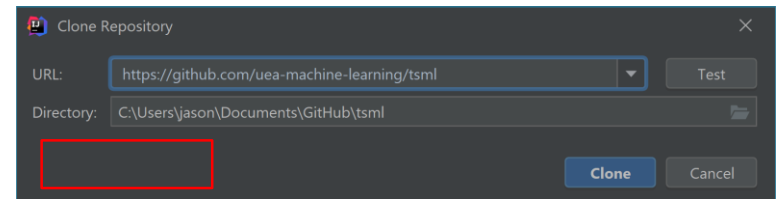
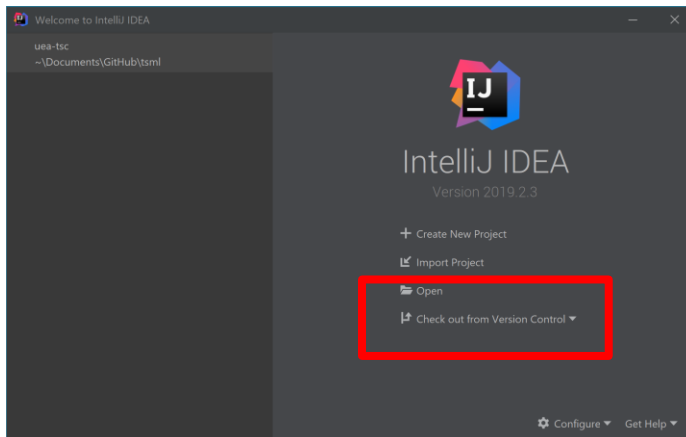


Building JARs with TSML in IntelliJ

Prerequisites

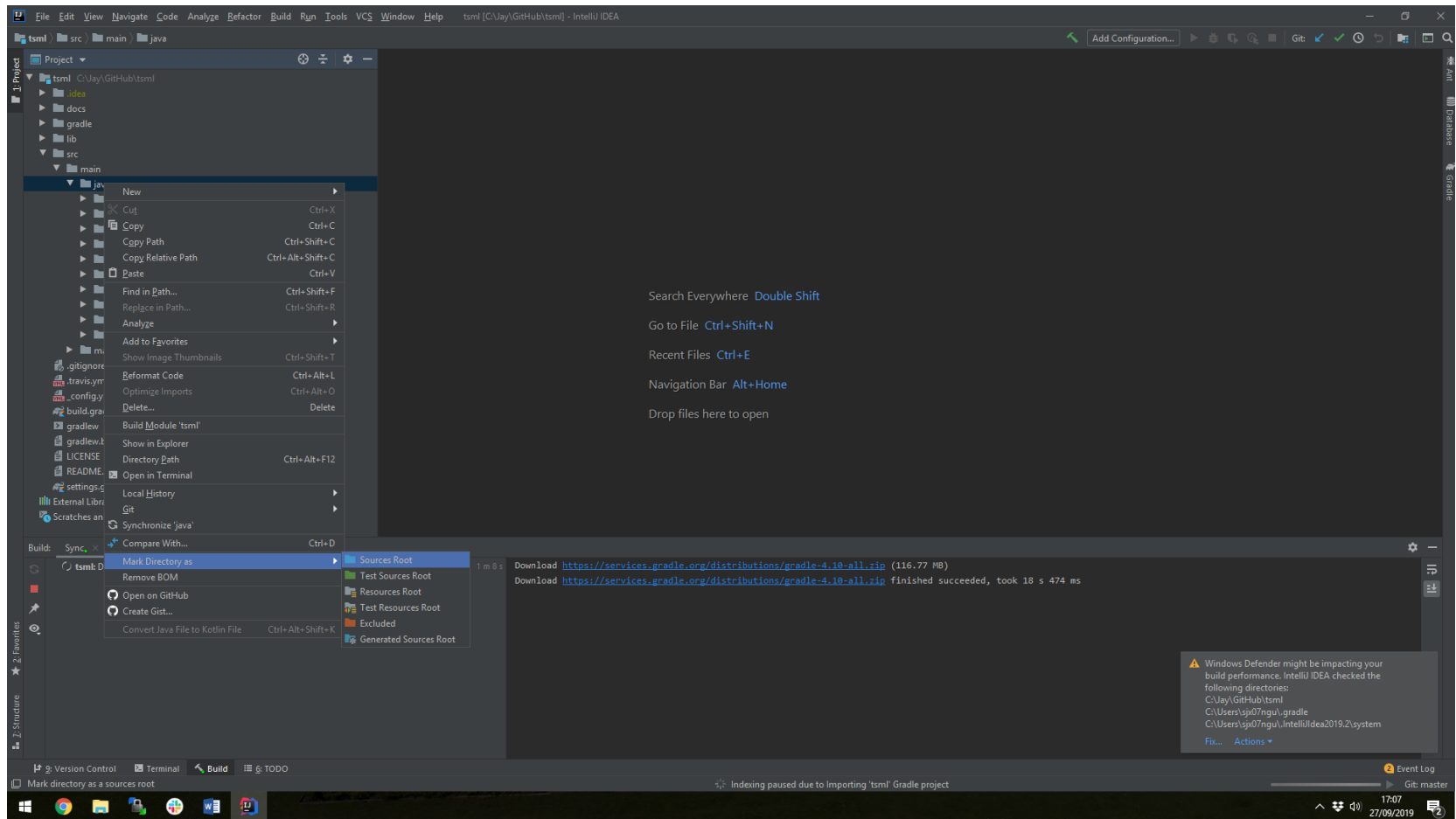
Either:

1. Download/check-out the tsml code manually and open in IntelliJ
2. Clone directly from the IntelliJ welcome screen

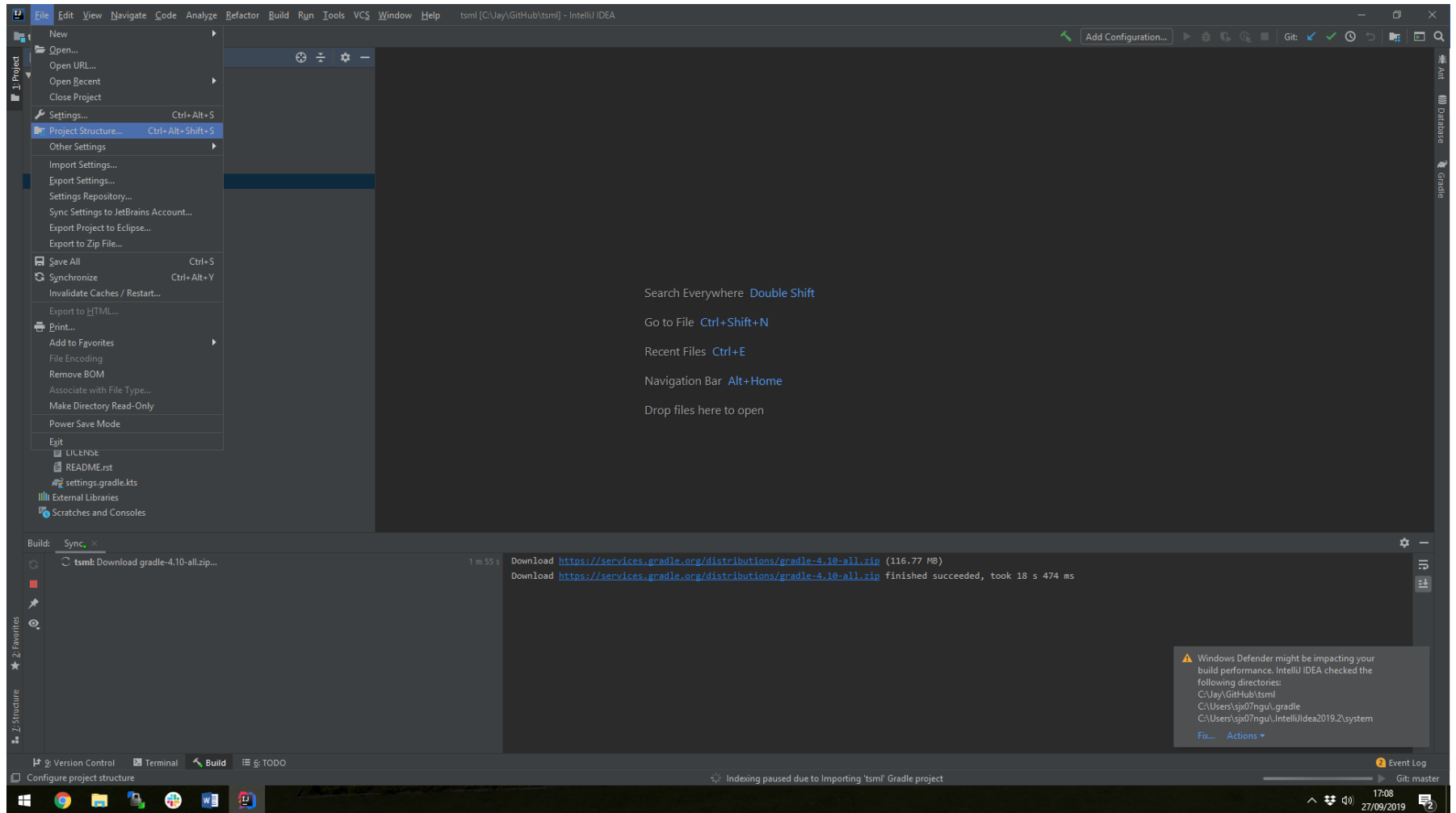


Note – I'm already logged into GitHub via IntelliJ.
If not logged in, a button to do so will be here (recommended)

