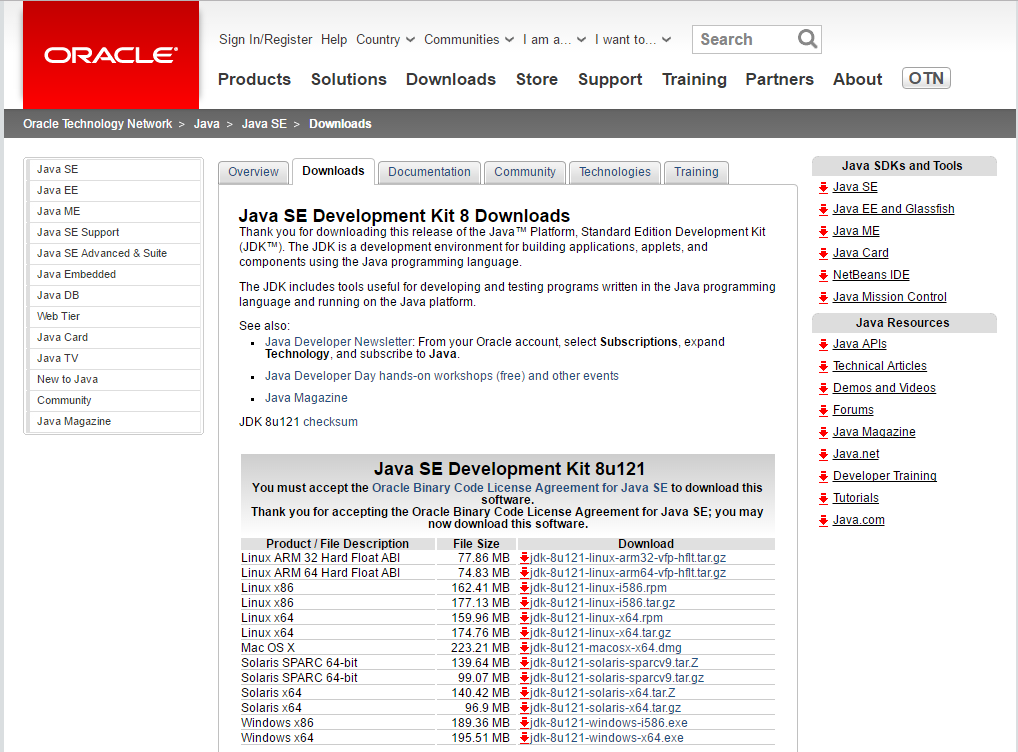
# How to Run Solution

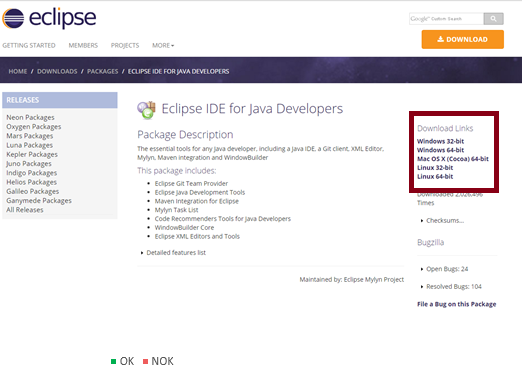
Please find below instructions on how to run the automated tests.

In the end of the document is also a brief description on how we can create easily additional automated tests.

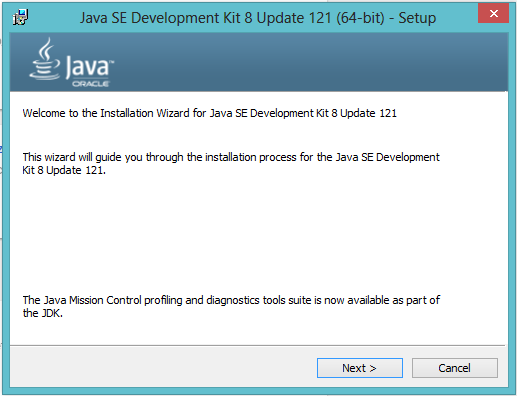
1 - Go to http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html and download java jdk accordingly with you system.

