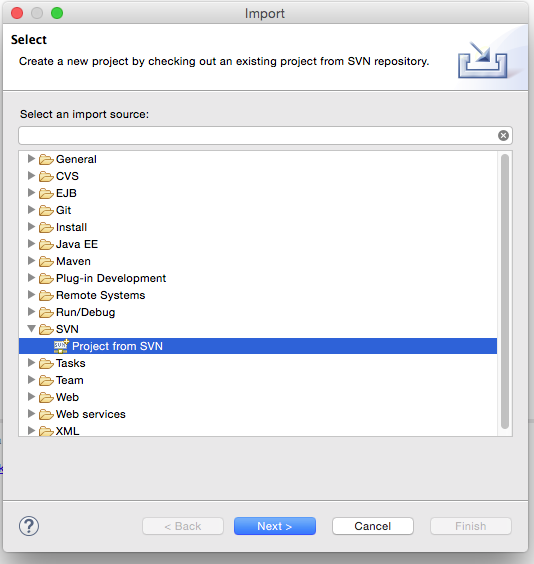
Setting Up a Mmowgli Project in Eclipse

(The most up-to-date version of this document is in the root of the SourceForge Mmowgli file tree.)

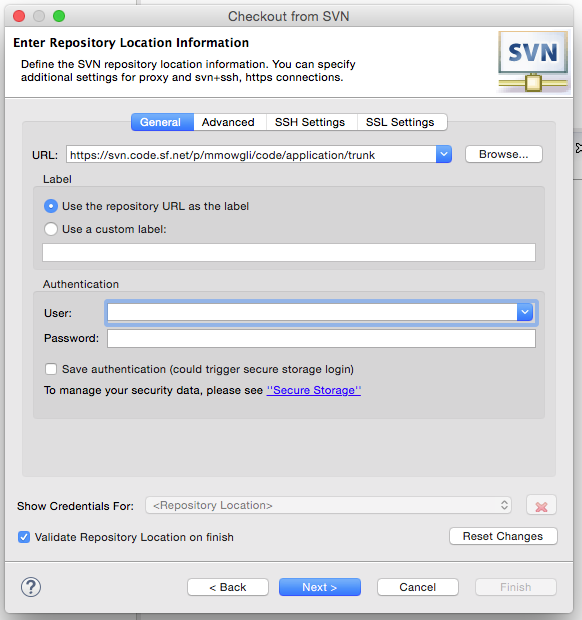
# Import mmowgli application from SourceForge

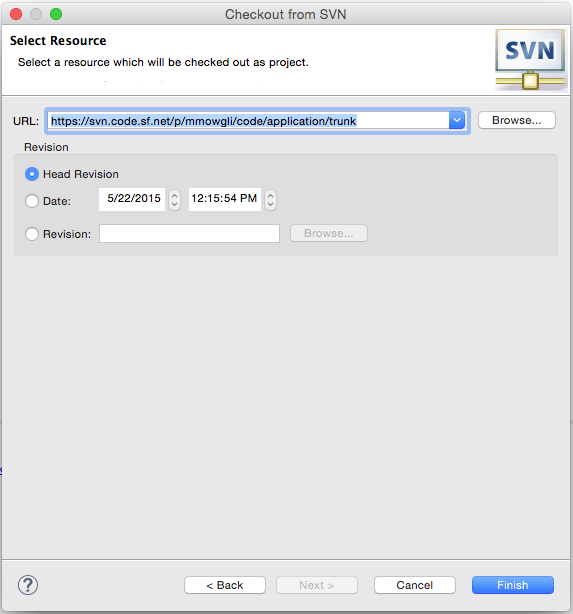
These instructions leave out the information about configuring the initial database.

