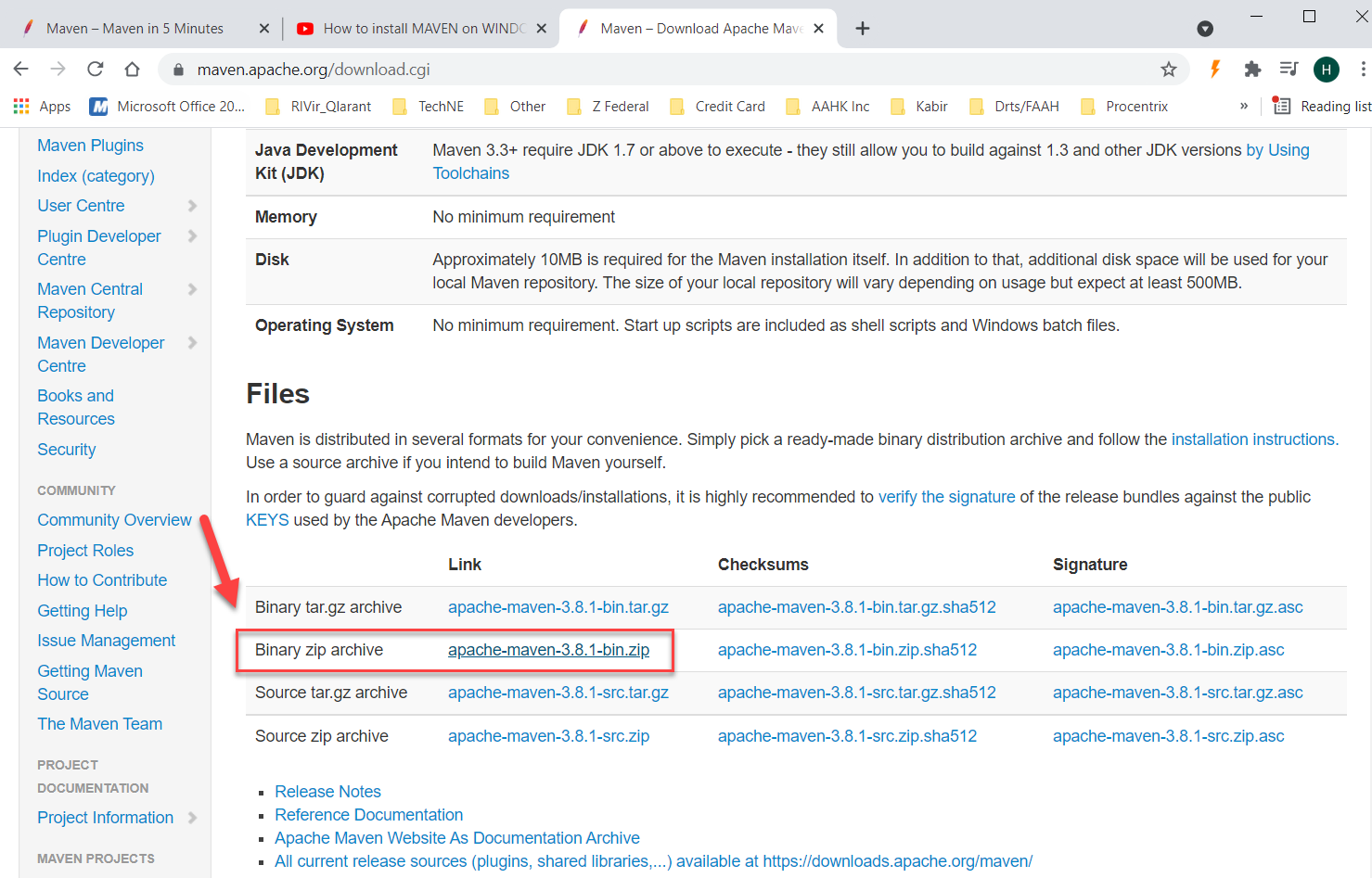
**Create a Maven Project in Eclipse**

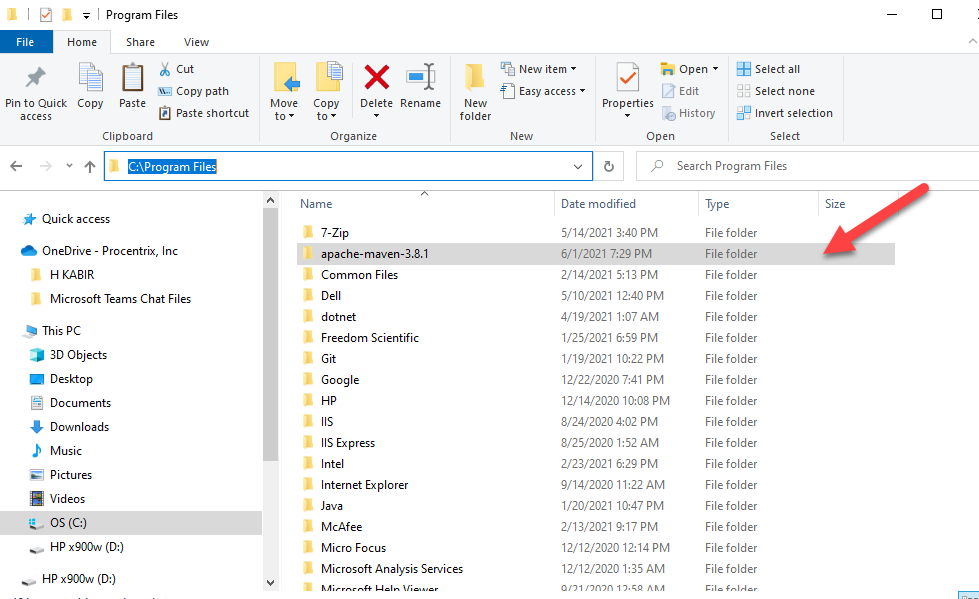
**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Install MAVEN on WINDOWS 10 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

* Download MAVEN -> Select “Binary zip archive”

[https://maven.apache.org](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbWo5OHpqNktxUGxJdDNrSXk1NzdWZnpIQlcwUXxBQ3Jtc0tuSk9XLUZXOXNuNGxyQzVXQTN2eG9yMUtaM0VydUcweEJtOWlGY1JEUElSYXd6TG85Y2liRnc1RnpsSk5qR2V5MGFYRnMtQzgwdllNaWVlOHZEanlBUEFuaWQ5VDFFU1ZWa1pyWV9XODF1UU1fRk9GQQ&q=https%3A%2F%2Fmaven.apache.org)



