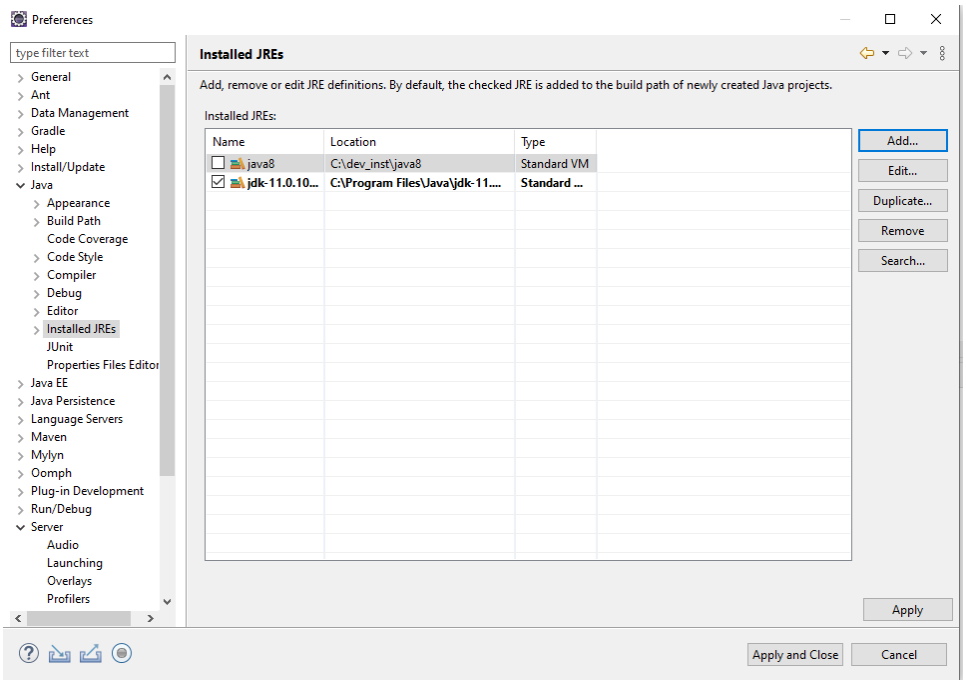
How to Setup a lite Grouper Development Environment for Grouper

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 or daemons.

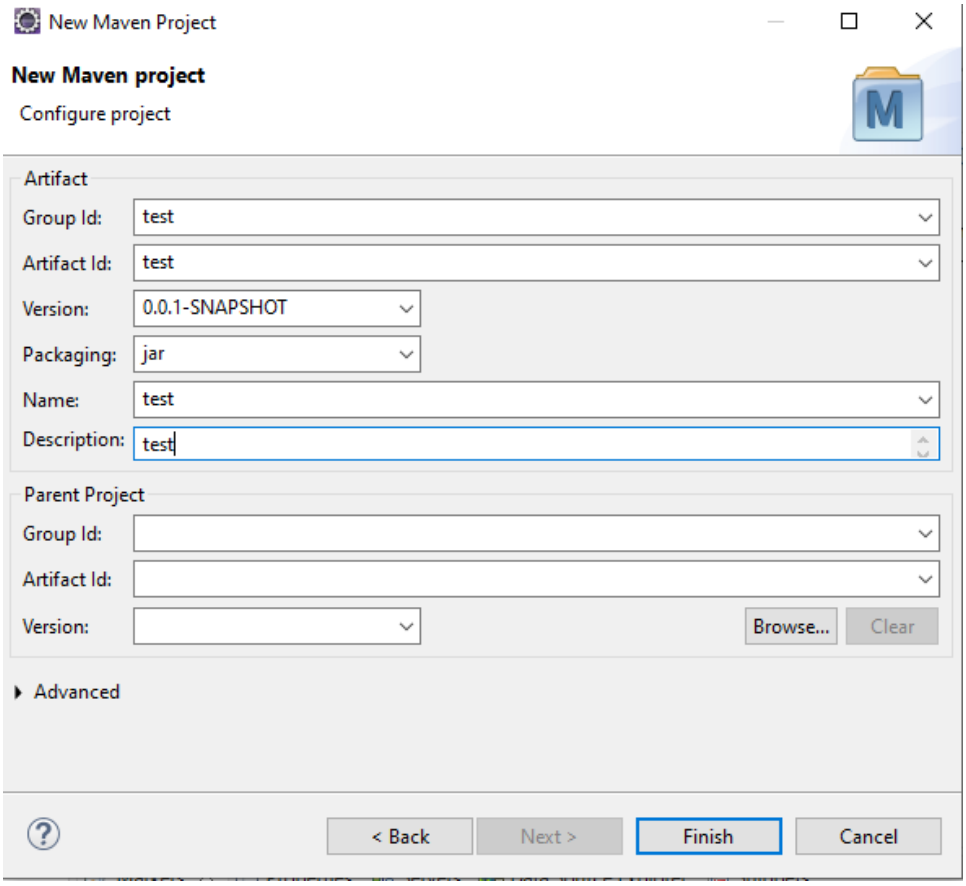
Note, if using Java 17, pass this argument to tests and tomcat

```
--add-opens java.base/java.lang=ALL-UNNAMED --add-opens java.base/java.util=ALL-UNNAMED --add-opens java.sql
/java.sql=ALL-UNNAMED
```