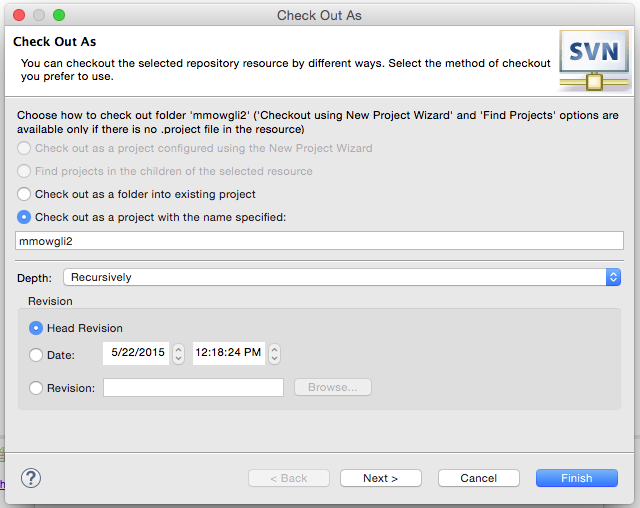
From the Eclipse File->Import… menu, choose to import a project from SVN.

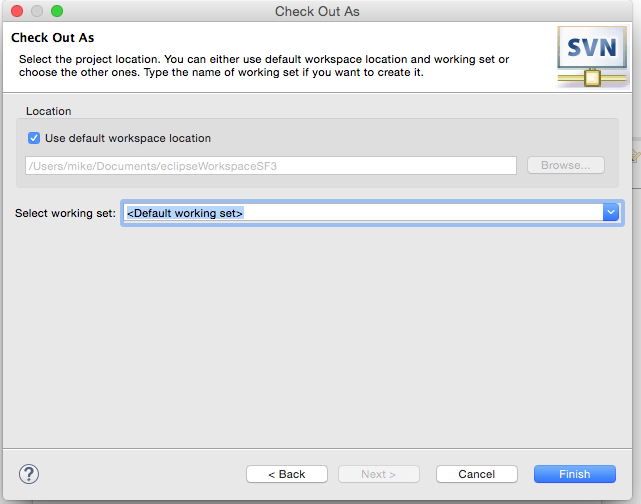


The repository URL is https://svn.code.sf.net/p/mmowgli/code/application/trunk





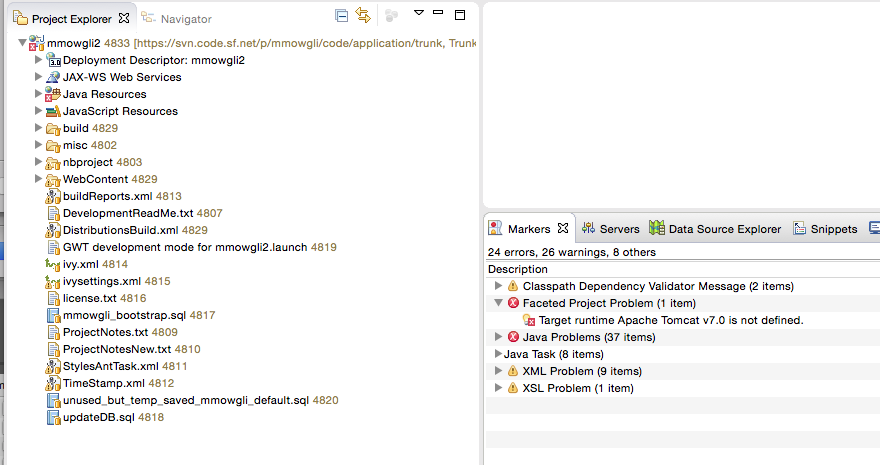




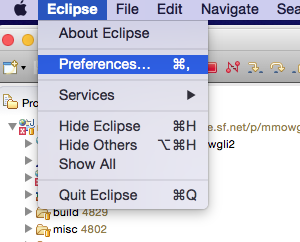
At this point, configure the WebContent/web.xml file with the proper information about your environment and database. Build the database and populate it with the mmowgli\_bootstrap.sql file found at the application project root. These steps are described in detail elsewhere.

