

```
*****
# Setting up Eclipse IDE for Vaadin 7
# Tutorial based on
https://github.com/maclomork/Vaadin/tree/master/vaadin/README.md
*****
```

I. Prerequisites

```
*****
Java;
Nodejs;
Maven;
Eclipse EE;
Vaadin library: clone https://github.com/maclomork/Vaadin

# I think the basic Eclipse Plugins are:
# Vaadin, GWT, eGit,
# Subclipse + m2e.connectors,

# However, there are so many plugins and features installed that I cannot
# remember all. The process described in the following should actually
# capture them all. If you encounter any dependency problems you can
# check the Install-Overview I posted at:
# https://github.com/maclomork/Vaadin/blob/master/Installed-Eclipse-
Packages-and-Plugins%20-%20Tabellenblatt1.pdf
*****
```

II. Setting up Develop Environment

```
*****
1. Git clone the Vaadin repo and move to where you want to have your
   working set

2. Start Eclipse and choose the 'Vaadin' folder as working space

3. Perform any installs/updates when prompted

   3.1. Verify that IvyDE is installed
       (http://www.apache.org/dist/ant/ivyde/updatesite)

       3.1.1. if not, select all, install, restart Eclipse

   3.2. Verify that Apache Ivy Ant Targets is installed

       3.2.1. Help --> Install New Software --> click 'what is
              already installed'
       3.2.2. Verify that the list includes Apache Ivy Ant Tasks

4. Set Eclipse preferences

   4.1. Eclipse --> (Window -->) Preferences --> General --> Workspace

       4.1.1. set: Text file encoding to UTF-8
       4.1.2. set: New text file line delimiter to Unix

   4.2. Eclipse --> (Window -->) Preferences --> General --> Workspace
       --> Linked Resources
```

- 4.2.1. Add a new Path Variable GWT_ROOT referring to the gwt folder containing the gwt project
- 4.3. Eclipse --> (Window -->) Preferences --> Java --> Build Path --> Classpath Variables
 - 4.3.1. add: GWT_TOOLS (to the gwt-tools folder containing the dependency jars ../Vaadin/gwt-tools)
 - 4.3.2. add: JDK_HOME (your jdk installation directory)
- 4.4. Eclipse --> (Window -->) Preferences --> Java --> Compiler
 - 4.4.1 set compliance level: 1.6
- 4.5. Eclipse --> (Window -->) Preferences --> XML --> XML Files --> Editor
 - 4.5.1. Line width: 72
 - 4.5.2. Format comments: true
 - 4.5.3. Join lines: true
 - 4.5.4. Insert whitespace before closing empty end-tags: true
 - 4.5.5. Indent-using spaces: true
 - 4.5.6. Indentation size: 4

III. Workflow

1. Start Eclipse
2. File -> Import -> General -> Existing Projects into Workspace
 - 2.1. Select workspace folder as root directory
 - 2.2. 'deselect all'
 - 2.3. select: 'gwt-dev' and 'gwt-user', click 'Finish'
3. File -> Import -> General -> Existing Projects into Workspace
 - 3.1. Select workspace folder as root directory
 - 3.2. 'deselect all'
 - 3.3. select: 'vaadin', click 'Finish'
 - 3.4. wait until Ivy resolves dependencies
4. Need to compile the vaadin widget set by executing the default target in build/ide.xml
 - 4.1. browse to build/ide.xml
 - 4.2. 'right-click' --> Run As --> Ant Build
5. Running UI test
 - 5.1. Browse to 'com.vaadin.launcher.DevelopmentServerLauncher'

5.2. Launch in debug mode (Eclipse) --> 'right click'
Debug As --> Java Application.

5.3. Built in Jetty server will launch (port 8888)
--> test UI classes

5.4. i.e.:
<http://localhost:8888/run/com.vaadin.tests.components.label.LabelModes>
(Add ?restartApplication to ensure)

5.5. If this does not work for you, configure a Tomcat Server v7

6. Running JUnit Test

6.1. go to 'gwt/' directory and run cmd: 'ant test'
--> testing with GWT JUnit

6.2. go to 'vaadin/' directory and run cmd: 'ant test'
--> testing with Vaadin JUnit

6.2.1. TestBench: required access/configured TestBench 2
cluster (in Vaadin)

7. Building Packages

7.1. build distribution files

7.1.1. 'gwt' project --> go to .gwt/ --> cmd: 'ant'

7.2. build elemental package

7.2.1. 'gwt' project --> go to .gwt/ --> cmd: 'ant elemental'

7.2.2. <!-- on Windows you must install GCC first -->
<!-- see instructions @ <http://mingw-w64.sourceforge.net/download.php> -->

7.3. Go to 'vaadin/' directory

7.3.1. unpack the previously built gwt jars, cmd: 'ant -f gwt-files.xml unpack.gwt'

7.3.2. build vaadin project, cmd: ant

Now you should be able to properly run Eclipse, create Vaadin/GWT projects and deploy them.