1) Install Java SE 8

from

https://www.oracle.com/java/technologies/javase/javase8u211-later-archive-downloads.html

find in page the string 8u291

then download

Java SE Development Kit 8u291

for windows 10 64 bit

jdk-8u291-windows-x64.exe

Pay attention!

You need to create an Oracle account, for downloading

jdk-8u202-windows-x64.exe

(before downloading you will be redirected to

https://login.oracle.com/mysso/signon.jsp)

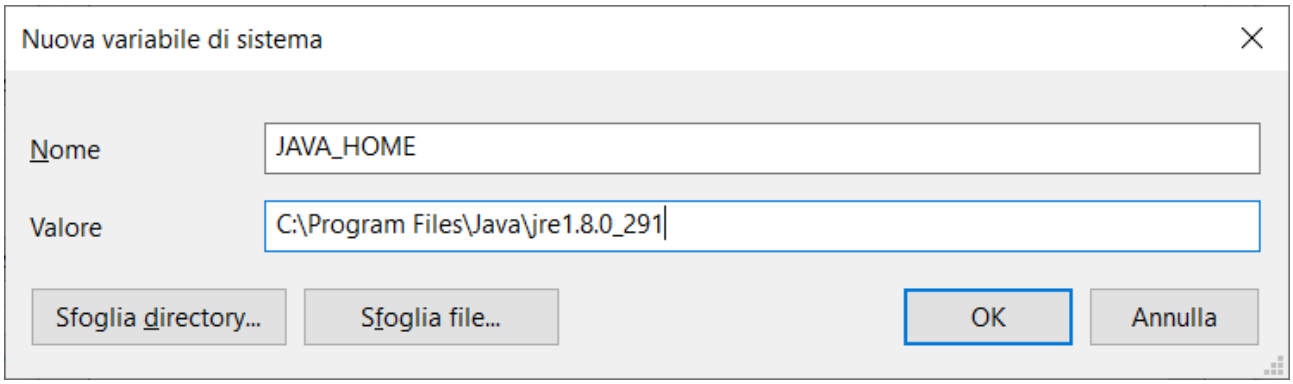
2) Set JAVA_HOME Environment Variable and Java Path on Windows 10

https://www.youtube.com/watch?v=yGxCQisOL1A

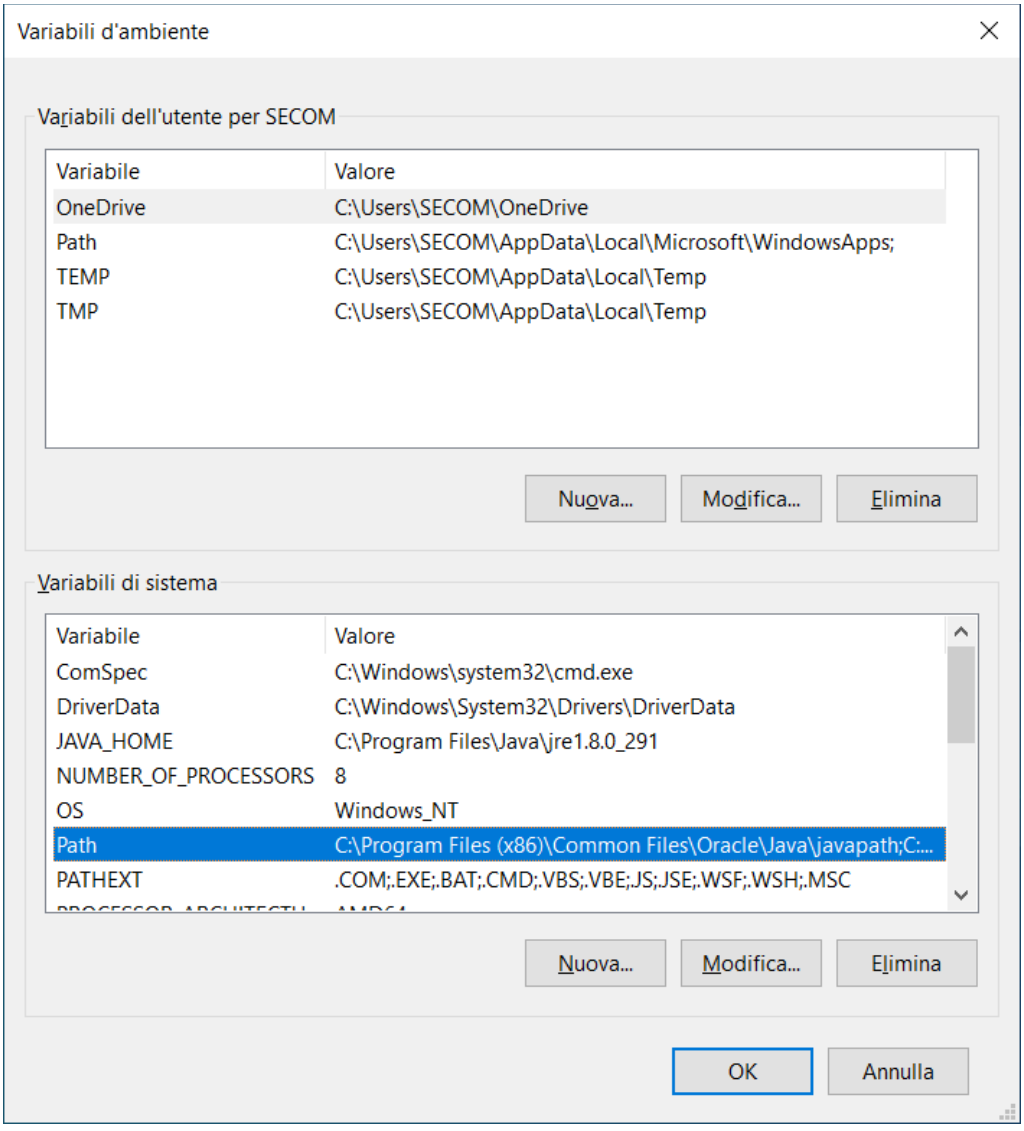
of course the path you have to insert is not the same as the one mentioned in the video:

but refer to this below:

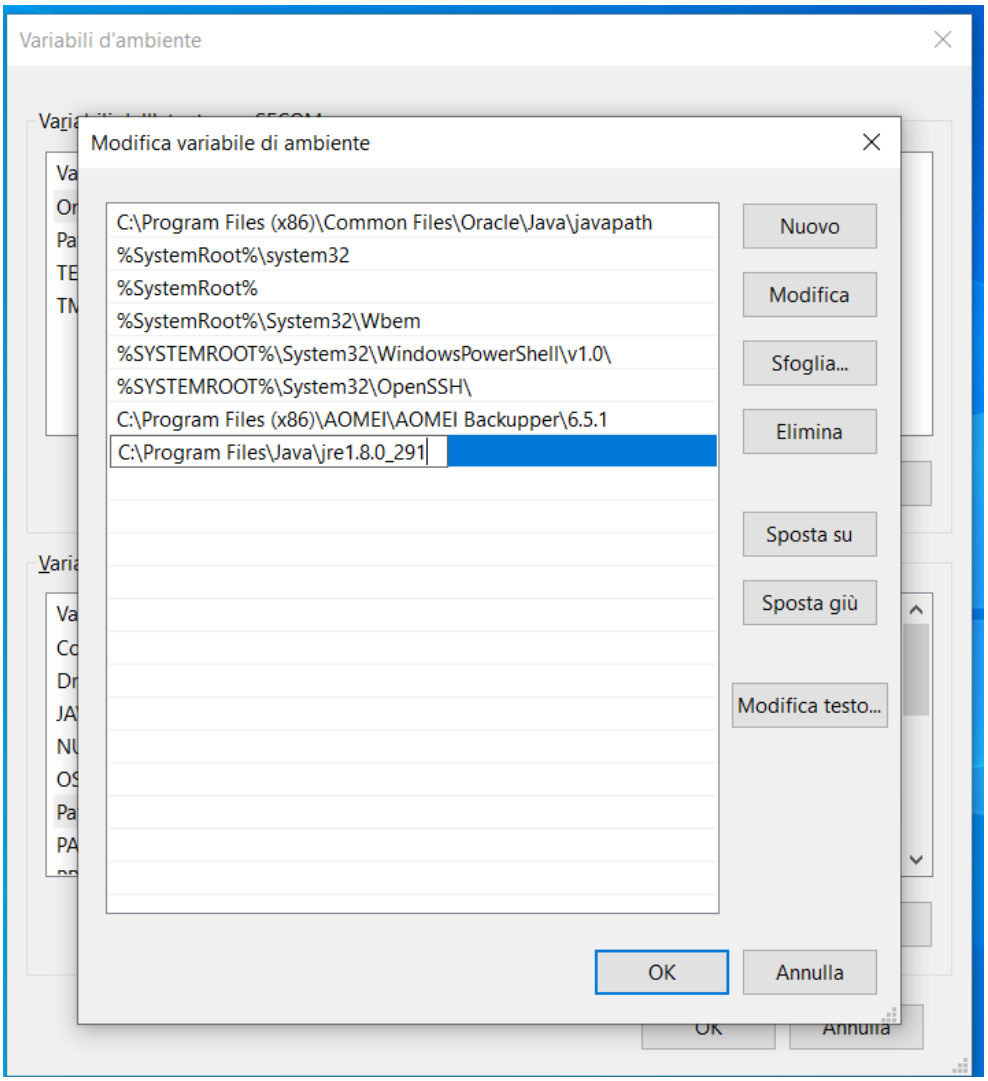
Nuova variabile di sistema (new Environment variable)



Then select Path and then Edit



Then New > Add jre path



To check if Java has been installed and configured correctly, go to CMD mode and exec DOS command

java –version

as shown below

```
C:\Users\SECOM>java -version
java version "1.8.0_291"
Java(TM) SE Runtime Environment (build 1.8.0_291-b10)
Java HotSpot(TM) 64-Bit Server VM (build 25.291-b10, mixed mode)
```

3) Install Eclipse on Windows 10

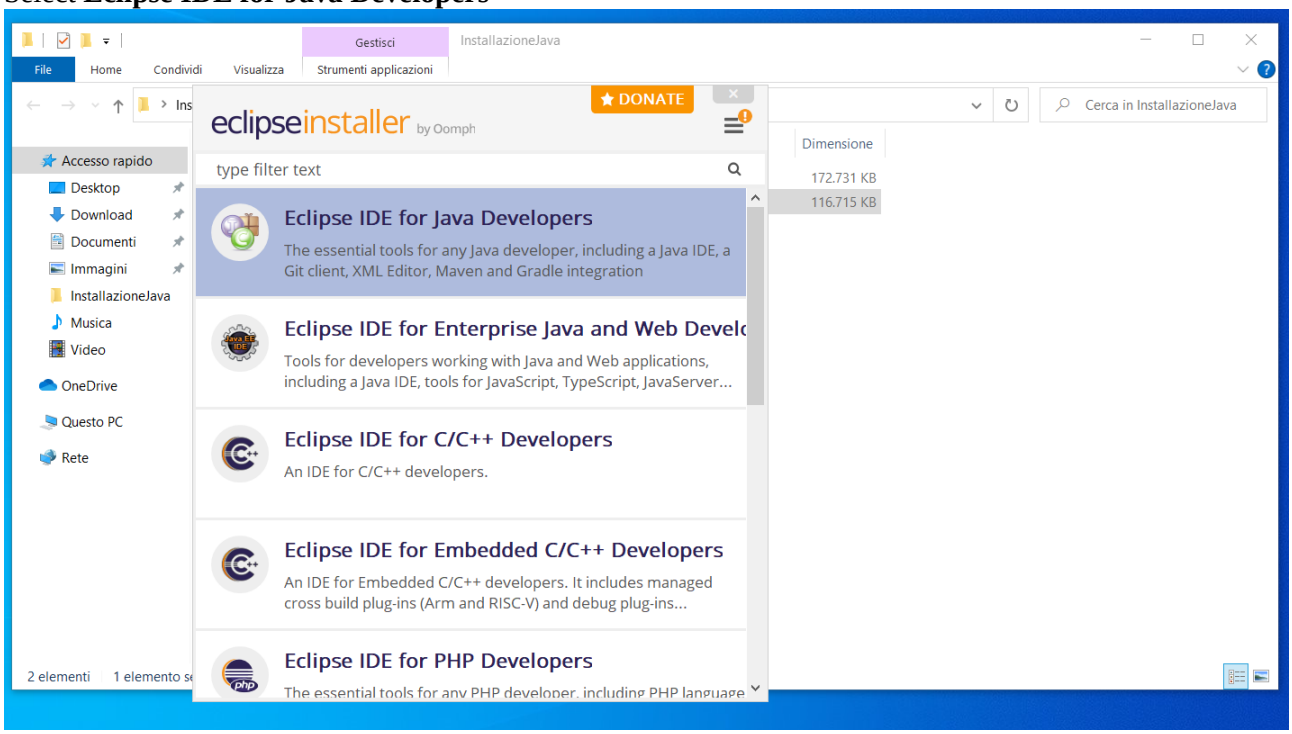
From https://www.eclipse.org/downloads/packages/release/2021-06/r

Go to the right side of the web page and click on the orange button below this text:

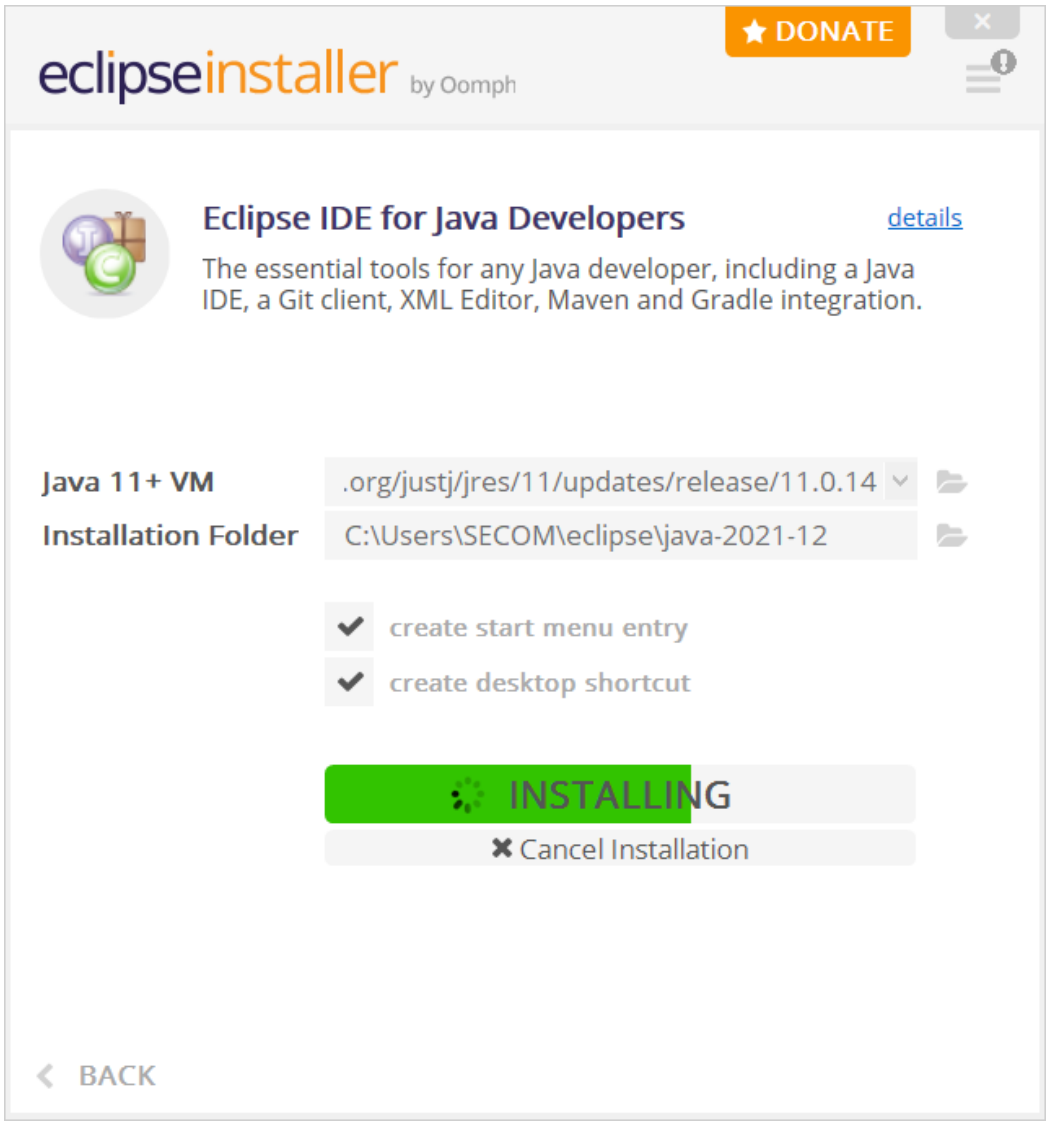
Get Eclipse IDE 2021-12
Install your favorite desktop IDE packages