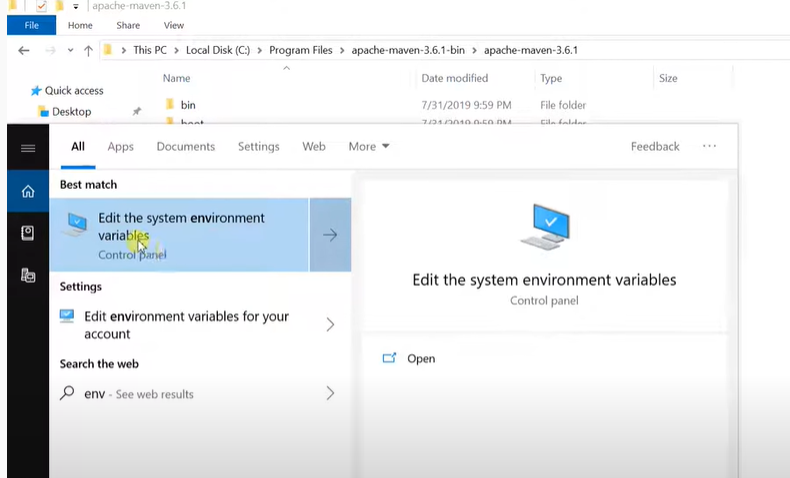
* Extract the Folders
* Copy the extracted folder into **“C:\Program Files”**



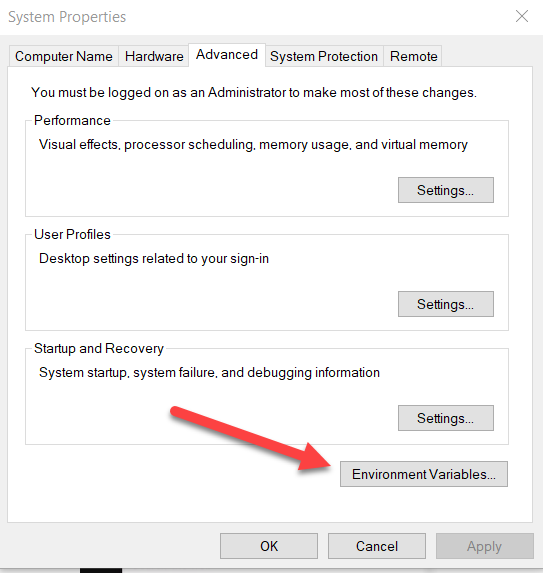
* Need to create Variables
  + User Variables:
    - **MAVEN\_HOME -> path for Maven Folder**
    - **M2\_HOME -> path for Maven Folder**

**Note: we need MAVEN\_HOME and M2\_HOME Environment Variable because some software use MAVEN\_HOME and some uses M2\_HOME**

* + System Variable:
    - Path -> path for Maven bin Folder
* In start menu type Environment



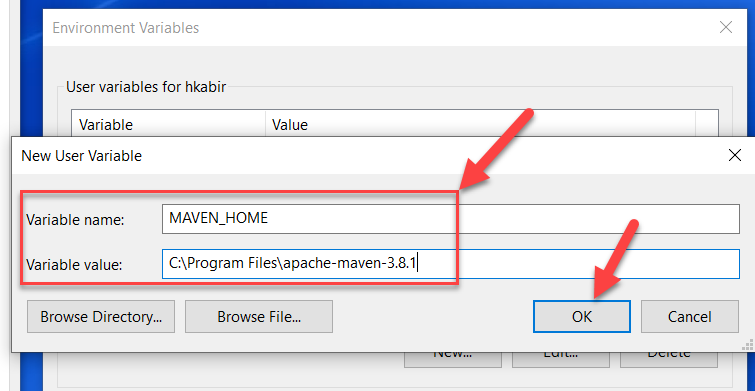
* Dialog box open -> click “Environment Variable”



* Enter below:

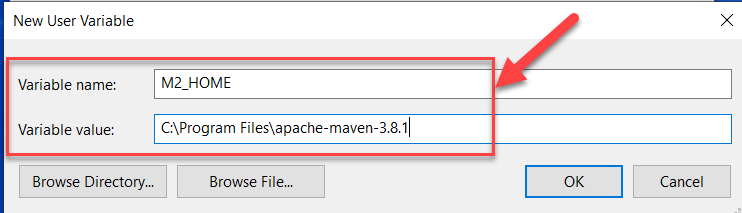
Variable Name: **MAVEN\_HOME**

Variable Value: **C:\Program Files\apache-maven-3.8.1**

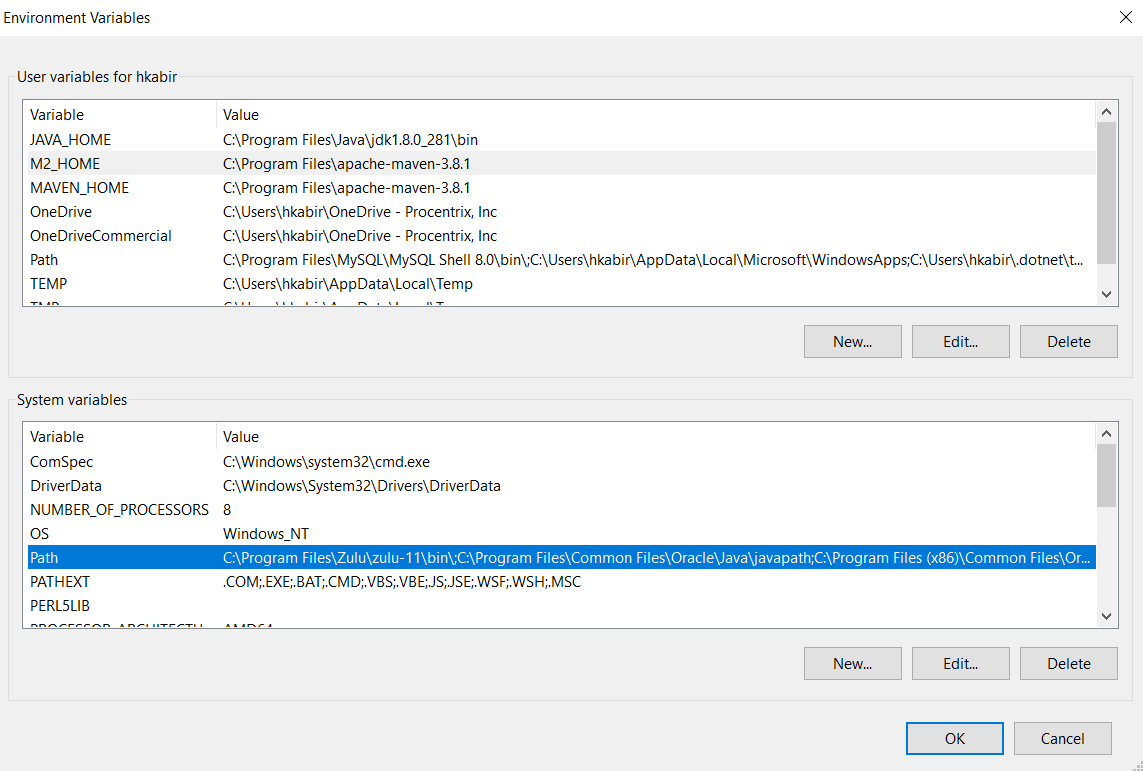


Variable Name: **M2\_HOME**

Variable Value: **C:\Program Files\apache-maven-3.8.1**

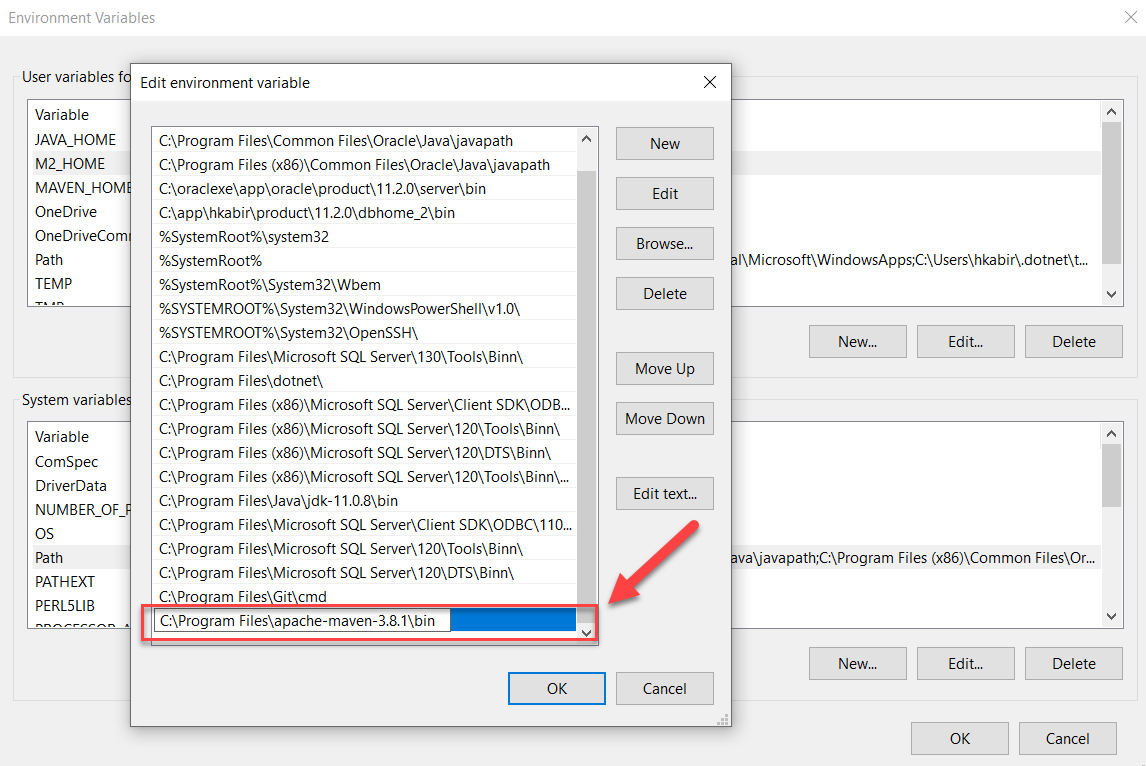


* Click Path under System Variable



* Click Edit -> New -> Enter maven folder path until bin -> click ok

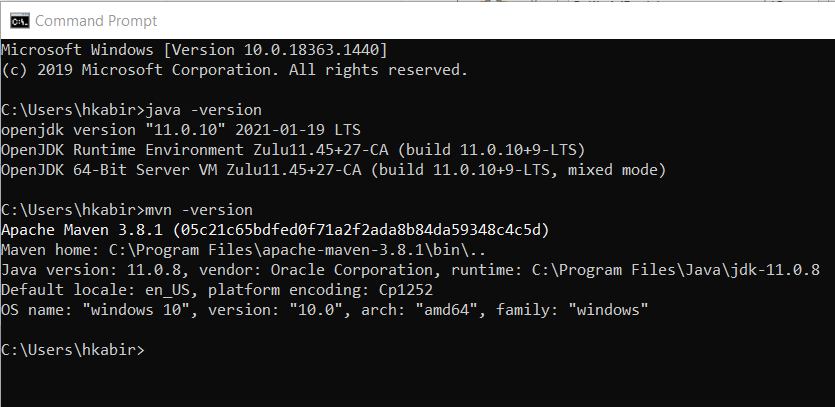
**“C:\Program Files\apache-maven-3.8.1\bin”**



* Click OK
* Close all the window of environment variable
* Open Command Line
* Type -below -> able to appropriate version