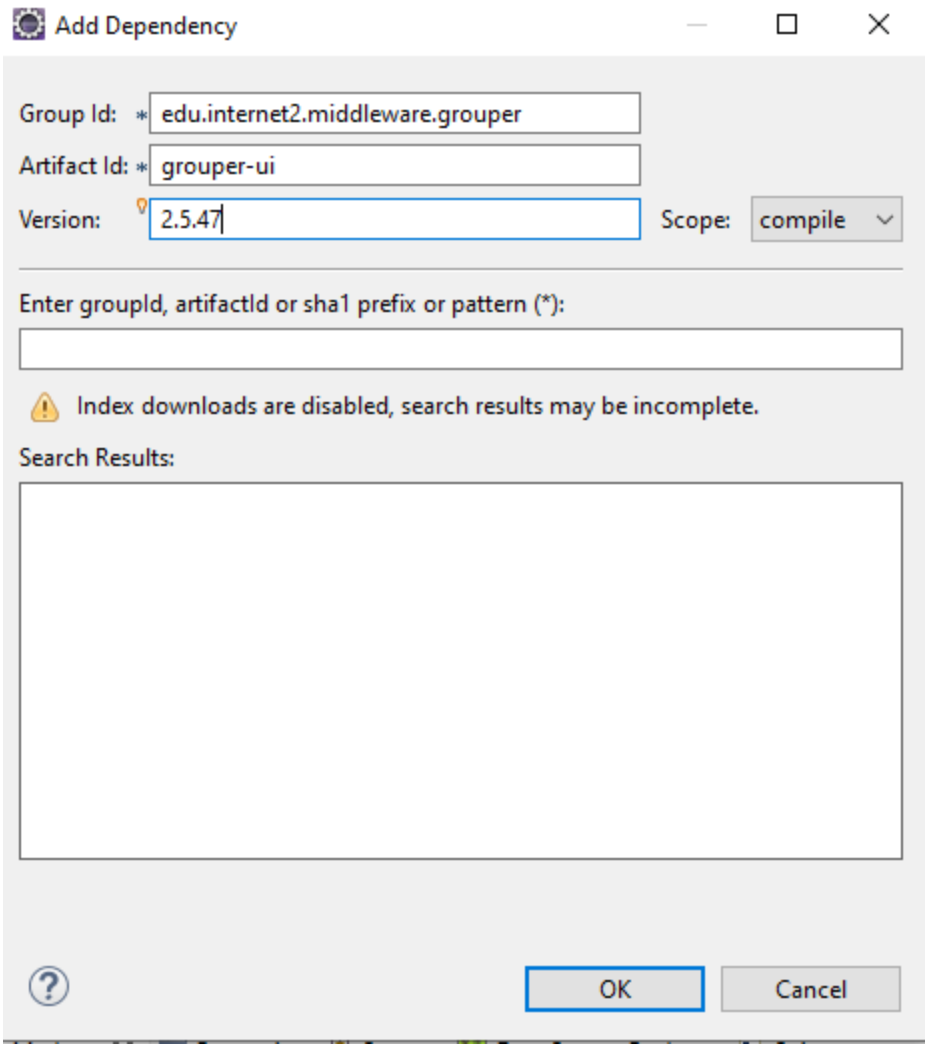
1. Install Java17 (or whatever version your container runs in)
2. Install eclipse, in installer select "Eclipse IDE for Enterprise Java and Web Developers", select the Java17 you just installed
    a. Make sure eclipse ini has at least 3 gig memory
3. Add Java17 JRE



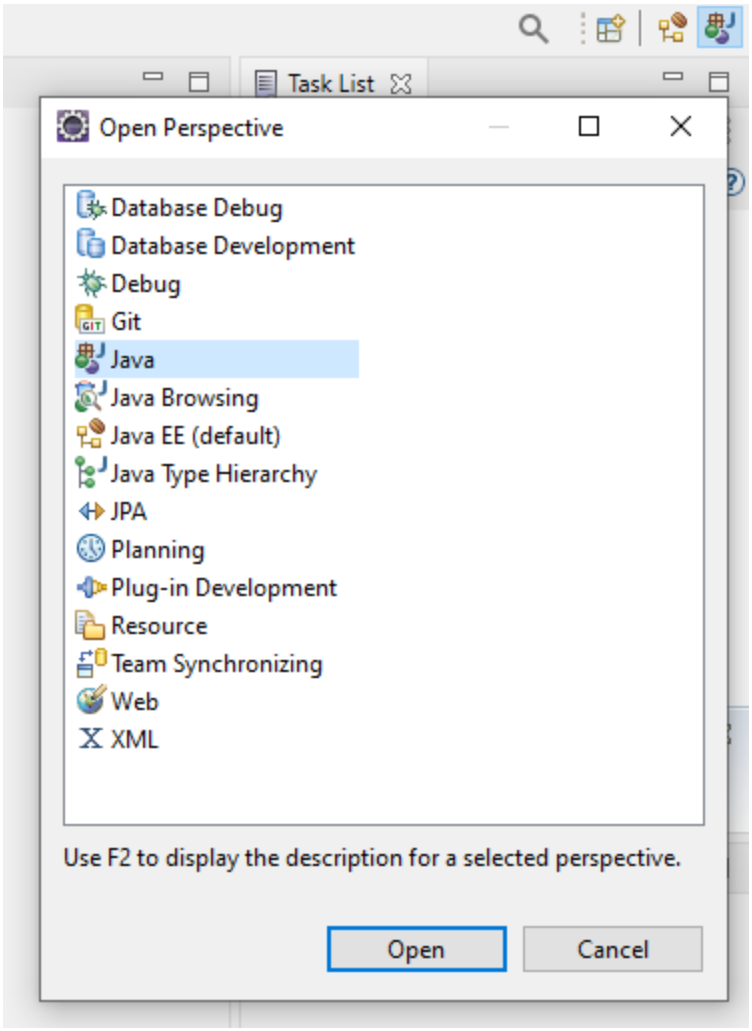
4. Make a new Maven project without archetype



