="com.oracle.ojdbc6">
  <resources>
    <resource-root path="ojdbc6.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

4. Open standalone.xml file that is located in as ECLIPSE_HOME \jboss-as-7.1.1.Final\standalone\configuration directory
5. Find "datasources" section and insert the following tags (change your username and password parameters):

```
<datasource jta="true" jndi-name="java:/jdbc/OracleDS" pool-name="OracleDS"
enabled="true" use-java-context="true" use-ccm="true">
  <connection-
url>jdbc:oracle:thin:@localhost:1521:myinstance</connection-url>
  <driver>oracle</driver>
  <transaction-isolation>TRANSACTION_READ_COMMITTED</transaction-
isolation>
  <pool>
    <prefill>true</prefill>
    <use-strict-min>>false</use-strict-min>
    <flush-strategy>FailingConnectionOnly</flush-strategy>
```

```
</pool>
<security>
  <user-name>myusername</user-name>
  <password>mypassword</password>
</security>
</datasource>
```

Add driver:

```
<driver name="oracle" module="com.oracle.ojdbc6">
  <xa-datasource-class>oracle.jdbc.OracleDriver</xa-datasource-class>
</driver>
```

The Boss AS 7 should be ready to go. The next step is to add it to Eclipse: File -> New -> Server and follow the wizard instructions.

The server should be available at localhost:myport.

Your JTA data source in the persistence.xml should be the same as the jndi-name (java:/jdbc/OracleDS).

Richfaces 4

Richfaces can be downloaded from <http://www.jboss.org/richfaces/download/stable>. In readme.txt there are instructions how to manually setup project:

1. Manually

After downloading the distribution package, extract its content in a folder of your choice.

Search for the 'artifacts' directory in the extracted files. There you will find 'framework' and 'ui' directories, each of them having the jars needed to use RichFaces inside your project.

- From 'ui' directory: Copy richfaces-components-api-4.x.jar and richfaces-components-ui-4.x.jar to your application libraries folder.

- From 'framework' directory: Copy richfaces-core-api-4.x.jar and richfaces-core-impl-4.x.jar to your application libraries folder.

In addition to the RichFaces jars, a number of dependencies are required in order to properly configure the framework.

a. *sac-1.3, cssparser-0.9.5 - required for components CSS work.*

b. *google-guava-11.0.2 - core runtime dependency.*

c. *annotations.jar - from org.richfaces.cdk.*

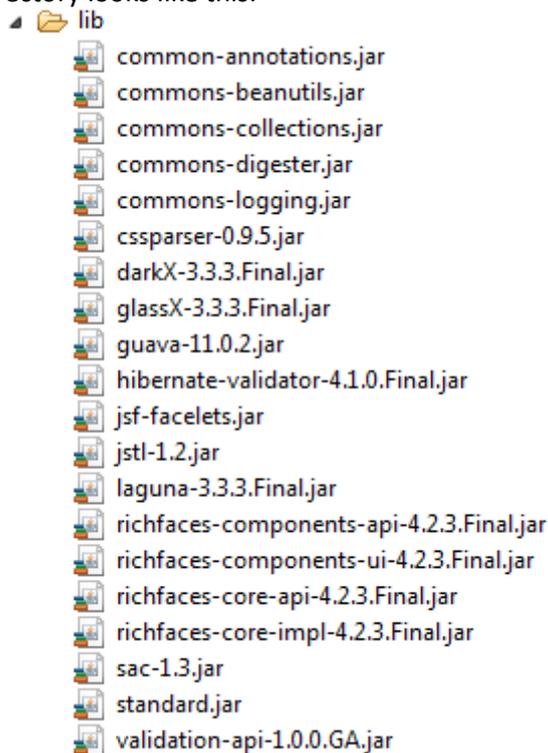
- It's optional and only needs to be added if RichFaces components will be created/accessed dynamically in your apps actions/listeners.

d. *validation-api.jar and any implementation like hibernate-validators.jar*

- It's optional and should be added if you using Client or Graph Validation. Should be added only if it's not provided by a server (Java EE 6 servers).

If you create a new JSF, JPA RichFaces project, at first xhtml pages with Richfaces tags will appear correctly. After restart of Eclipse, the 404 error will appear.

This is where the frustration begins. It turns out, the solution is to download and add `jstl-1.2.jar`. Also, the RichFaces jars with word „source“ should not be copied into lib directory. My final lib directory looks like this:



If you are creating a new application, you should be ready to go.

If you are upgrading the old application, you will find out that some of the RichFaces tags are missing. Some of the tags can be changed to new tags (ex. `Datadefinitionlist = list`). The solution is to redesign you GUI using new RichFaces tags.

Hibernate 4

Hibernate 4 can be downloaded from <http://www.hibernate.org/downloads>. Extract Hibernate zip file into workspace directory.

You will need to add all of the required JARs to your project. To do this:

1. Right click on your project, select "Properties" -> "Java Build Path"
2. Select "Libraries" tab
3. Click "Add External JARs"
4. Select all JARd that are located in the "..\hibernate-release-4.1.7.Final\lib\required" directory

If you are migrating from the earlier version of migrate you will notice notification and errors that some of the methods and classes have been deprecated. For example:

```
SessionFactory sessionFactory = new Configuration().configure().buildSessionFactory();
```

Can be changed:

Declare

```
private static ServiceRegistry serviceRegistry;
```

In your methods, change:

```
SessionFactory sessionFactory = new Configuration().configure().buildSessionFactory();
```

To

```
Configuration configuration = new Configuration();
serviceRegistry = new
ServiceRegistryBuilder().applySettings(configuration.getProperties()).buildService
Registry();
SessionFactory sessionFactory =
configuration.buildSessionFactory(serviceRegistry);
```

You will need to check for errors all of the classes that are using Hibernate.