**java -version**

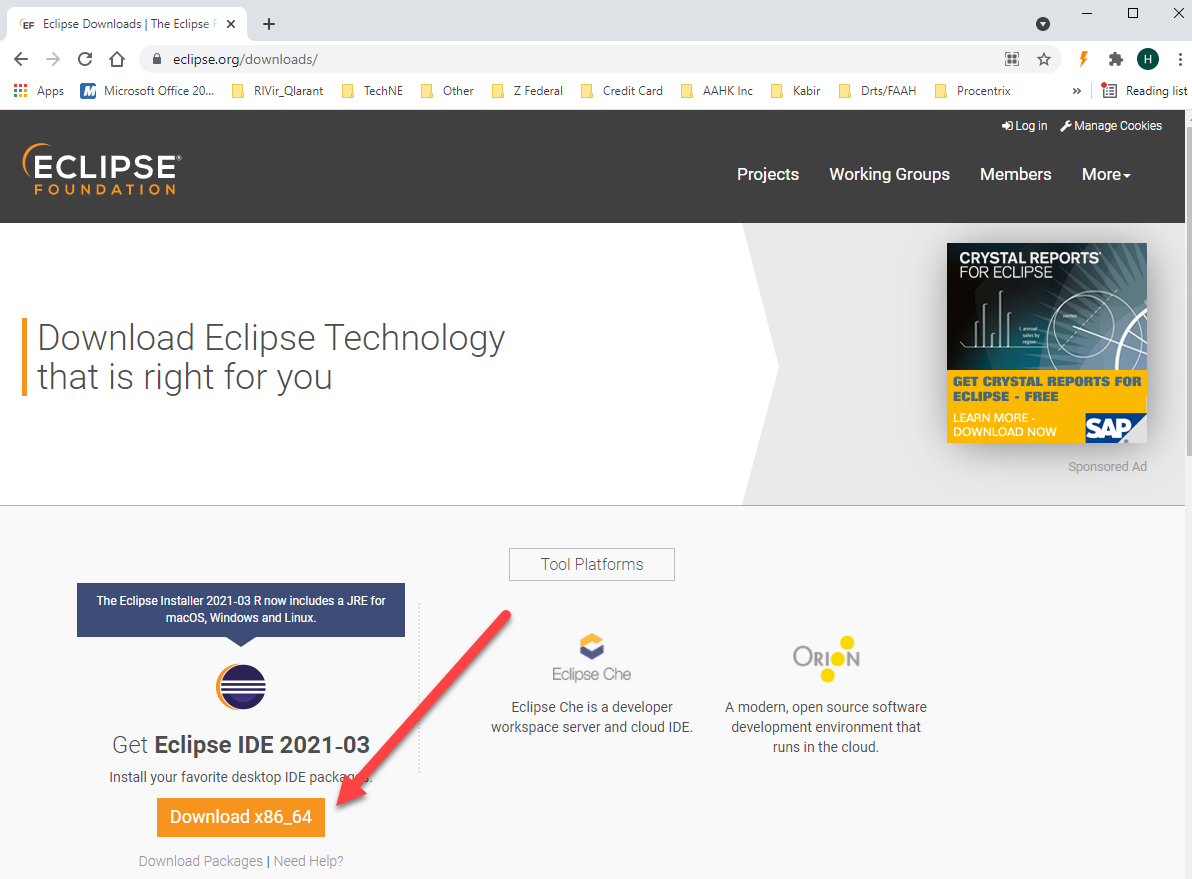
**mvn -version**



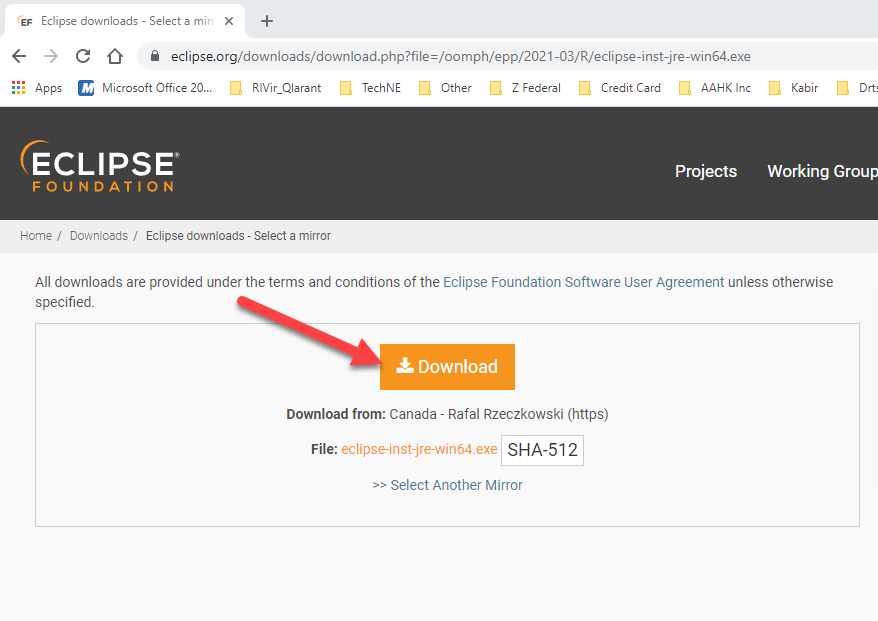
**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Install Eclipse \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

* Install Eclipse -> Select appropriate version -> click “Download x86\_64”