Then launch (Alway with Administrator user) the installation setup **eclipse-inst-jre-win64.exe**

Select **Eclipse IDE for Java Developers**



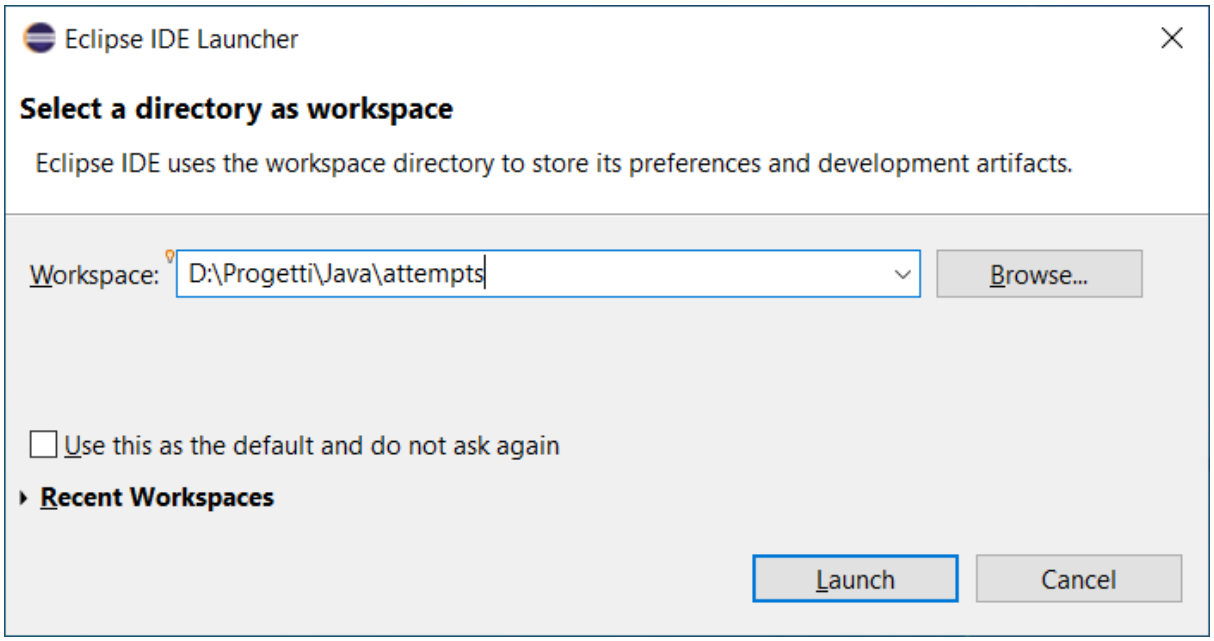
Accept default settings and click on Install



Once installation has finished,

Click on LAUNCH

And then set a test Workspace (I suggest D:\Progetti\Java\attempts) and then go on



Close Introductory page

Welcome to the Eclipse IDE for Java Developers

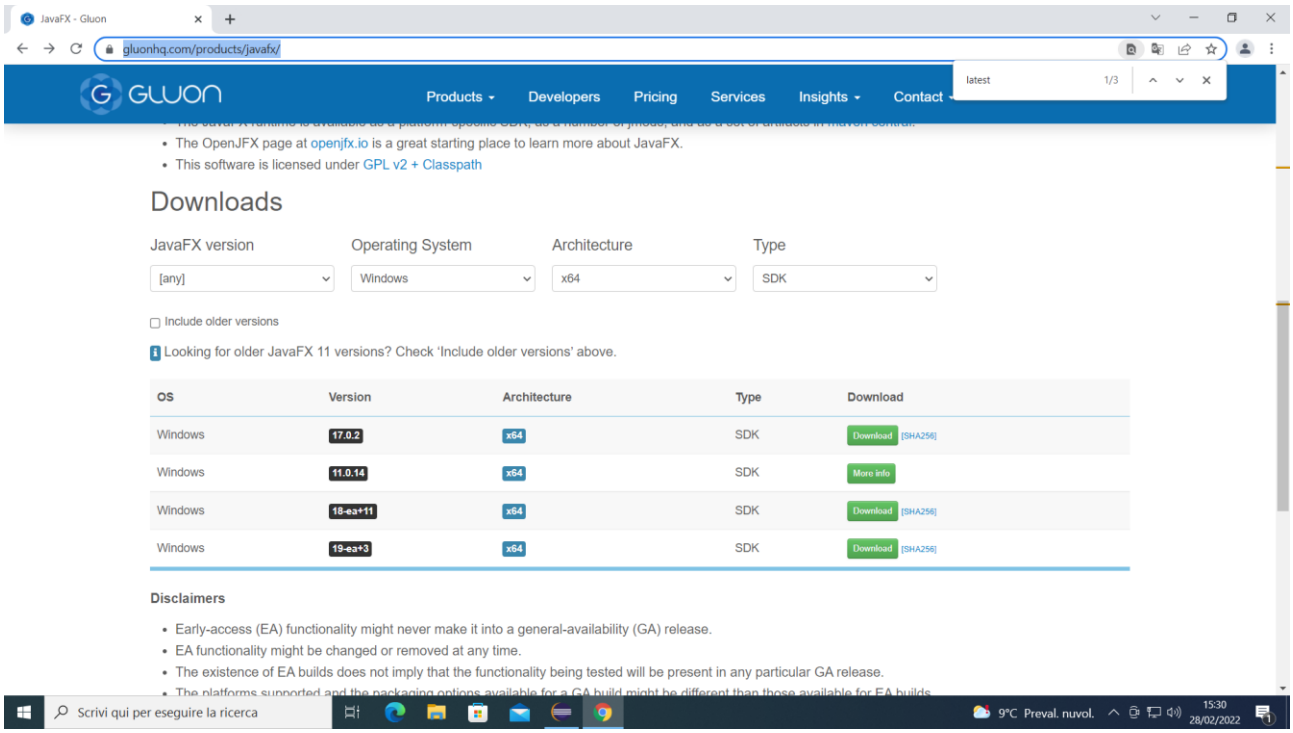
3) Now INSTALL JAVAFX according to the instructions contained in this video:

https://www.youtube.com/watch?v=bC4XB6JAaoU&t=129s

NOTE: you can download Javafx from this link:

https://gluonhq.com/products/javafx/

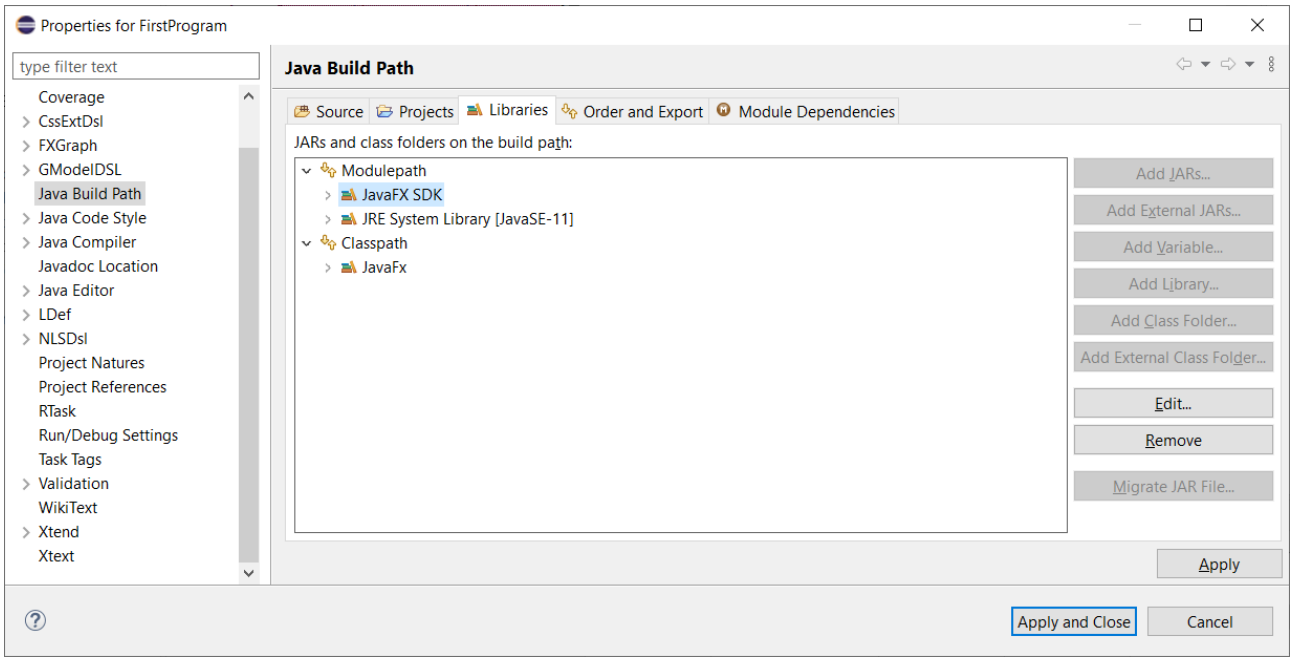
then filter how indicated below:



And download version 17.0.2 (javafx-sdk-17.0.2)

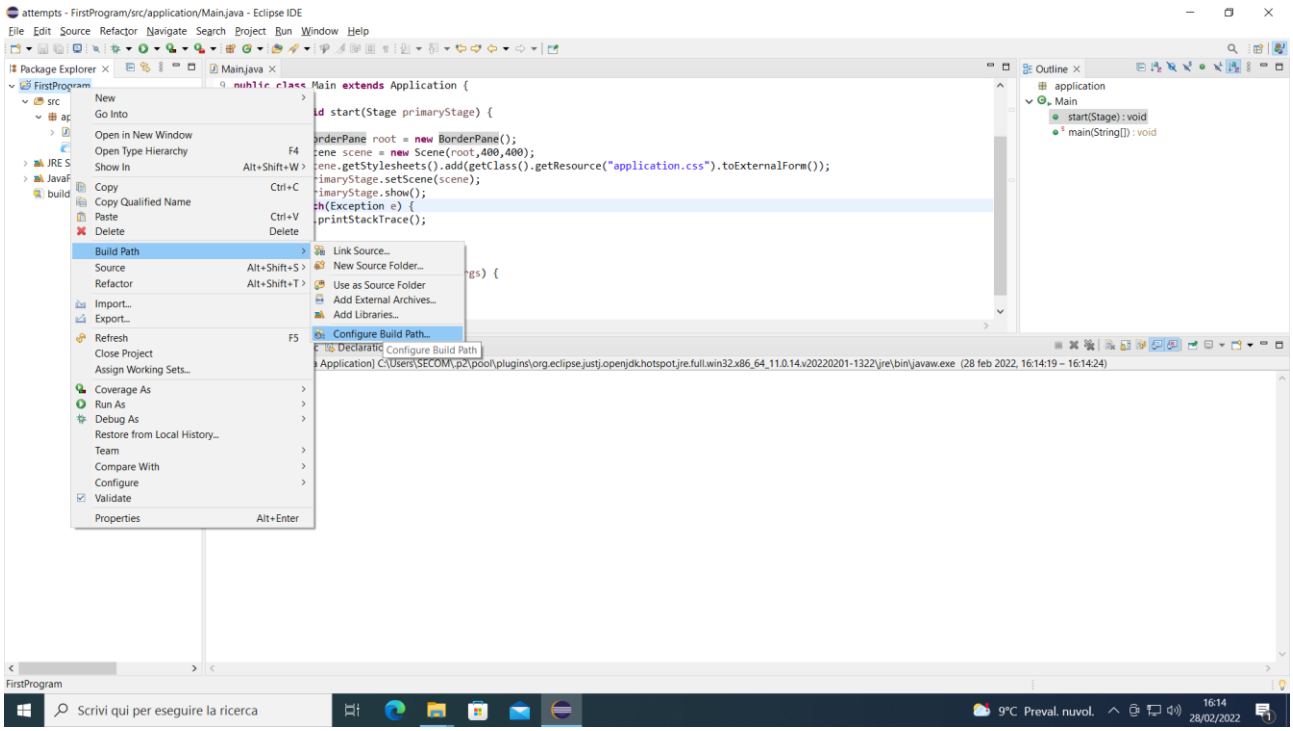
Follow the instructions in the video above.

Remember to remove JavaFX SDK if you find it under Modulepath:



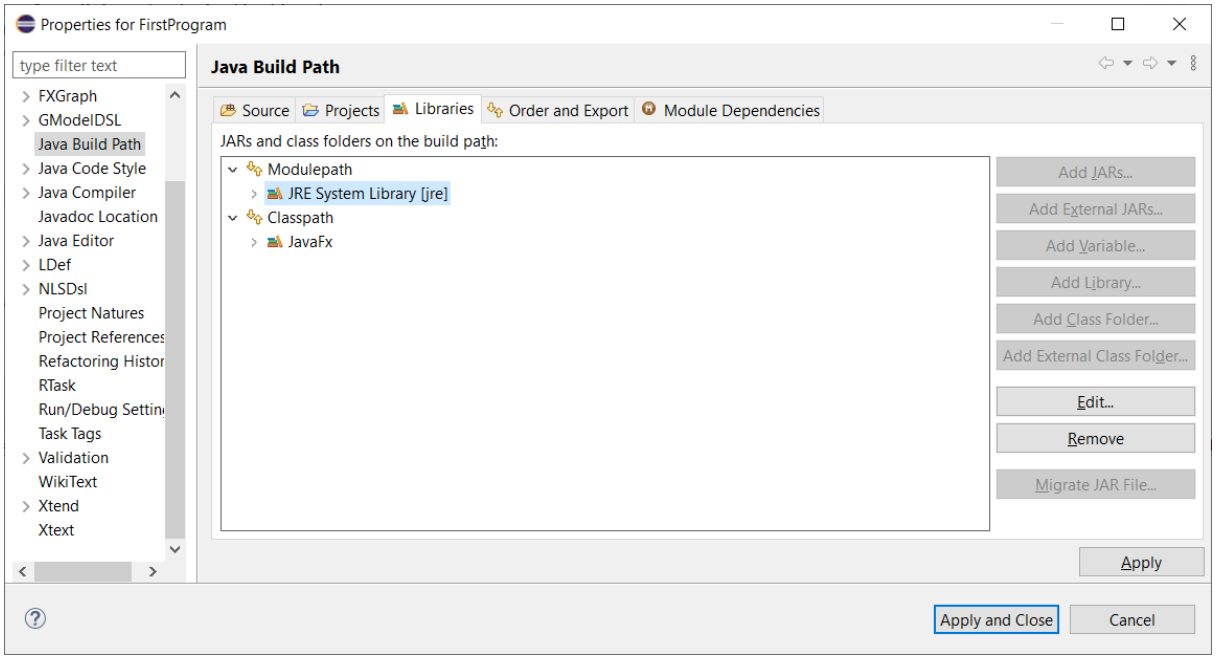
Remember that by default Java 11 is installed, but we decided to use Java 1.8.xx