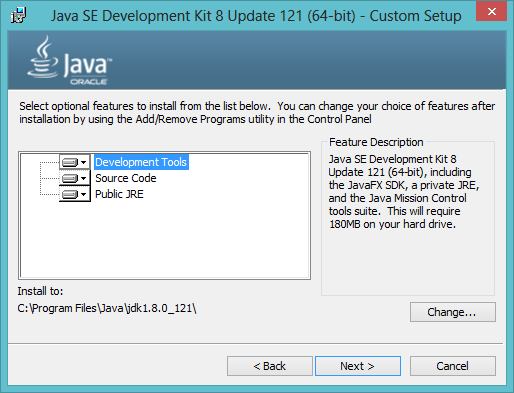


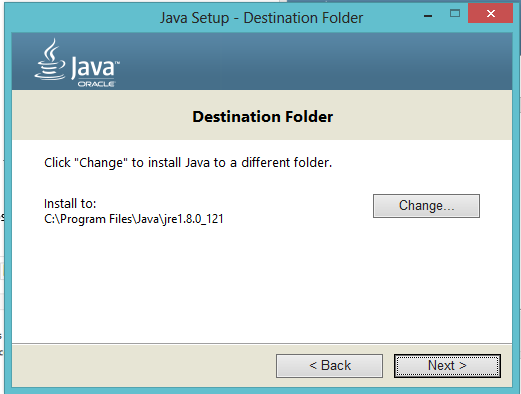
2 - Go to http://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/marsr and download eclipse, accordingly with you system, from the right side "Download Links" section.



3 - Install Java JDK with the default options



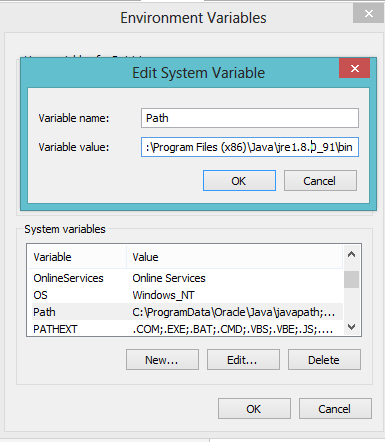




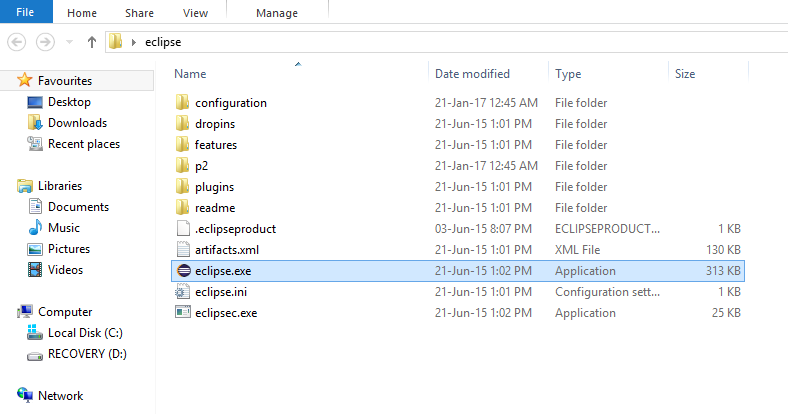


4 – Add installation folder in the Path under “Environment Variables” 🡪 “Systems Variables”:

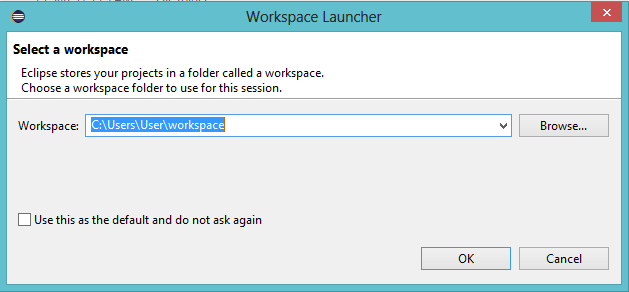
*C:\Program Files (x86)\Java\jre1.8.0\_91\bin*