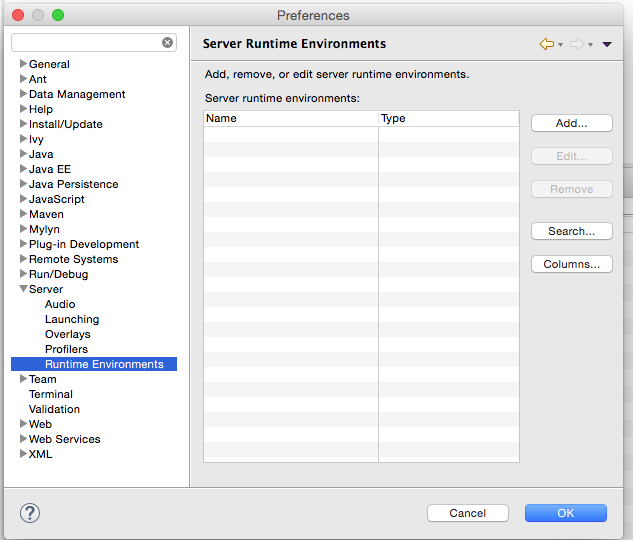
After successfully retrieving the application files, you have a project tree as below. Note the red badge meaning there are errors which must be resolved. Depending on the existing Eclipse configuration, these errors may involve:

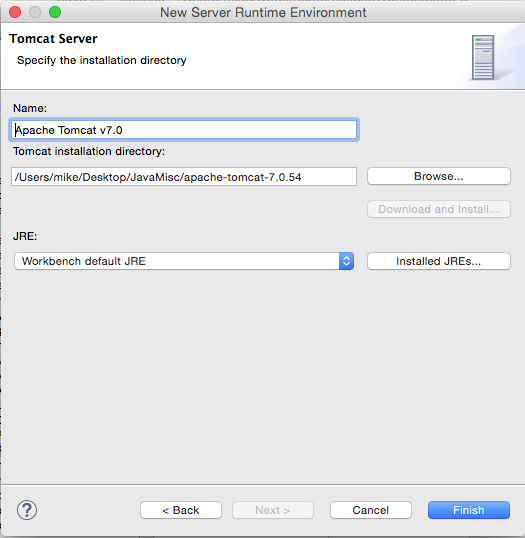
1. A missing Apache Tomcat server.
2. An alternately-named Java runtime environment



## To add the appropriate local server runtime to Eclipse:

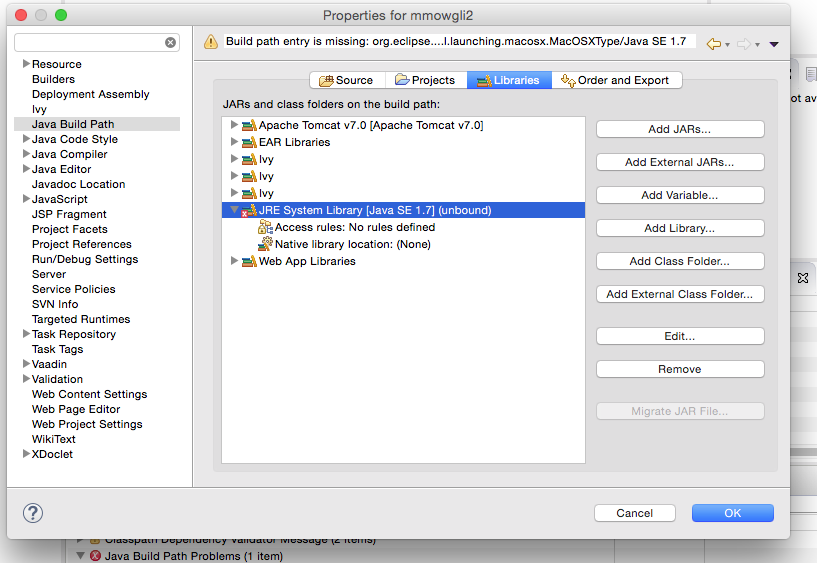


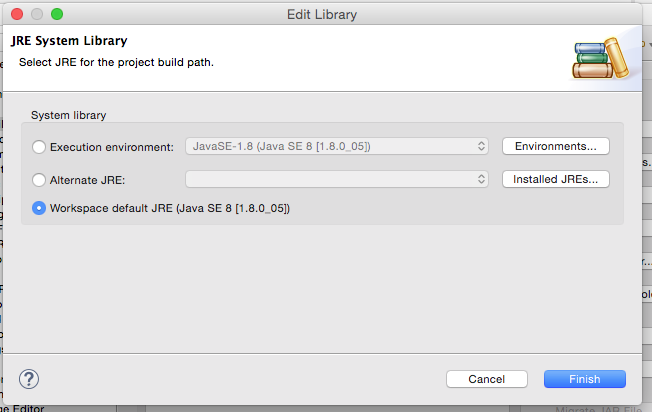




## Adjusting the Java Run-time environment

If you see an error in the project properties like the following, correct the java runtime environment by clicking the “Edit…” button.





