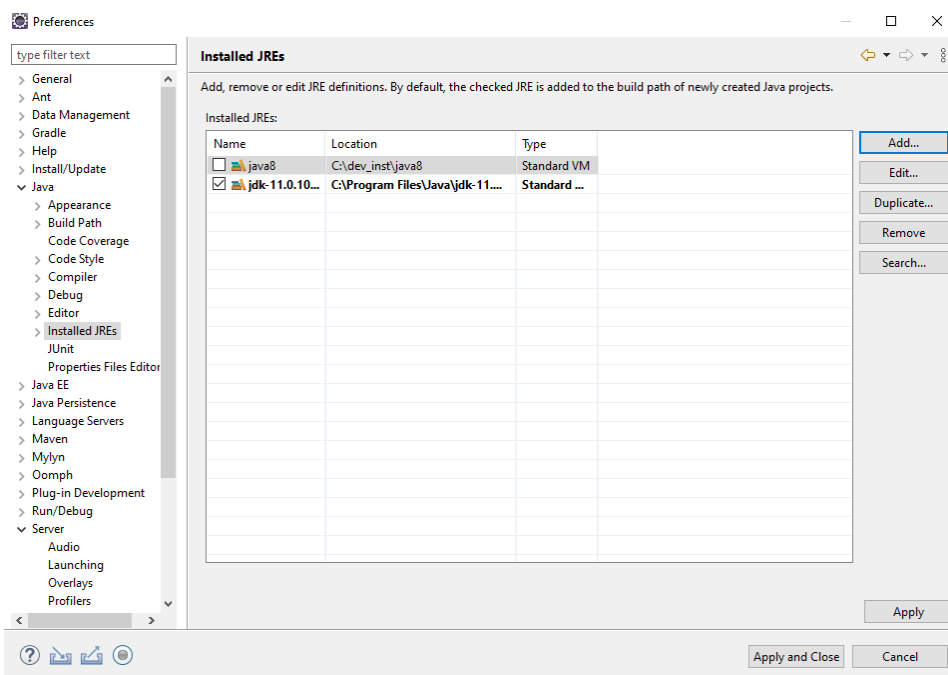


How to Setup a lite Grouper Development Environment for Grouper v2.5

[Wiki Home](#)[Download Grouper](#)[Grouper Guides](#)[Community Contributions](#)[Developer Resources](#)[Deployment Guide](#)

This is considered a "lite" dev env since we are not cloning git, or making pull requests. We just want to run grouper and make GSH templates or hooks or provisioners.

1. Install Java11
2. Install Java8
3. Install [eclipse](#), in installer select "Eclipse IDE for Enterprise Java and Web Developers", select the Java11 you just installed
 - a. Make sure eclipse ini has at least 3 gig memory
4. Add Java8 JRE



5. Make a new Maven project [without archetype](#)

New Maven Project

New Maven project

Configure project

Artifact

Group Id: test

Artifact Id: test

Version: 0.0.1-SNAPSHOT

Packaging: jar

Name: test

Description: test

Parent Project

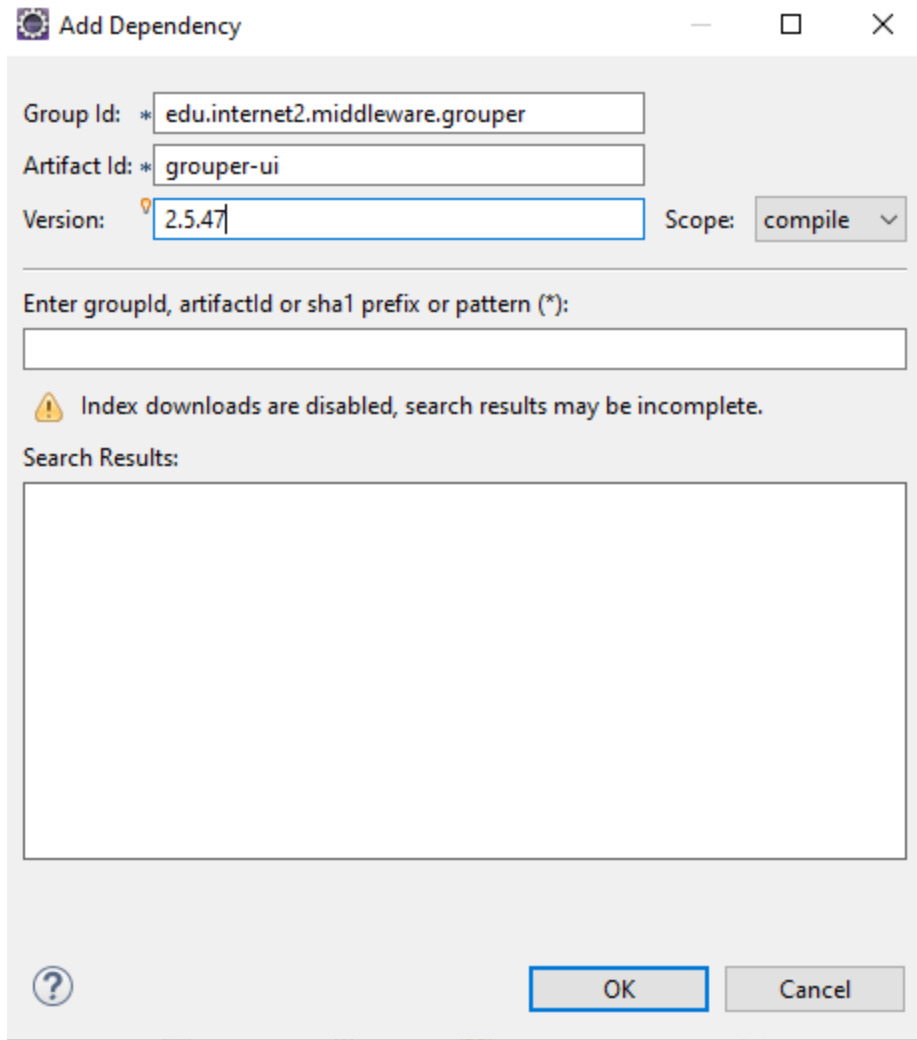
Group Id:

Artifact Id:

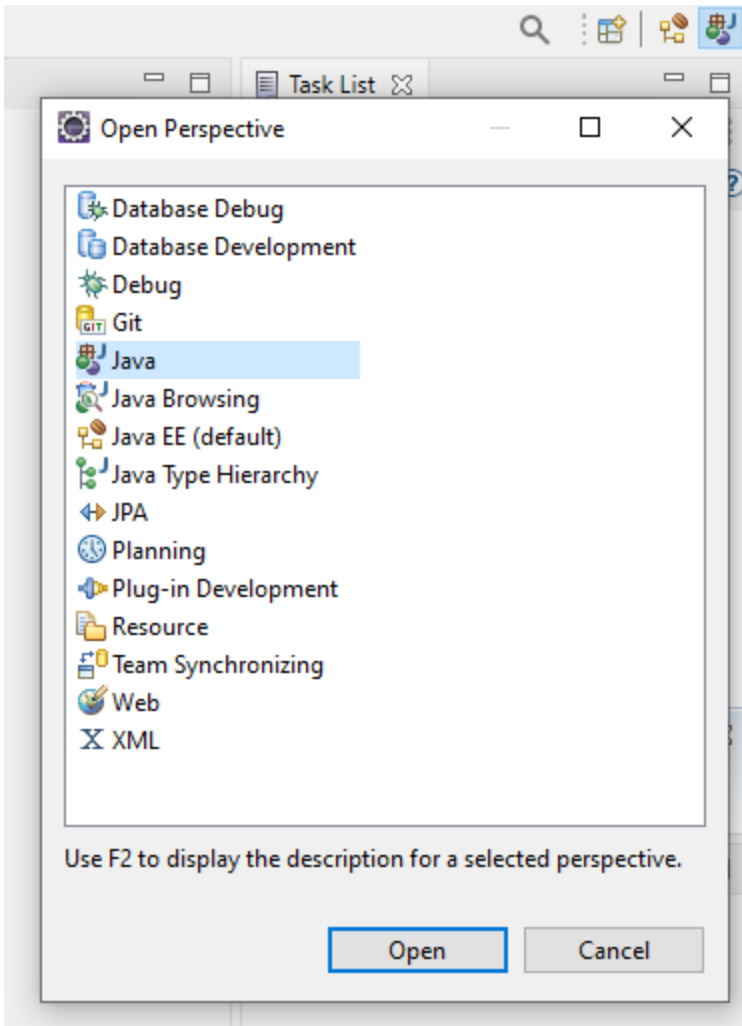
Version:

Advanced

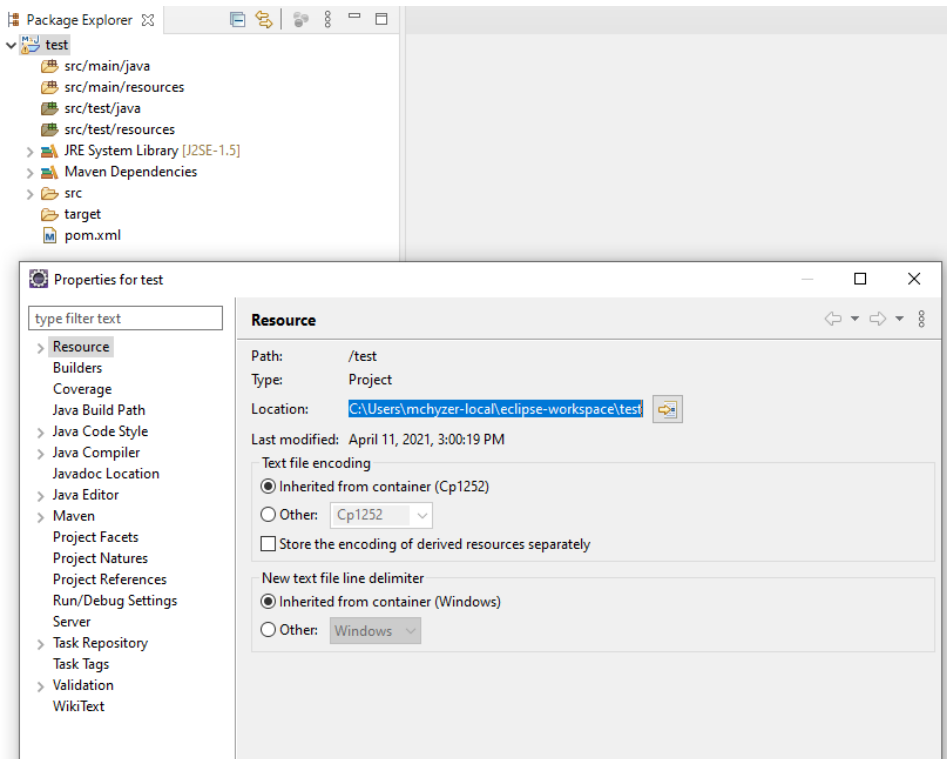
- 6. Right click on Project, Maven Add dependency
 - a. GroupId: edu.internet2.middleware.grouper
 - b. ArtifactId: grouper-ui
 - c. Version: 2.5.47