So, what you need to do is:



Select **Configure Build Path**

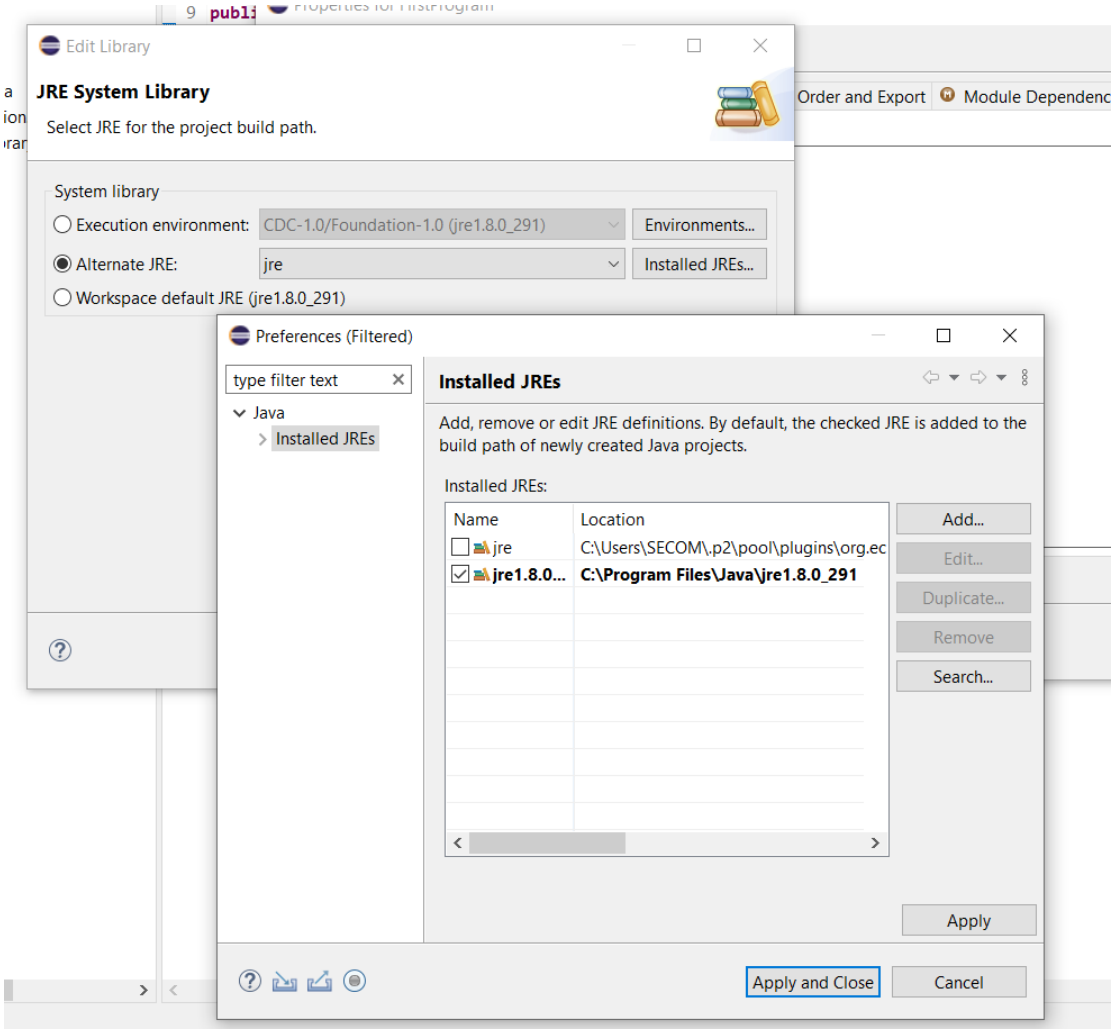
Then change **JRE System Library from 11 to 1.8.**



To do this, you need to select

JRE System Library

Then Click on Edit, select Alternate JRE and select the path to **jre1.8.0_291**.



4) Install Scene Builder

Download Scene Builder from

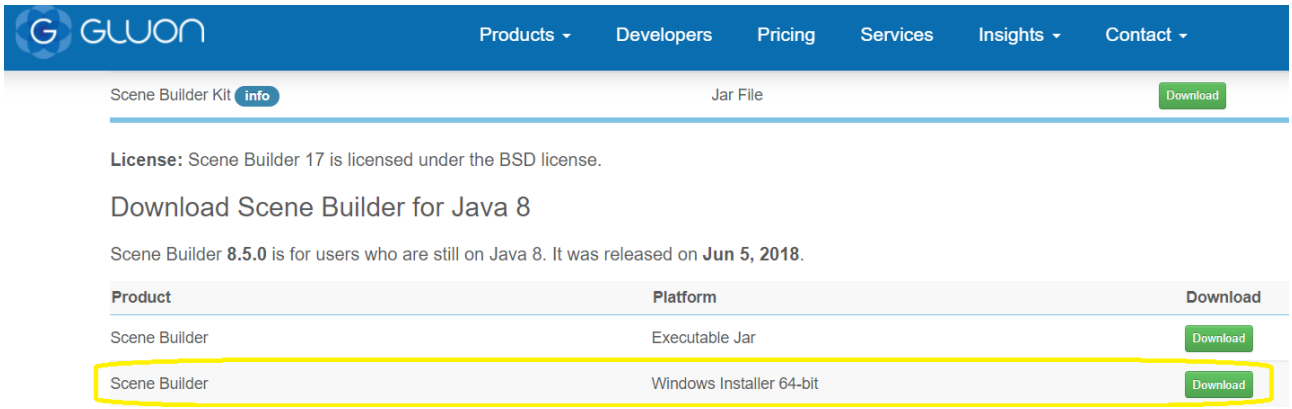
https://gluonhq.com/products/scene-builder/

In the page goto

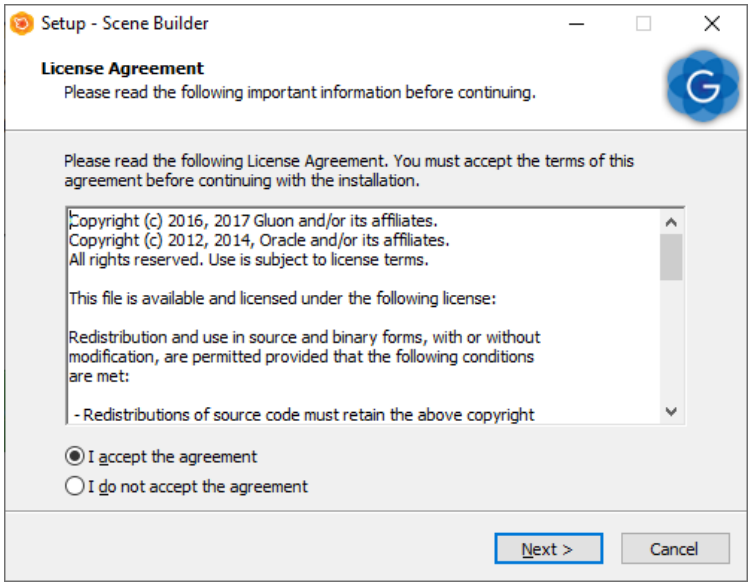
Download Scene Builder for Java 8

Scene Builder **8.5.0** is for users who are still on Java 8. It was released on **Jun 5, 2018**.

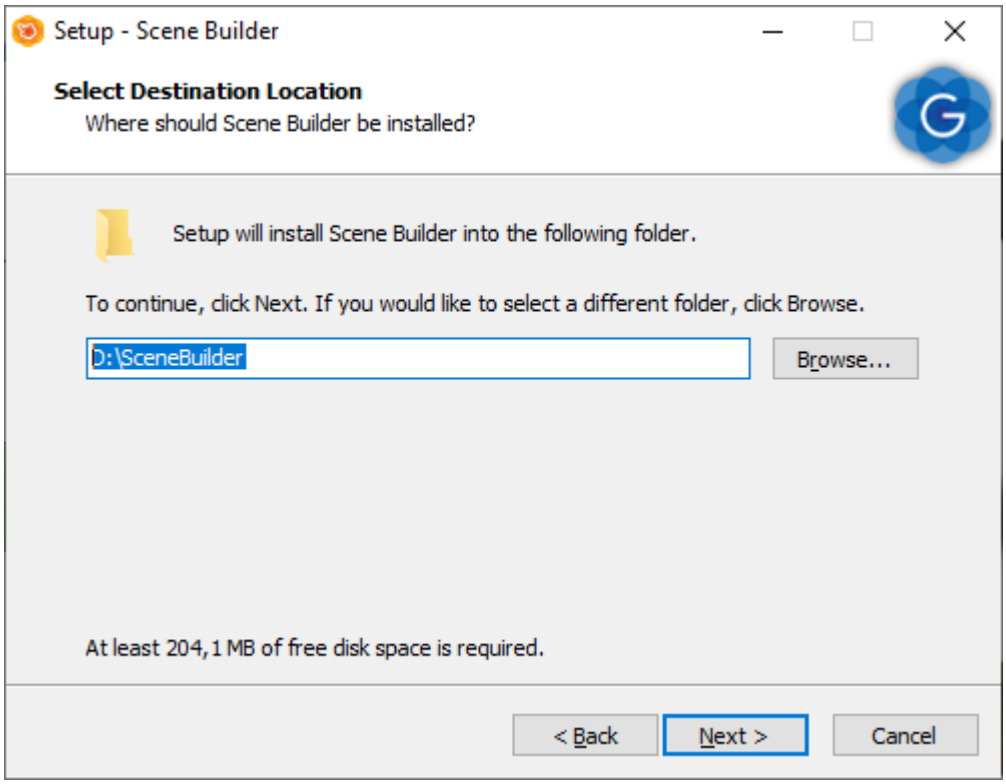
And download Scene Builder Windows Installer 64-bit



Download and install

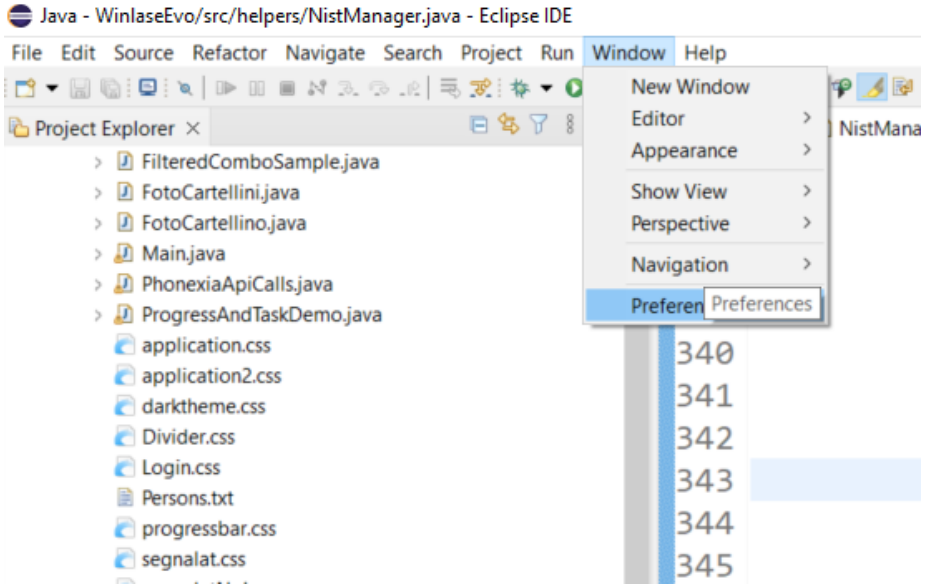


Install it into D:\SceneBuilder

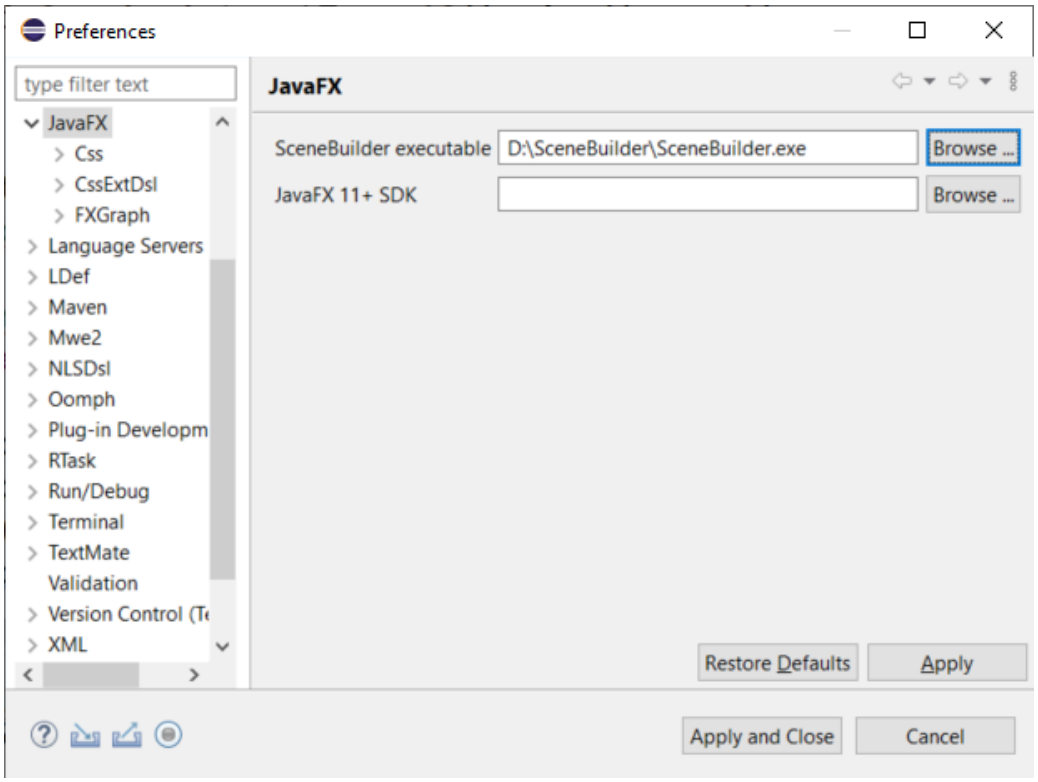


Once SceneBuilder has been donwloaded and installed in D:\SceneBuilder

Open Eclipse and goto Window>Preferences



Then select JavaFx and set the SceneBuilder exe Path



Now you should be able to open an FXML file into a Javafx program.