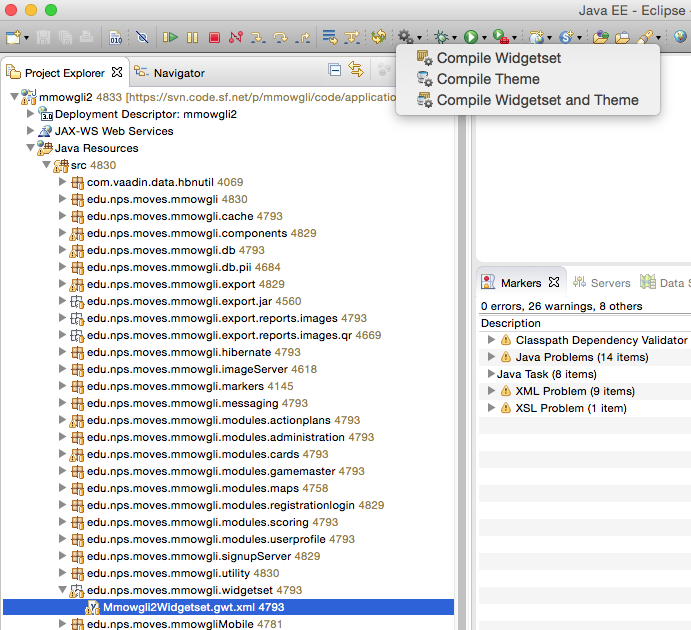
There should be no red badges in the Markers tab at this point.

Compile the Theme and Widget-Sets

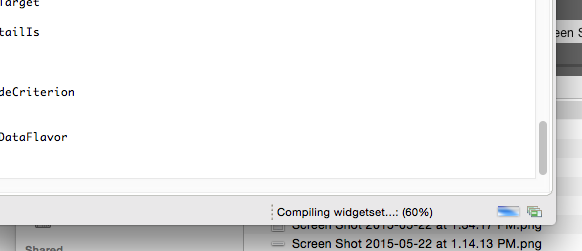
Mmowgli uses the Vaadin framework. Vaadin themes are written in the Sass stylesheet language and need to be compiled before use. Similarly, Vaadin used GUI widgets build from GWT objects, and this library also needs to be built.

Mmowgli has 2 widget sets: web application and mobile application. They are compiled from the tool bar “gear” icon. Compile the theme and app widgetset with one menu hit of “Compile Widgetset and Theme” after selecting the gwt.xml file, then “Compile Widgetset” after selecting the

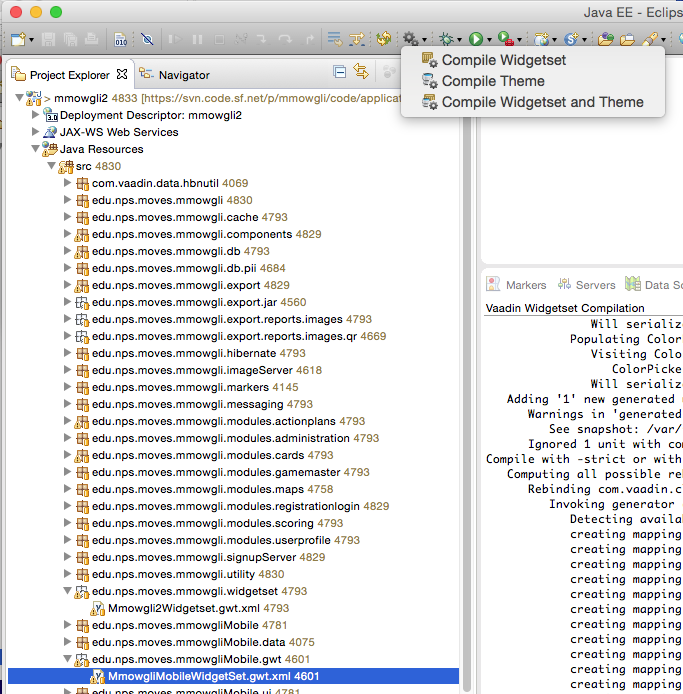
similar file in the mobile branch.