4 - Unzip Eclipse to a chosen folder



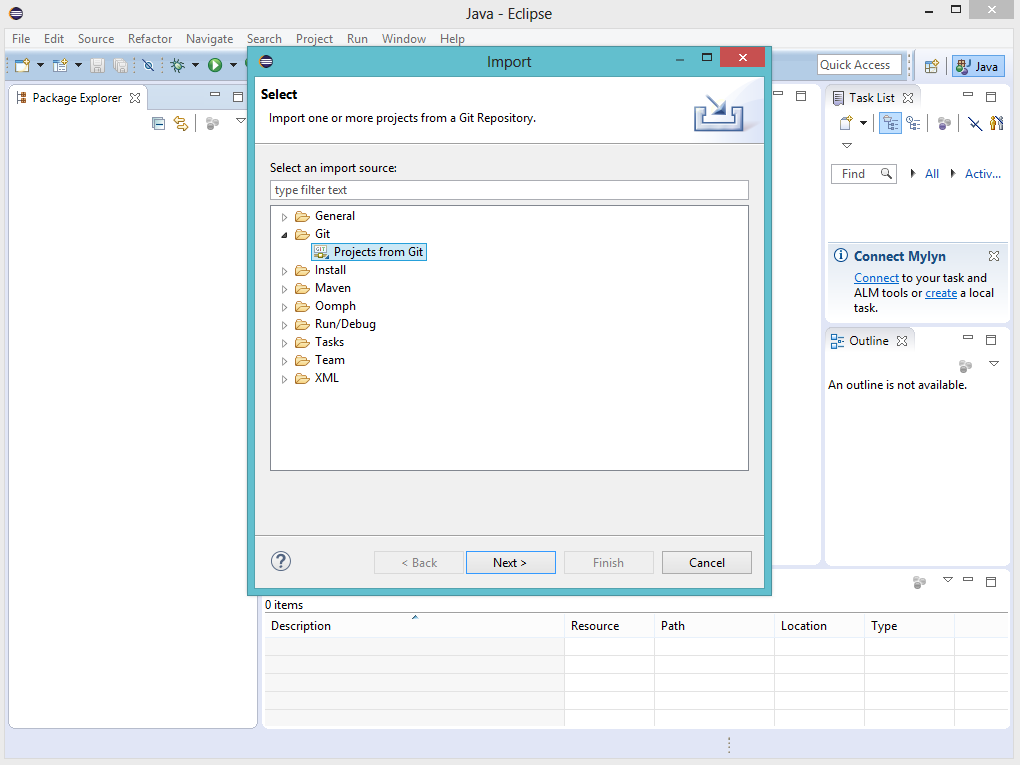
5 - Open eclipse.exe and chose a workspace destination folder



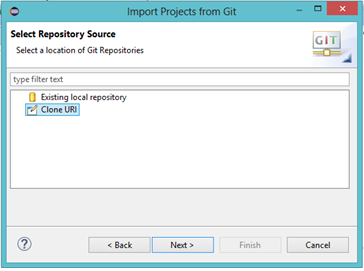
6- Click on the Workbench icon



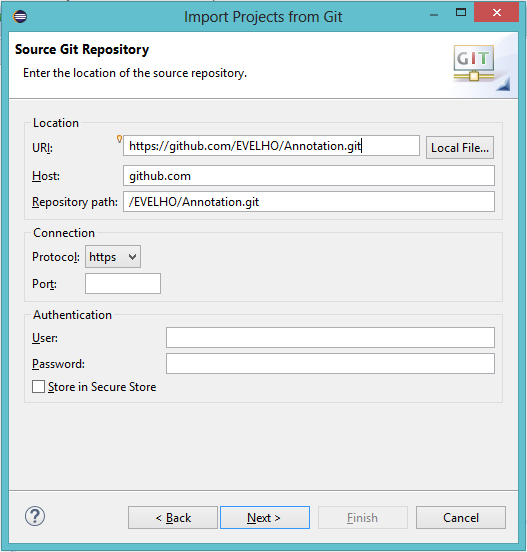
7 – Go to File🡪 Import 🡪 Git 🡪 Projects from Git and click the “Next” button