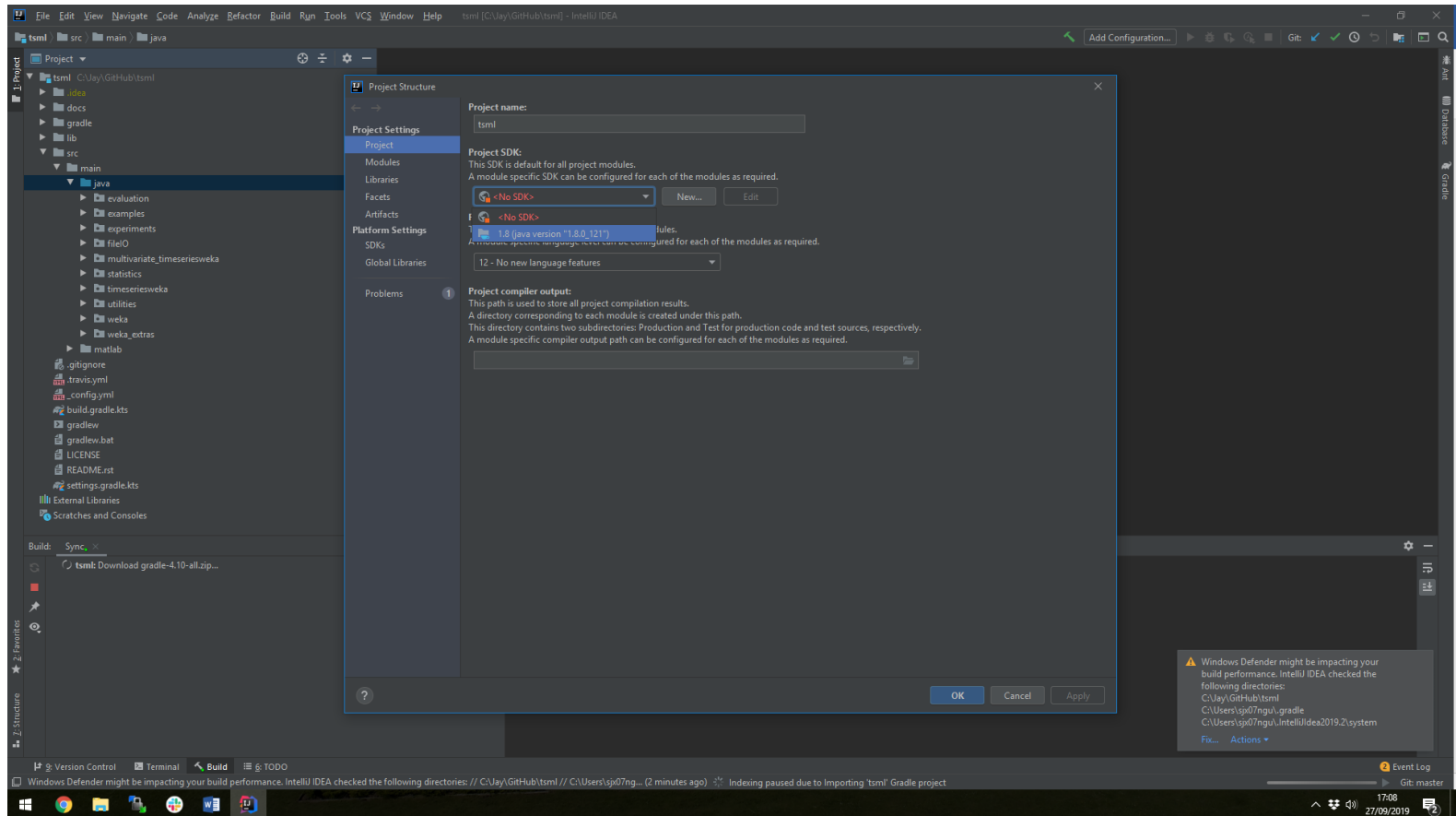
Once opened, right click `tsml/src/main/java` and click **Mark Directory as Source Route** (it should go blue afterwards - if it wasn't already)



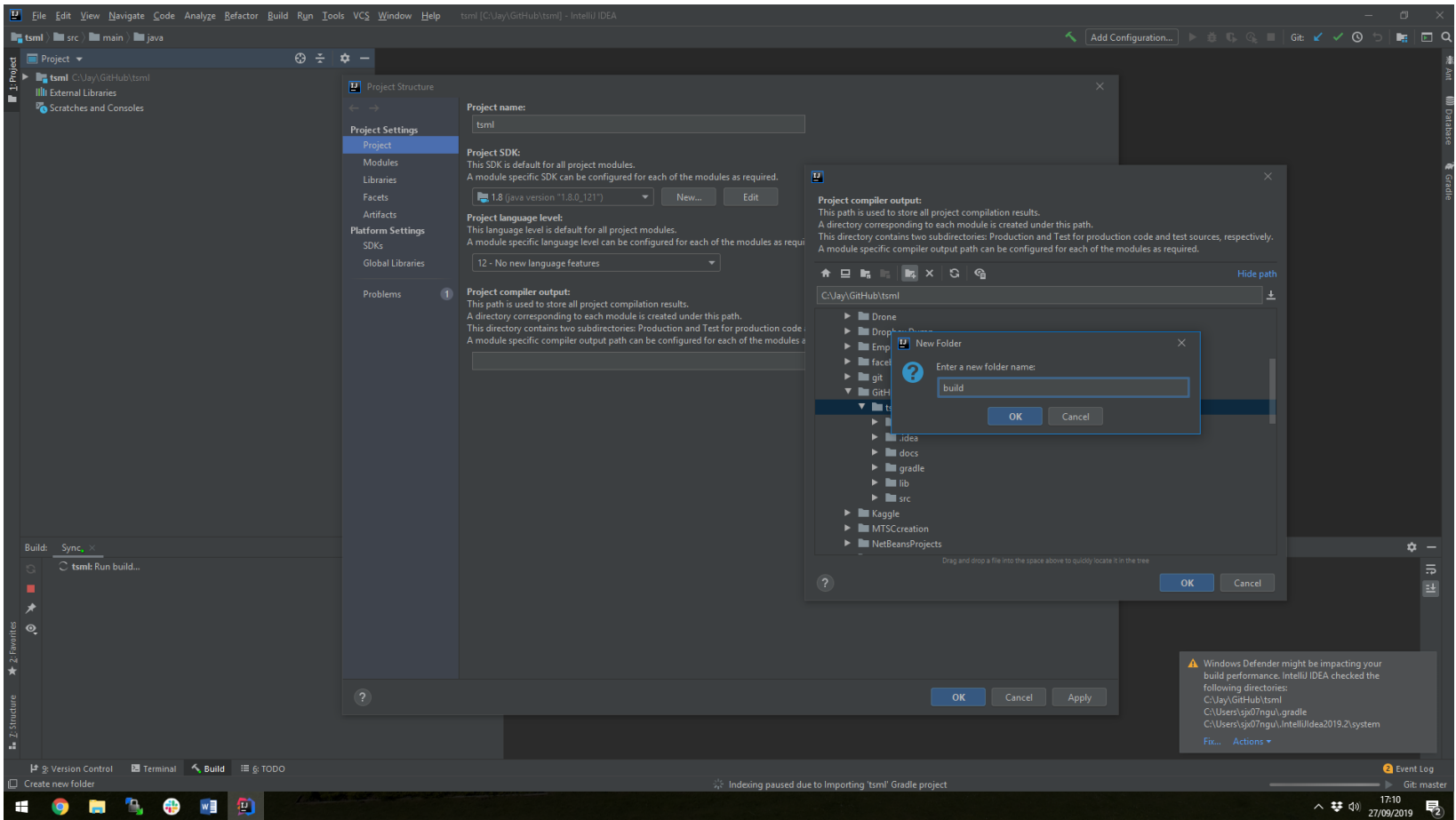
Click File -> Project Structure



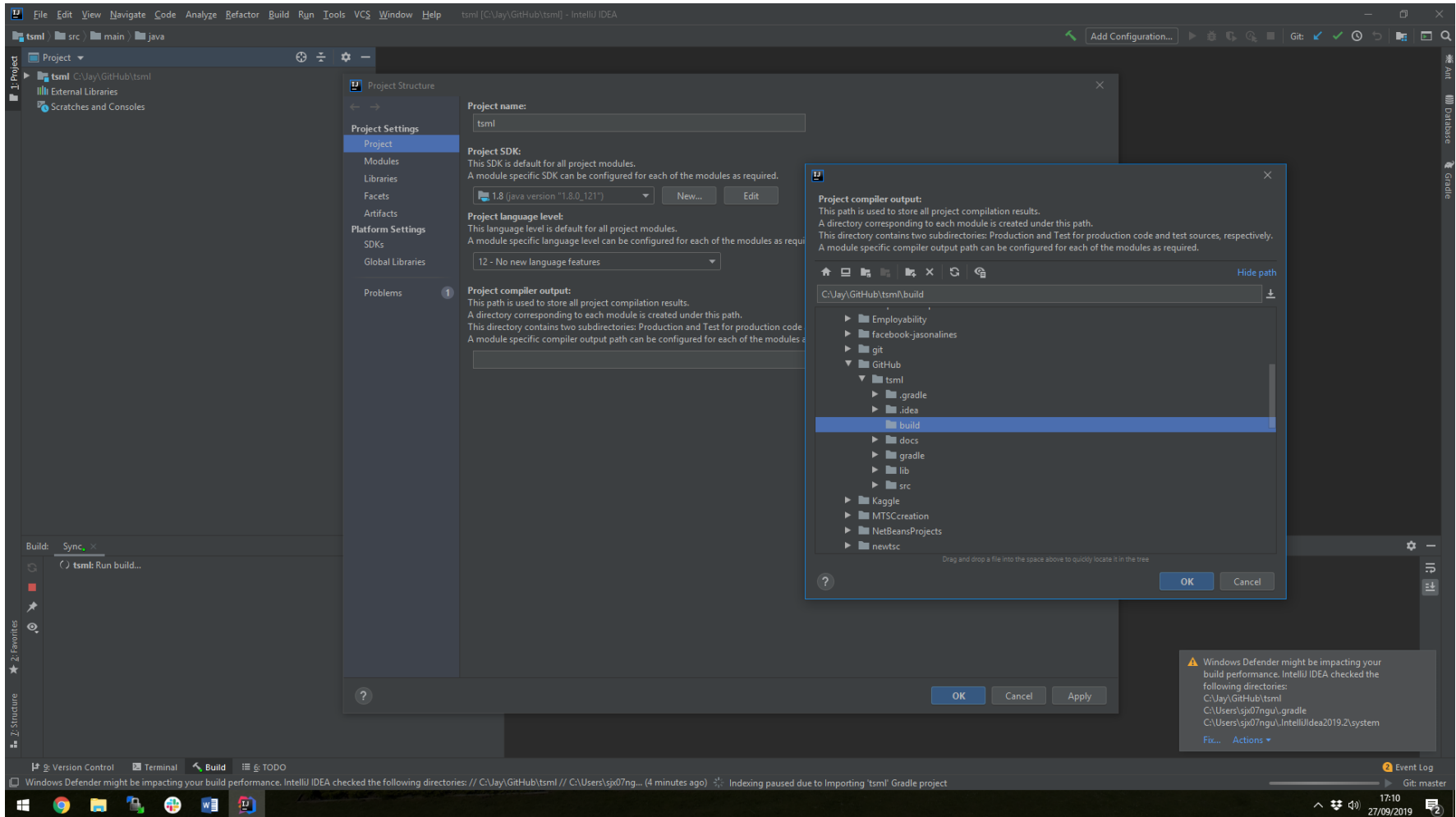
Set SDK (if not already set)