Watch the footer for completion of the task



And the mobile widget set:

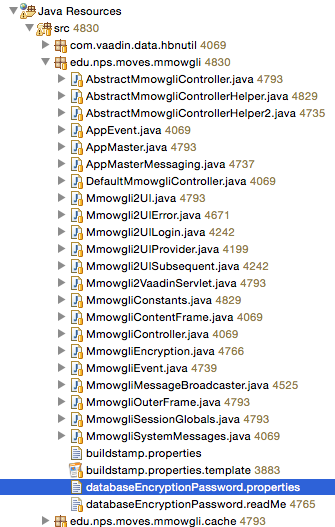


## Choose an Encryption Password

Read the file src/edu/nps/moves/mmowgli/databaseEncryptionPassword.readme and make a databaseEncryptionPassword.properties file right next to it.

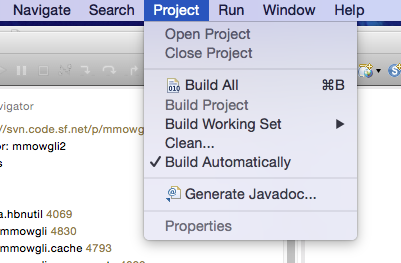
## ~~Create the Build Stamp File~~

~~Copy the file src/edu/nps/moves/mmowgli/buildstamp.properties.template and make a buildstamp.properties file right next to it. Select mmowgli2 in the navigator, right-click and choose “Refresh”. (wrong-build task does that…fix)~~

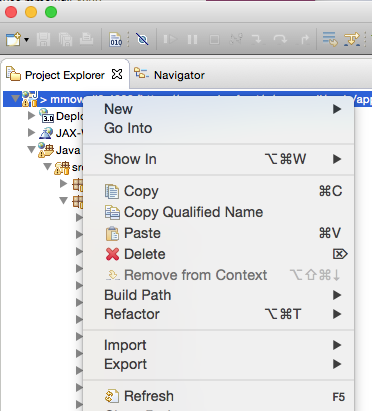


## Clean and Refresh

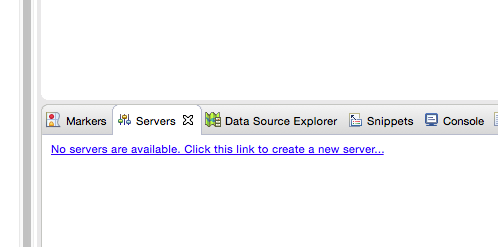
Clean (and rebuild) the application. This adds a build-stamp which is used to mark the time of each build. It will remain in place from this point on, so futher cleans are not required unless otherwise desired.

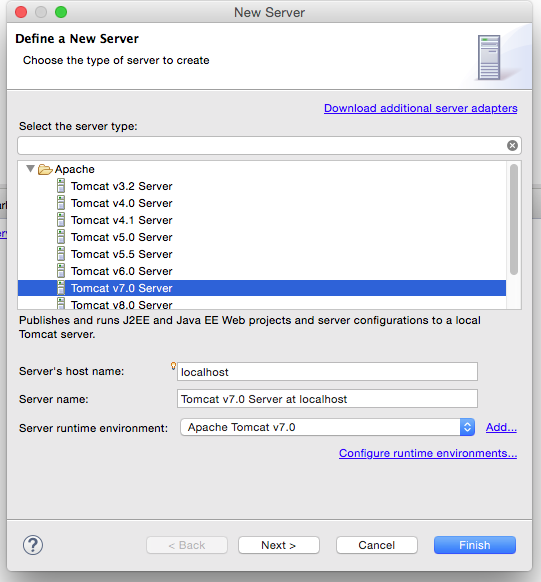


Refresh the project tree, so Eclipse knows about the new file.