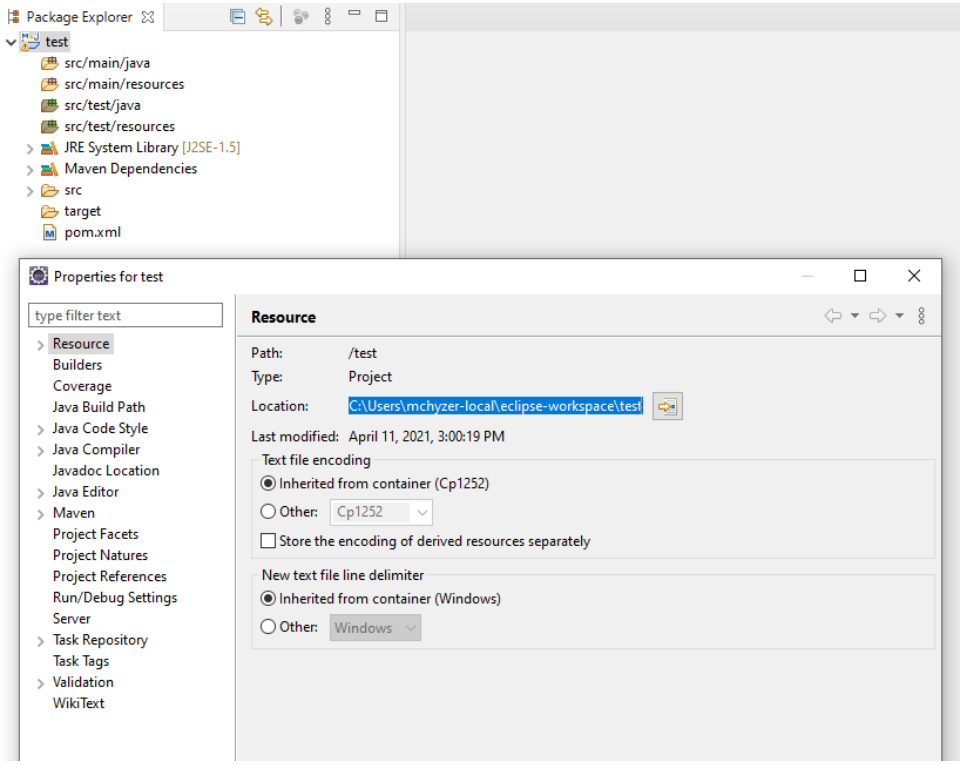
5. Right click on Project, Maven Add dependency
    a. GroupId: edu.internet2.middleware.grouper
    b. ArtifactId: grouper-ui
    c. Version: (whatever you use): e.g. 4.12.0



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

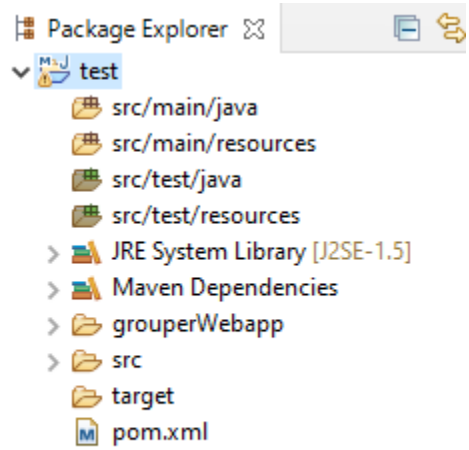

Task List
Open Perspective
Database Debug
Database Development
Debug
Git
Java
Java Browsing
Java EE (default)
Java Type Hierarchy
JPA
Planning
Plug-in Development
Resource
Team Synchronizing
Web
XML
Use F2 to display the description for a selected perspective.
Open
Cancel