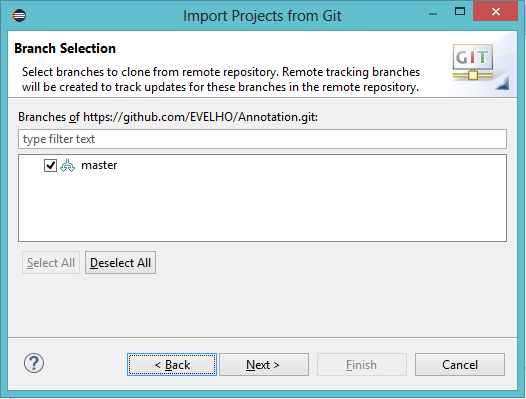
8 – Select “Clone URI” and click the “Next” Button



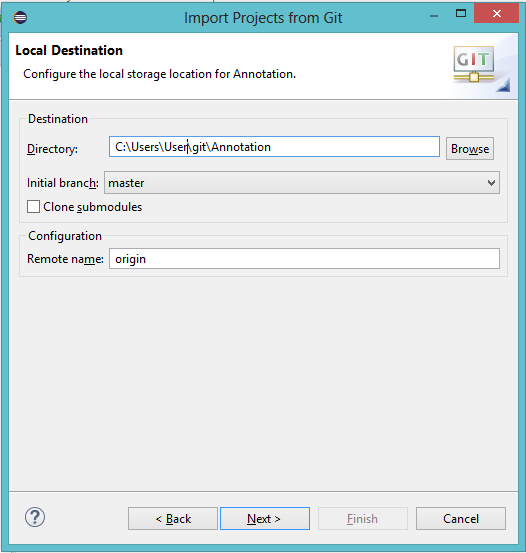
9 – Set URI = <https://github.com/EVELHO/Annotation.git> and hit the “Next” button:



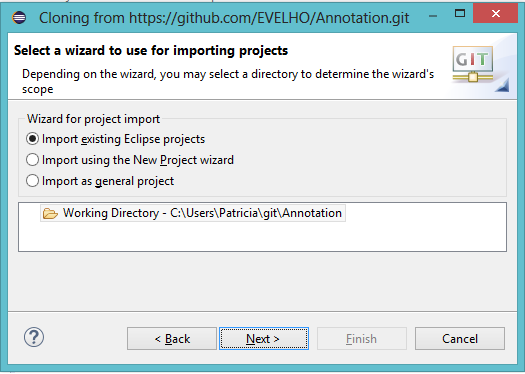
10 – Select “Master Branch” and hit the “Next” Button



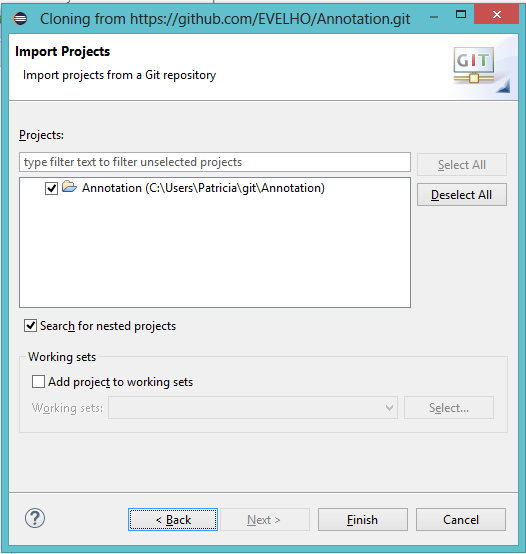
11 – Select the Directory Destination and hit “Next” button