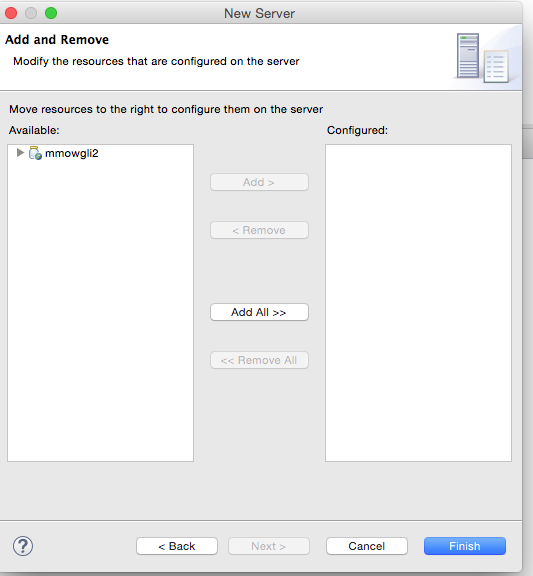


## Define the Local Server



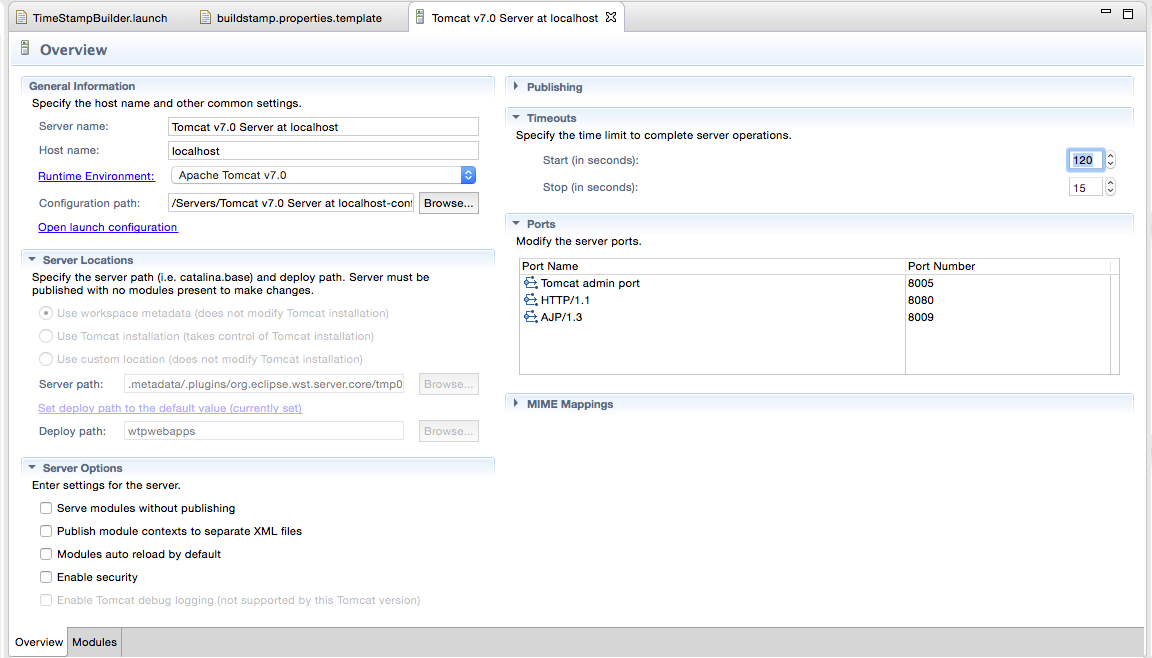


The mmowgli application needs to be installed on the server.



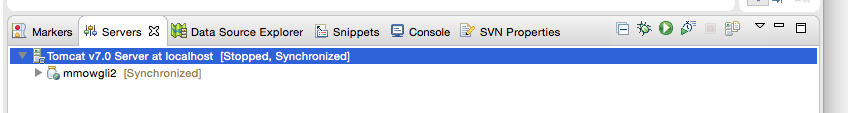
Two server setting may need to be adjusted, depending on the developer’s preference. Startup timeout may need to be bumped. The first Mmowgli developers use 120 seconds.

Server options to automatically reload and “serve without publishing” are turned off.