8. You can stop here if just coding GSH. If coding against the UI, continue: Get the webapp dir out of container

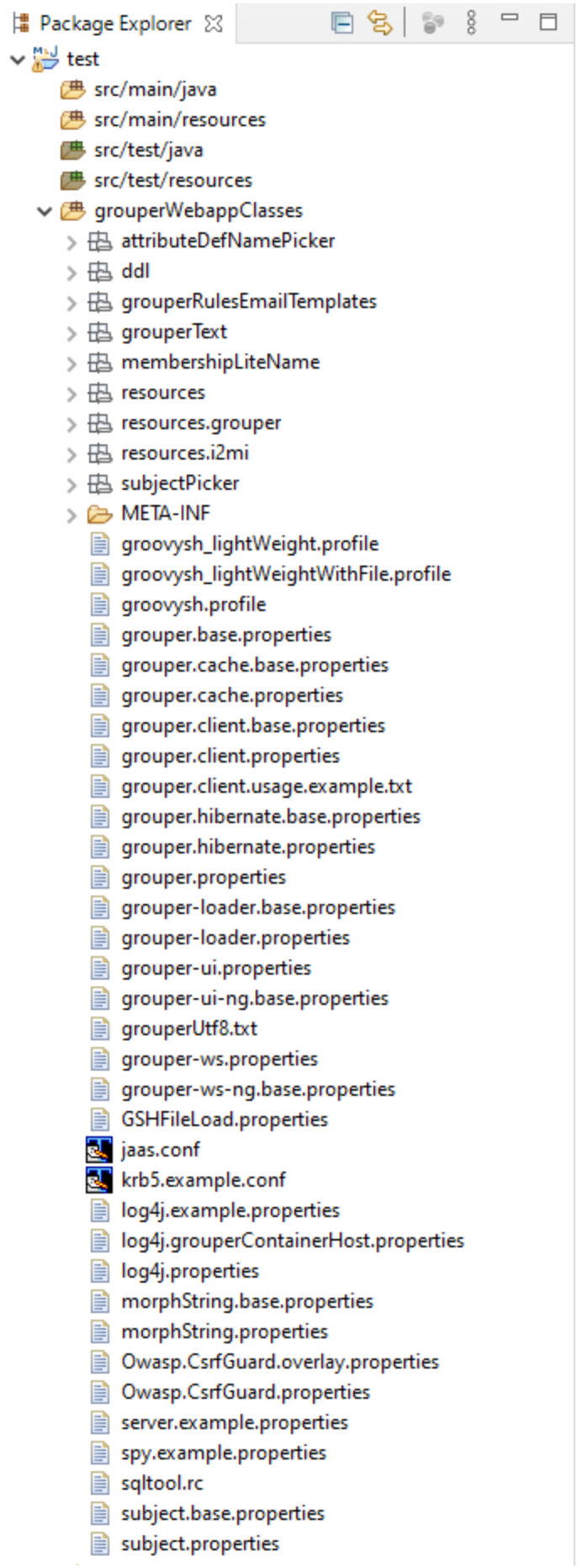


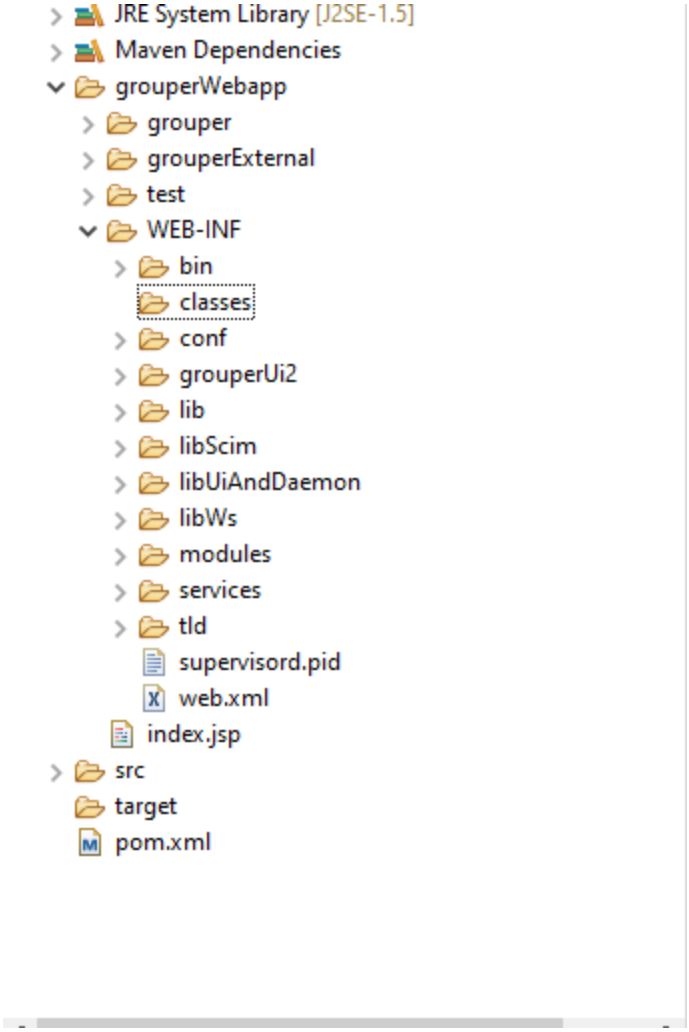

```
PS C:> cd C:\users\mchyzer-local\eclipse-workspace\test
PS C:\users\mchyzer-local\eclipse-workspace\test> docker run --detach -e GROUPER_LOG_TO_HOST=true --name
grouperFiles i2incommon/grouper:2.5.47 ui
62149d4d5f784949c635ba3ebc4276fb91b11e2bc39fe77d2ce7100f4780f405
PS C:\users\mchyzer-local\eclipse-workspace\test>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

9. Refresh your eclipse project and see grouperWebapp

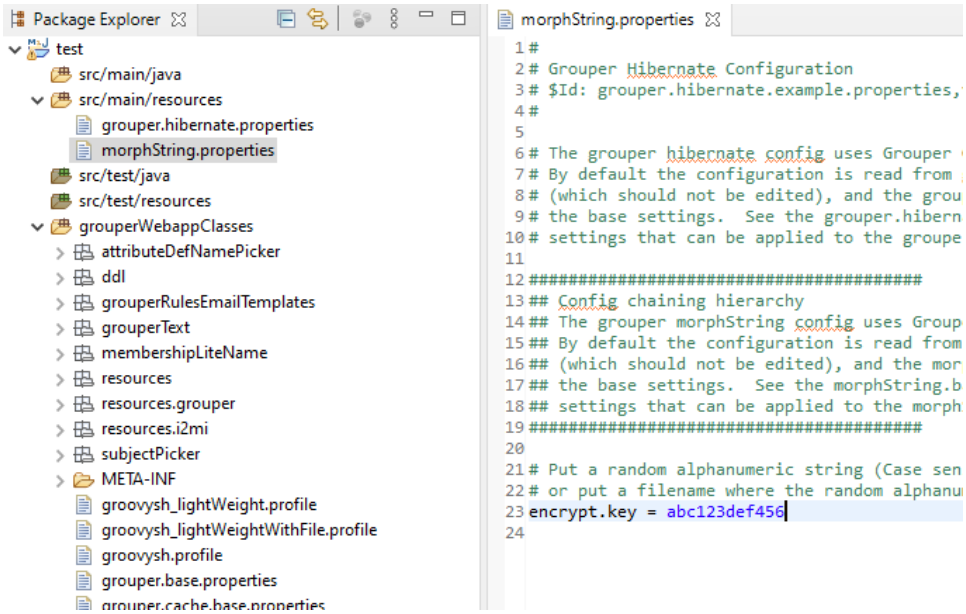


10. Lets move the existing grouperWebapp/WEB-INF/classes dir contents into a new source folder named grouperWebappClasses


Package Explorer
test
src/main/java
src/main/resources
src/test/java
src/test/resources
grouperWebappClasses
attributeDefNamePicker
ddl
grouperRulesEmailTemplates
grouperText
membershipLiteName
resources
resources.grouper
resources.i2mi
subjectPicker
META-INF
groovysh_lightWeight.profile
groovysh_lightWeightWithFile.profile