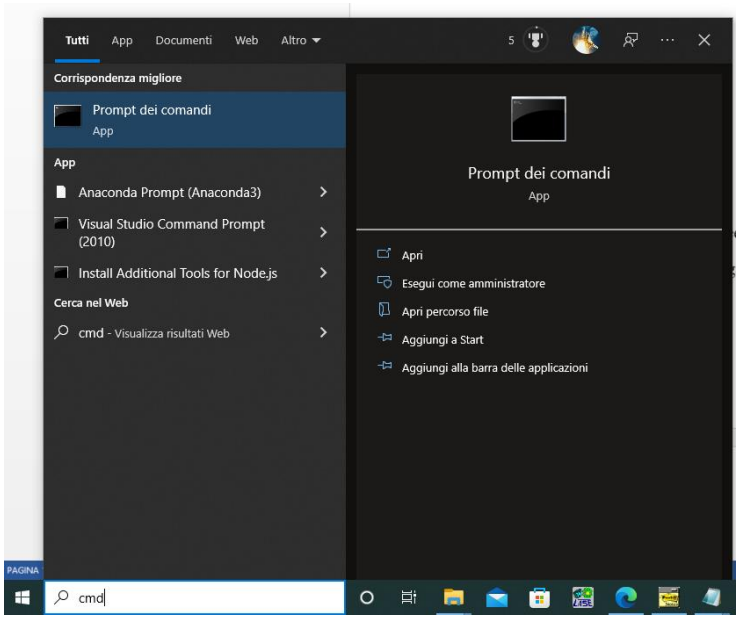
To understand better what a FXML file is, I suggest you to read the following introductions:

https://www.vojtechruzicka.com/javafx-fxml-scene-builder/

https://docs.oracle.com/javase/8/javafx/api/javafx/fxml/doc-files/introduction_to_fxml.html

5) Connect to Secom Git Repository

Open **Prompt** (through CMD command)



check git version using the following command:

git --version

if no Git version is available then install 64-bit Git from

https://git-scm.com/download/win

Once **Git-2.35.1.2-64-bit.exe** file has been downloaded, install it

Then close CMD window and open it again

Check git version

git --version

Now the following string should be returned

git version 2.35.1.windows.2

Then go to D:\ and create the following folder

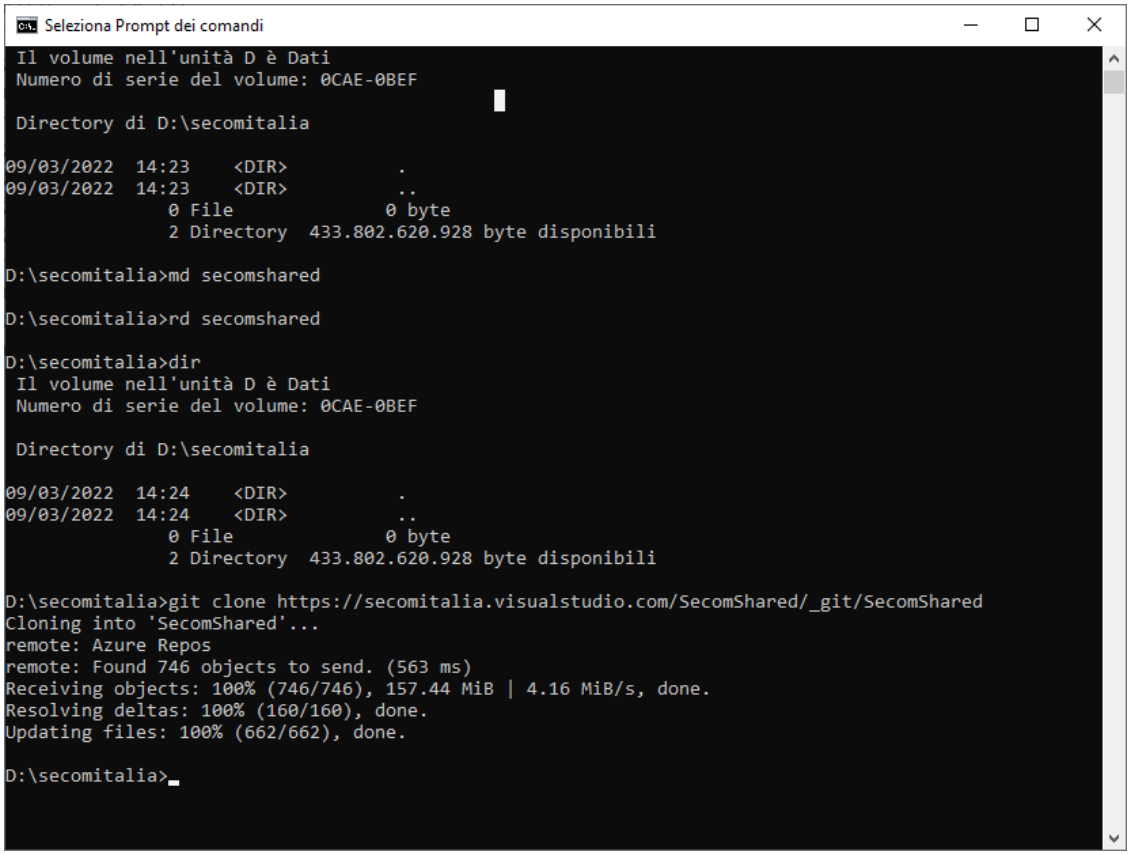
D:\secomitalia

The access SecomShared folder through dos command **cd**

then launch this command

git clone https://secomitalia.visualstudio.com/SecomShared/_git/SecomShared

you will be requested to authenticate



See below the DOS commands to execute (in order)

```
Microsoft Windows [Versione 10.0.19044.1566]
(c) Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\SECOM>git --version
git version 2.35.1.windows.2

C:\Users\SECOM>cd\
```

Open Dos modality (through CMD command)
check git version using the following command:

```
git --version
```

in no git version is available

install 64-bit Git from
https://git-scm.com/download/win

Once Git-2.35.1.2-64-bit.exe file
has been downloaded
install it

close CMD window and open it again
check git version

git --version
Now it should return the following string

git version 2.35.1.windows.2

Now go to D:\
create this path

D:\secomitalia

then do:

cd secomitalia

then launch this command

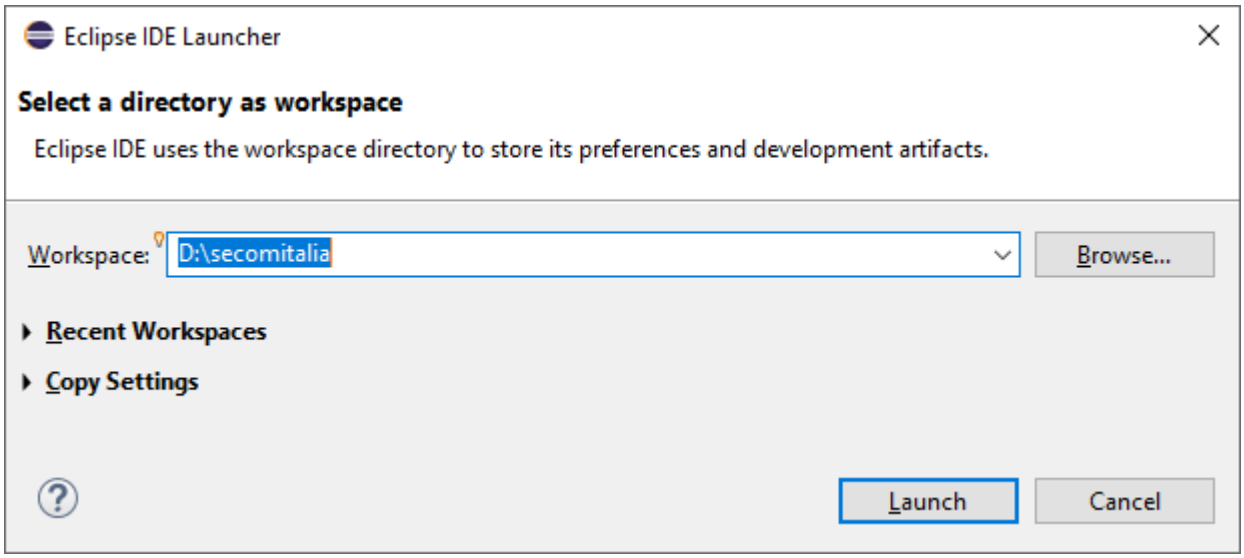
git clone
https://secomitalia.visualstudio.com/SecomShared/_git/SecomShared

you will be requested to authenticate

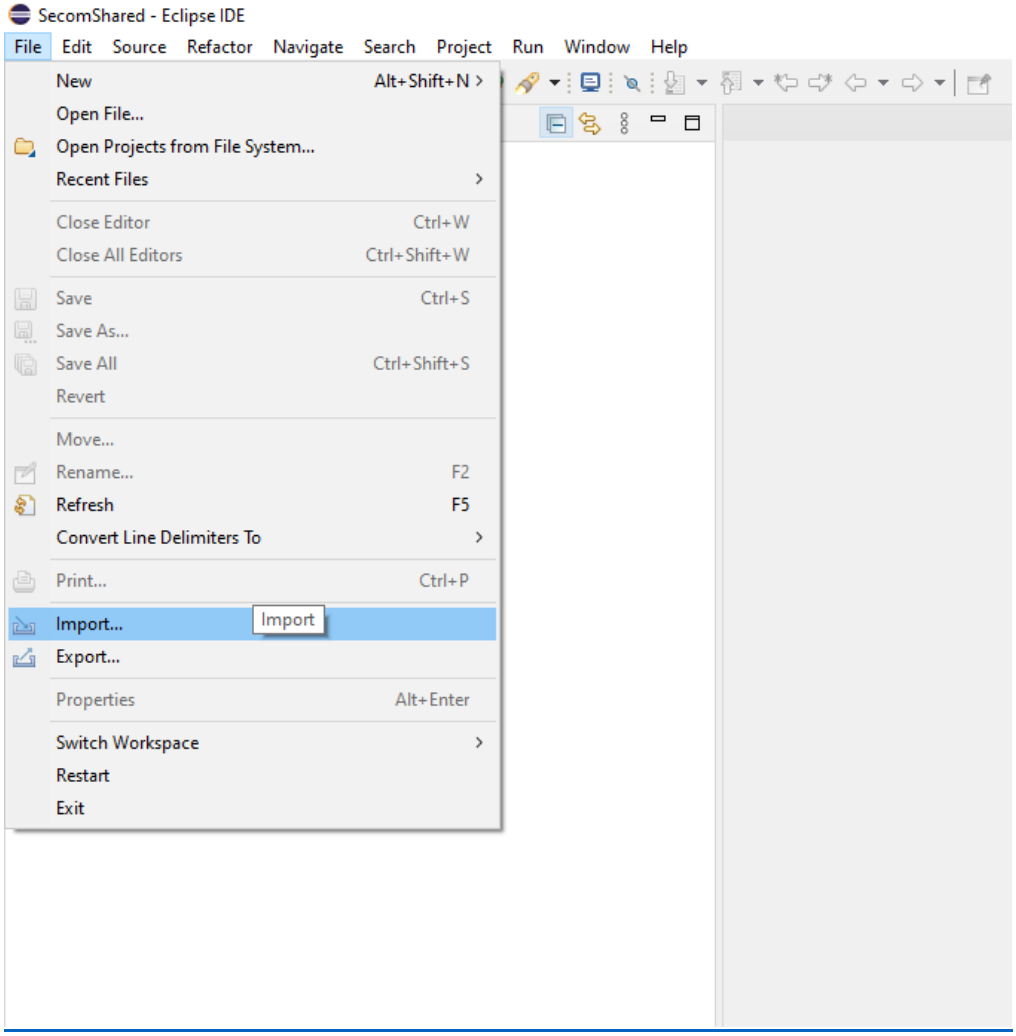
```
C:\secomitalia>git clone
https://secomitalia.visualstudio.com/SecomShared/_git/SecomShared
Cloning into 'secomitalia'...
remote: Azure Repos
remote: Found 410 objects to send. (36 ms)
Receiving objects: 100% (410/410), 100.13 MiB | 4.57 MiB/s, done.
Resolving deltas: 100% (87/87), done.
**********************************
```

Now open eclipse (Run as Administrator, better) and set the following workspace:

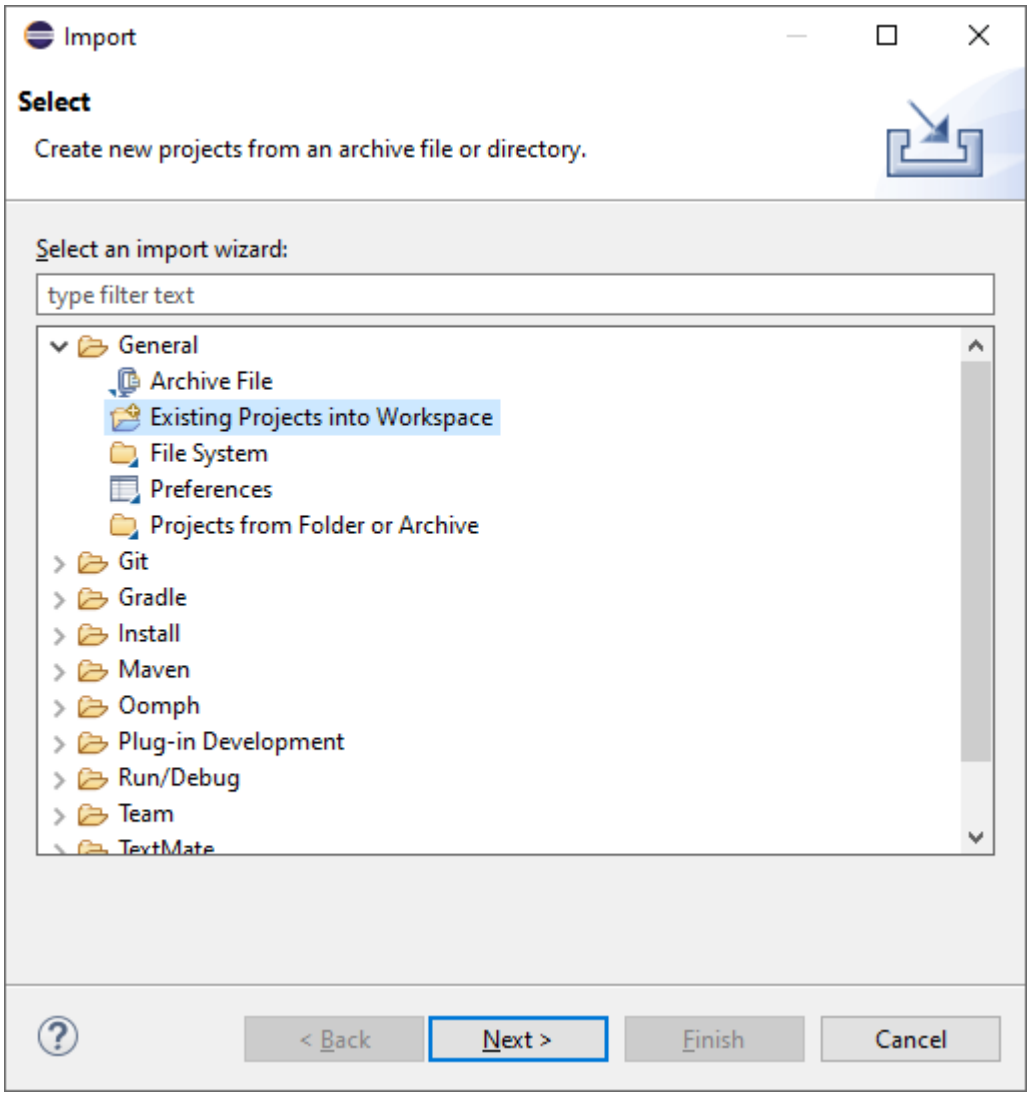
Close Welcome page



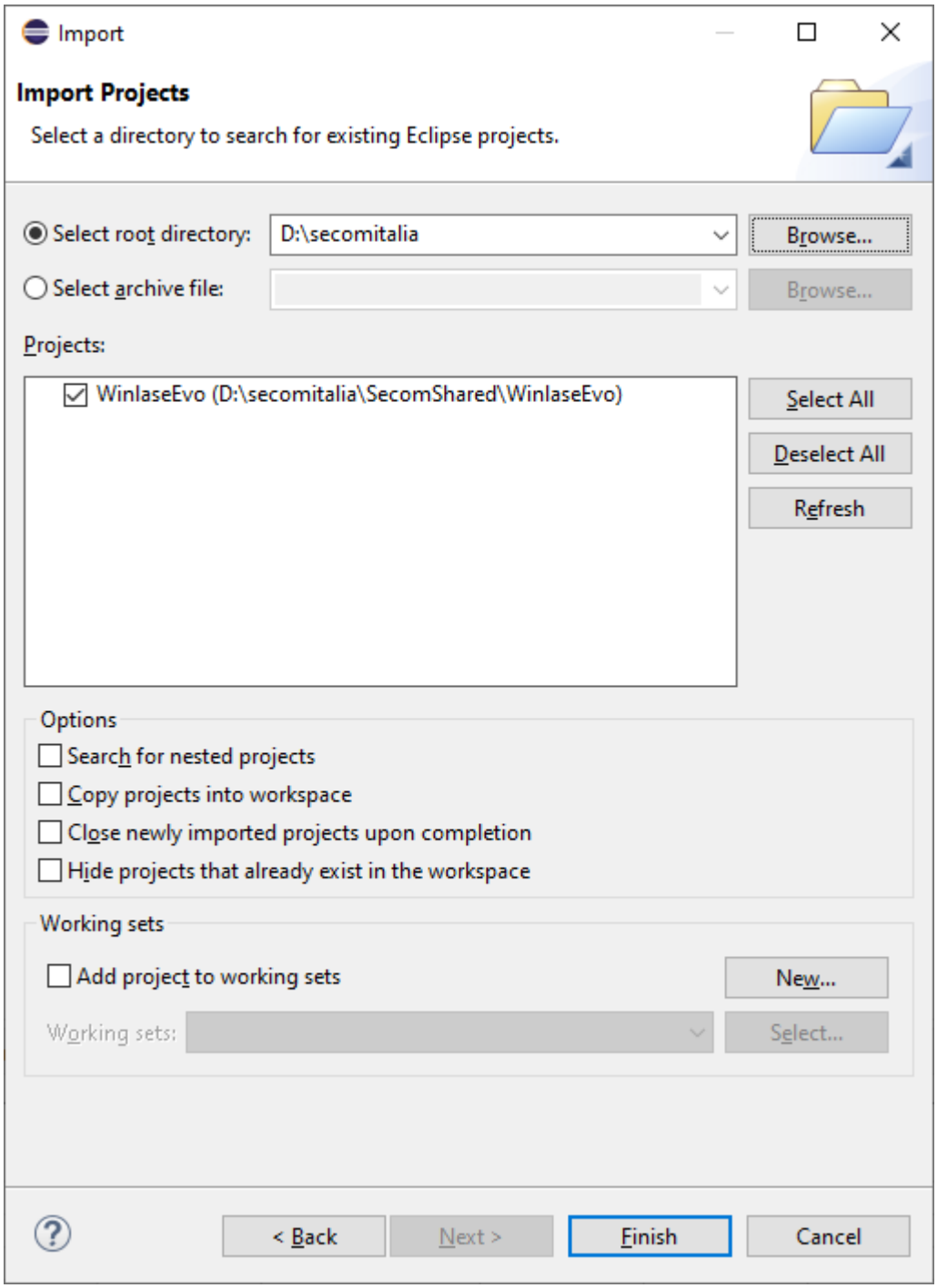
After closing the starting page, click on File>Import



