 **LIFERAY.** Enterprise. Open Source. For Life.

Profile Friends Blog Documents

Recent Bloggers

- 

[Ronald Sarayudej](#)
137 Posts
March 7, 2012
- 


[Olaf Kock](#)
35 Posts
March 6, 2012
- 

[Alice Cheng](#)
192 Posts
February 29, 2012
- 

[James Falkner](#)
43 Posts
February 28, 2012
- 

[Mika Koivisto](#)
9 Posts
February 27, 2012
- 

[Caleb Fong](#)
1 Posts
February 23, 2012
- 

[Thiago Leão Moreira](#)
8 Posts
February 23, 2012
- 

[Francisco Fernandez](#)
9 Posts
February 23, 2012
- 

[Igor Beslic](#)
5 Posts
February 23, 2012
- 

[Josef Šustáček](#)
1 Posts
February 23, 2012

Getting started with Liferay Maven SDK

February 1, 2012 By [Mika Koivisto](#)

« Back

This will be the first in series of posts on how to develop Liferay plugins with Maven. In this post we'll start by creating a new parent project for your plugins and add a portlet project to it. You need to have your maven environment setup with maven and java installed. If you don't know how to do it I would recommend reading [Maven: The Complete Reference](#) from Sonatype, Inc. The chapter 2 has good instructions on how to install maven.

- 1) Download and install Liferay 6.1.0 bundle. In these posts we assume it's tomcat bundle but you can use any bundle. I'll refer to the bundle install location is LIFERAY_HOME from now on. If you need instructions on how to install bundle please refer to [Liferay 6.1 User Guide](#).
- 2) Create a new directory which will be your project root. This is the location where you would extract Liferay plugins SDK if you were using Ant. Then in that directory create a pom.xml file.

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.liferay.sample</groupId>
  <artifactId>sample-project</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>pom</packaging>

  <name>sample-project</name>
  <url>http://www.liferay.com</url>

  <properties>
    <liferay.auto.deploy.dir>/path/to/liferay/bundle/deploy</liferay.auto.deploy.dir>
    <liferay.version>6.1.0</liferay.version>
  </properties>
</project>
```

[This Gist](#) brought to you by [GitHub](#).

[pom.xml](#) [view raw](#)

Now you should adjust groupId and artifactId to match you project. Also set the value of liferay.auto.deploy.dir to LIFERAY_HOME/deploy. This is where the plugin is copied for Liferay to deploy. The liferay.version property is set to version of Liferay you are using.

- 3) Open command prompt or terminal and go to your project directory. Next we'll going to create a portlet project using a liferay portlet project template. Run
- mvn archetype:generate

That command will create a list of available project templates like below:

February 22, 2012

```
...
21: remote -> com.liferay.maven.archetypes:liferay-ext-archetype (Provides an archetype to
create Liferay extensions.)
22: remote -> com.liferay.maven.archetypes:liferay-hook-archetype (Provides an archetype to
create Liferay hooks.)
23: remote -> com.liferay.maven.archetypes:liferay-layouttpl-archetype (Provides an archetype
to create Liferay layout templates.)
24: remote -> com.liferay.maven.archetypes:liferay-portlet-archetype (Provides an archetype to
create Liferay portlets.)
25: remote -> com.liferay.maven.archetypes:liferay-servicebuilder-archetype (Provides an
archetype to create Liferay Service Builder portlets.)
26: remote -> com.liferay.maven.archetypes:liferay-theme-archetype (Provides an archetype to
create Liferay themes.)
27: remote -> com.liferay.maven.archetypes:liferay-web-archetype (Provides an archetype to
create Liferay webs.)
...
Choose a number or apply filter (format: [groupId:]artifactId, case sensitive contains): 171:
```

[This Gist](#) brought to you by [GitHub](#).

[templates snippet](#) [view raw](#)

Choose number 24 or what ever the number you have for com.liferay.maven.archetypes:liferay-portlet-archetype

Next you will be asked to choose the template version:

```
Choose version:
1: 6.0.2
2: 6.0.3
3: 6.0.4
4: 6.0.5
5: 6.0.6
6: 6.1.0
7: 6.2.0-SNAPSHOT
```

[This Gist](#) brought to you by [GitHub](#).

[version snippet](#) [view raw](#)

Choose number 6 or what ever you have for 6.1.0 version.

Next you will be asked to provide groupId, artifactId and version:

```
Define value for property 'groupId': : com.liferay.sample
Define value for property 'artifactId': : sample-portlet
Define value for property 'version': 1.0-SNAPSHOT: :
Define value for property 'package': com.liferay.sample: :
Confirm properties configuration:
groupId: com.liferay.sample
artifactId: sample-portlet
version: 1.0-SNAPSHOT
package: com.liferay.sample
Y: :
```

[This Gist](#) brought to you by [GitHub](#).

[template properties snippet](#) [view raw](#)

For groupId use the same as in the first pom.xml. In my case it would be com.liferay.sample. For artifactId I chose sample-portlet as this is the directory it will create. Version should be the same as the project parent. Once you have confirmed the values maven will create the portlet project and add it to you parent project as module automatically.

Now you project structure should be something like this:

```

pom.xml
sample-portlet
sample-portlet/pom.xml
sample-portlet/src
sample-portlet/src/main
sample-portlet/src/main/java
sample-portlet/src/main/resources
sample-portlet/src/main/webapp
sample-portlet/src/main/webapp/css
sample-portlet/src/main/webapp/css/main.css
sample-portlet/src/main/webapp/icon.png
sample-portlet/src/main/webapp/js
sample-portlet/src/main/webapp/js/main.js
sample-portlet/src/main/webapp/view.jsp
sample-portlet/src/main/webapp/WEB-INF
sample-portlet/src/main/webapp/WEB-INF/liferay-display.xml
sample-portlet/src/main/webapp/WEB-INF/liferay-plugin-package.properties
sample-portlet/src/main/webapp/WEB-INF/liferay-portlet.xml
sample-portlet/src/main/webapp/WEB-INF/portlet.xml
sample-portlet/src/main/webapp/WEB-INF/web.xml

```

[This Gist](#) brought to you by [GitHub](#).

[project file listing](#) [view raw](#)

4) Go to sample-portlet directory and run

```
mvn package
```

This will compile any classes and packages the portlet war file in target directory.

5) To deploy the portlet into your Liferay bundle you can run

```
mvn liferay:deploy
```

Now you have created your first Liferay plugin project with maven and deployed it to your Liferay bundle.

Trackback URL:

Tweet 7

Like 2

4

0



submit

[BlinkList](#)

[del.icio.us](#)

[Newsvine](#)

[Technorati](#)

2125 Views, 14 Comments

Showing 14 Comments



Jens Meinecke

2/1/12 11:23 PM

Thanks Mika! I will give it a try!



Jonas Yuan

2/2/12 8:54 AM

Nice! Thanks Mika!



Bijan Vakili

2/2/12 2:06 PM

Thanks!



Igor Beslic

2/3/12 12:31 AM

Perfect!



Saswat Sahoo

2/6/12 4:44 AM