groovysh.profile
grouper.base.properties
grouper.cache.base.properties
grouper.cache.properties
grouper.client.base.properties
grouper.client.properties
grouper.client.usage.example.txt
grouper.hibernate.base.properties
grouper.hibernate.properties
grouper.properties
grouper-loader.base.properties
grouper-loader.properties
grouper-ui.properties
grouper-ui-ng.base.properties
grouperUtf8.txt
grouper-ws.properties
grouper-ws-ng.base.properties
GSHFileLoad.properties
jaas.conf
krb5.example.conf
log4j.example.properties
log4j.grouperContainerHost.properties
log4j.properties
morphString.base.properties
morphString.properties
Owasp.CsrfGuard.overlay.properties
Owasp.CsrfGuard.properties
server.example.properties
spy.example.properties
sqltool.rc
subject.base.properties
subject.properties



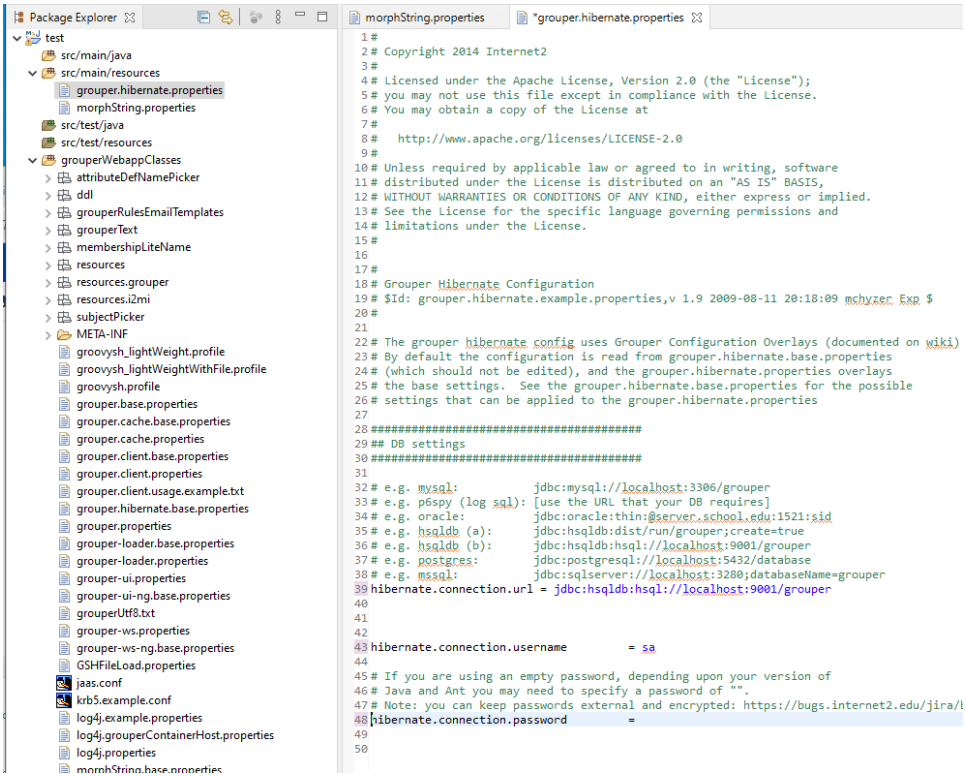
11. Move any properties files from there to the resources folder and customize. Note if you are connecting to an existing database, the morphString secret must match



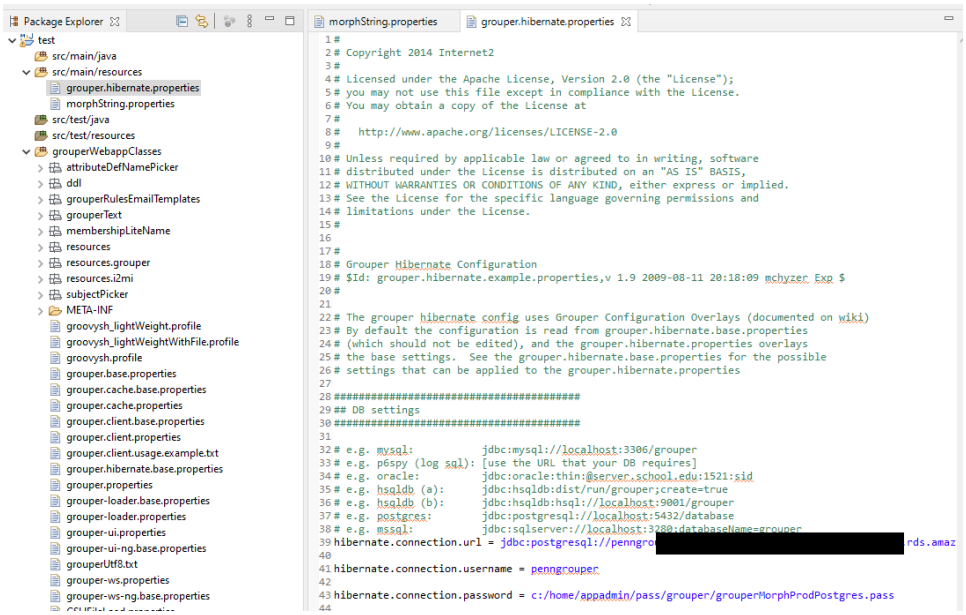
12. If you are using a new database
    a. Start hsql