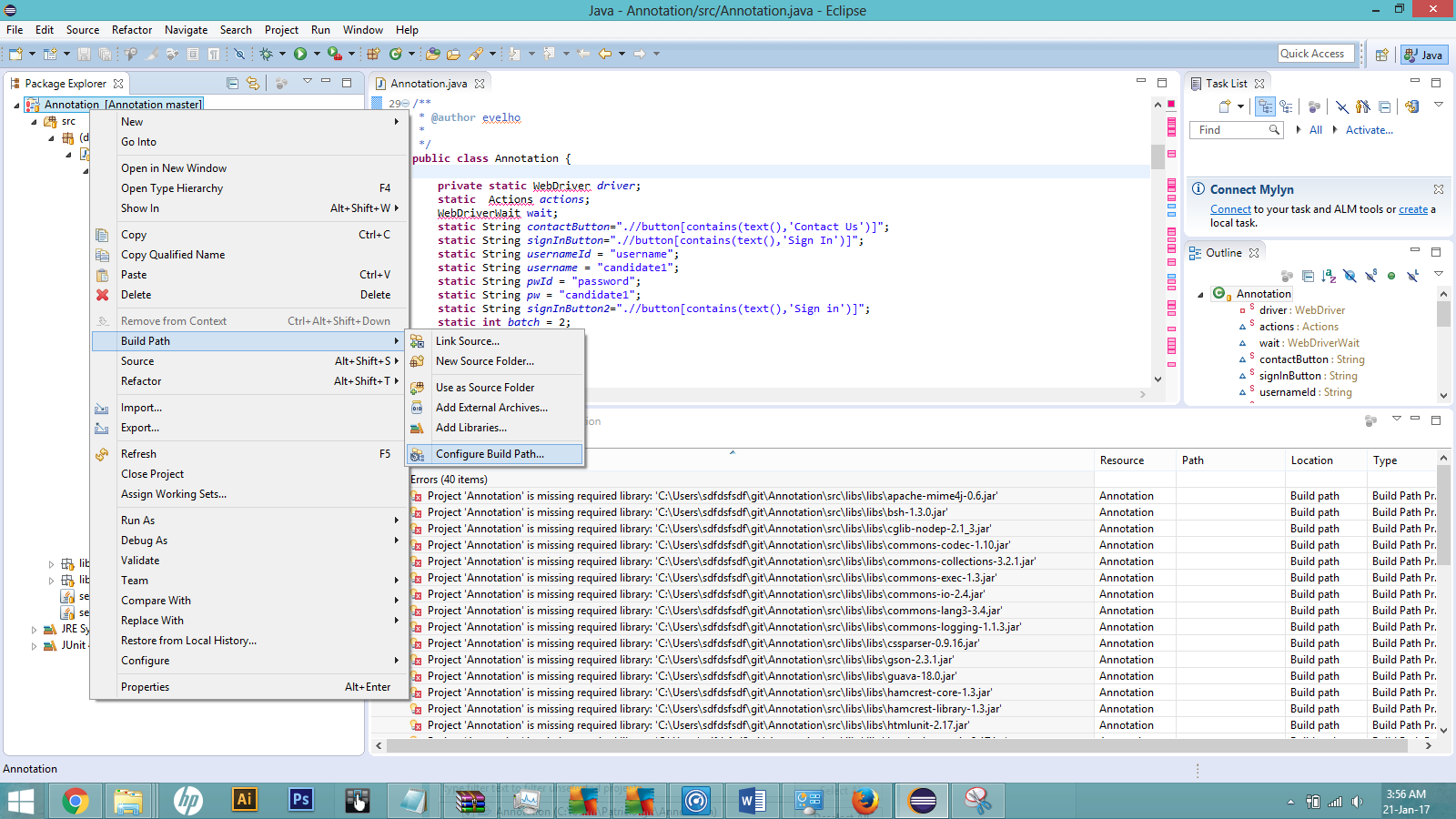
12 – Select the “Import existing Eclipse projects” option and hit “Next”



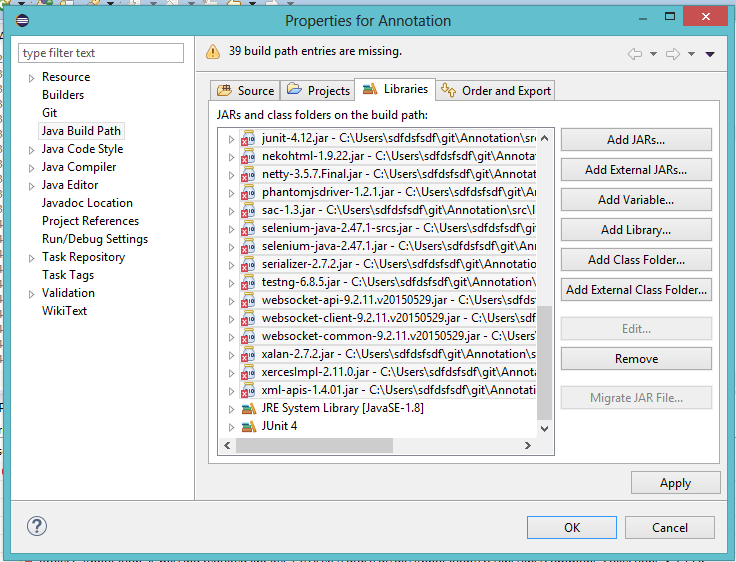
13 – Click on the “Finish” button



14 - Right Click on the “Project” 🡪 “Build Path” 🡪 “Configure Build Path”