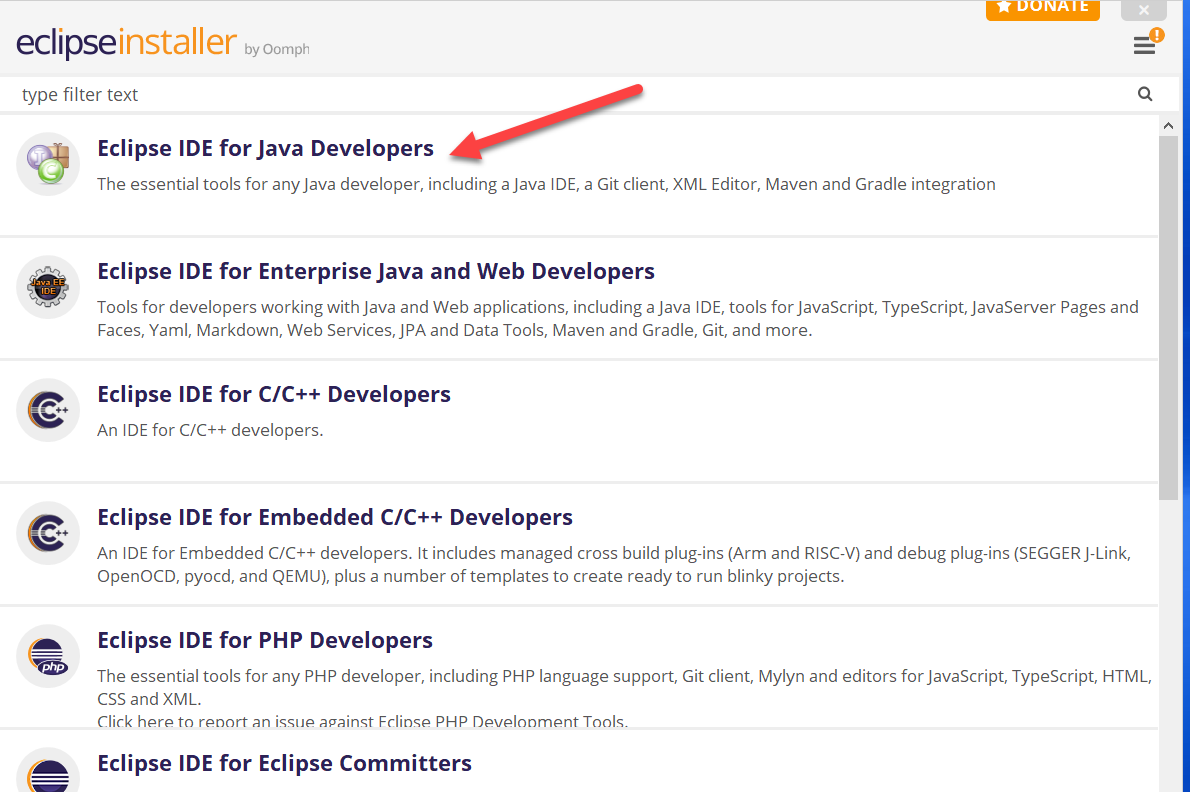
<https://www.eclipse.org/downloads/>



* Click Download



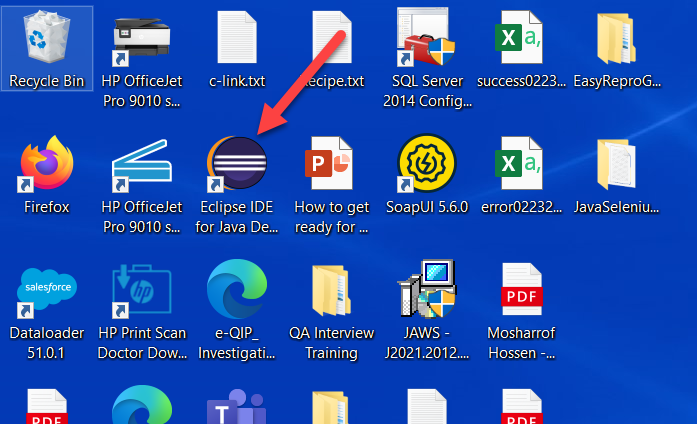
* Extract the zip file -> Select either **“Eclipse IDE for Java Developers”** or **“Eclipse IDE for Enterprise Java and Web Developers”**



* Click Install

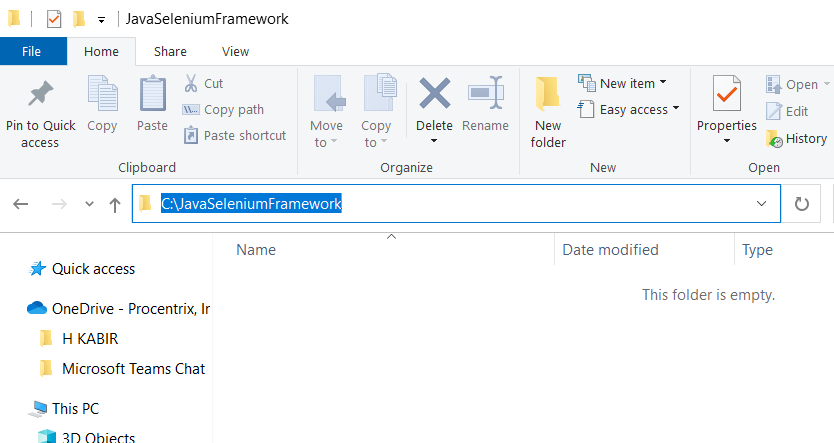


* After installation Eclipse shortcut will be shown in Desktop

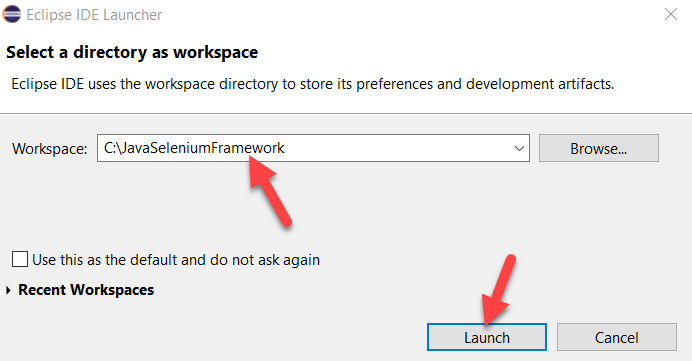


**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Create Maven Project \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

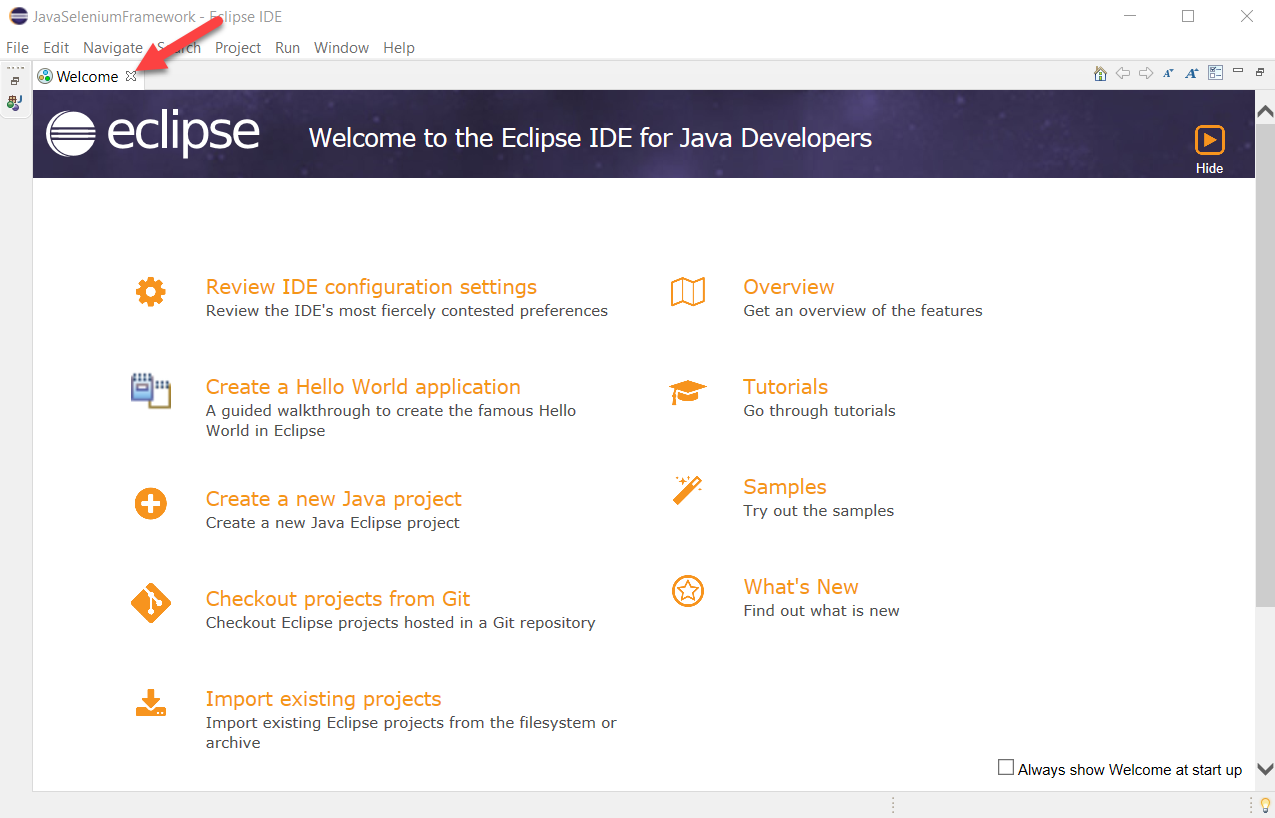
* Create a folder where the workspace directory to store its preferences and development artifacts



