Developing a Portlet using Eclipse

The Portlet support is experimental. Feedback is appreciated!

Index

Step-by-Step Tutorial

Introduction

This tutorial focuses on how to create an Eclipse development environment to create and debug portlets. It does not cover portal deployment (which is different for each portal) and cheats a bit by using the struts2-archetype-portlet Maven 2 archetype. We can get away with not talking about portlet development, because Struts 2 portlet support allows a carefully-written portlet to be ran as a regular servlet application, drastically simplifying our development setup. Of course, to fully test your portlet, you will need to deploy the war into a portal server, but by running it in pure Tomcat, we can get 90% of the way there. Therefore, this tutorial requires Eclipse, Maven 2, and Tomcat to be installed.

Installing Eclipse

In the tutorial, we will be using Eclipse 3.3 which can be downloaded from http://www.eclipse.org. I recommend the JEE package, which contains the popular (and required for this tutorial) Web Tools Project.

Installing Maven 2

Apache Maven 2 can be found at http://maven.apache.org.

Installing Tomcat

Apache Tomcat can be found at http://tomcat.apache.org. To install, simply unzip the distribution to a known location on your hard drive.

Creating the project

To start, we will use the Struts 2 portlet Maven archetype, struts2-archetype-portlet, which will create a new portlet project will all the necessary files for a simple "hello world"-style portlet. Open a terminal and type:

```
mvn archetype:create -DgroupId=com.mycompany.myportlet \
                     -DartifactId=myportlet \
                     -DarchetypeGroupId=org.apache.struts \
                     -DarchetypeArtifactId=struts2-archetype-portlet \
                     -DarchetypeVersion=2.0.9-SNAPSHOT \
                     -DremoteRepositories=http://people.apache.org/repo/m2-snapshot-repository
```

Aternatively, you can use Archy, a tool that provides an interactive command-line front-end to Maven Archetypes.

This will result in the following layout:

```
./src/main/java/com/mycompany/myportlet/view/IndexAction.java
./src/main/java/com/mycompany/myportlet/edit/IndexAction.java
./src/main/webapp/WEB-INF/web.xml
./src/main/webapp/WEB-INF/view/index.jsp
./src/main/webapp/WEB-INF/portlet.xml
./src/main/webapp/WEB-INF/edit/index-success.jsp
./src/main/webapp/WEB-INF/edit/index-input.jsp
./src/main/webapp/WEB-INF/help/index.jsp
./src/main/webapp/index.jsp
./src/main/resources/struts.xml
./pom.xml
```

As you can see, this portlet comes with a simple "hello world" view, but also default edit and help screens.

Eclipse project generation

A nice benefit of the Maven 2 build that new project comes equipped with is Maven 2 has a plugin that will generate all our Eclipse configuration. Move into the 'myportlet' directory and type:

```
mvn -Dwtpversion=1.0 -DdownloadSources=true eclipse:eclipse
```

First Eclipse and Maven project?

If this is your first time using Eclipse and Maven, Eclipse needs to know the path to the local maven repository. Therefore the classpath variable M2_REPO has to be set. Execute the following command:

```
mvn -Declipse.workspace=<path-to-eclipse-workspace> eclipse:add-maven-repo
```

You can also define a new classpath variable inside eclipse: From the menu bar, select Window > Preferences. Select the Java > Build Path > Classpath Variables page.
For more information on the Maven Eclipse plugin, see its homepage

Now, your generated 'myportlet' Eclipse module is ready to be imported into your workspace. In Eclipse, go to "File -> Import... -> General -> Existing Projects into Workspace", select the your 'myportlet' directory, and follow the prompts.

Deploying your portlet as a servlet

Since the Eclipse project was generated with the 'wtpversion' flag, it will be immediately recognized as a web application by Eclipse. If this is your first time deploying web applications in Eclipse, you will need set up your Tomcat server. To do this in Eclipse:

1. Navigate to "Window -> Show View -> Other..."
2. Open "Server" and select "Servers". This will open a "Servers" tab, probably in your bottom tab panel.
3. Right-click in the new "Servers" tab and select "New -> Server"
4. Select the version of Tomcat you installed and click "Next"
5. Click "Browse" and locate your Tomcat installation, and click "Next"
6. If "myportlet" isn't already in the "Configured projects" column, move it over and click "Finished"

Before we can run our portlet in Eclipse, I've found that you need to add the portlet jar to Tomcat. To do this:

1. Right-click on the 'myportlet' project in the "Project Explorer" and select "Properties"
2. Click on "J2EE Module Dependencies
3. Click on the checkbox next to "M2_REPO/portlet-api/portlet-api/1.0/portlet-api-1.0.jar"
4. Click "OK"

Now, you should be able to run and debug your project in Tomcat. The way I prefer to do this is to:

1. Right-click on the 'myportlet' project in the "Package Explorer" and select "Run As -> Run on Server"
2. Select the Tomcat server you set up and click "Finish"

Eclipse will now run your portlet application as if it was a servlet.

Additional Tips

- "View", "Edit", "Help" mode actions are mapped to the "/view", "/edit", "/help" namespaces respectively
- The default action in each namespace is titled the "index" action
- To add actions, just add the Action class and jsp following the shown conventions. No `struts.xml` modification needed.
- Use the Eclipse option "Debug As..." instead of "Run As..." to enable step through debugging