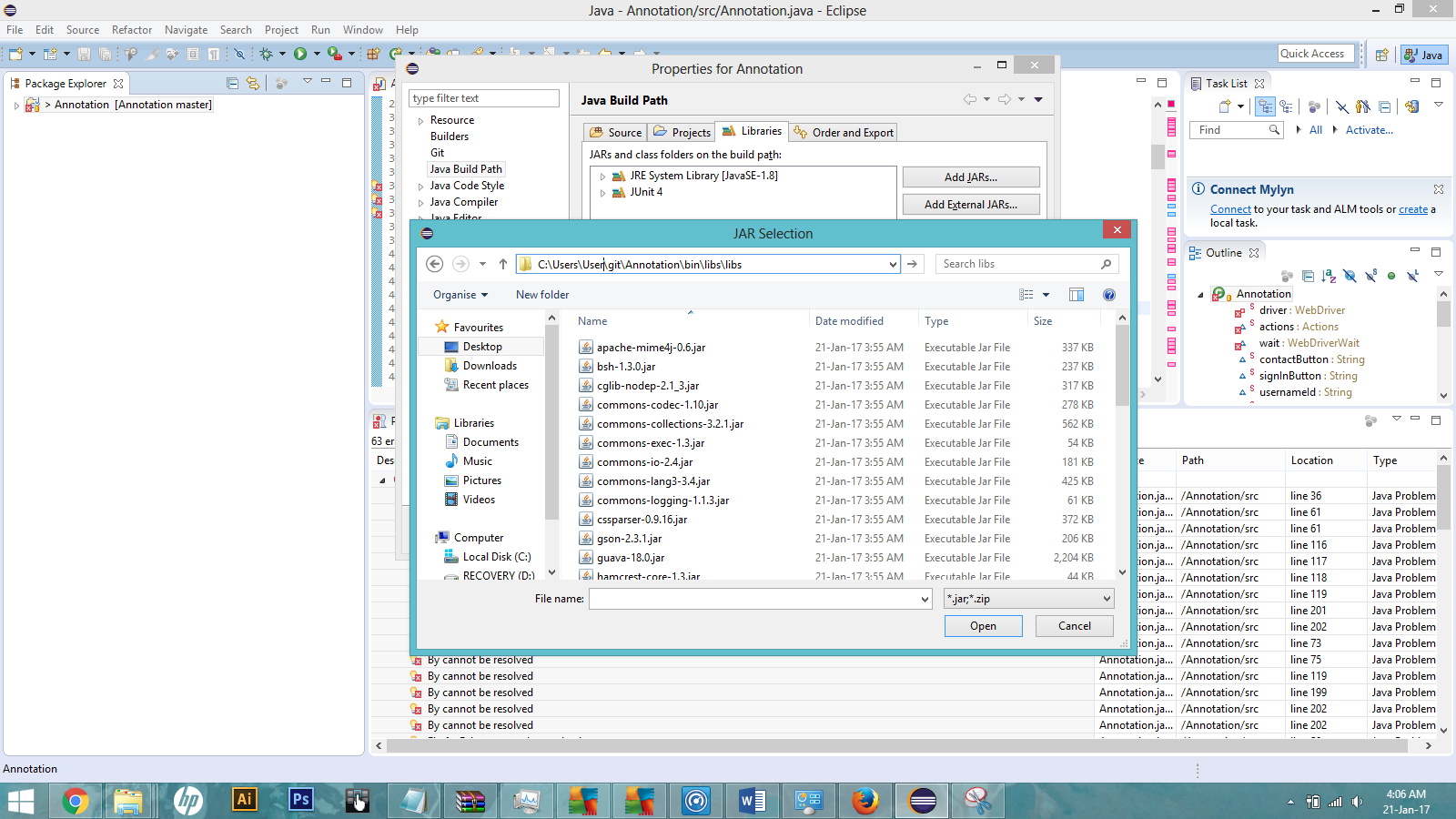


15 – Remove all the missing JARs that are missing (red “x” icon) – select with shift+click. Click on the “Apply” button.