7. Right click on project, Maven, update project
8. I use the java perspective, so switch to that

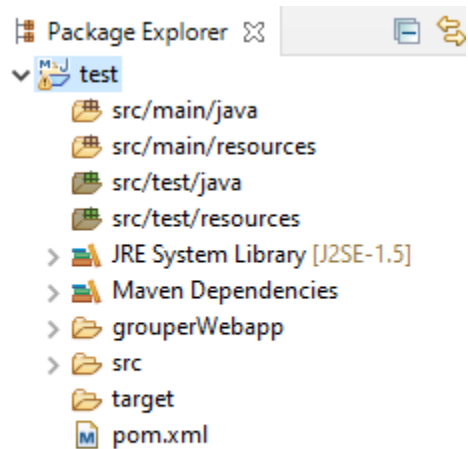


9. Get the webapp dir out of container

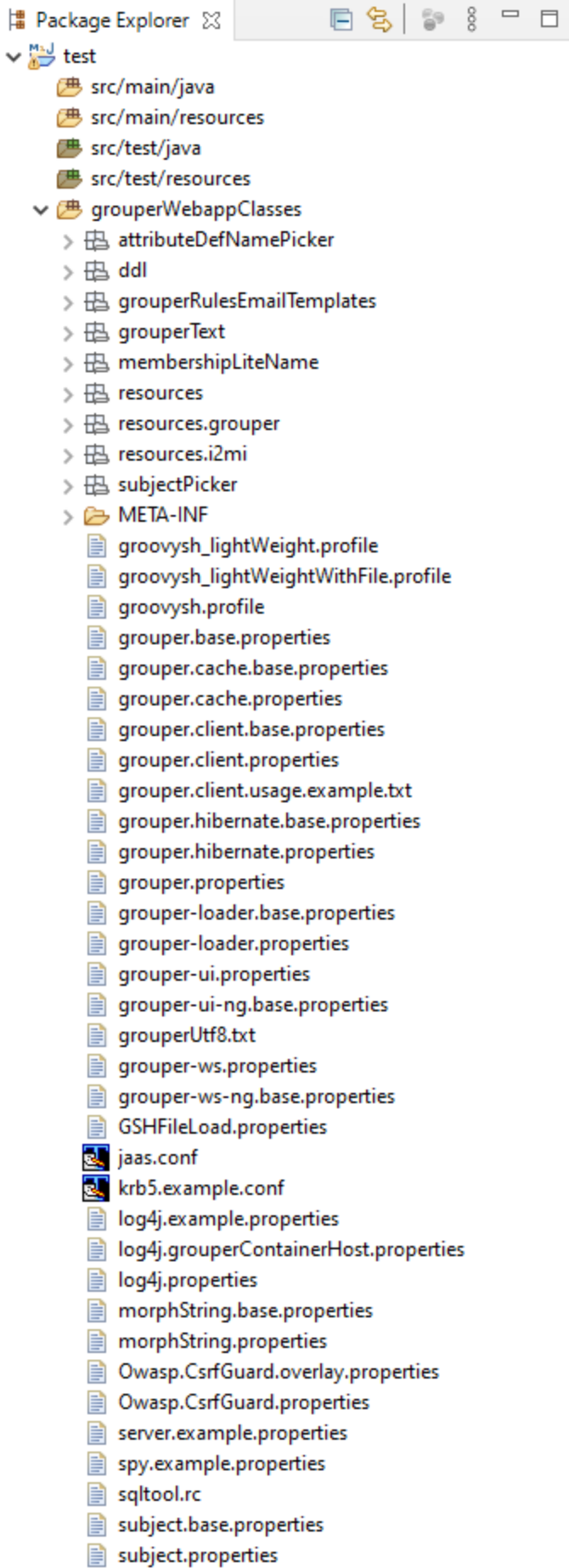


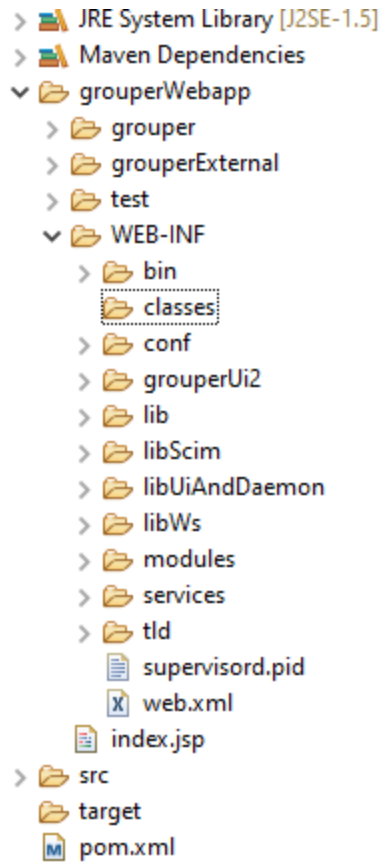
```
PS C:> cd C:\users\mchyzer-local\workspace\test
PS C:\users\mchyzer-local\workspace\test> docker run --detach -e GROUPER_LOG_TO_HOST=true --name
grouperFiles i2incommon/grouper:2.5.47 ui
62149d4d5f784949c635ba3ebc4276fb91b11e2bc39fe77d2ce7100f4780f405
PS C:\users\mchyzer-local\workspace\test> docker cp grouperFiles:/opt/grouper/grouperWebapp .
PS C:\users\mchyzer-local\workspace\test> docker rm -f grouperFiles
grouperFiles
PS C:\users\mchyzer-local\workspace\test>
```