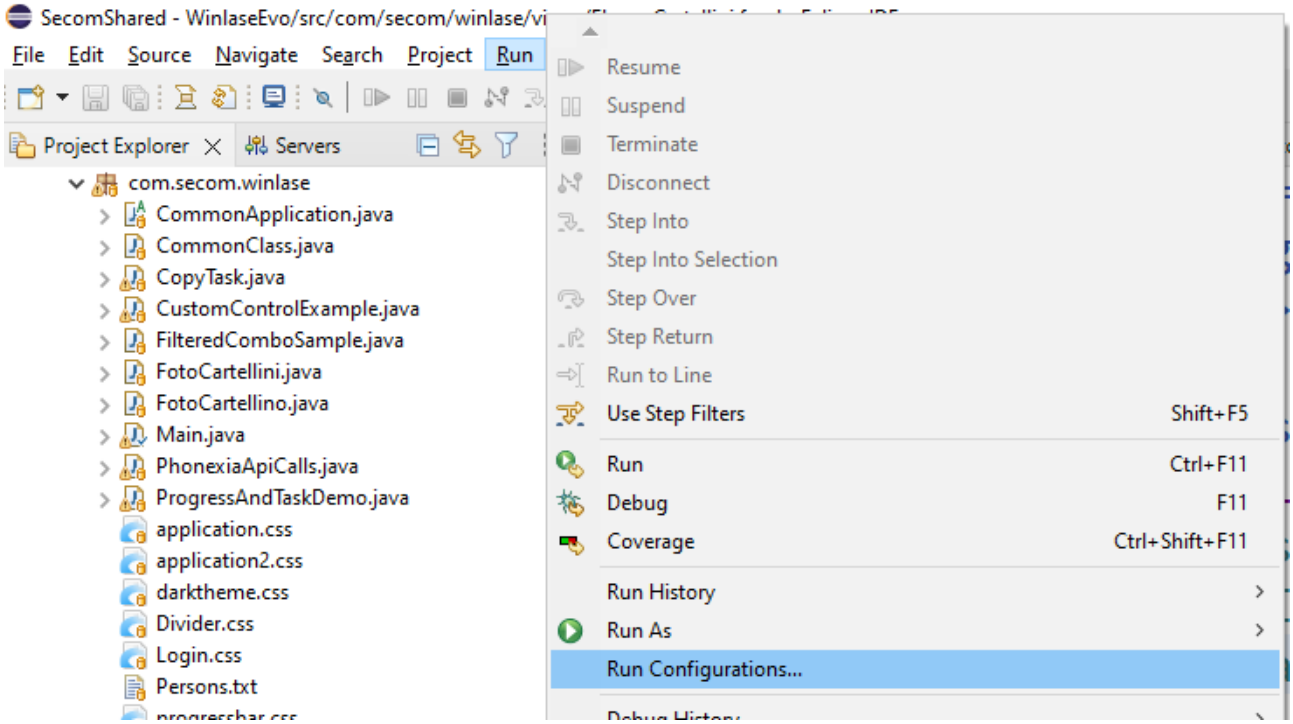
Then click on General>Existing Projects into Workspace



Select existing workspace as follows and then click on Finish

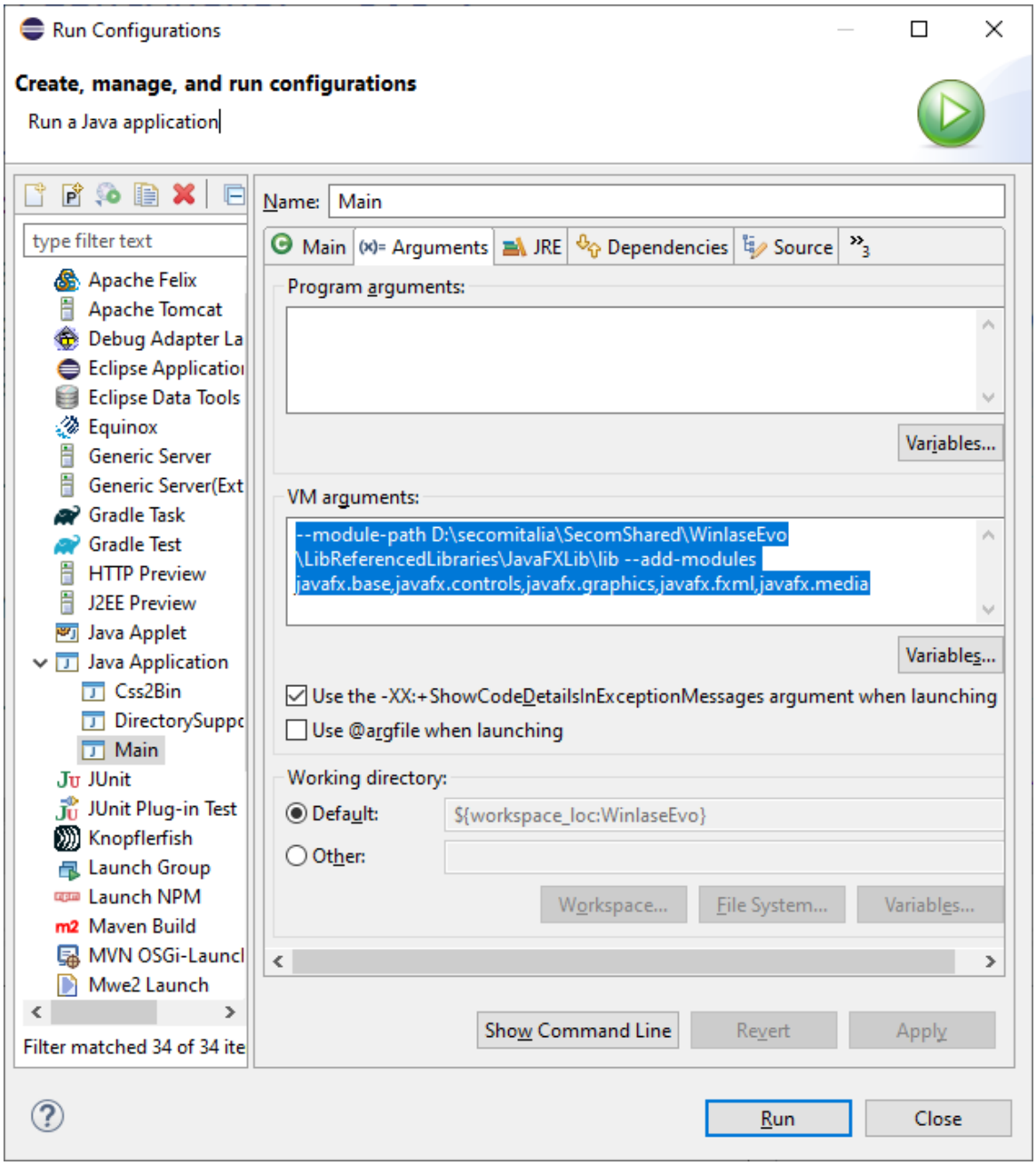


Once the project (WinlaseEvo) has been imported go to Run> Run Configurations



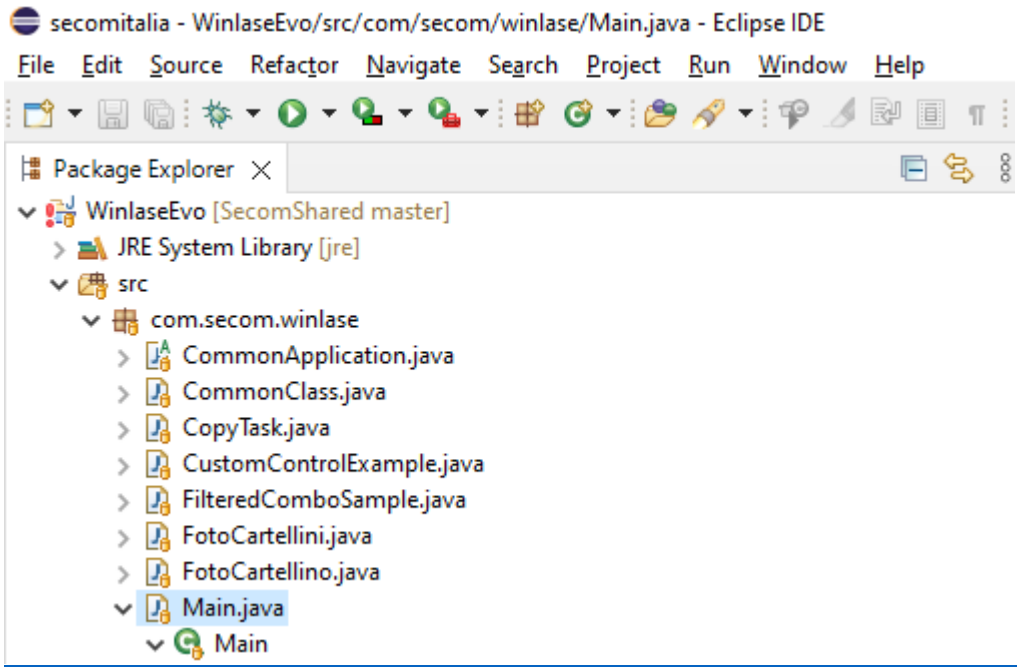
And paste the following string into the VM Arguments textbox, as shown in the picture below.

--module-path D:\secomitalia\SecomShared\WinlaseEvo\LibReferencedLibraries\JavaFXLib\lib --add-modules javafx.base,javafx.controls,javafx.graphics,javafx.fxml,javafx.media



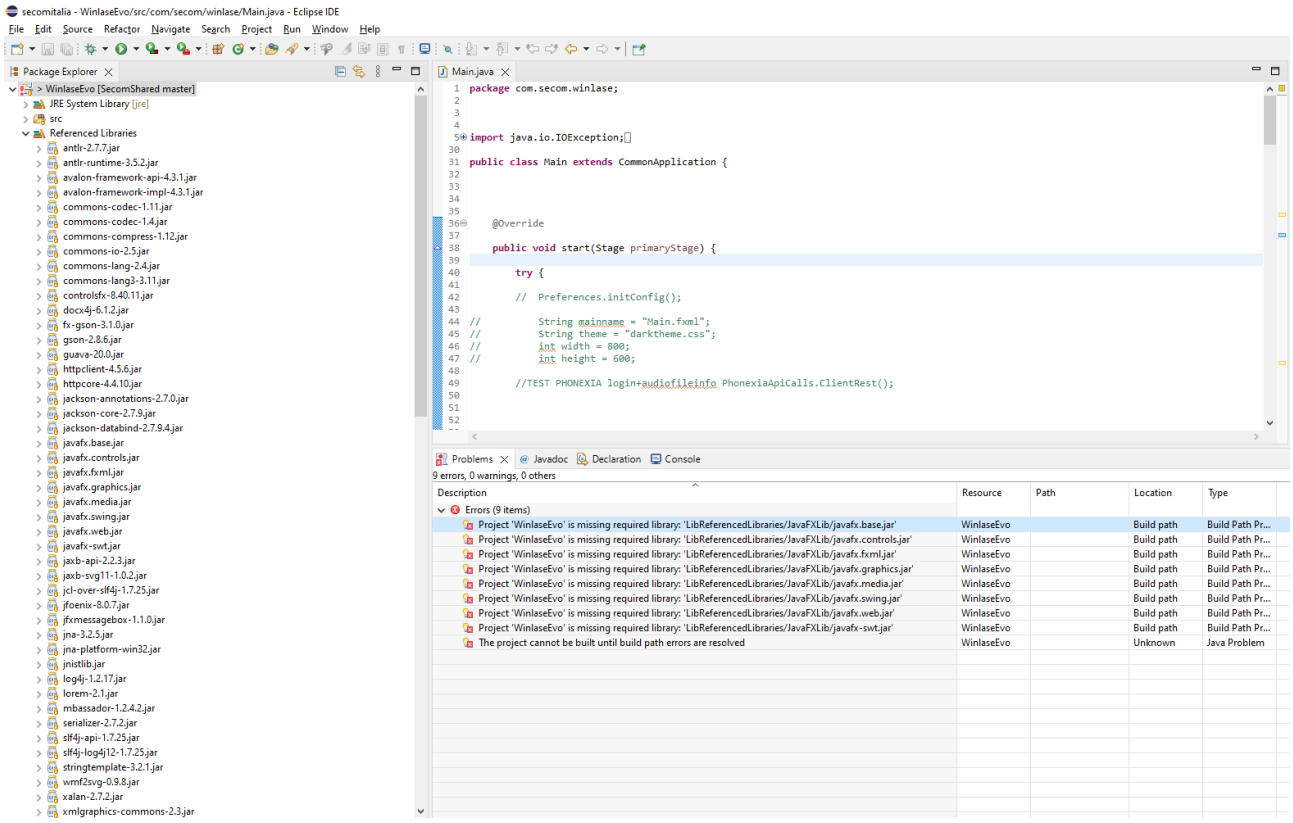
Click on Apply and then close.