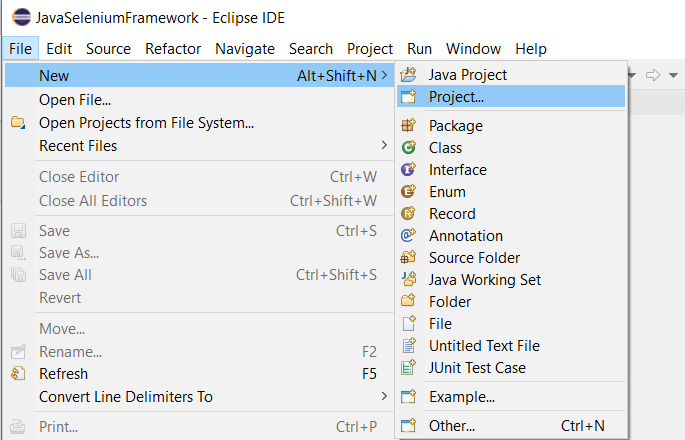
* Open Eclipse and browse appropriate workspace location -> Launch



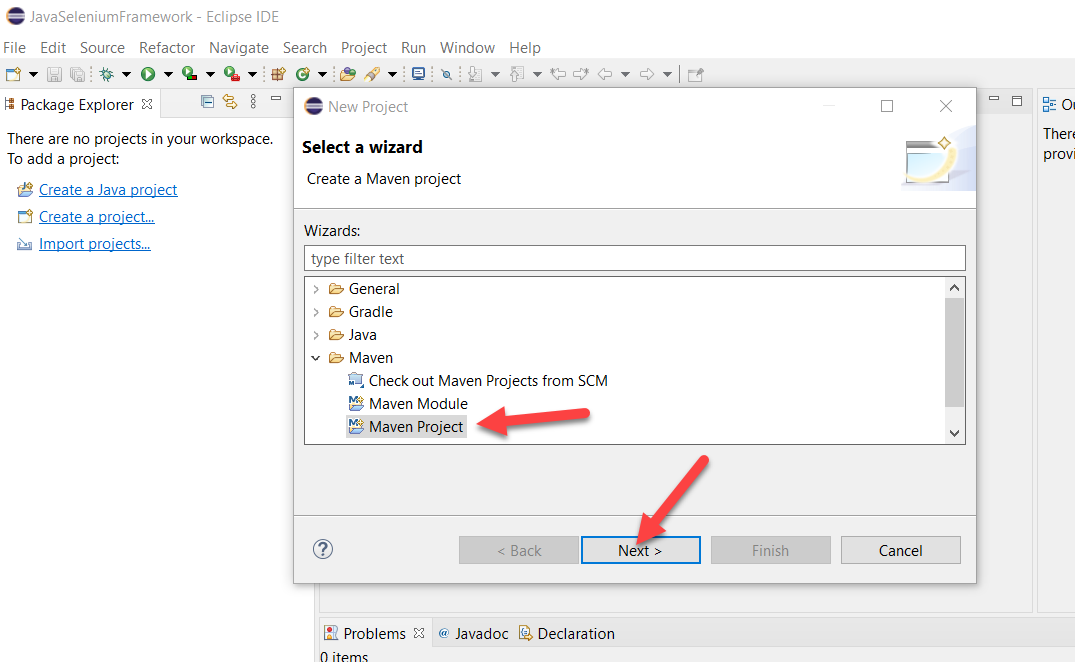
* Close the welcome screen (if appears)



* Create a Maven Project: File -> New -> Project



* The dialog box that appears on the screen will display different types of projects -> Select the Maven Project option -> Click on Next