10. Refresh your eclipse project and see grouperWebapp

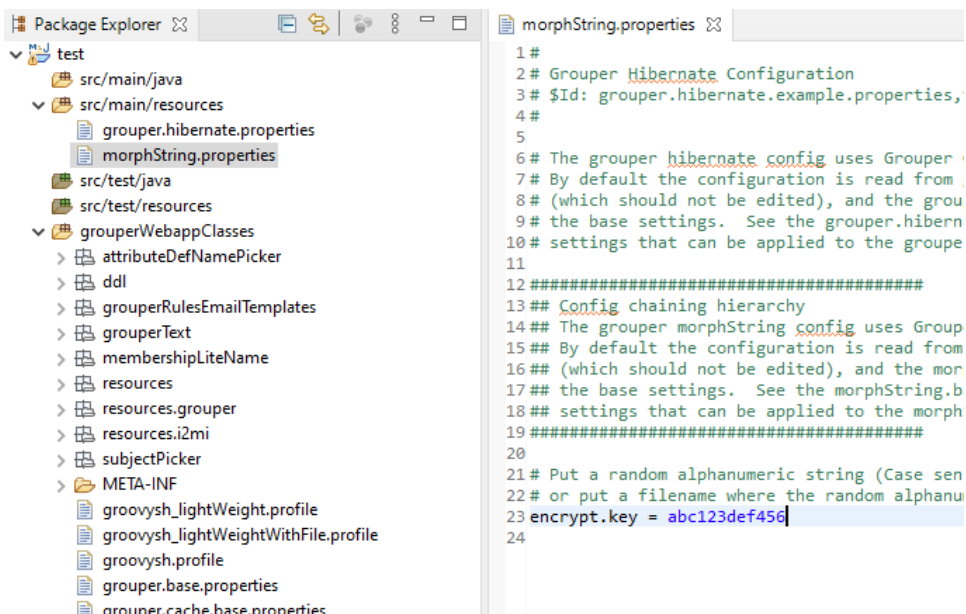


11. Lets move the existing grouperWebapp/WEB-INF/classes dir contents into a new source folder named grouperWebappClasses

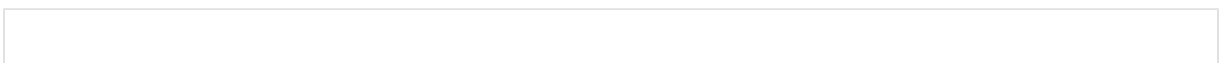




12. Move any properties files from there to the resources folder and customize. Note if you are connecting to an existing database, the morphString secret must match