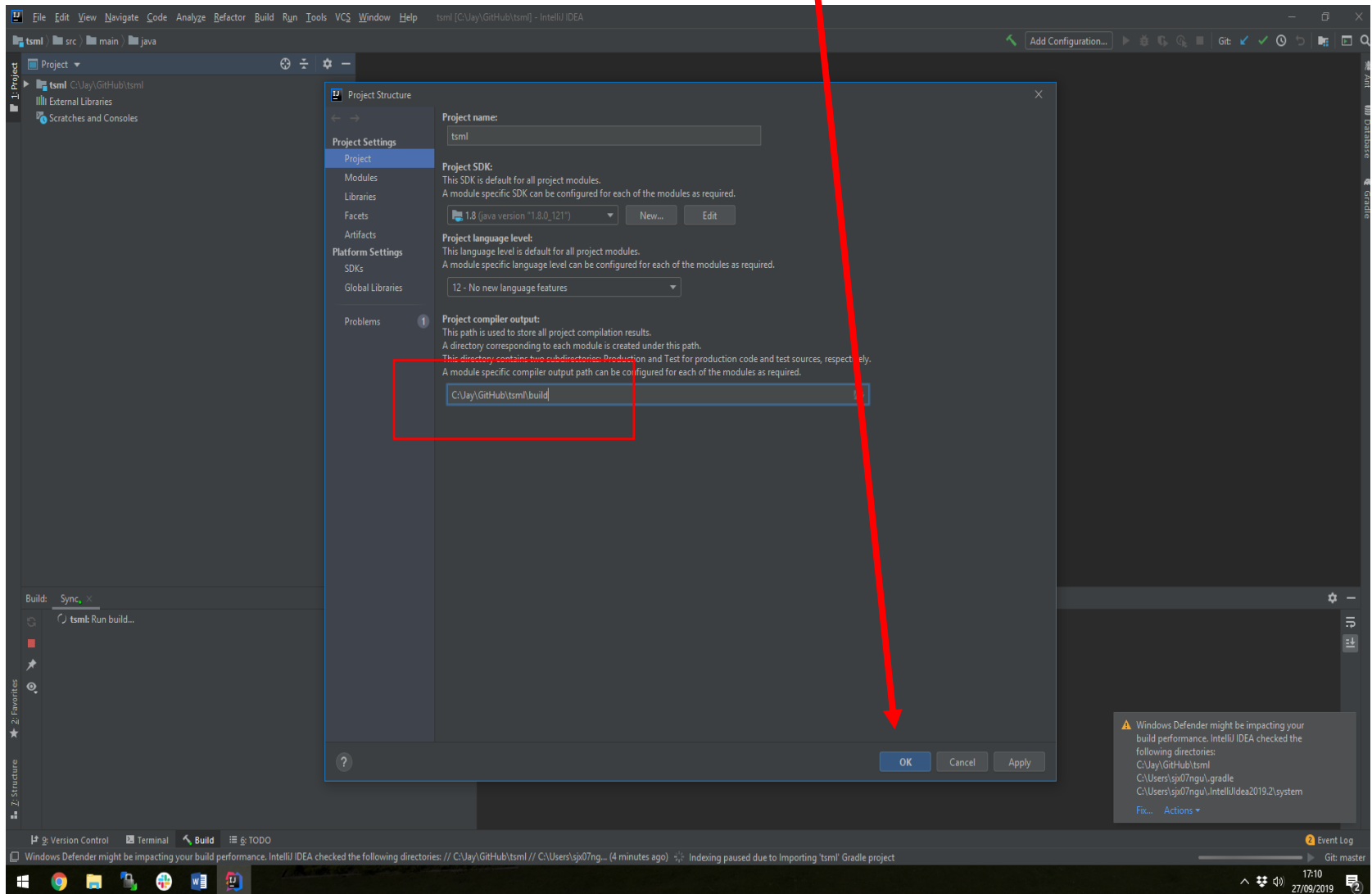
Project Compiler Output - set location for compiled class files
(this example creates a folder called build to mirror NetBeans)



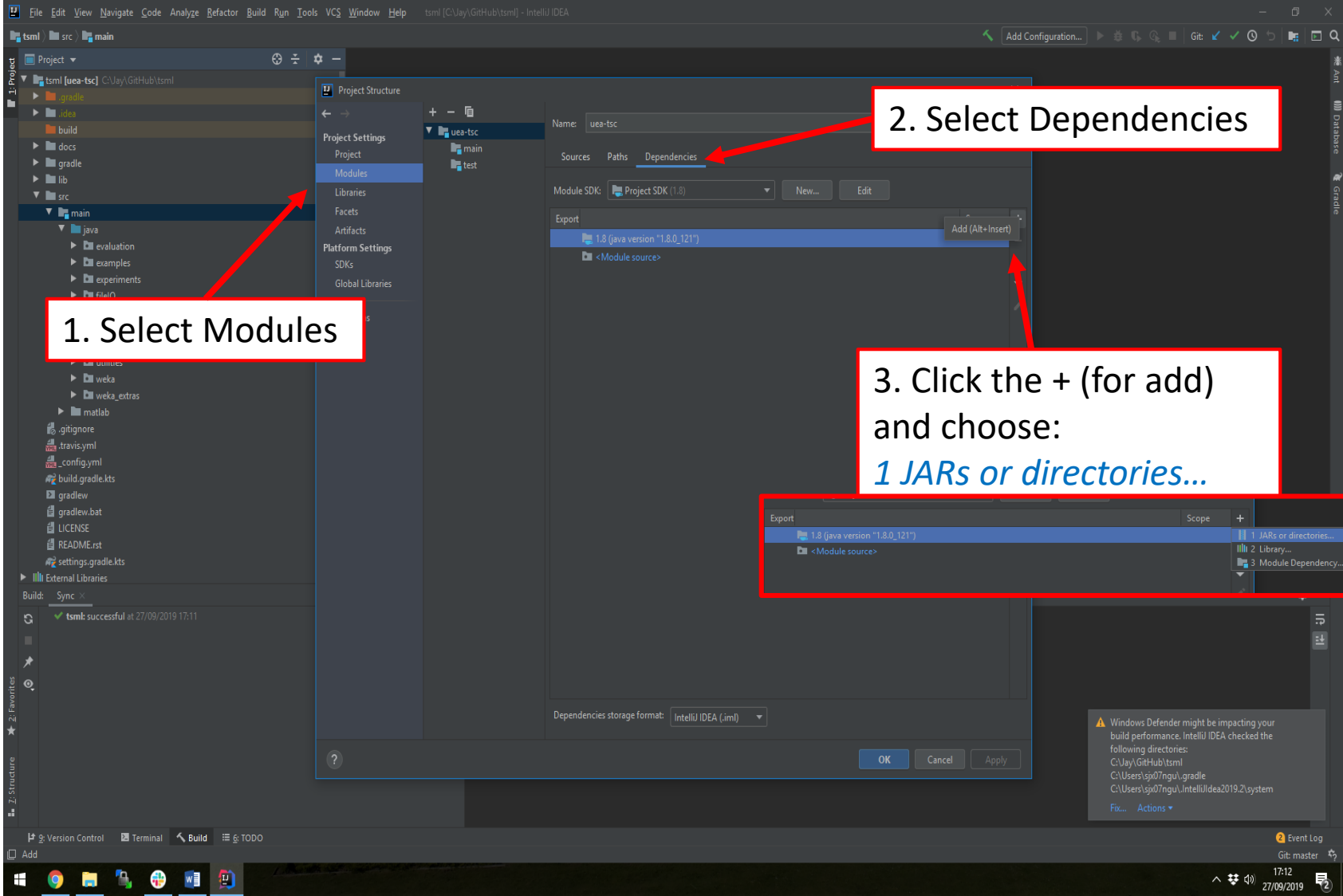
Confirm compiler output folder (build in this example)



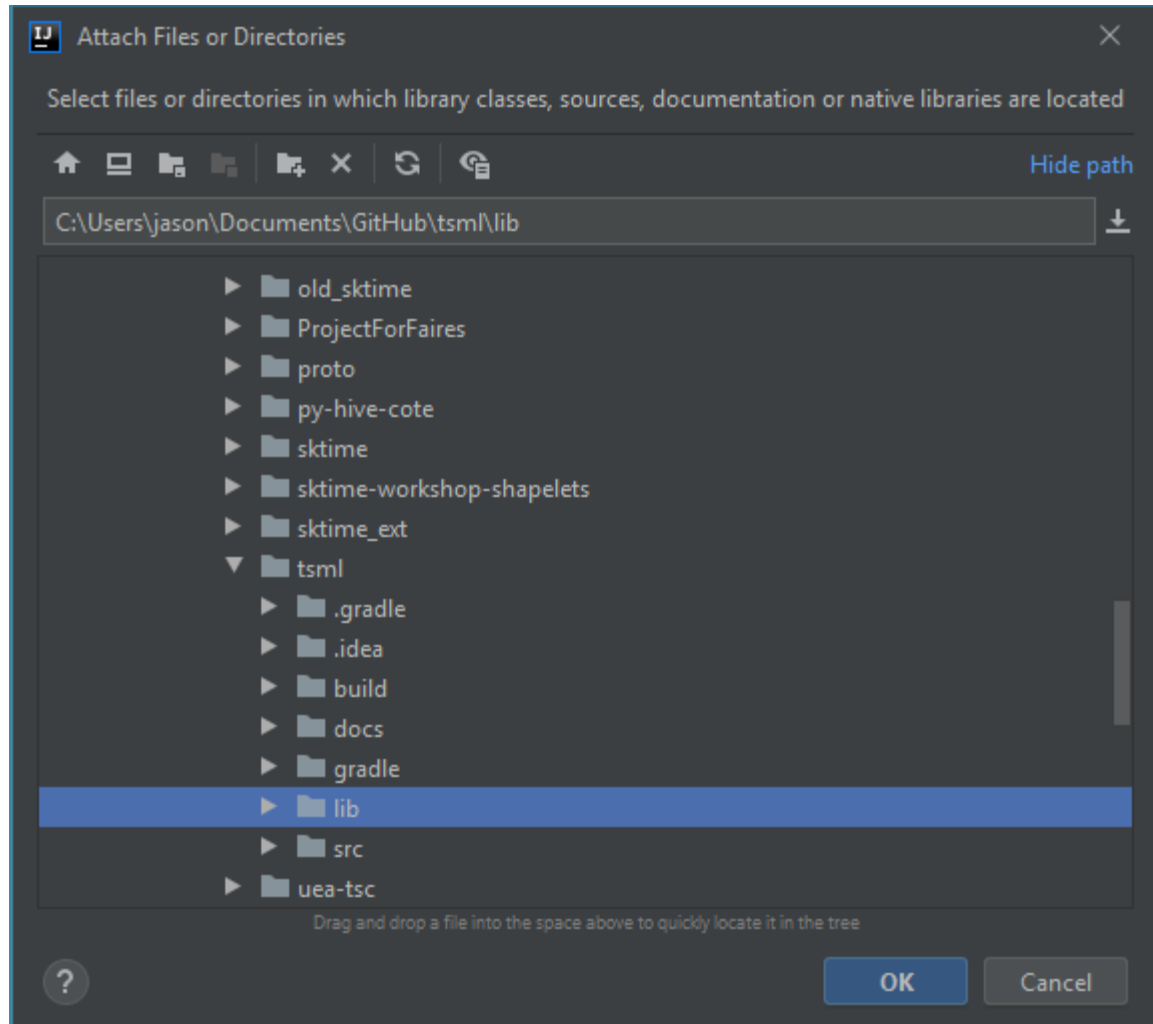
Output location should be updated now. Press OK to confirm