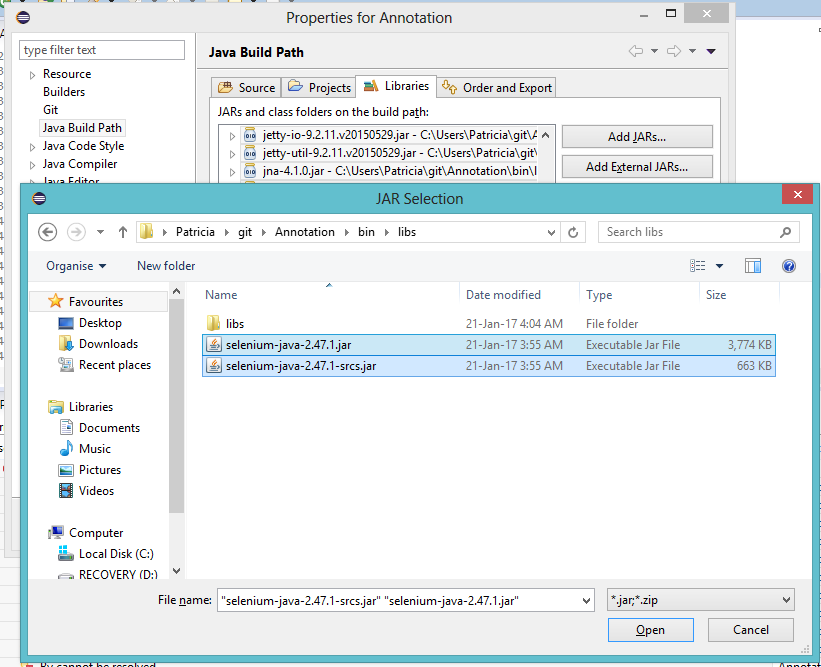


­ 16 – Click on Add external JARs 🡪 go to the folder where you cloned the git repo (ex. **C:\Users\User\git\Annotation)** and browse C:\Users\User\git\Annotation \bin\libs\libs.

Add all jars – select them all with ctrl + shift .