```
PS C:\users\mchyzer-local\eclipse-workspace\test> mkdir temp
PS C:\users\mchyzer-local\eclipse-workspace\test> cd temp
PS C:\users\mchyzer-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)



13. If you are using an existing database

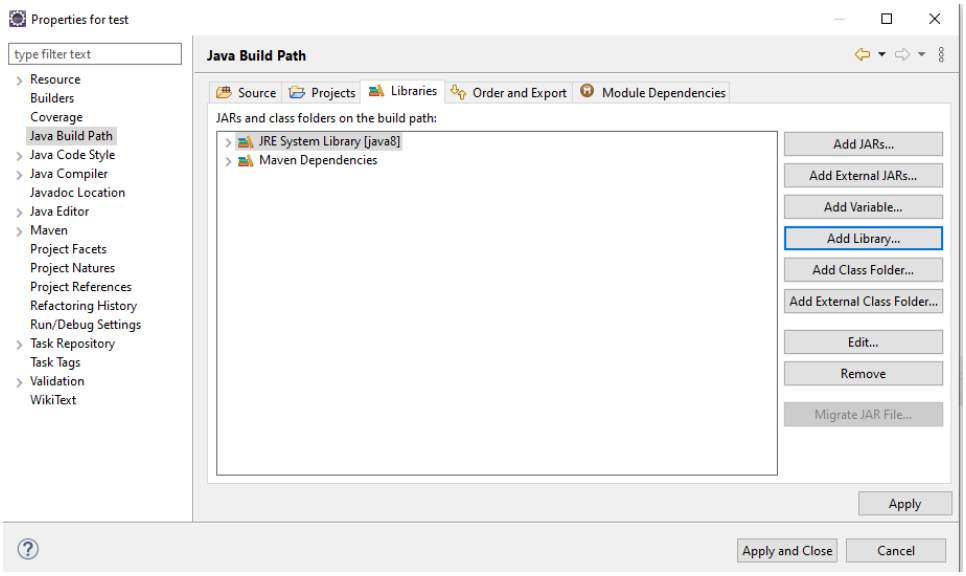


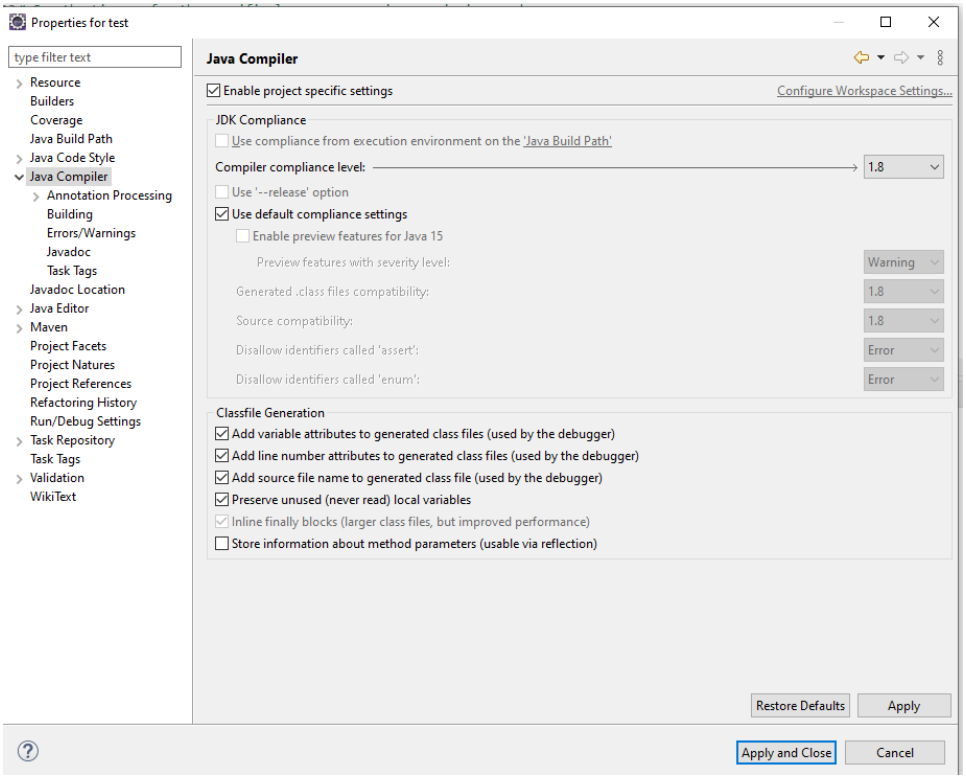
14. Add in to grouper.hibernate.properties that it is ui, and put in a local pass for a subject (remote database) or GrouperSystem (hsql database or remote)