13. If you are using a new database
 - a. Start hsql

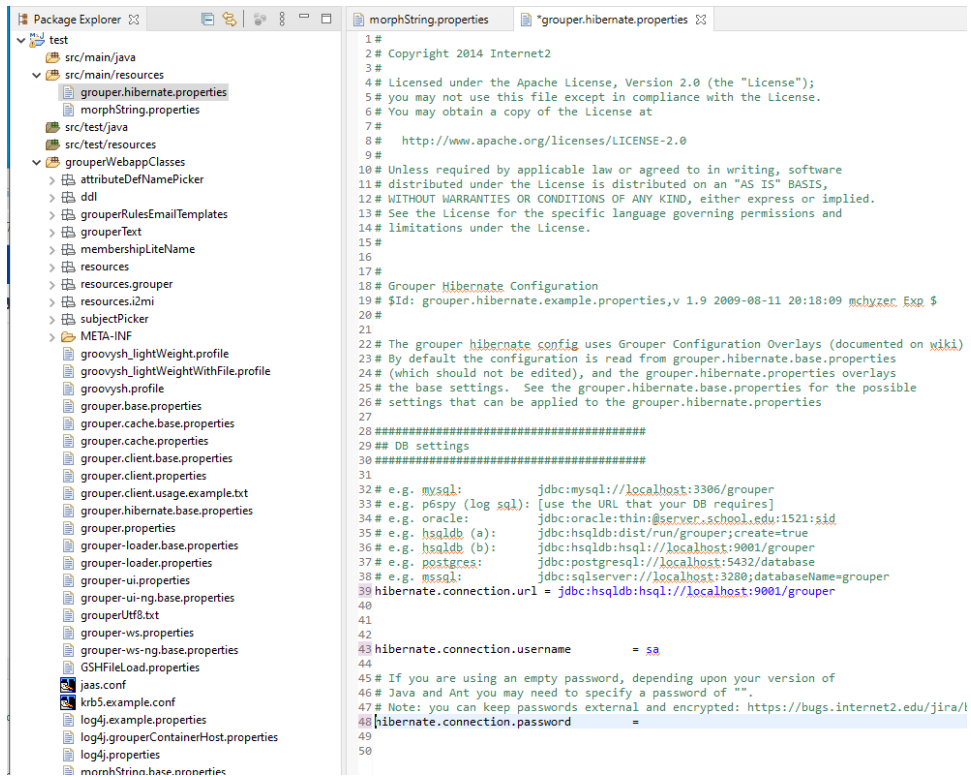



```

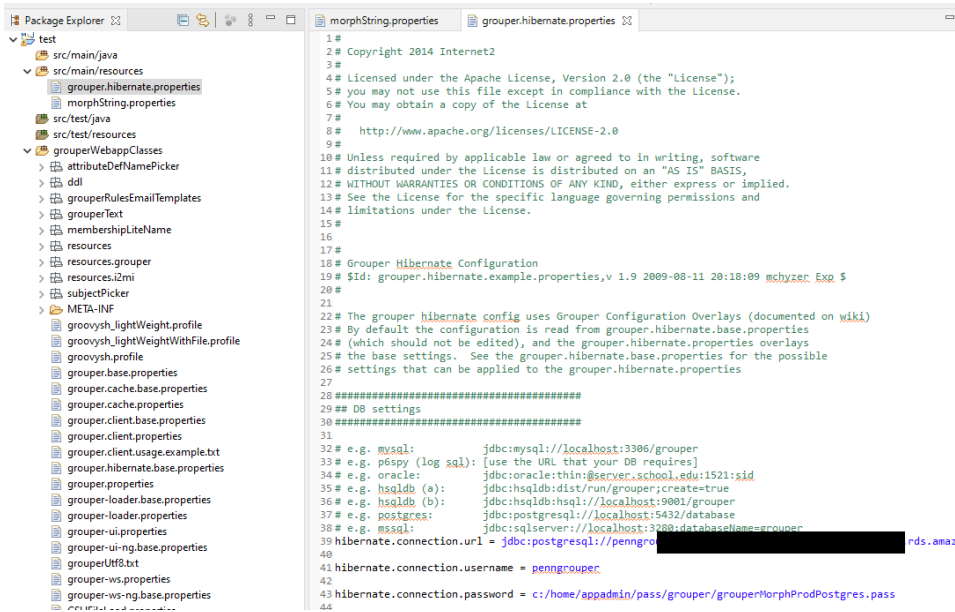
PS C:\users\mchzyer-local\eclipse-workspace\test> mkdir temp
PS C:\users\mchzyer-local\eclipse-workspace\test> cd temp
PS C:\users\mchzyer-local\eclipse-workspace\test> java -cp .\grouperWebapp\WEB-INF\lib\hsqldb-2.3.5.jar org.hsqldb.Server -port 9001 -database file:temp/grouper

```

b. grouper.hibernate.properties (moved from grouperWebappClasses)



14. If you are using an existing database



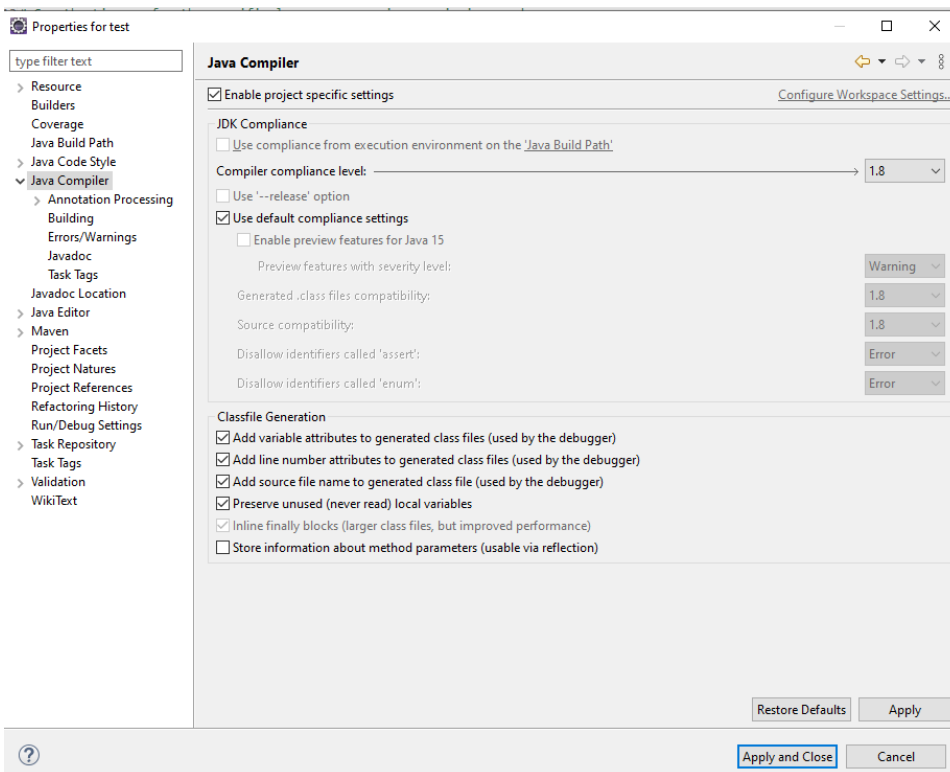
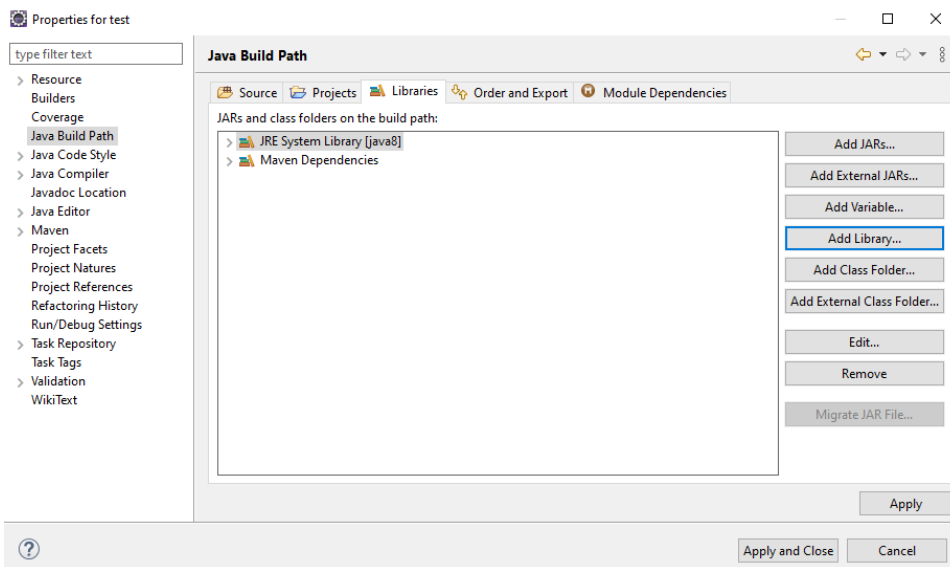
15. Add in to grouper.hibernate.properties that it is ui, and put in a local pass for a subject (remote database) or GrouperSystem (hsqldb database or remote)

```
grouper.is.ui = true
```