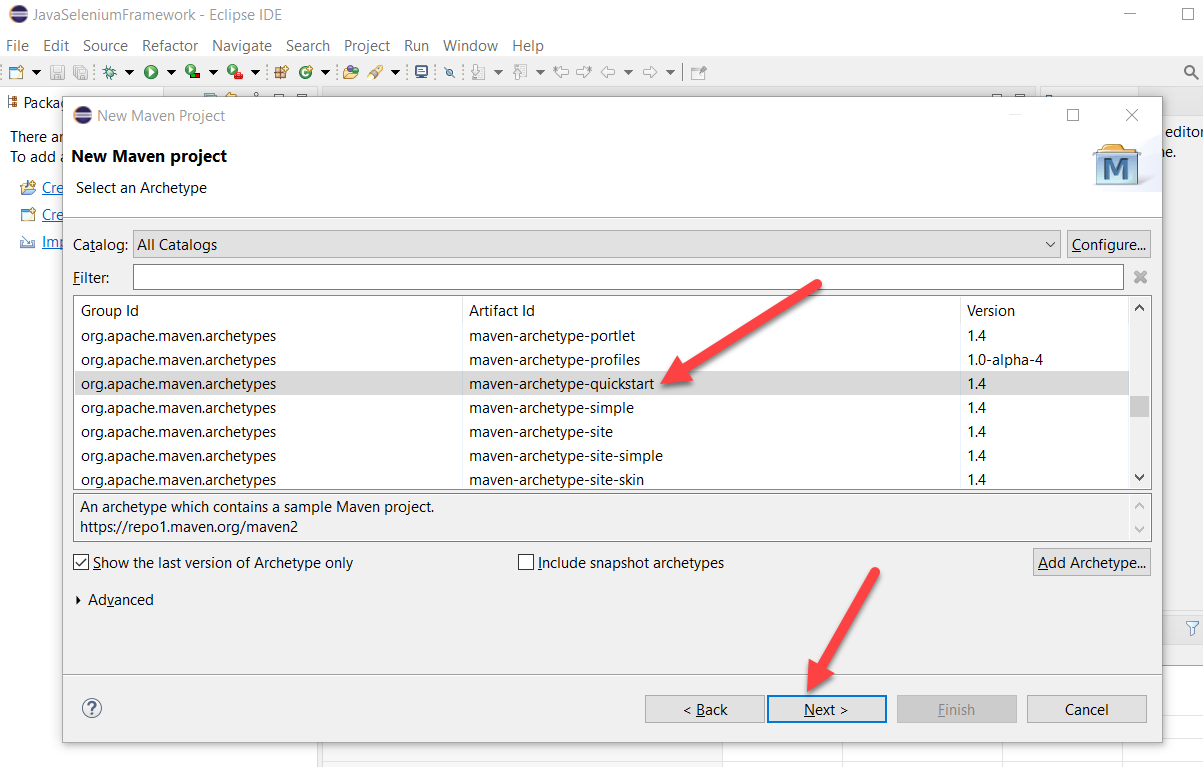


* Dialog box will appear. Select the default workspace -> Click on “Next”



* Select a plugin there and click on “Next”

Example: Artifact IDs: **maven-archetype-quickstart**



* Project Identifiers:

groupId – a unique base name of the company or group that created the project

artifactId – a unique name of the project

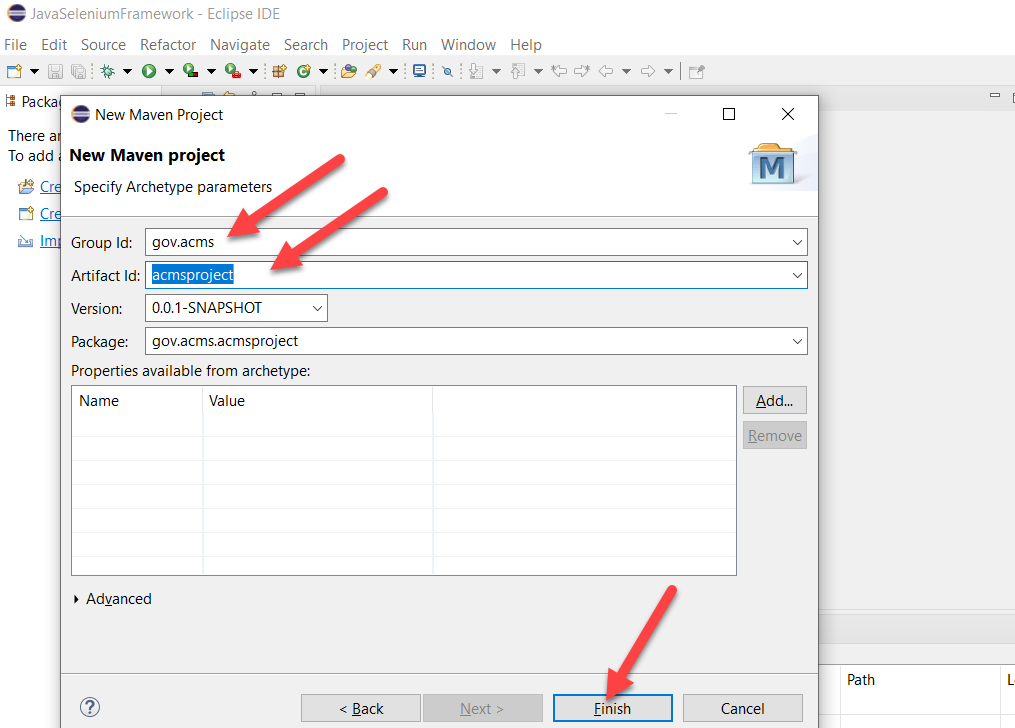
version – a version of the project

packaging – a packaging method (e.g. WAR/JAR/ZIP)