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

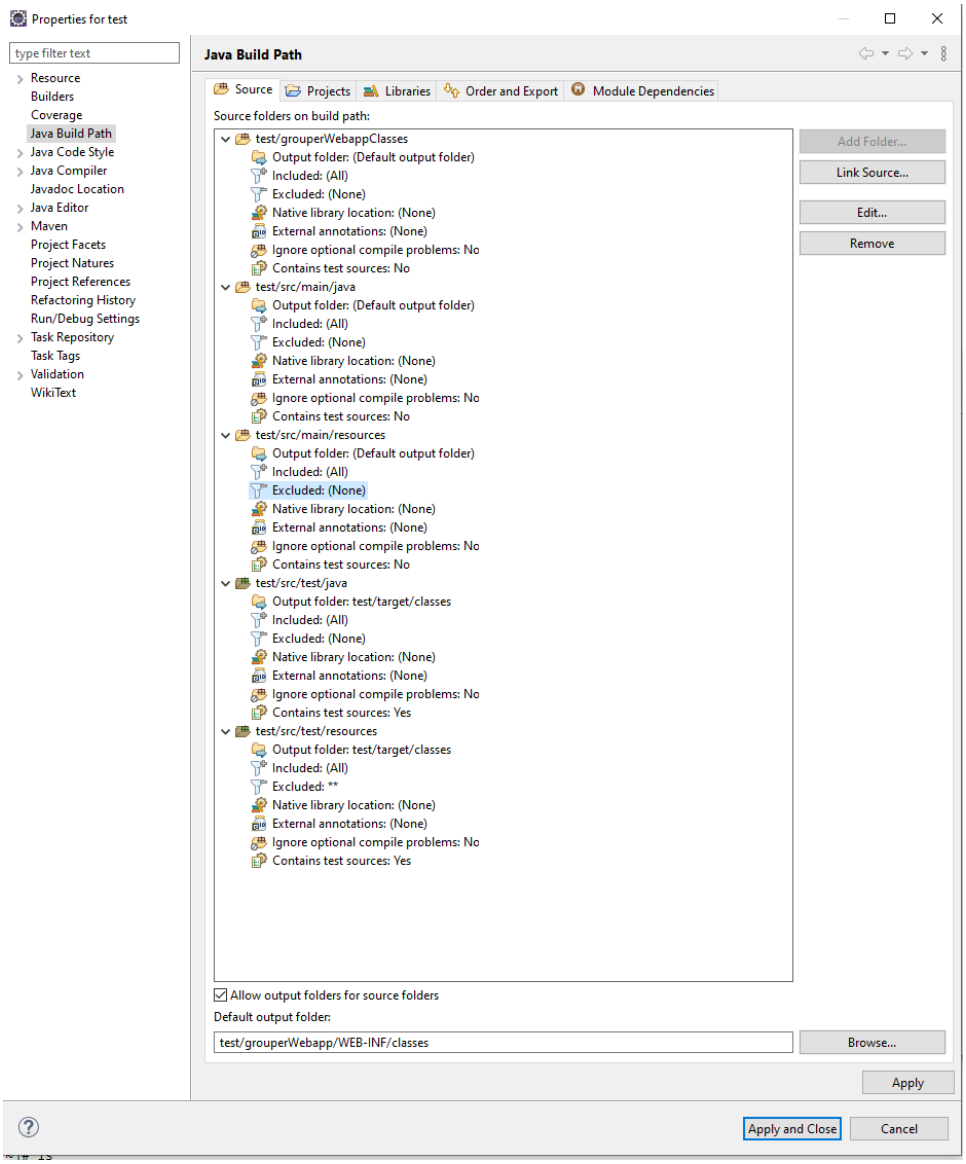
```
# UI basic auth is for quick start. Set to false when you migrate to shib or something else
grouper.is.ui.basicAuthn = true

grouperPasswordConfigOverride_UI_mchyzer_pass = pass
grouperPasswordConfigOverride_UI_GrouperSystem_pass = pass
```

15. Set java17 for project

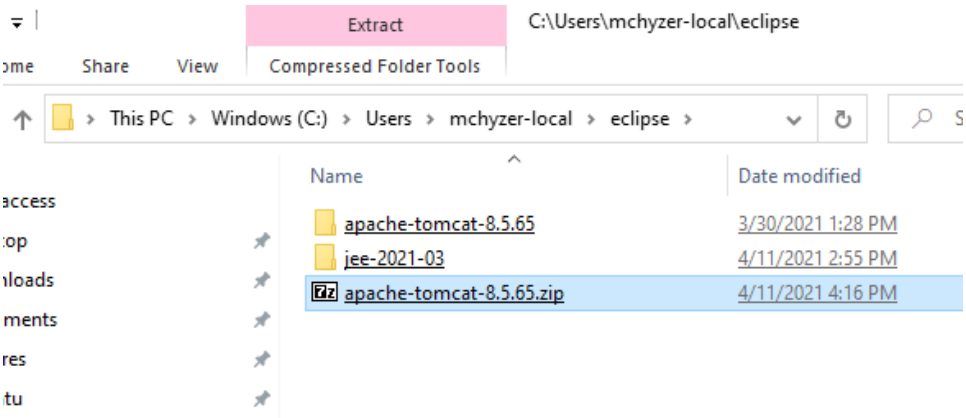




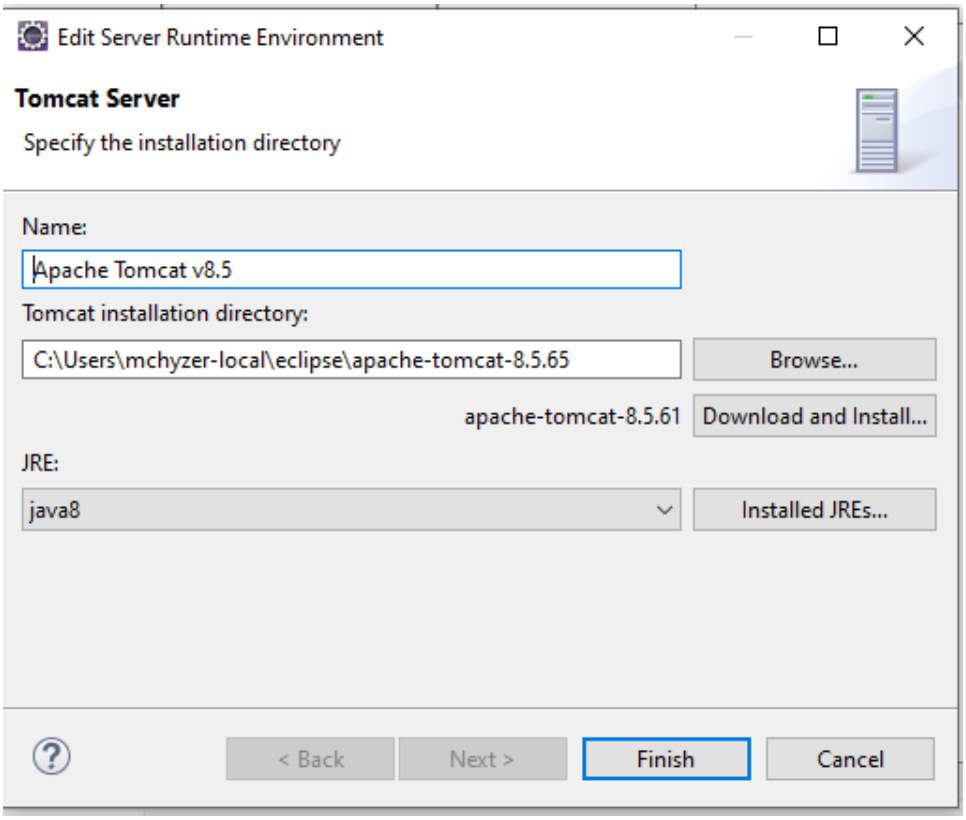
16. Change build path to compile to grouperWebapp/WEB-INF/classes (except the test source and test resources). Also dont exclude any resources. Note, in future if you do Maven Update project, you might have reset some of these settings