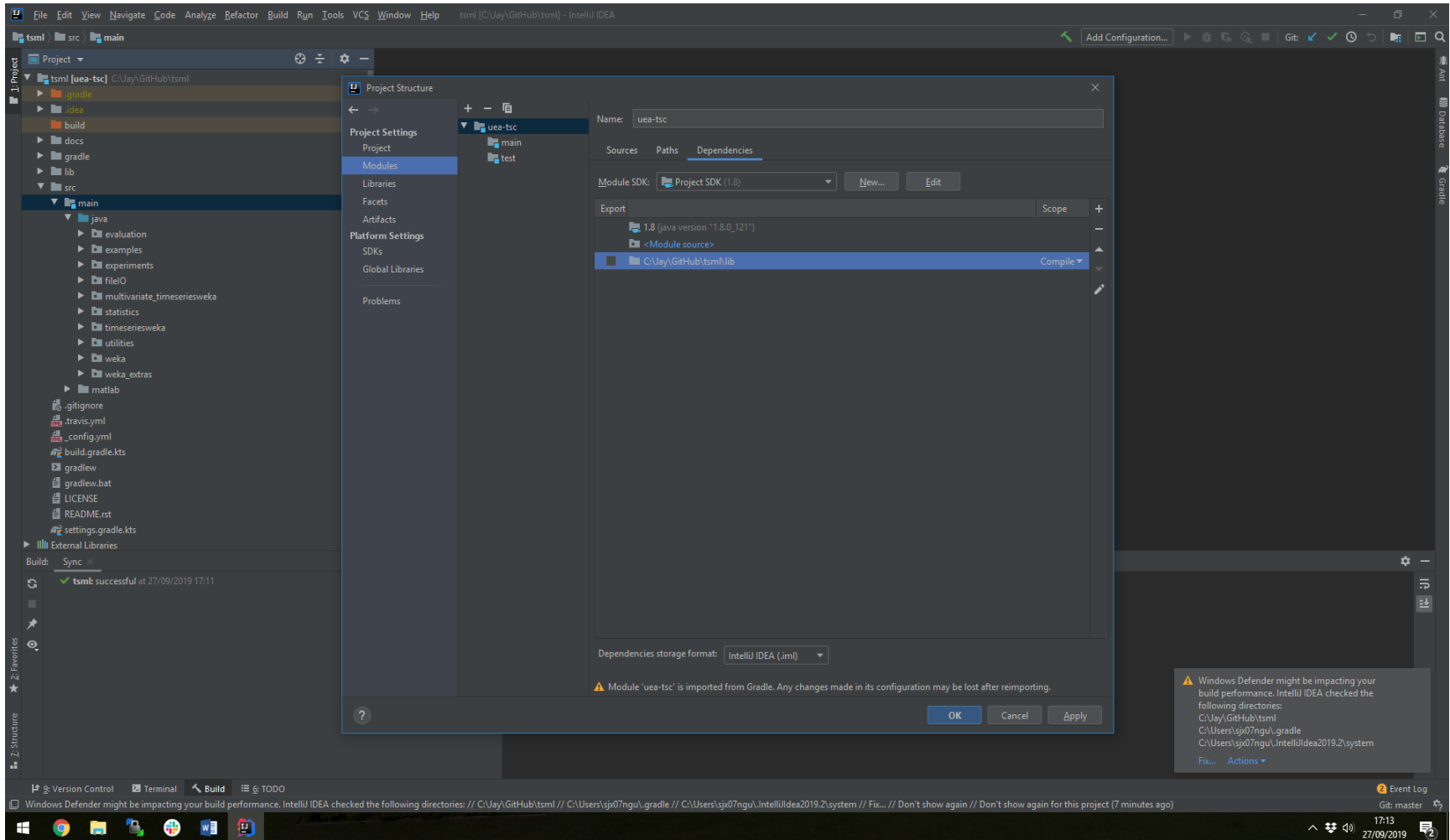
Click File -> Project Structure again, this time going to the **Modules** tab



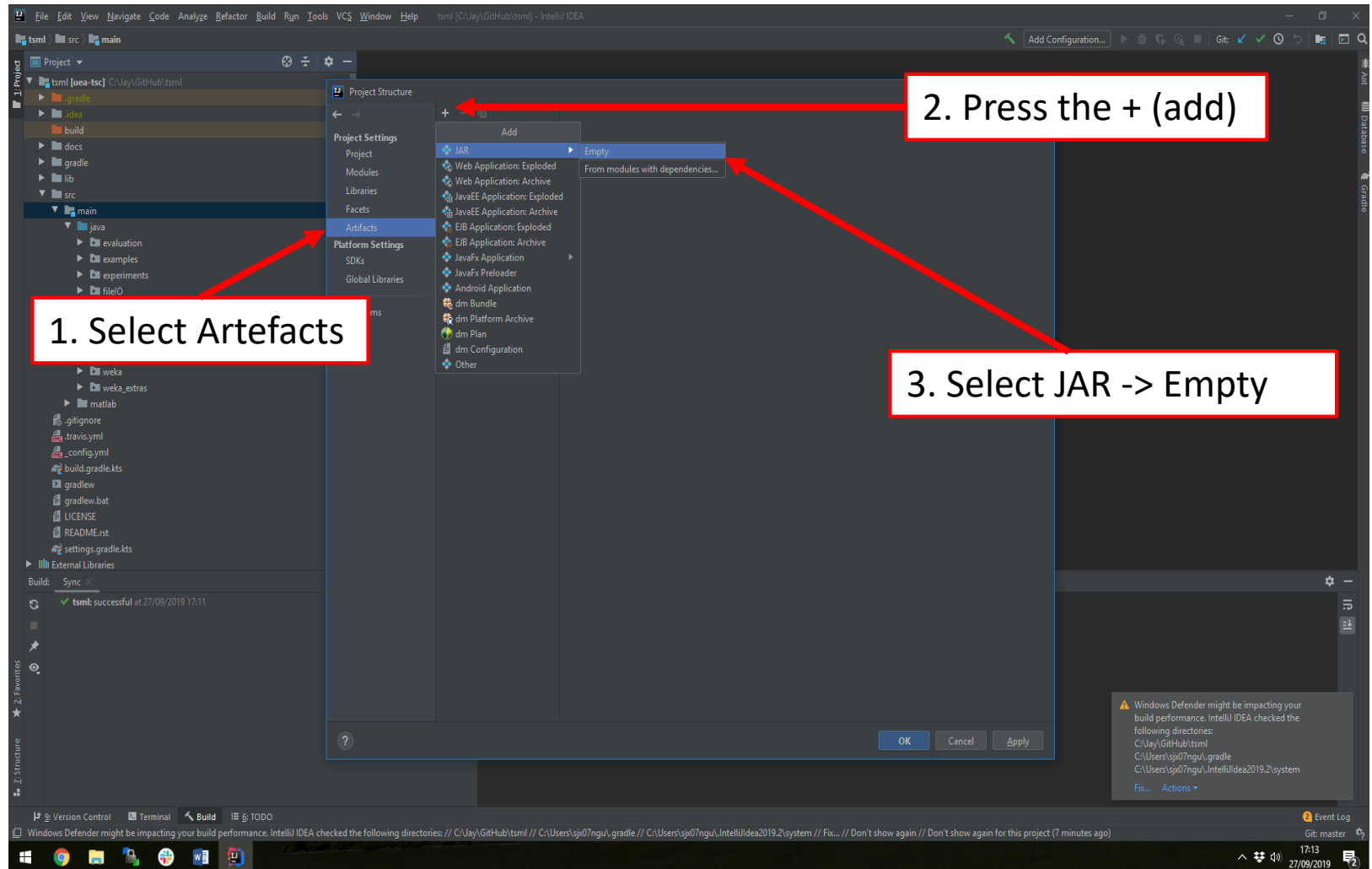
Choose the lib folder from tsml and press OK



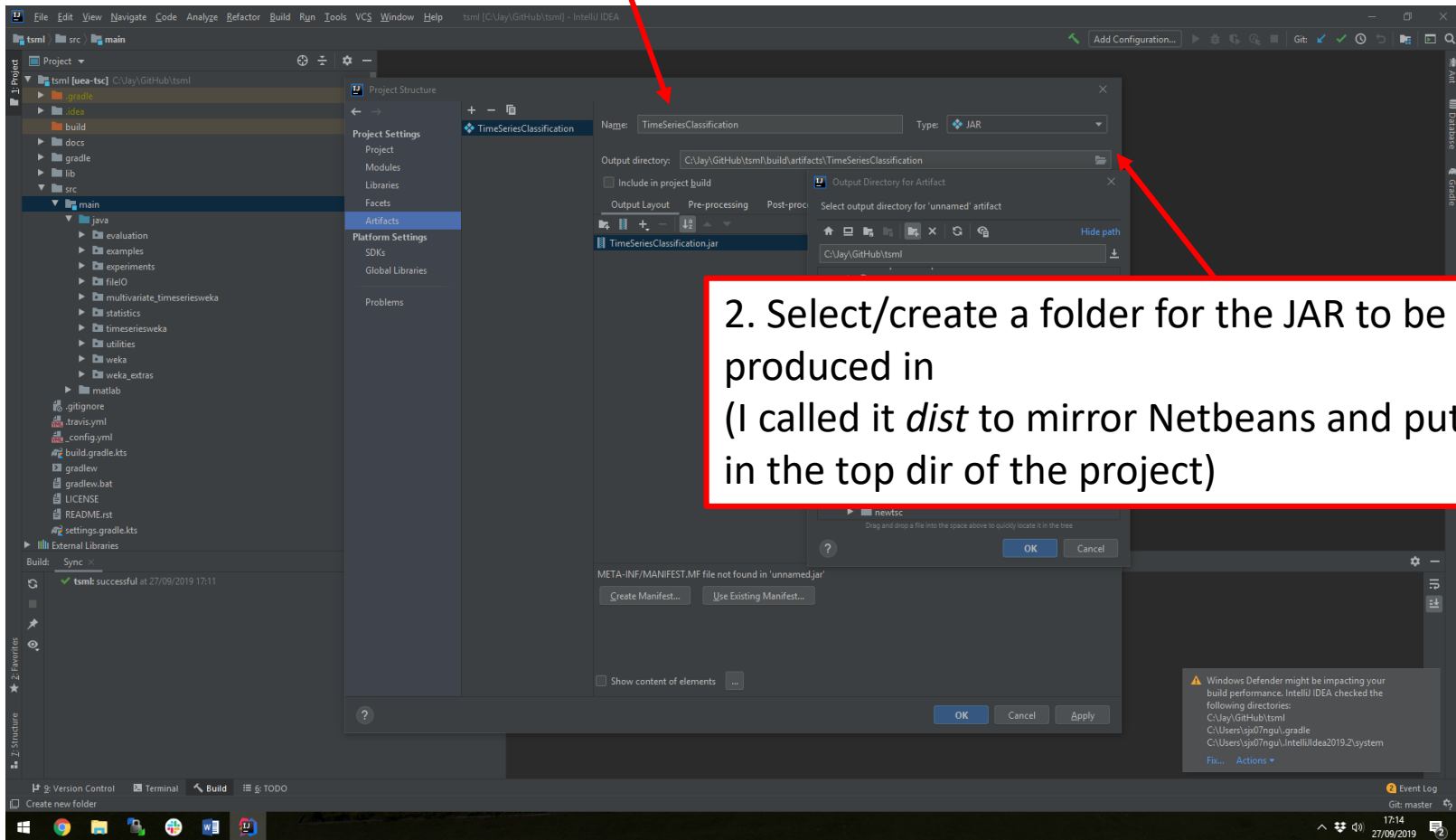
Lib should now be added to the list:



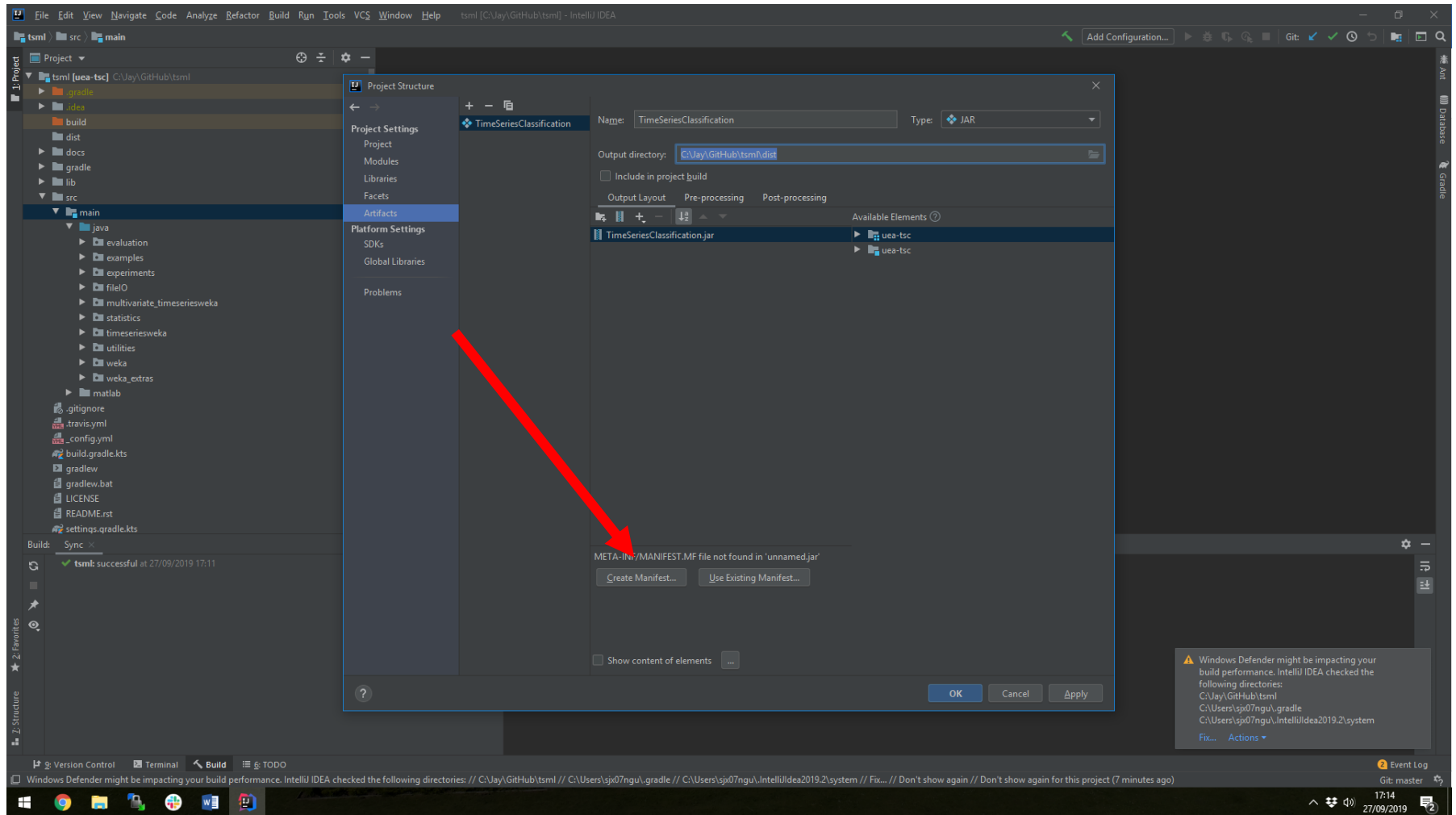
Now change to the **Artefacts** tab



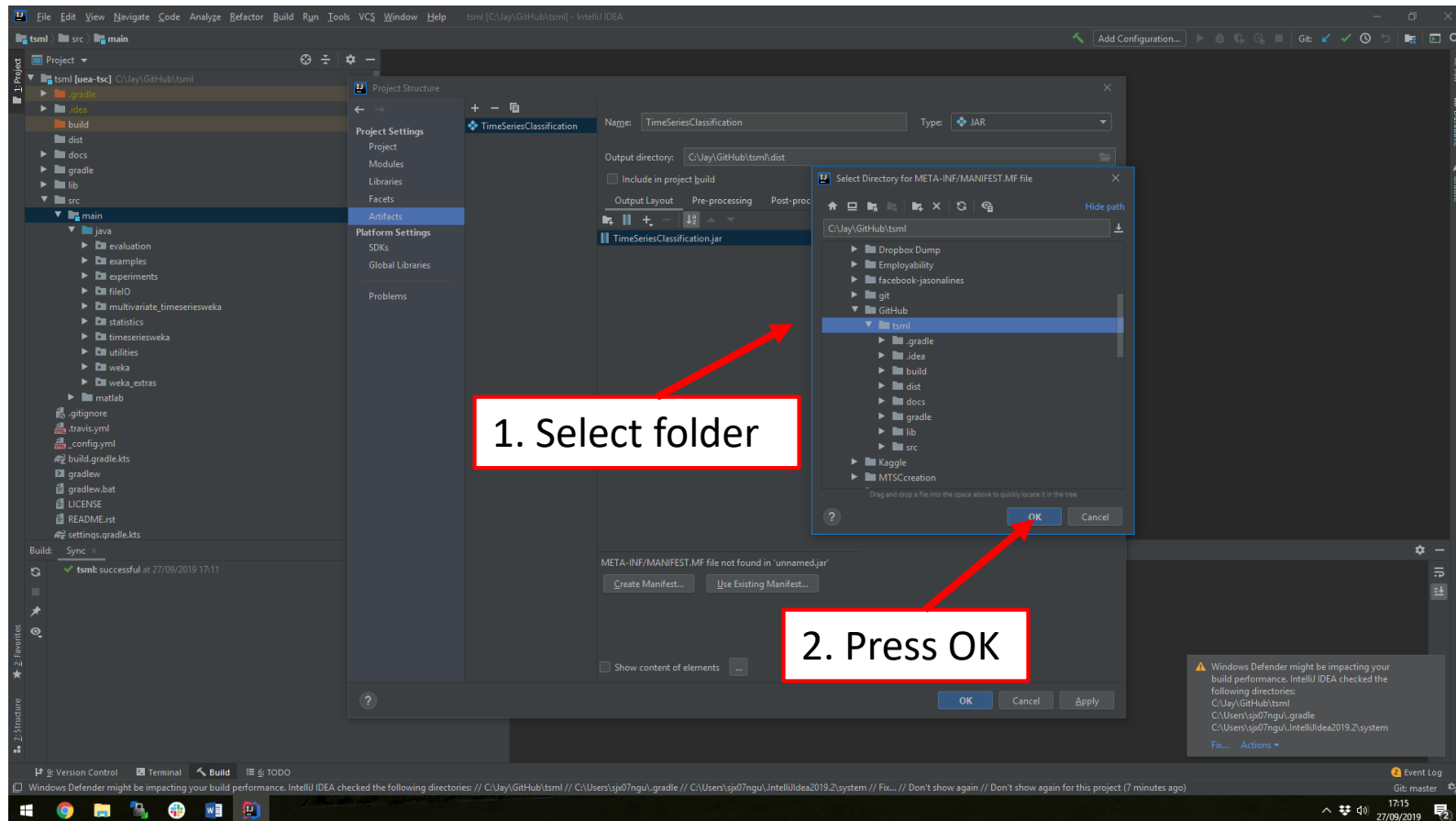
1. Rename what the output jar will be called
(I went with TimeSeriesClassification.com)