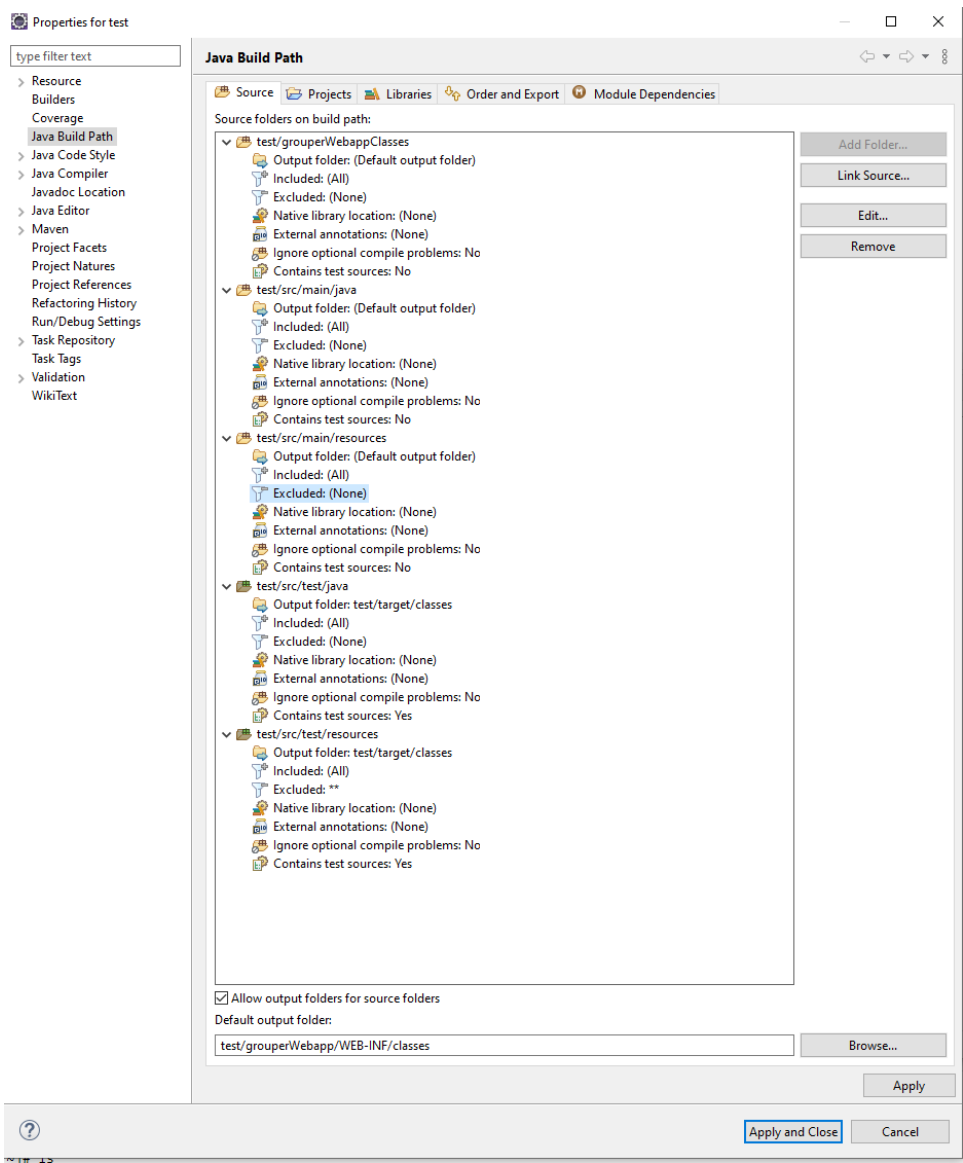
```
# UI basic auth is for quick start. Set to false when you migrate to shib or something else
grouper.is.ui.basicAuthn = true

grouperPasswordConfigOverride_UI_mchzyer_pass = pass
grouperPasswordConfigOverride_UI_GrouperSystem_pass = pass
```