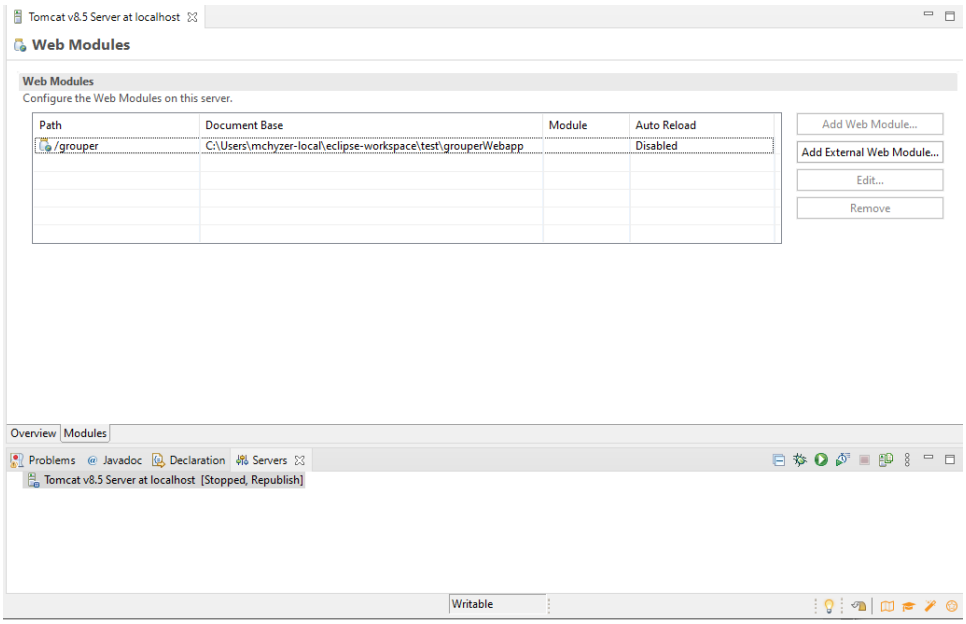
17. Download tomcat 9



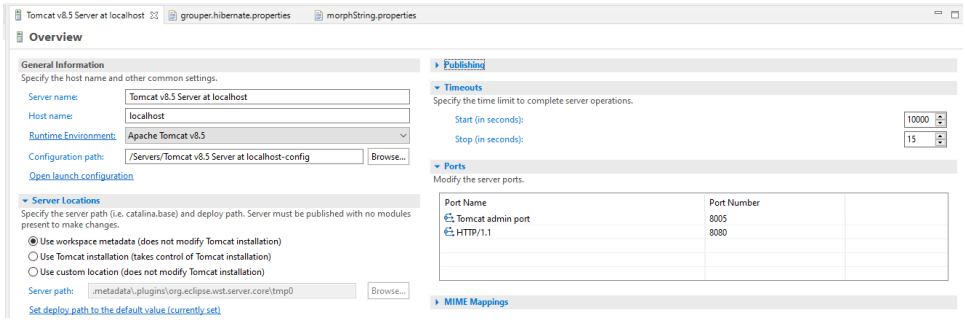
18. Add a new server in eclipse



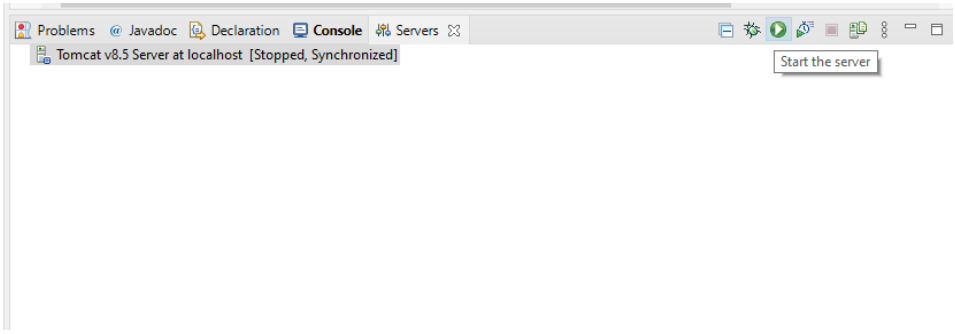
19. Window Show view Servers, add a module



20. Make a large startup timeout



21. Start the server



22. 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
   cd .....\test. (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
   cd .....\test. (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
   cd .....\test. (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!)