Now, select the main module as shown below

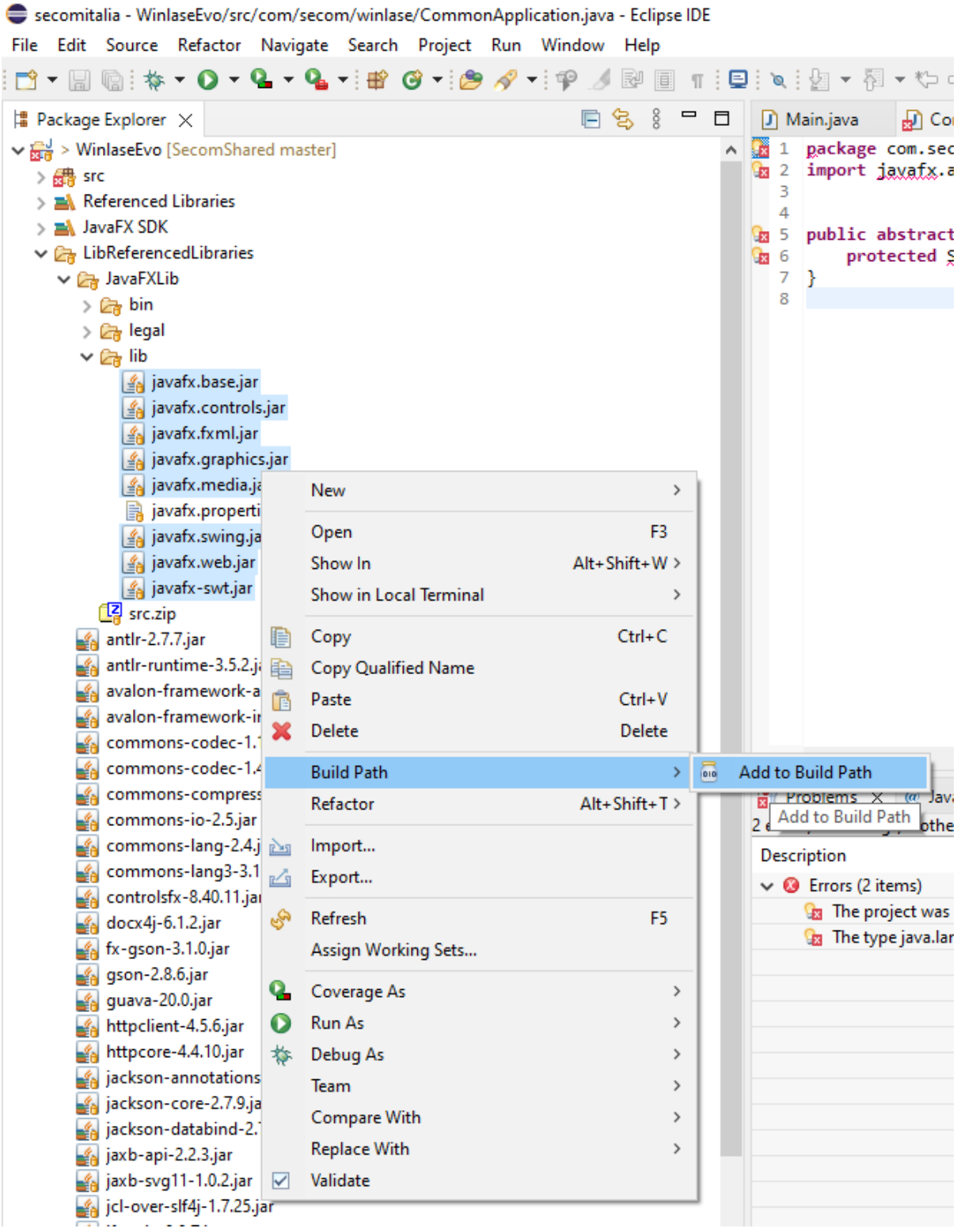


And then try to run the program.

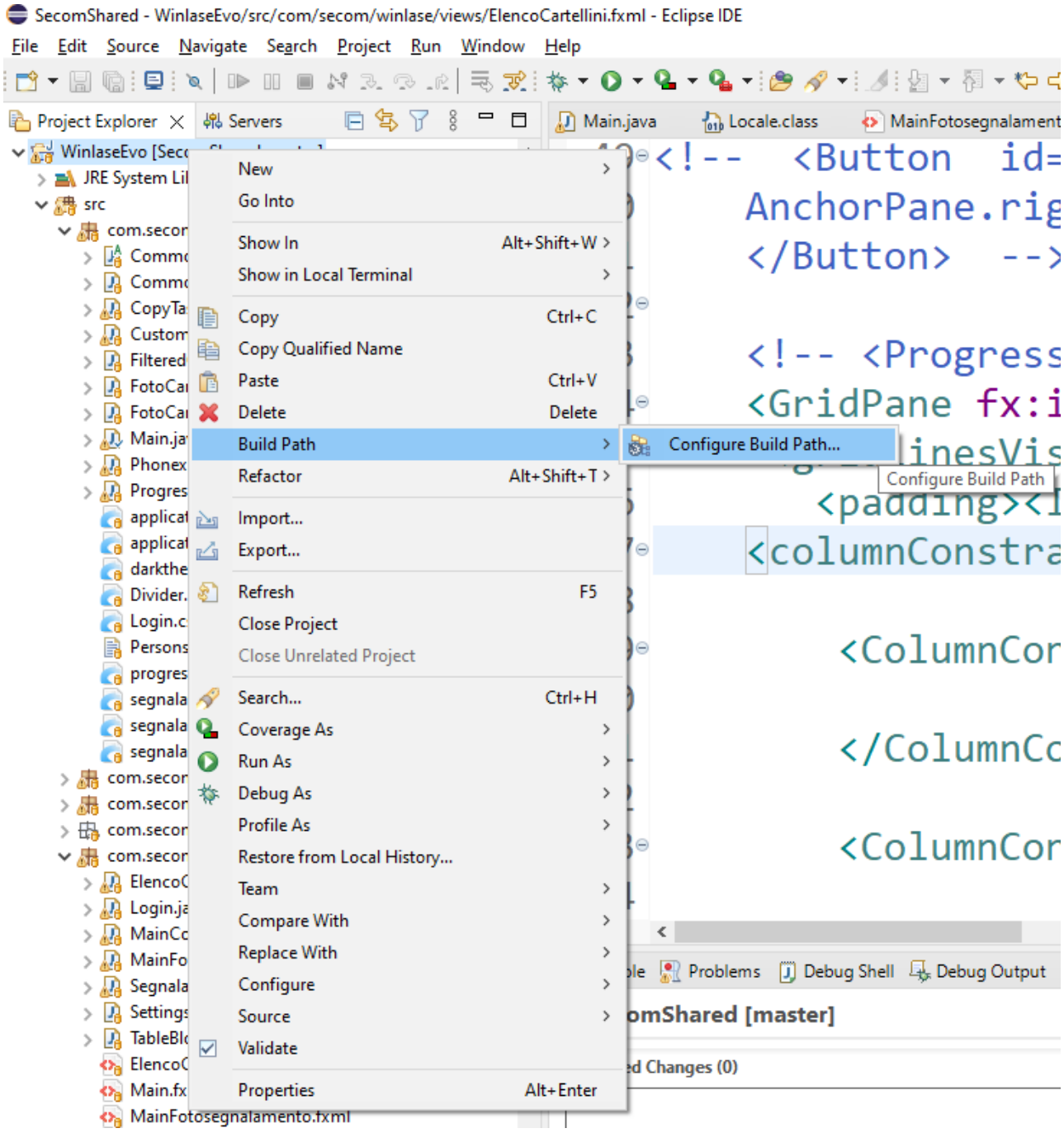
If you receive this error:



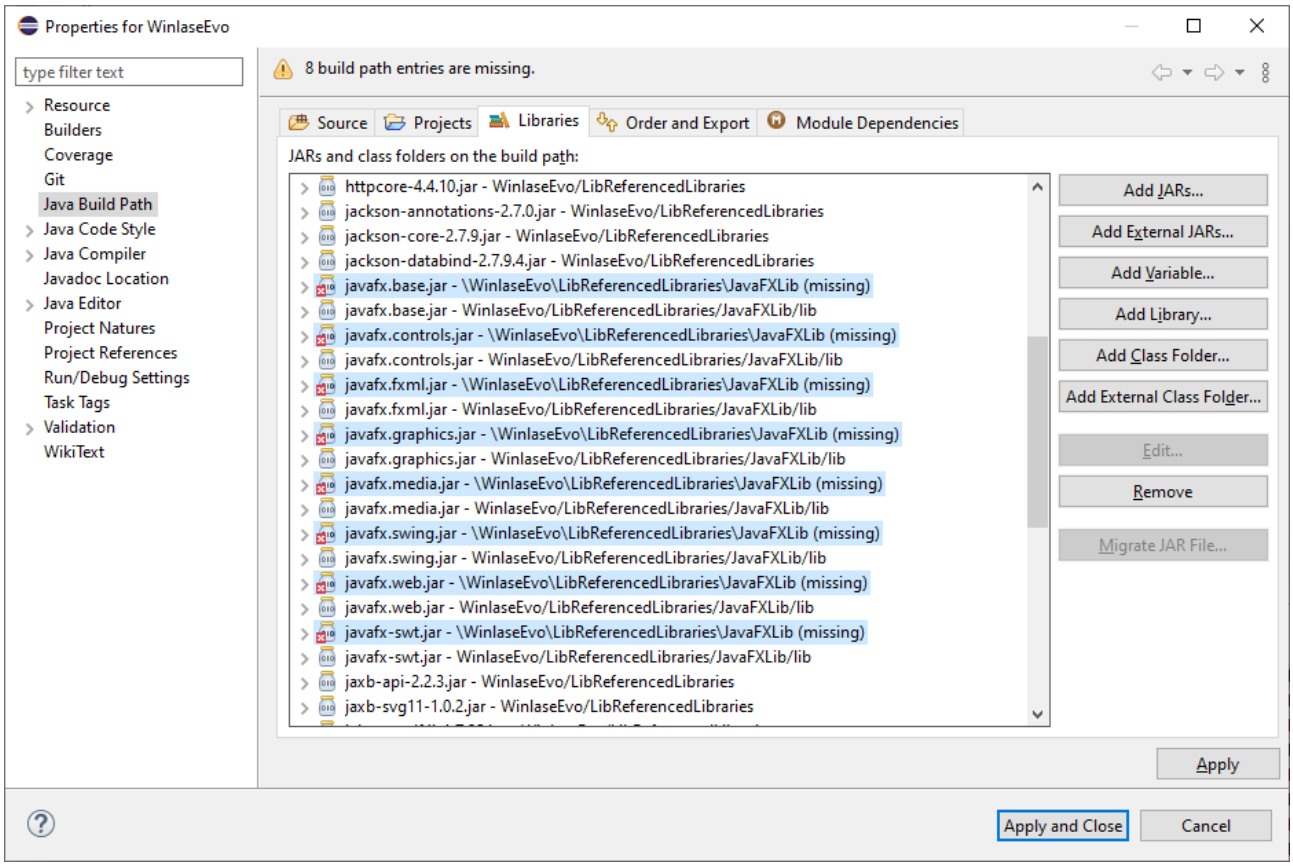
Then add the following (only selected) files to Build path as follows



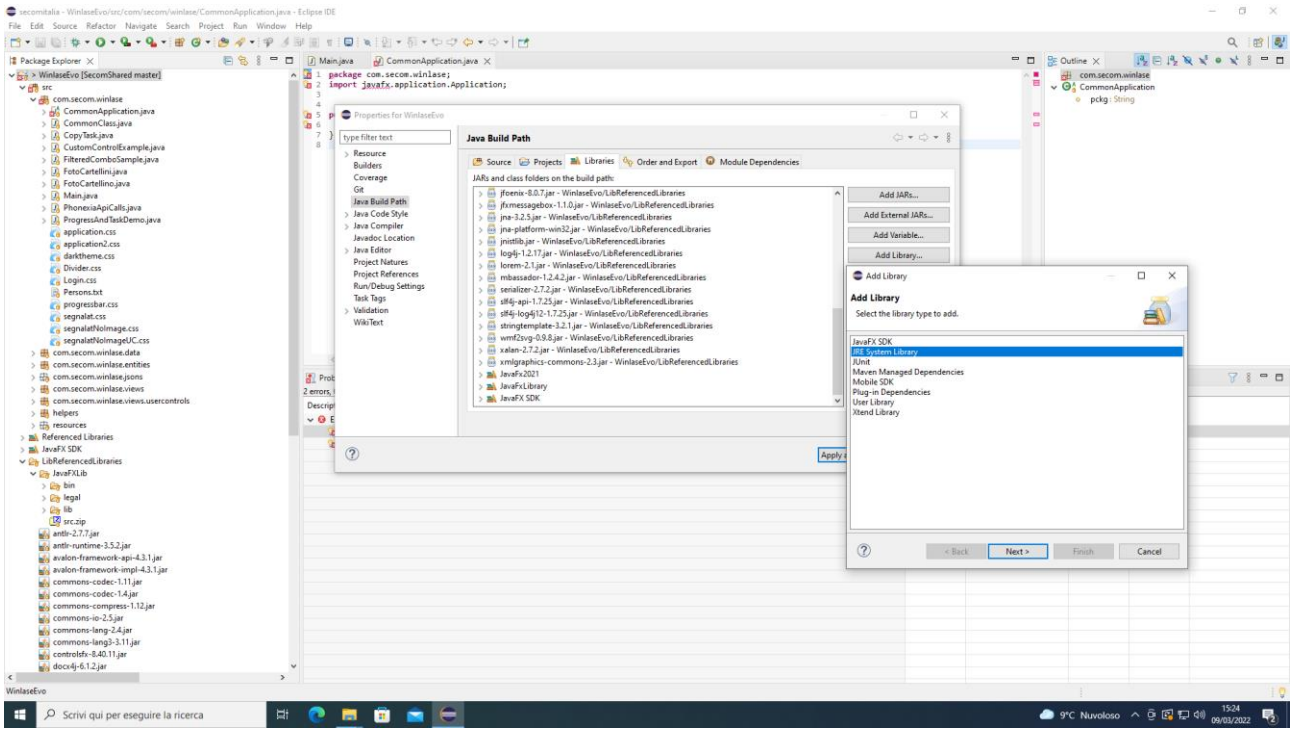
Then select WinlaseEvo > Build Path > Configure Build Path

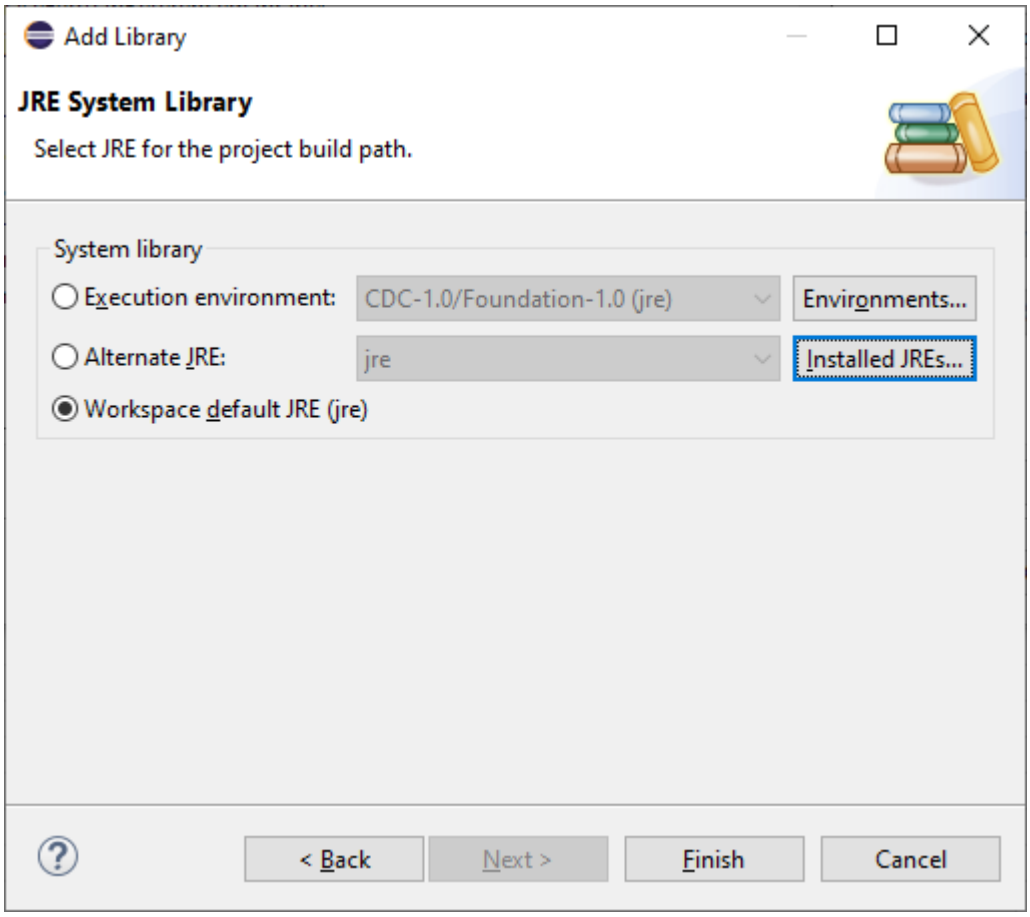


Then go to Libraries and remove Missing files



If JRE library is missing then add library as follows





And then click on Finish.