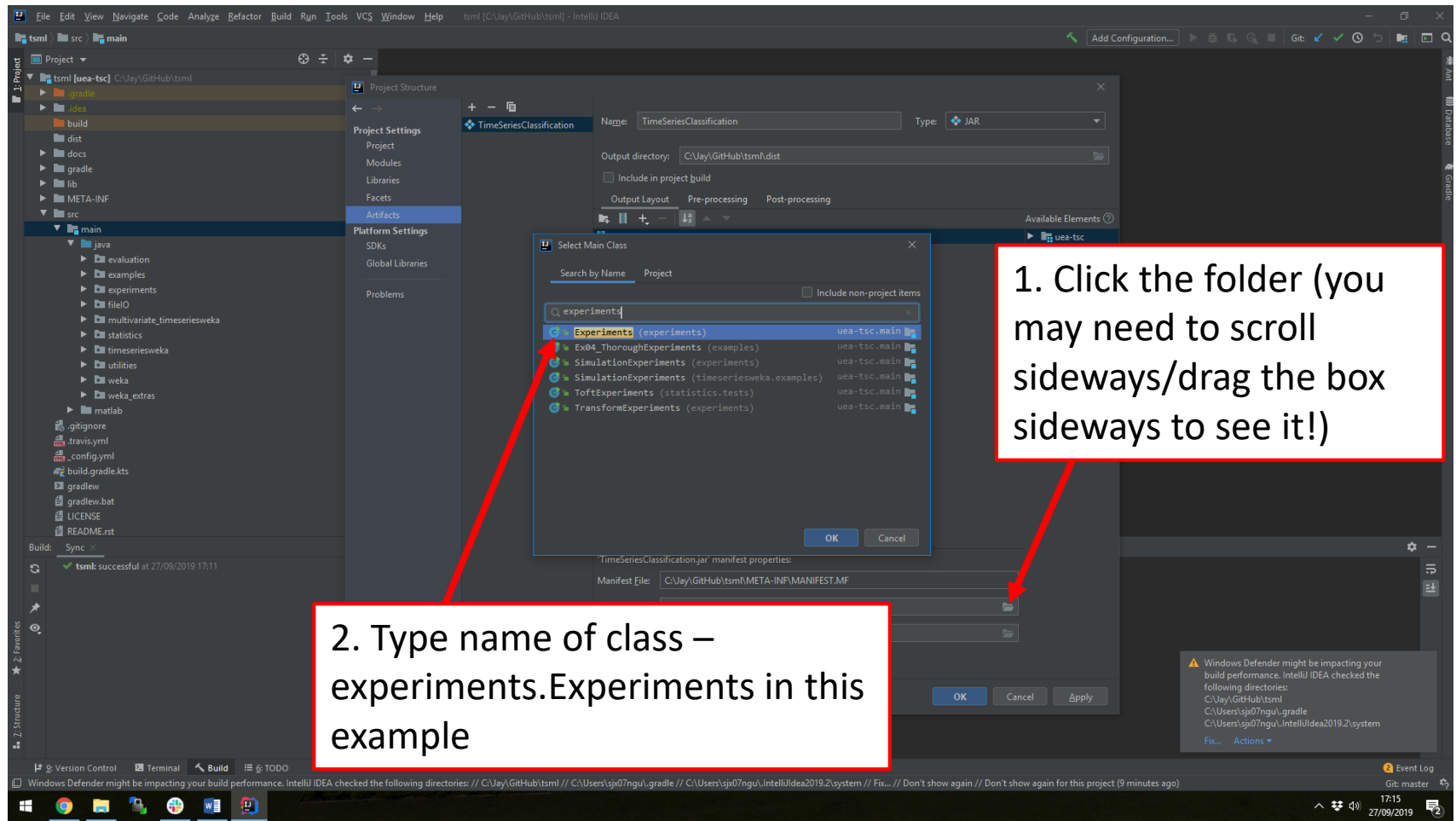
Now click *Create Manifest...*



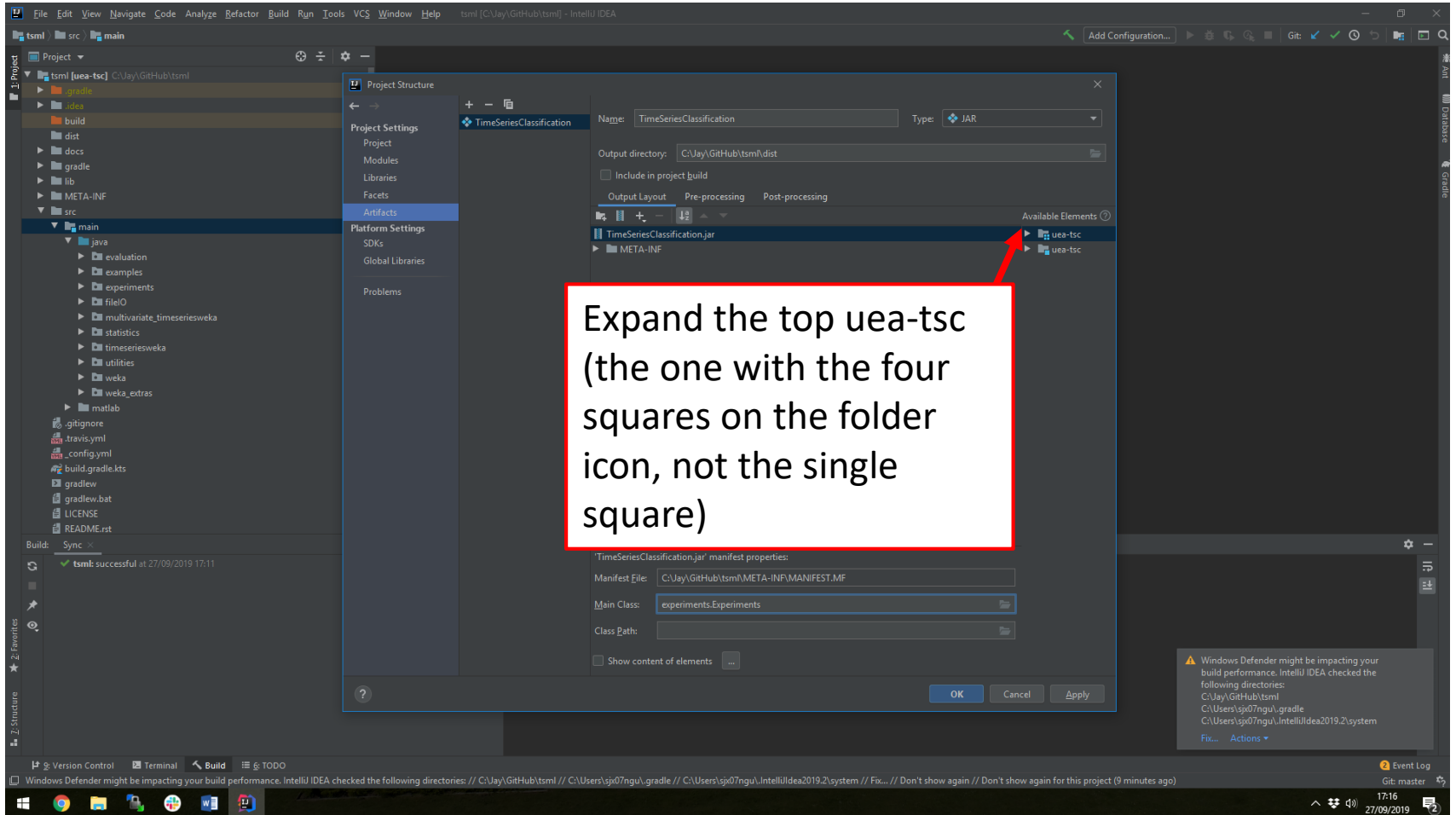
Select where the manifest should go (I put it in the top directory to mirror Netbeans)

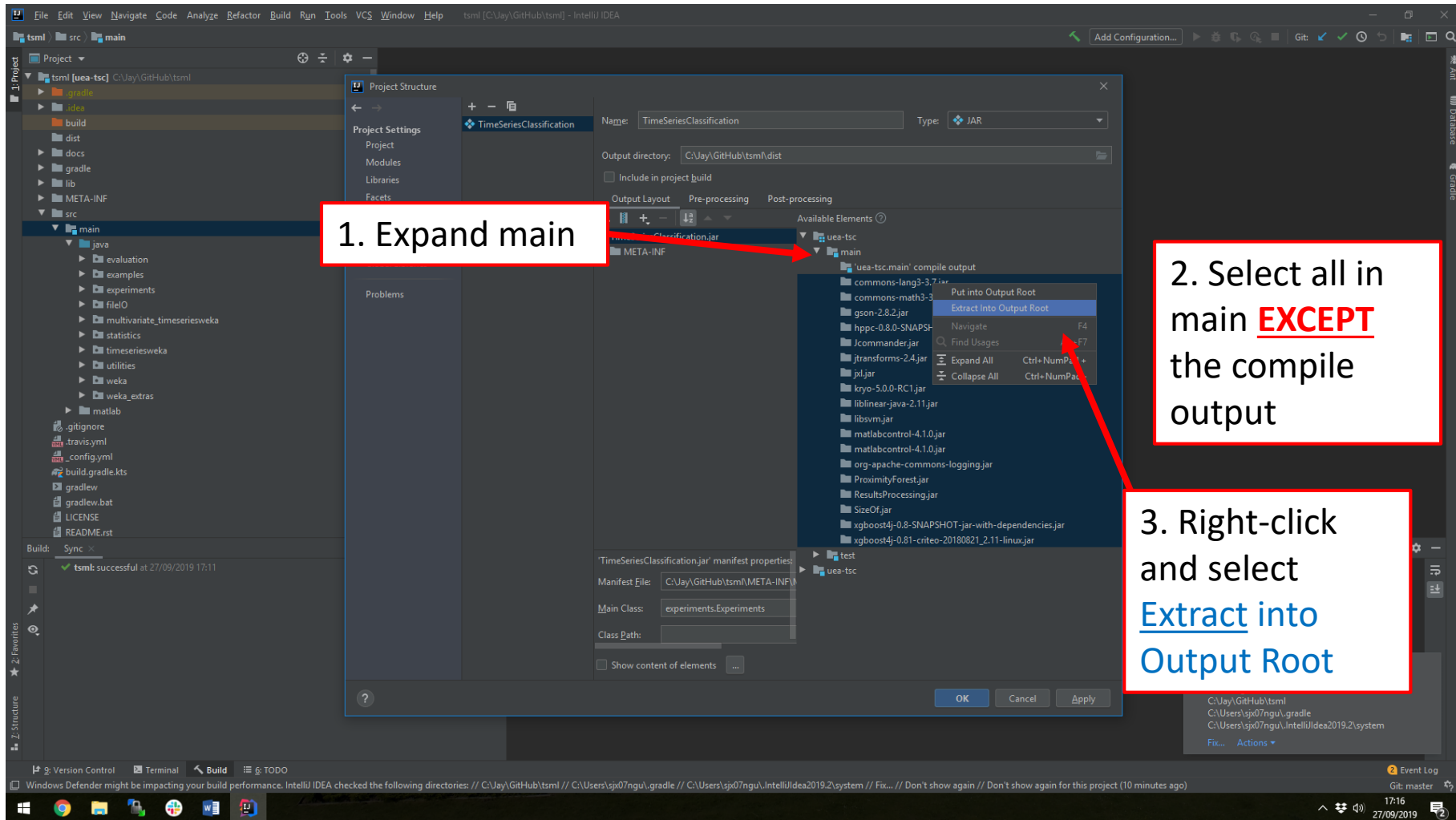


Now set main class:

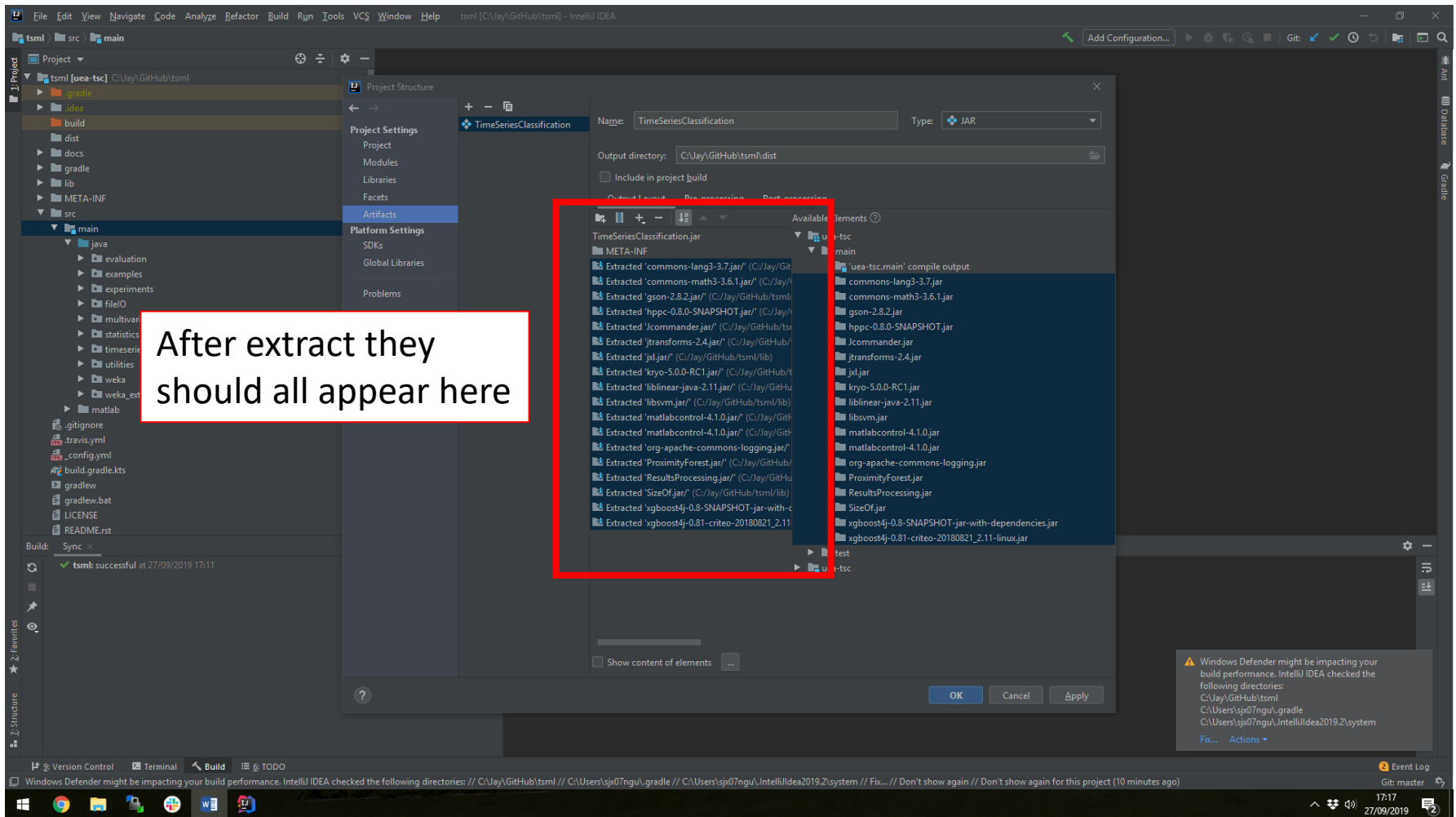


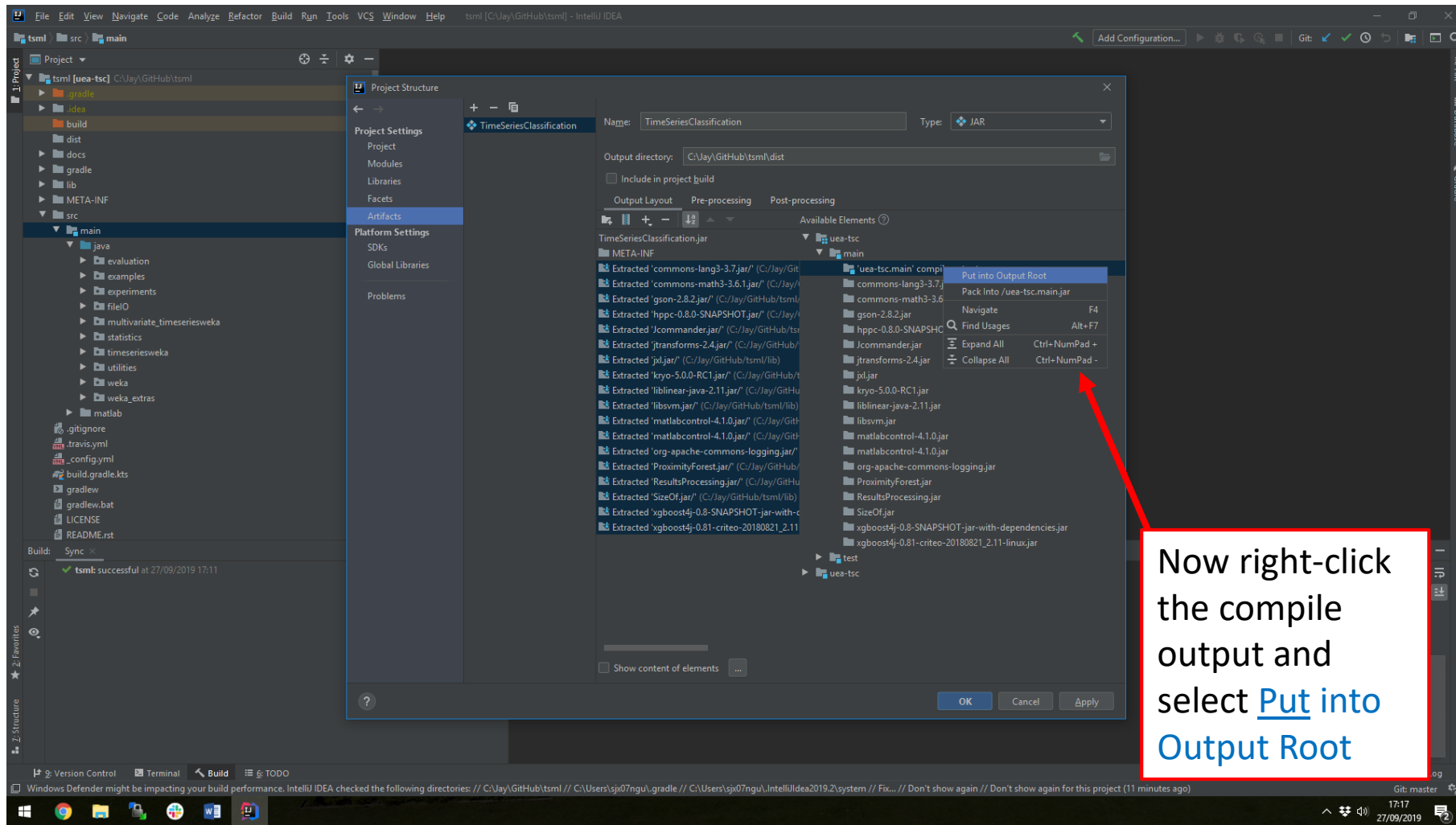
Next we need to look at the Available Elements column (again, you may need to resize if necessary)





* I believe this is because the .jars are all archives and require extracting. The compile output is not an archive however so we handle that slightly different in a couple of slides

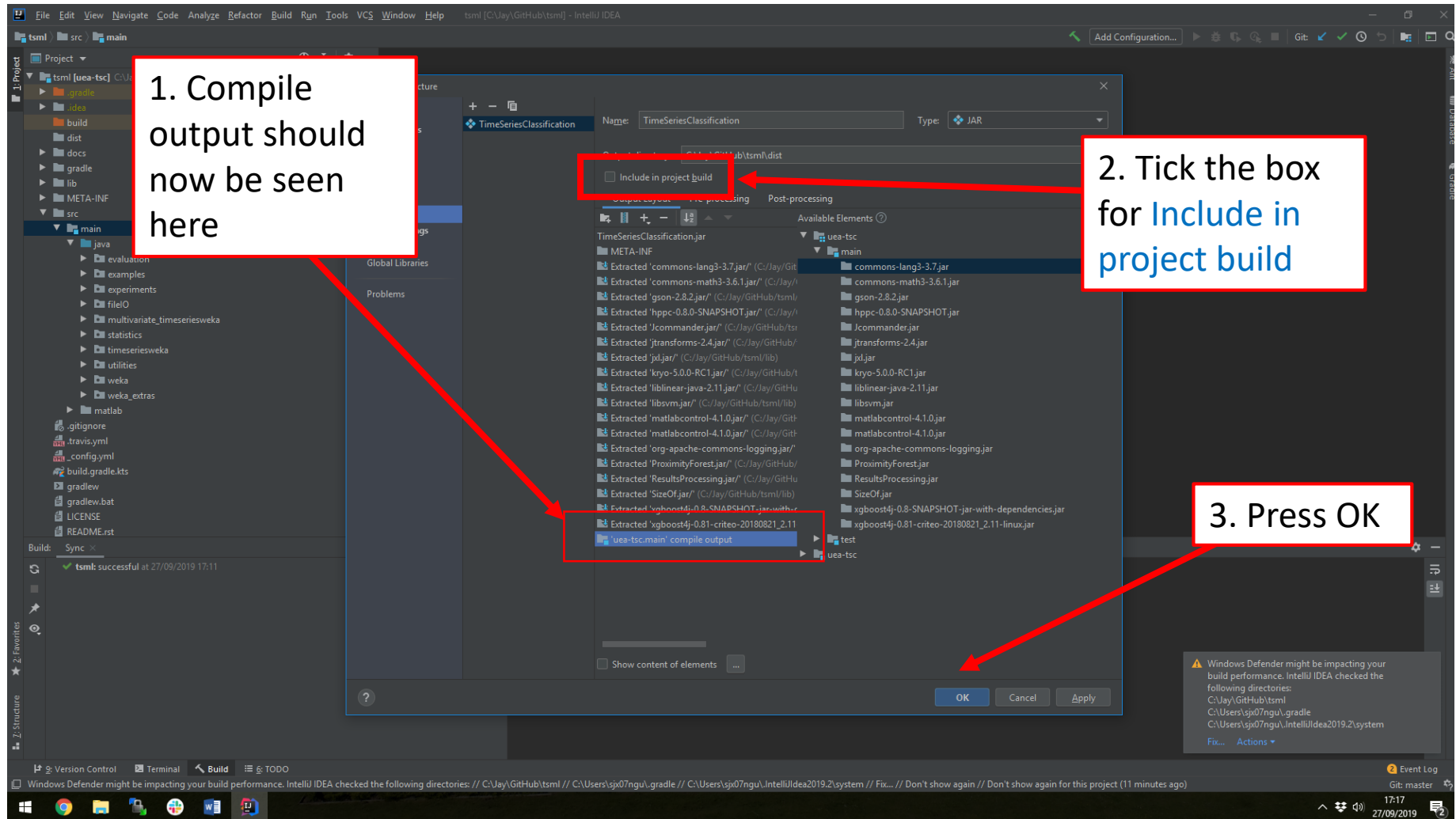




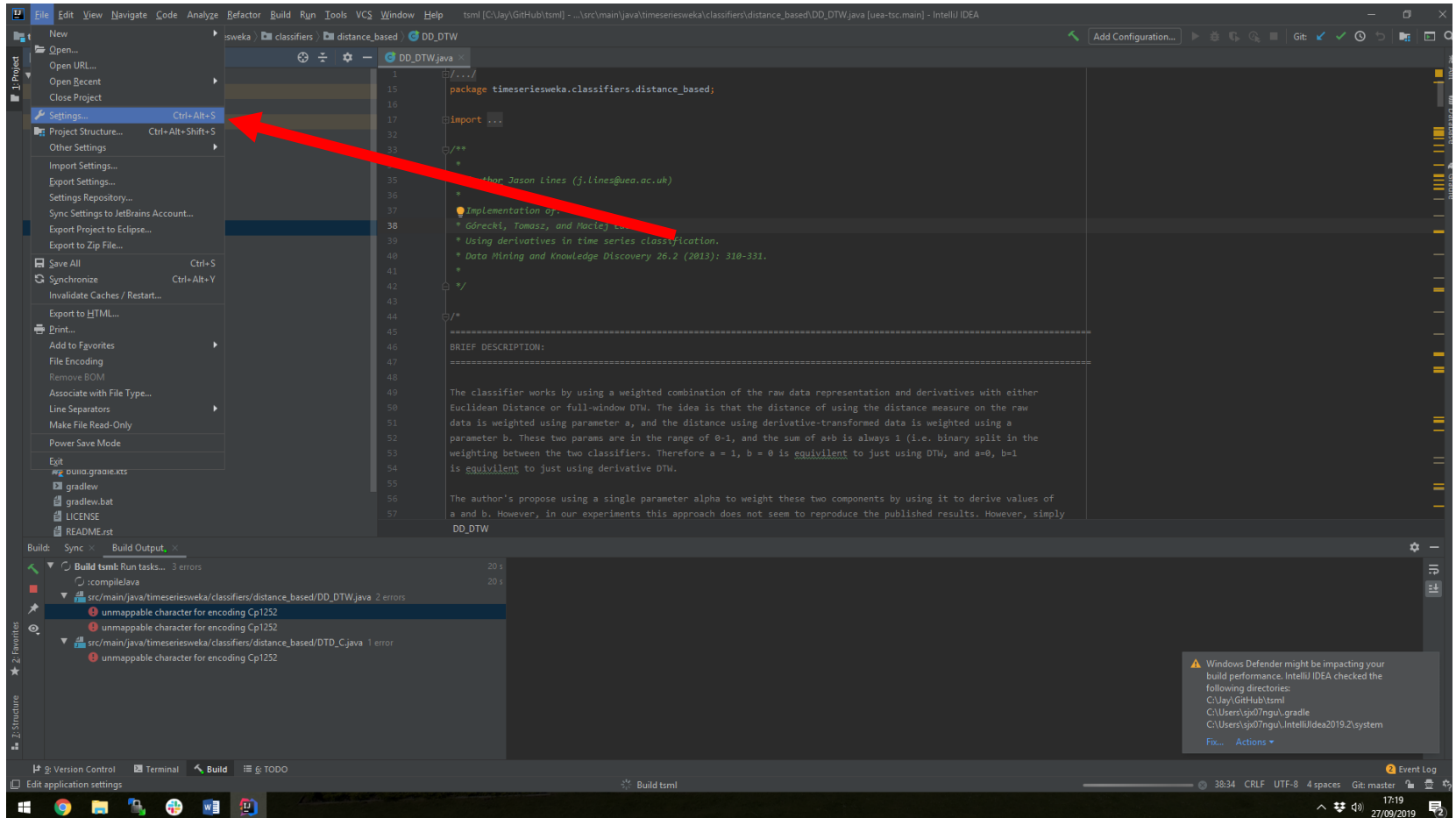
1. Compile output should now be seen here

2. Tick the box for Include in project build

3. Press OK



Click File -> Settings...