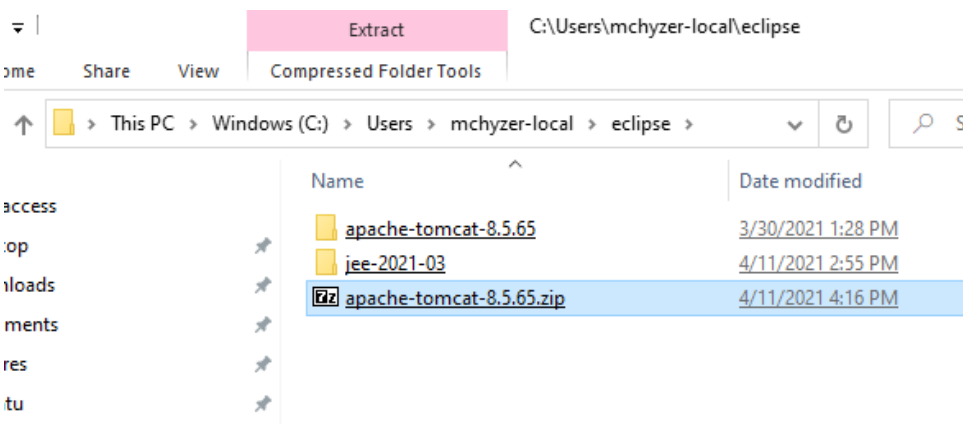
16. Set java8 for project



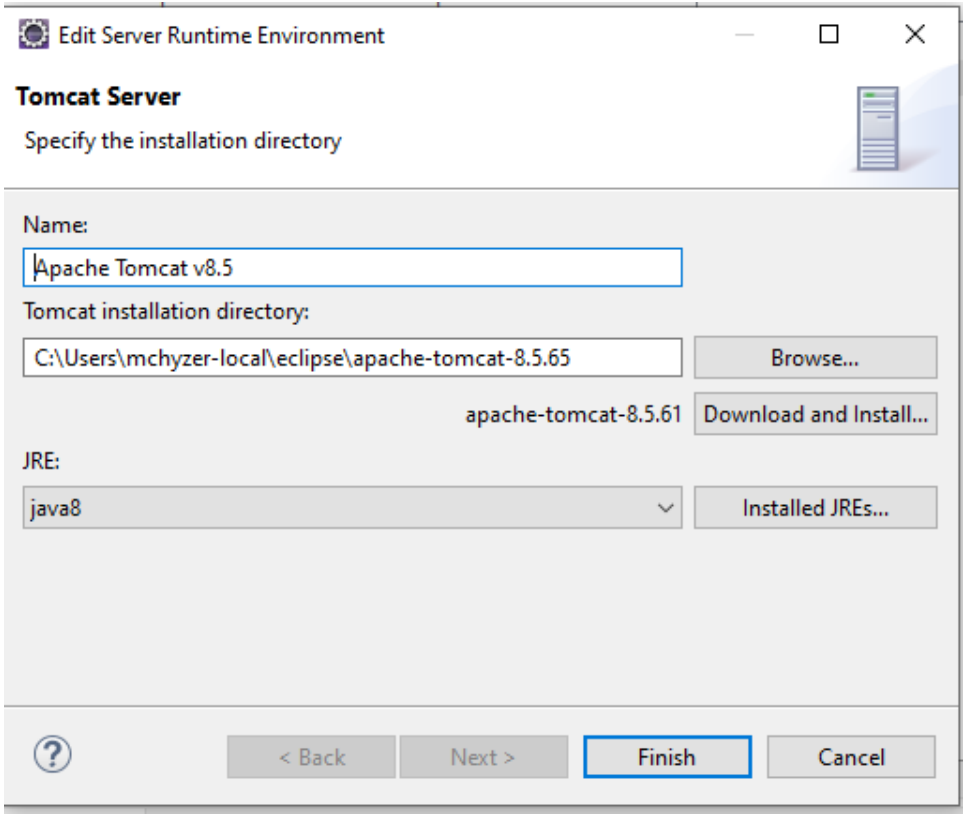
17. Change build path to compile to grouperWebapp/WEB-INF/classes (except the test source and test resources). Also dont exclude any resources. Note, in future if you do Maven Update project, you might have reset some of these settings