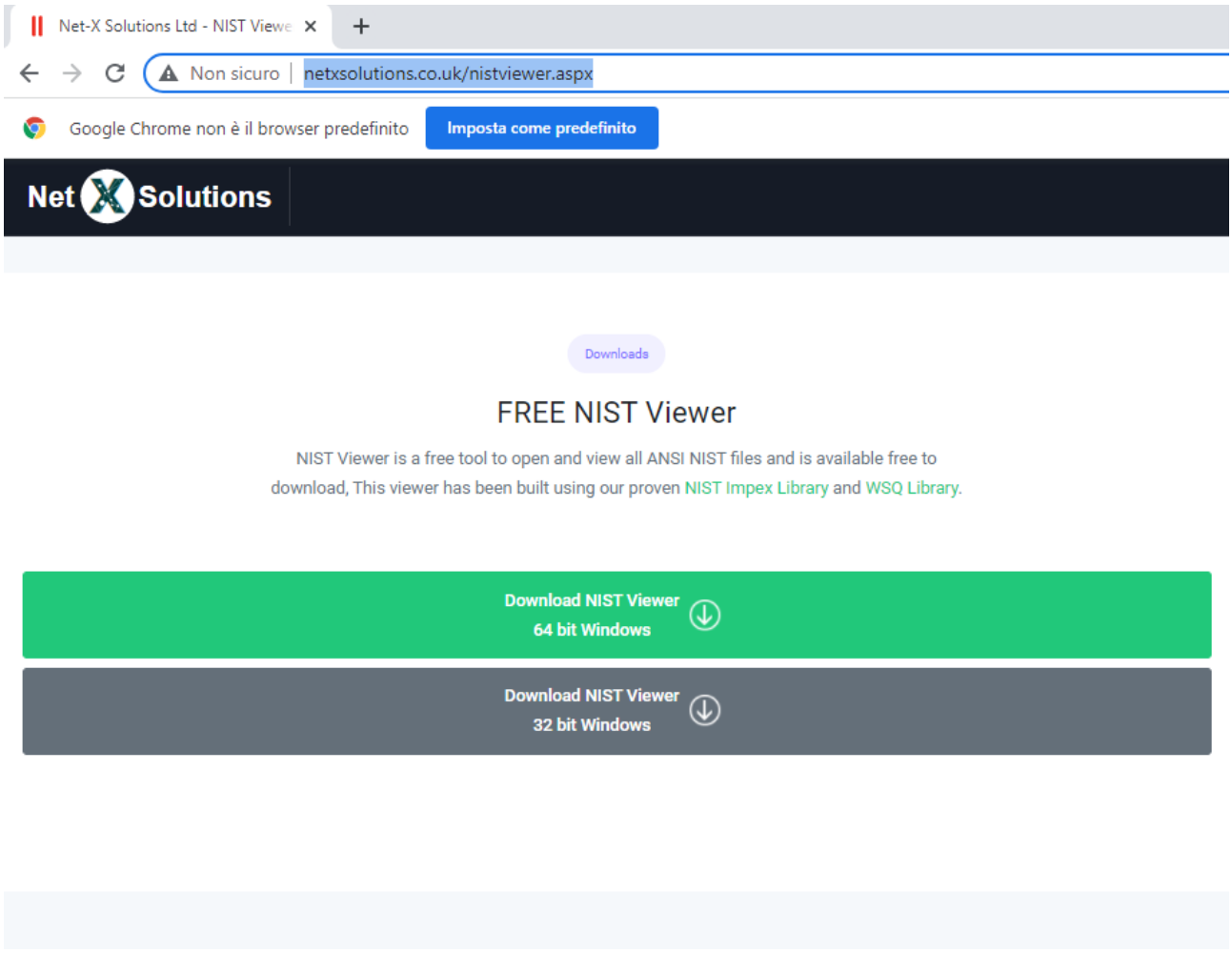
6) Install the following NistViewers

http://www.cognaxon.com/index.php?page=nistviewer

https://download.cnet.com/IDX-BioSuite-NIST-Viewer/3000-20418_4-75925897.html

if the last link doesn't work as expected, please download the 64-bit version of the IDX-Biosuite from here

http://www.netxsolutions.co.uk/nistviewer.aspx

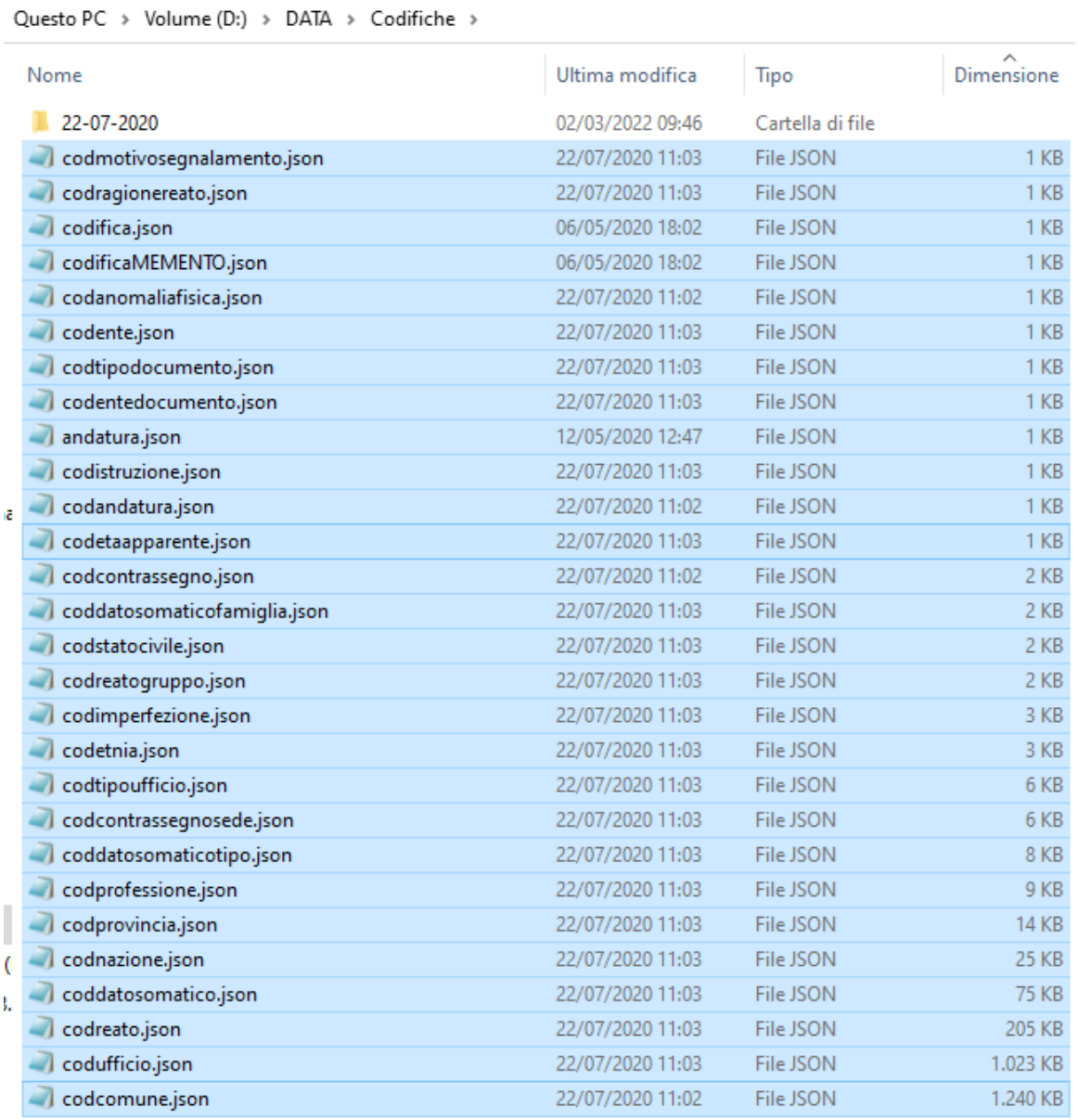


7) create in D:\ the following paths

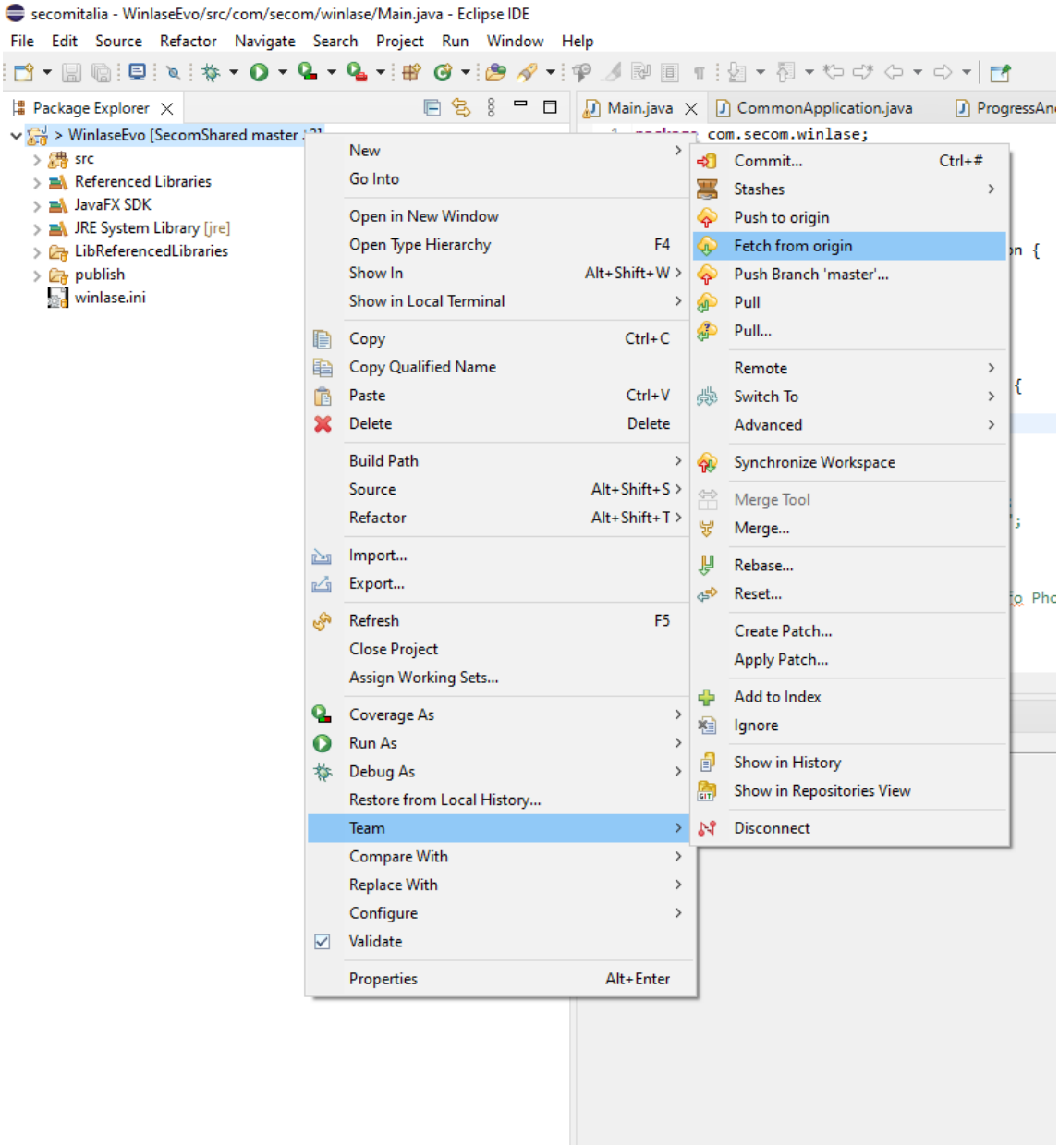
D:\DATA\Codifiche

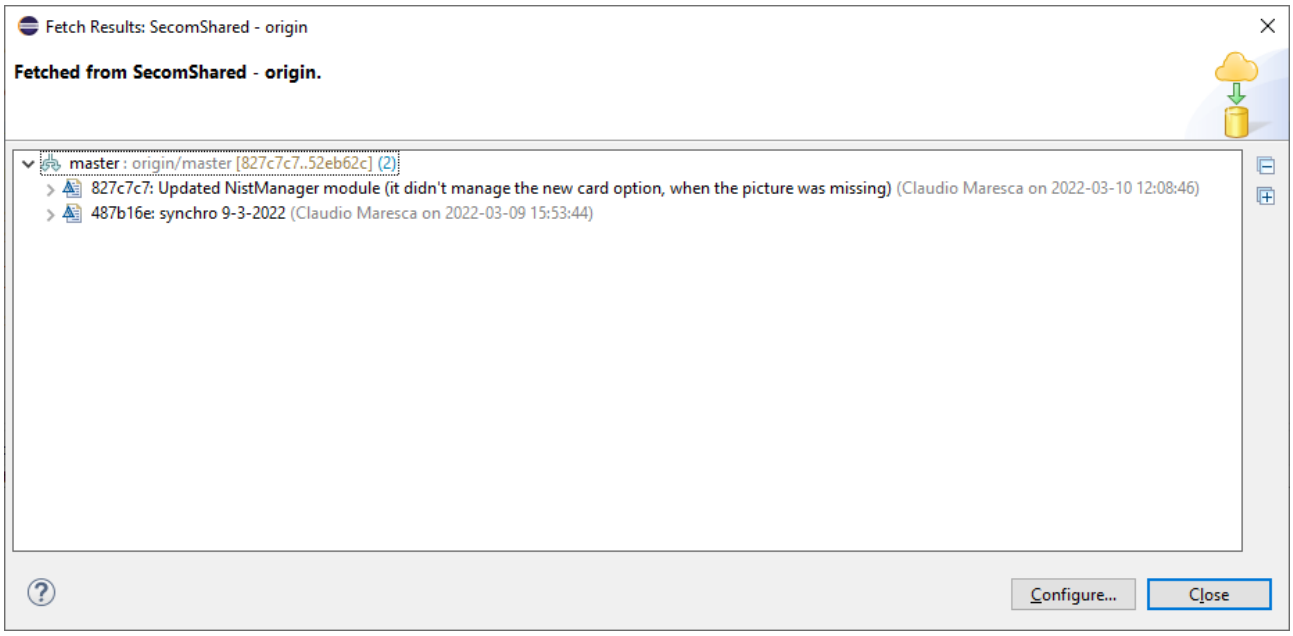
D:\DATA\xml\out

Copy into codifiche the files (after downloading them)



8) Open project (from the WinlaseEvo root), right click and fetch last updates from origin (as follows)


secomitalia - WinlaseEvo/src/com/secom/winlase/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
> WinlaseEvo [SecomShared master
src
Referenced Libraries
JavaFX SDK
JRE System Library [jre]
LibReferencedLibraries
publish
winlase.ini
Main.java
CommonApplication.java
com.secom.winlase;
New
Go Into
Open in New Window
Open Type Hierarchy F4
Show In Alt+Shift+W
Show in Local Terminal
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
Refresh F5
Close Project
Assign Working Sets...
Coverage As
Run As
Debug As
Restore from Local History...
Team
Compare With
Replace With
Configure
Validate
Properties Alt+Enter
Commit... Ctrl+#
Stashes
Push to origin
Fetch from origin
Push Branch 'master'...
Pull
Pull...
Remote
Switch To
Advanced
Synchronize Workspace
Merge Tool
Merge...
Rebase...
Reset...
Create Patch...
Apply Patch...
Add to Index
Ignore
Show in History
Show in Repositories View
Disconnect

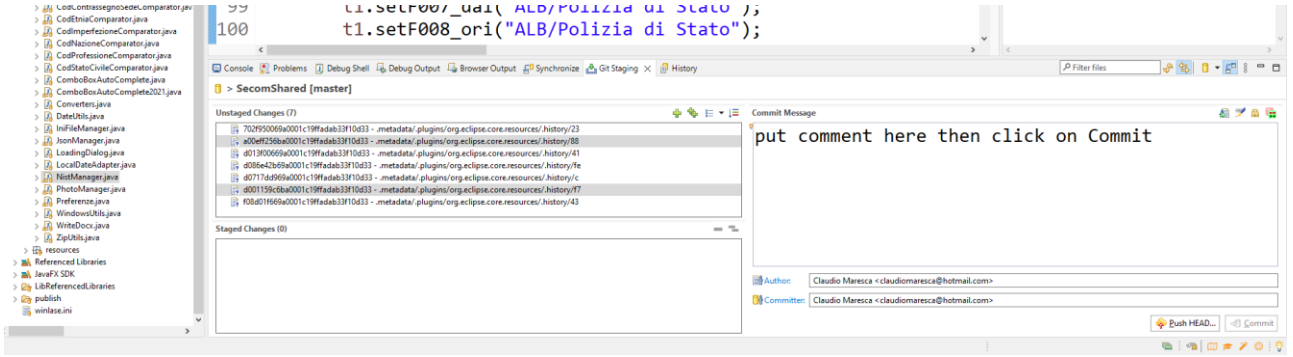


Then Click on WinlaseEvo>(right click)>Team>Pull

Now the source files should be aligned.

