**Chapter 2. Using the library with Vaadin7**

You have to make the steps as follows:

1. Download the latest version of our library Freecode Charts and D3 Wrapper
2. Create your Vaadin 7 project and place our library into your project
3. Configure your widgetset.gwt.xml file and at least once run the Vaadin widget compilation (which also includes updating the web.xml if it is necessary).
4. In your code, create an instance (or many instances) of ua.net.freecode.chart.Chart component, set its properties, and add this component to Vaadin panel, layout or window.
5. Set up the colors in your custom css file.
6. Write the other code of your application and run the application

In this chapter we will discuss in detail steps 1,2 and 3. The steps 4 and 5 are discussed in the next chapters. Step 6 is up to you.

1. *Download the latest version of our library Freecode Charts and D3 Wrapper*

You can find the latest version of our library at <https://vaadin.com/directory>

We are going to provide new versions with new features, so you can verify the site from time to time.

1. *Create your Vaadin 7 project and place our library into your project*

Upon creating Vaadin 7 project, copy the following files from our library to your project as follows:

1. Copy chartlib/libforvaadin7/fchart\_1.0.jar to <YOUR PROJECT>/WebContent/WEB-INF/lib/

Copy chartlib/libforvaadin7/fchart\_1.0.src.jar to <YOUR PROJECT>/WebContent/WEB-INF/lib/

If you use Eclipse, left-click on the project -> F5(refresh), right-click on the project ->Properties -> Java Build Path -> Libraries -> Add Jars -> <YOUR PROJECT>/WebContent/WEB-INF/lib/fchart\_1.0.jar,

Right-click on the project ->Properties -> Java Build Path -> Libraries -> Add Jars -> <YOUR PROJECT>/WebContent/WEB-INF/lib/fchart\_1.0.src.jar

In Order and Export (here in Java Build Path), check the fchart\_1.0.jar and leave unchecked fchart\_1.0.src.jar.

Attention: fchart\_1.0.jar should be used for both compilation and exporting to the final jar or ear or war.

Fchart\_1.0.src.jar should be used for widget compilation and as a source code, but it is not necessary to include this file into the final jar, ear or war (or export it as it is declared in Eclipse).

1. *Configure your widgetset.gwt.xml file and at least once run the Vaadin widget compilation.*

When you created the Vaadin7 application, the file which name ended with widgetset.gwt.xml is created automatically, but if it is not so, create a file in some of your source folders.

For example, we can decide to create this file in src/ua/net/freecode/vaadin7/widgetset and call it MyWidgetset.gwt.xml. (If you have no own widget-level code, it does not matter where you place this file (provided that it is in the source folder) and does not matter how you call this file (provided that it ends with Widgetset.gwt.xml).

The content of this file should be as follows:

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

<!DOCTYPE module PUBLIC "-//Google Inc.//DTD Google Web Toolkit 1.7.0//EN" "http://google-web-toolkit.googlecode.com/svn/tags/1.7.0/distro-source/core/src/gwt-module.dtd">

<module>

<inherits name=*"com.vaadin.terminal.gwt.DefaultWidgetSet"* />

<inherits name=*"ua.net.freecode.chart.widgetset.FreecodeWidgetset"* />

</module>

If you have other widget-level libraries, they also should be referenced here.

Now first compile your Java code in eclipse as usual, make sure that there is no errors in your code during the compilation. If you have internal validation error, please see our Trouble-Shooting Guide. If it is ok, in Eclipse click the button Compile Widgets. It is necessary to compile it only once if you do not add other libraries. Also it will be necessary to recompile it again if you receive an update of our library in the future.

It is very important to make changes in your main UI class as follows:

@VaadinServletConfiguration(productionMode = **false**, ui = ChartUI.**class**,

widgetset="ua.net.freecode.vaadin7.demo.widgetset.MyWidgetset")

where the widgetset is your own reference to the newly created widgetset file (you should replace it with your own path).

The next step is the use of the Chart component in your code. It will be discussed in chapter 3. In Chapter 4 it will be discussed how to manage chart colors.