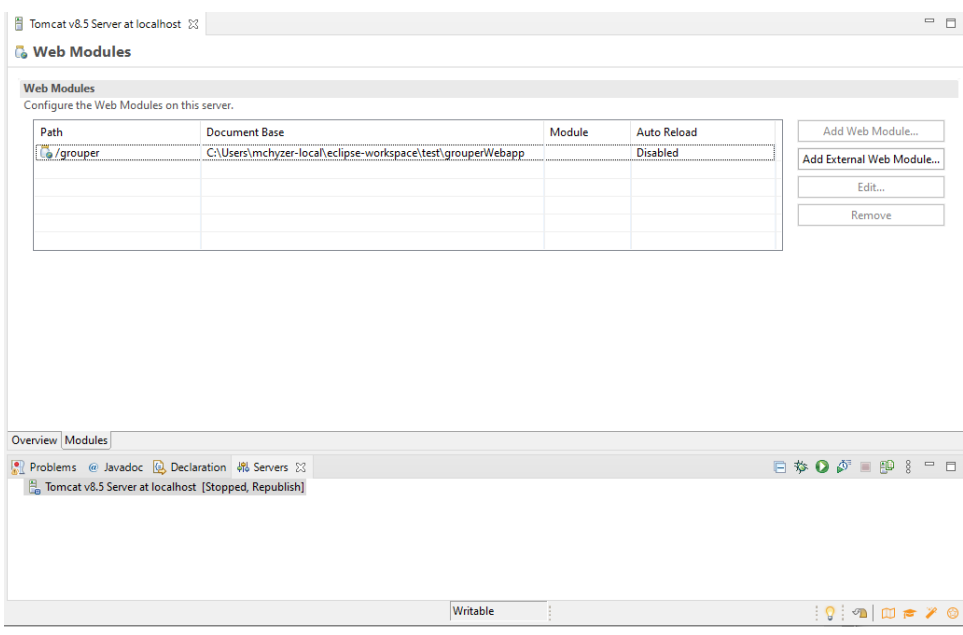
18. Download tomcat 8.5



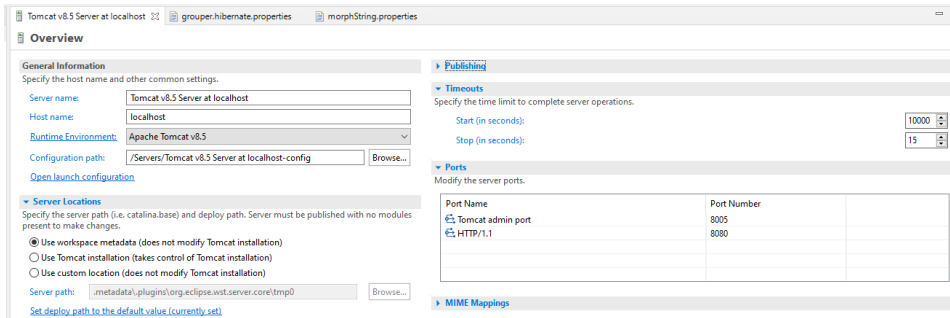
19. Add a new server in eclipse



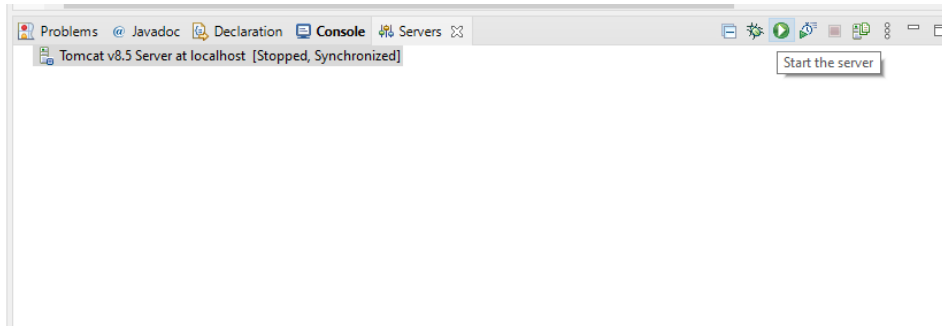
20. Window Show view Servers, add a module



21. Make a large startup timeout



22. Start the server



23. Go to local grouper: <http://localhost:8080/grouper> (userName/pass) or whatever you put in grouper.hibernate.properties

Steps to move lite env to new container version

1. Close eclipse
NOTE: This is important so that Eclipse can get a consistent state after the following changes!
2. cleanup dynamic folders from older container content
cdtest. (Be in the Eclipse project's root directory)
rm -rf .grouperWebapp
rm -rf .grouperWebappClasses
3. Get new container to copy files from (Note reusing the same container name as before. You may need to 'docker rm -f grouperFiles' first.)
docker run --detach -e GROUPER_LOG_TO_HOST=true --name grouperFiles i2incommon/grouper:2.5.NN ui
4. Copy the new container's files to the local filesystem
cdtest. (Be in the Eclipse project's root directory)
docker cp grouperFiles:/opt/grouper/grouperWebapp .
5. Move the existing grouperWebapp/WEB-INF/classes dir contents into the local source folder named grouperWebappClasses
cdtest. (Be in the Eclipse project's root directory)
mv grouperWebapp/WEB-INF/classes/* grouperWebappClasses
6. Remove any "non base" config files from grouperWebappClasses (or anything your going to override with files in src/main/resources)
likely examples: grouper.hibernate.properties , morphString.properties, grouper-ui.properties, log4j.properties, etc....
7. Start Eclipse
8. select the project, right click and "refresh" (or press the "F5" key)
9. Do a "Project"(menu) "Clean..." (not maven clean!)