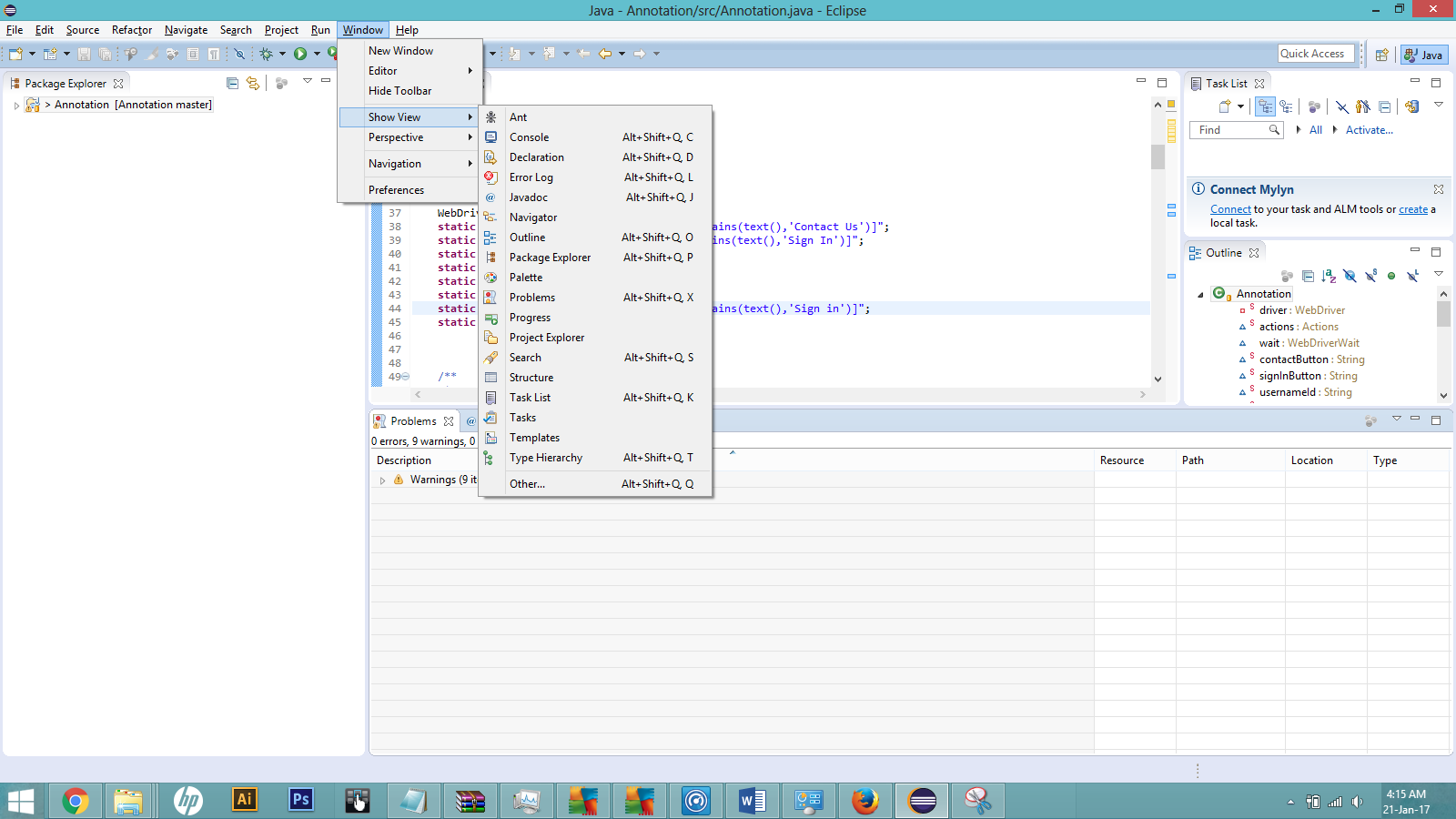


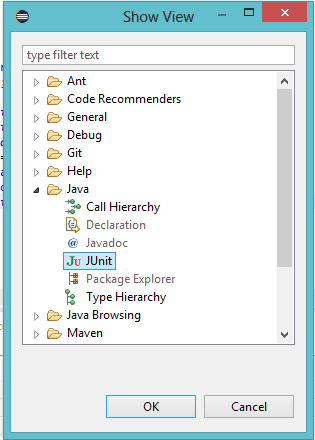
20 – Click again on “Add External Jars”, navigate to “C:\Users\User\git\Annotation\bin\libs” and add the selenium jars.



21 Click on “Apply” and then OK. Now we are able to run the tests.

Go to Window 🡪 Show View 🡪 Other and then, in the new window Java 🡪 JUnit. Press OK button.





22 – Check if you have installed Firefox version 40.0.2 – otherwise uninstall current version and Download Firefox 40.0.2 <http://filehippo.com/download_firefox/62757/>

23 – Now you are ready to run the automated tests. If you want you can create some Unit tests on the application, it’s somewhat simple:

@Test

**public** **void** testA() {

// **TODO** Auto-generated method stub

**try**{

*login*();

*selectBatch*(5);

*createAnnotation*(4);

*selectError*("False Friend");

*selectSeverity*(1);

*finalizeAnnotation*("add");

*assertFalse*(*QT21*.equals(*find*("xpath", "/html/body/div/div[1]/div/div[1]/li[2]").getText()));

}

**catch**(Exception e){

e.printStackTrace();

}

## Brief description on the methods we can use to create tests:

Login() – performs the login, as only one account is available I set the values directly as class variables.

selectBatch(int batchNumber) – Navigate to the a batch indicated by the number.

*createAnnotation*(int translationTaskPosition) – Create an annotation of half of the text on the task.

*selectError*("String error") – this method has as input the error type to be set.

*finalizeAnnotation*("String Action") – Adds (Add) or clears (Exit) the current annotation

*assertFalse*(*QT21*.equals(*find*("xpath", "/html/body/div/div[1]/div/div[1]/li[2]").getText())); - On here I was just checking if the QT21 changed when creating the annotations. Just checks if they are different after the new Annotation.

**PS. Do not forget of the @Test notation right above the method.**