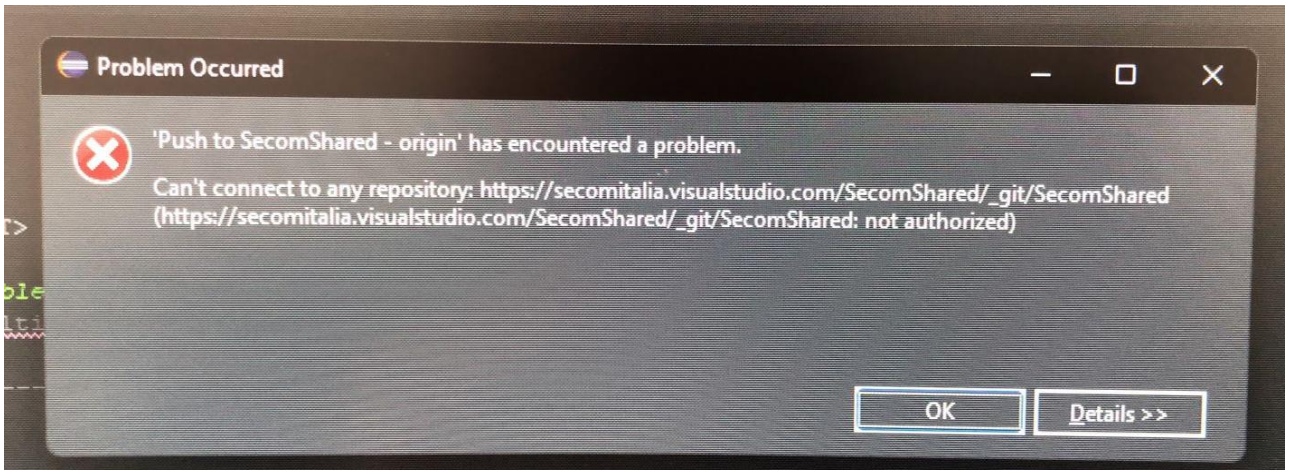
9) Commit operation after source changed

After modifying the software, do the same thing (WinlaseEvo> Right Click), choosing the Commit option instead. Put a comment as shown below, and click on Commit.



10.1) **Error Can't connect to any repository**

If, while synchronizing the sources (via commit / push / pull), you get the following error,



Then try to perform the following steps.

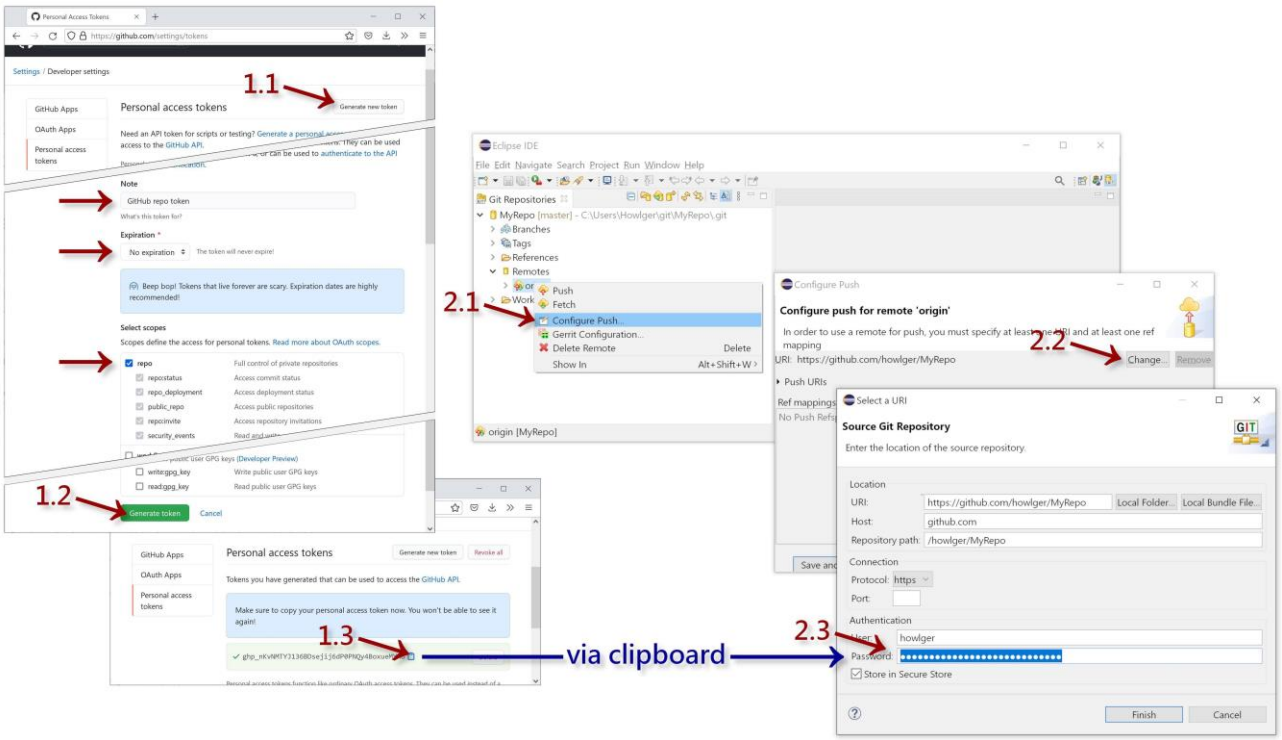
Consider that

Since August 13, 2021, **GitHub does not support** authentication via **HTTPS** with your **GitHub account password** for security reasons **anymore**. Instead, in Eclipse, when pushing to a GitHub repository or when fetching from a private repository, you will get a git-upload-pack not permitted on 'https://github.com...' error.

As solution, use either

- a GitHub specific **Personal access tokens** as password instead of your previously used GitHub account password or
- **SSH with an SSH key** of which the private and public key is on your local machine and configured in Eclipse and the public key is uploaded to your GitHub account instead.

Personal access token (GitHub specific)

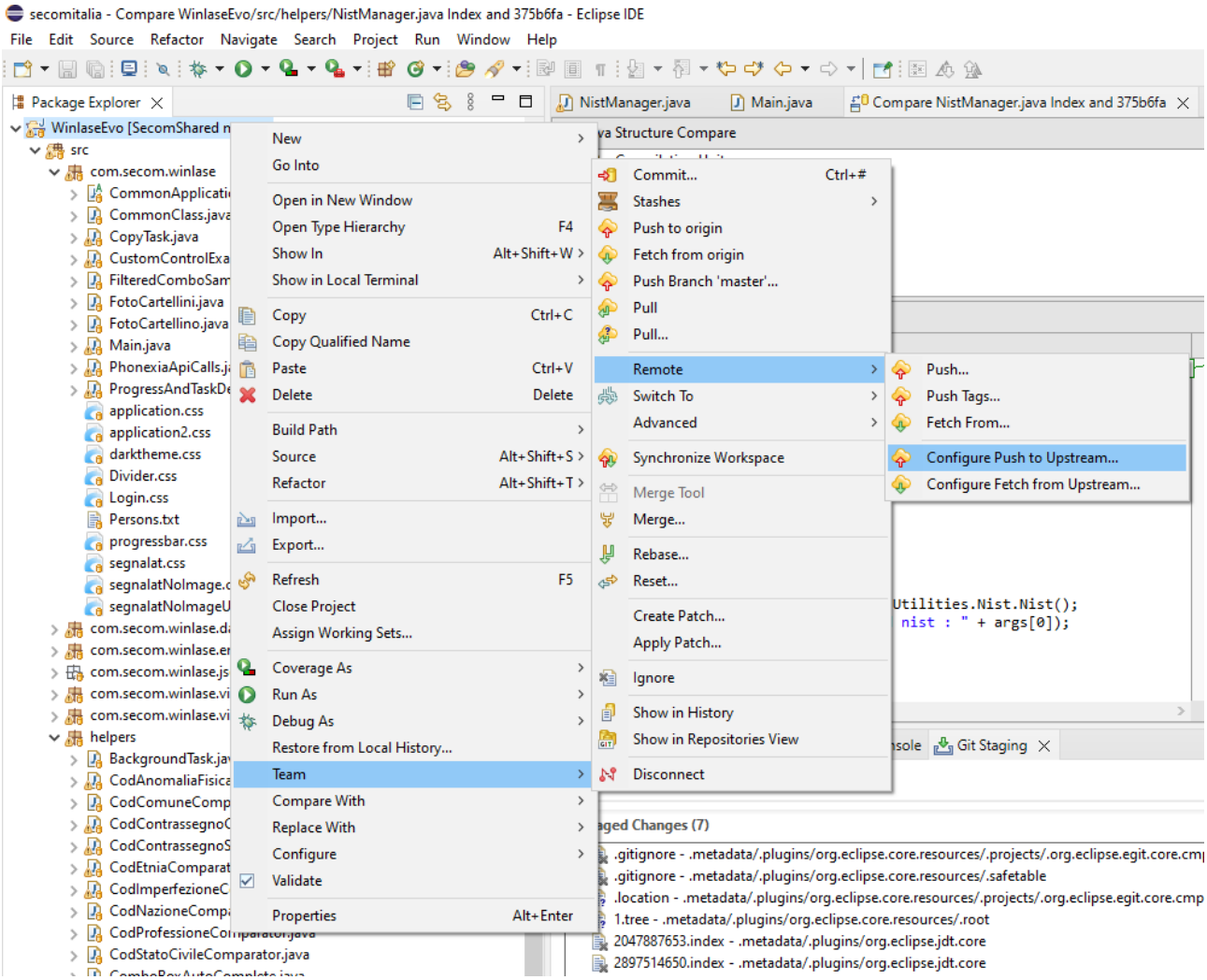


1. Go to your **GitHub account** to ***Settings > Developer settings > Personal access tokens*** website:
    1. Click the ***Generate new token*** button in the upper right
        - Enter a ***Note***, e.g. GitHub repo token
        - Choose ***Expiration***, e.g. ***No expiration***
        - Tick the checkbox ***repo***
    2. Click the ***Generate token*** button at the bottom
    3. **Copy** the generated **token to the clipboard**
2. In Eclipse, in the ***Git Repositories*** view:
    1. **Right-click** the ***Remote***s sub-**node for GitHub** (origin or the name you have chosen when you have cloned the repository) and choose ***Configure Push...***
    2. Click the ***Change...*** button to change the URI in the upper right
    3. **Replace the password with** with the copied generated GitHub **token**
    4. Click ***Finish*** and ***Save*** to apply the changes

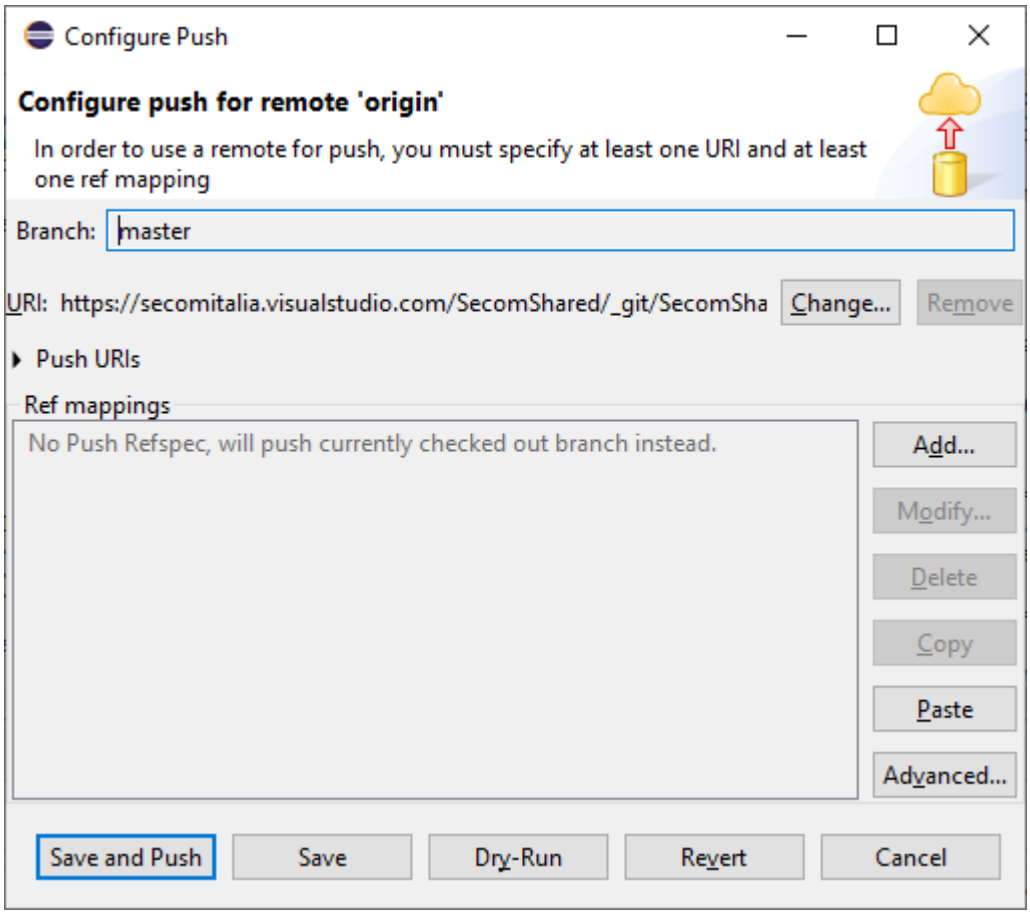
(extracted from the following article:

https://stackoverflow.com/questions/68790276/pushing-from-eclipse-to-my-github-repository-via-https-stopped-working-git-rec/68802292#68802292)

10.2) **After generating the new token, go to WinlaseEvo > Team > Remote > Configure push to Upstream…**



Click on “change”



Try to replace password with token