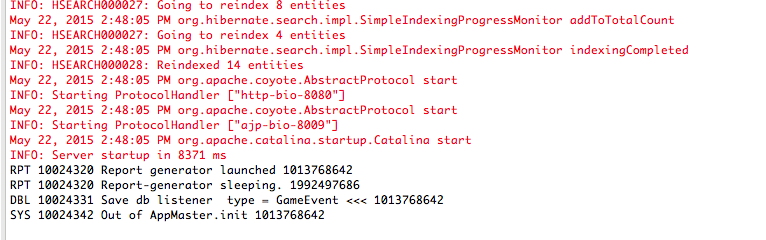
## Run the Application

Make sure all changes are published by clicking the “Publish to the Server” button at the top right.



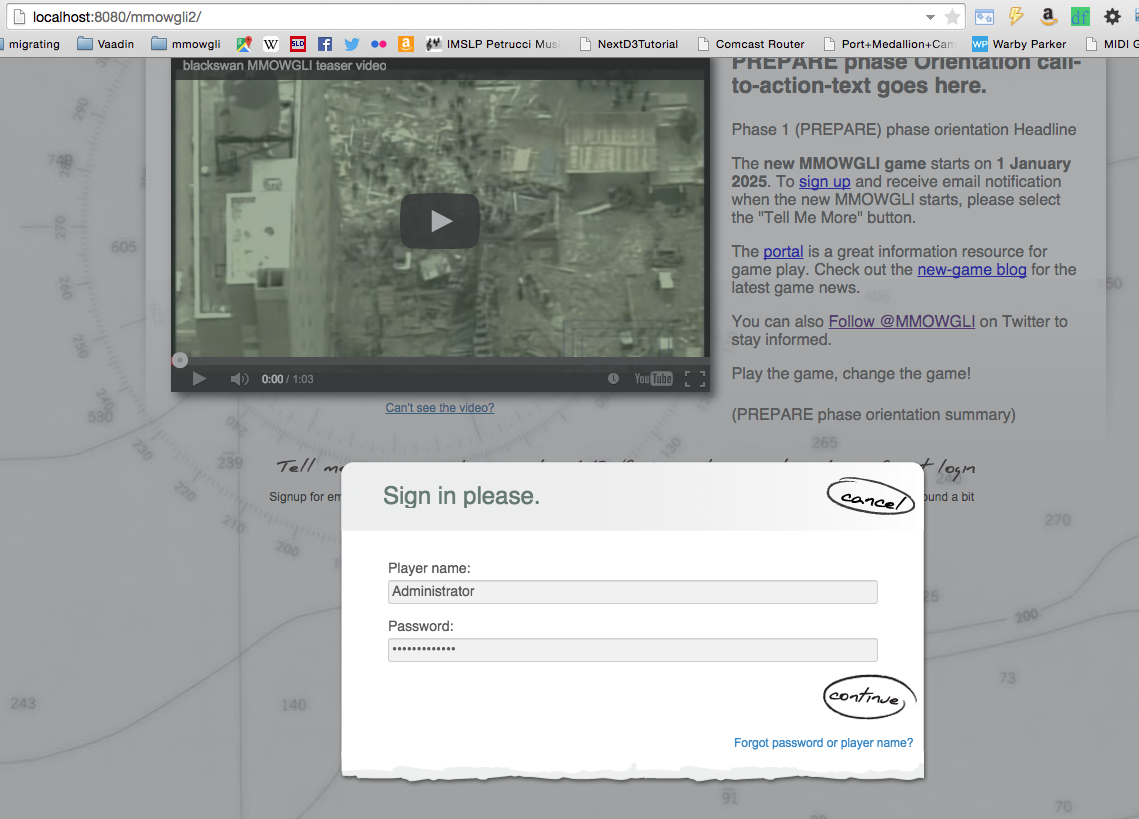
Launch the server, and thus the application, by clicking the green arrow.

Watch the Output tab for errors. If the game is successfully launched, you should see something like the following at the tail end of the console output.



## Log into Mmowgli

The default server port will be 8080 and the default application context is “mmowgli2”. Enter <http://localhost:8080/mmowgli2/> in your browser. A successfully install should show you something like the following.



Congratulations!

The first log on in a deployed game would be to Administrator/Administrator. There are 3 other initial users: GameMaster/GameMaster, GameBuilder/GameBuilder, and SeedCard/SeedCard. The Administrator would typically change the 4 passwords, logout, then create a regular account. Then logout of the regular account, and back in as Administrator/newpw. Go to the Game Administrator -> Player administration and give the just-made regular accounts the desired permissions.