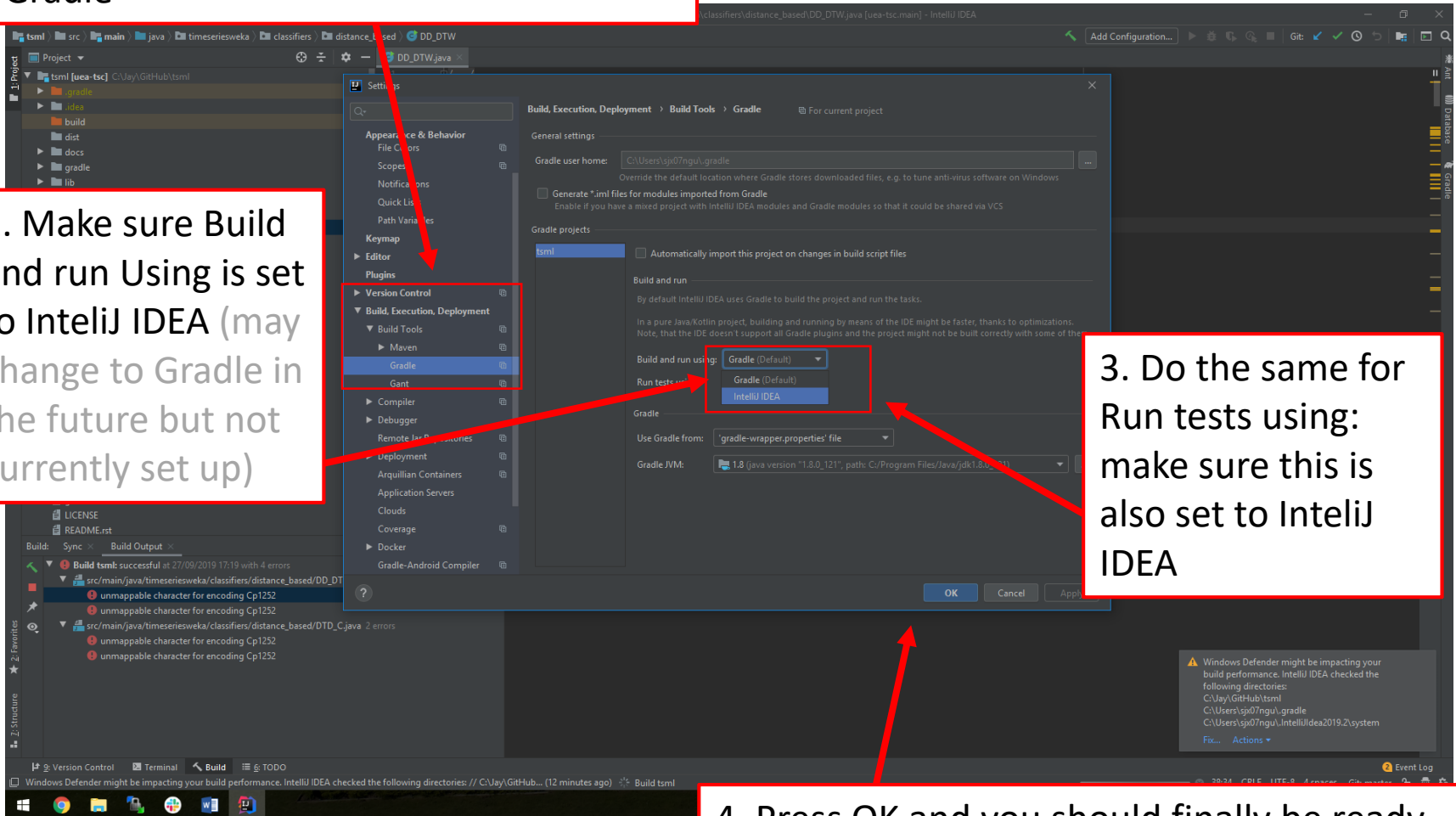


1. Go to
Build, Execution, Deployment ->
Build Tools ->
Gradle

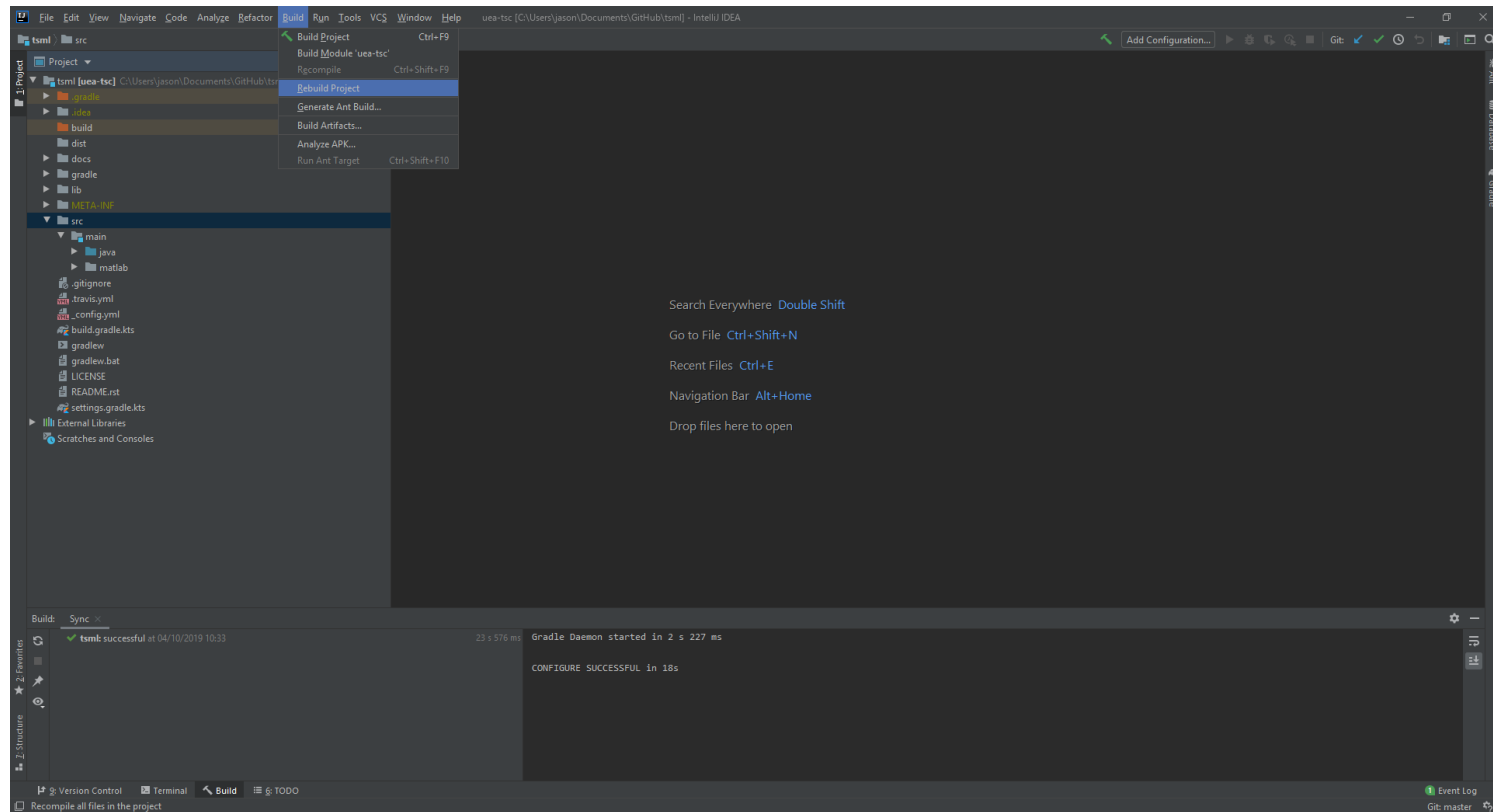
2. Make sure Build
and run Using is set
to IntelliJ IDEA (may
change to Gradle in
the future but not
currently set up)

3. Do the same for
Run tests using:
make sure this is
also set to IntelliJ
IDEA

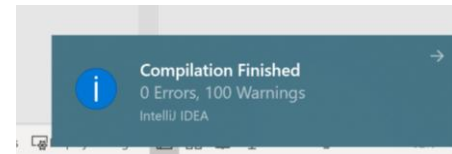
4. Press OK and you should finally be ready
to build!



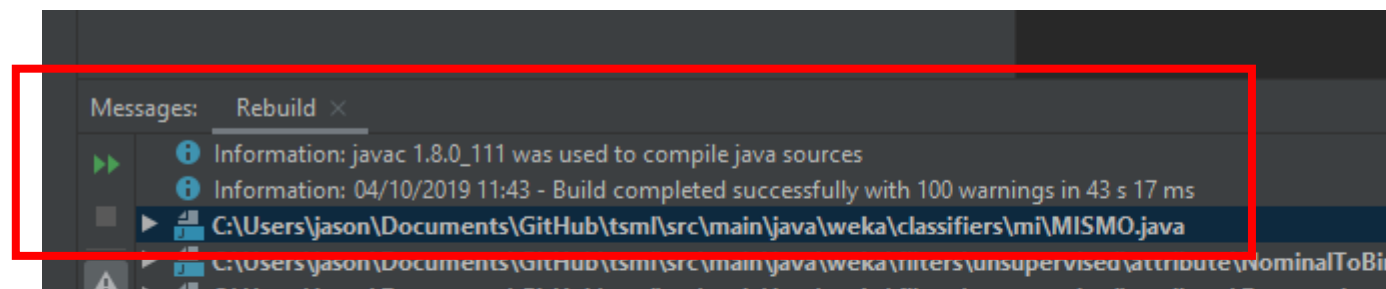
1. Click Build -> Rebuild Project




2. On completion of the build you may get a windows notification:



3. And also confirmed in the messages within IntelliJ:



FINALLY, you should now have a .jar in the location that you specified for the output

Clipboard		Organise		New		Open		Select	
<div>📁 > This PC > Documents > GitHub > tsml > dist</div>									
Name ^		Date modified		Type		Size			
 TimeSeriesClassification.jar		04/10/2019 11:43		Executable Jar File		43,367 KB			

Note that this is a “fat JAR” and you do not need to copy libs to the cluster, other machines, etc. to run directly from the jar