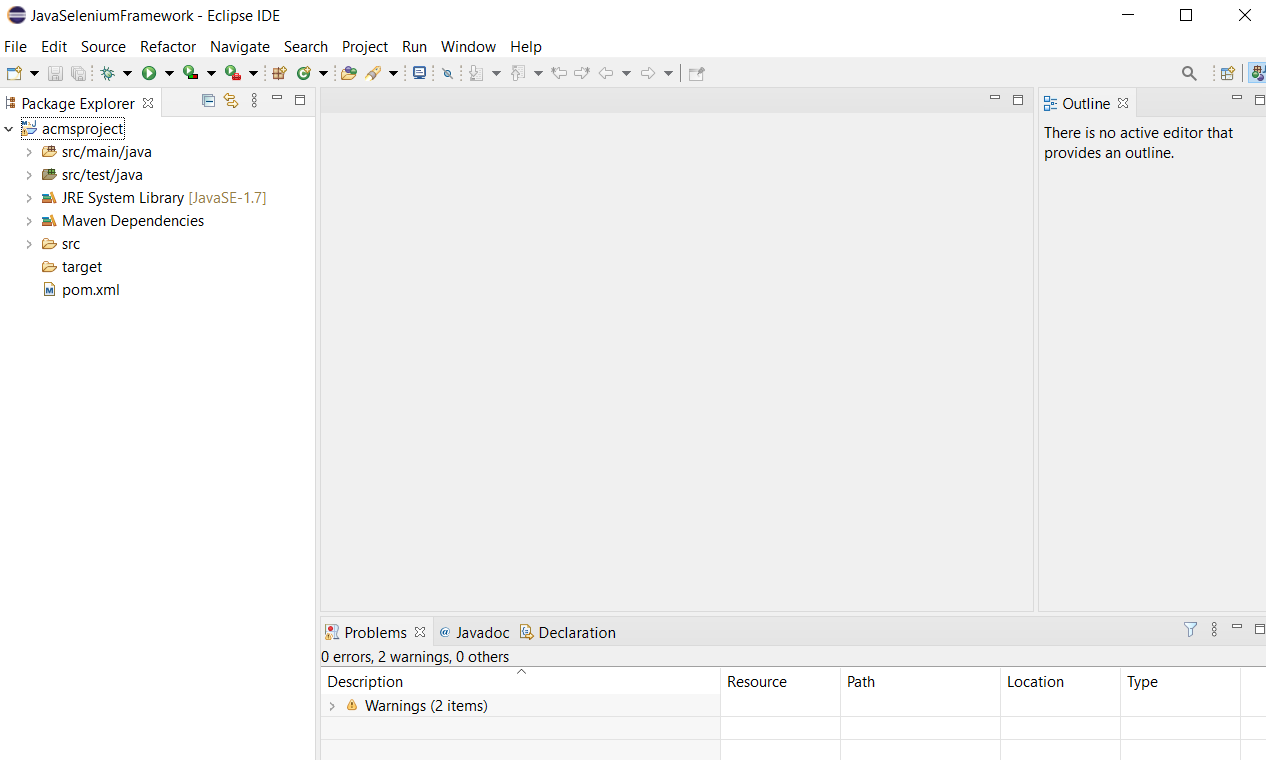
Enter the Group ID: **“gov.acms”**

Enter the Artifact ID: **“acmsproject”**

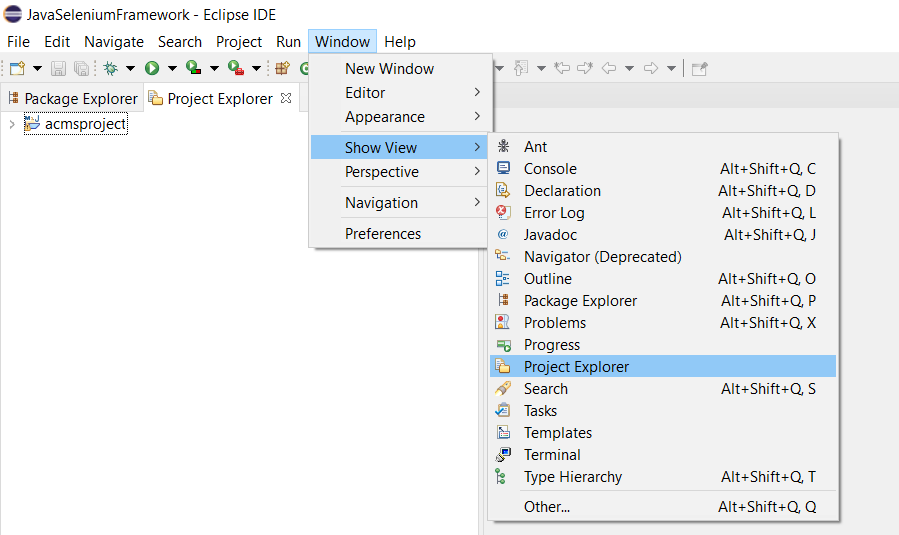
Click on “Finish”



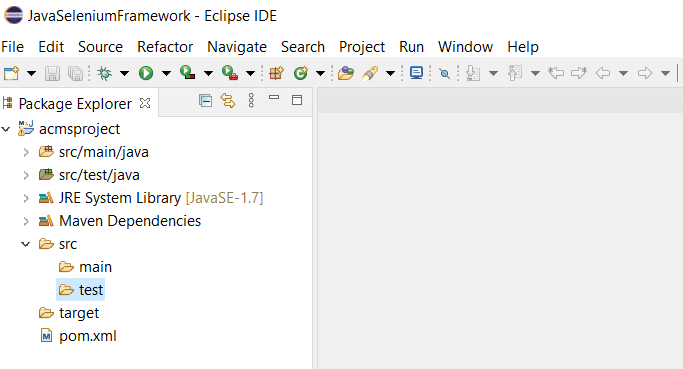
* The project is now created.



* Click Window -> Show View -> Project Explorer



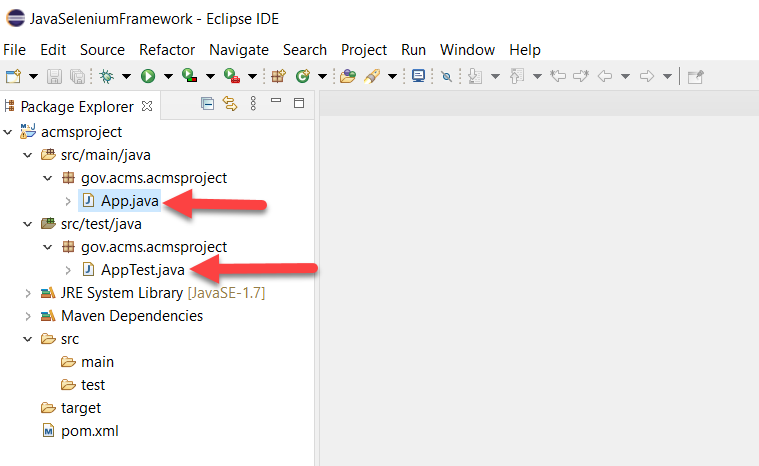
* There are two folders options available in the project under **src: main and test**

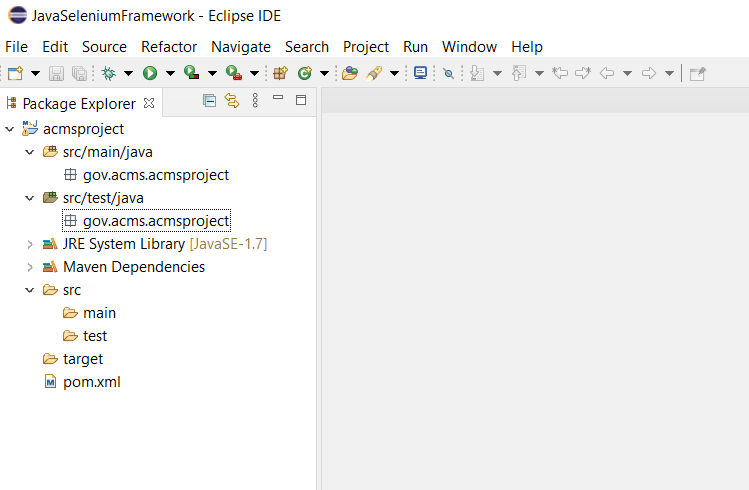


**Test Folder: All the Test Cases will be here**

**Main Folder: All the page object, utilities, or any kind of logging things or resources will be here**

* We can delete “App.Java” file





* Add TestNG: Open “pom.xml” file -> add TestNG dependency under dependencies section

<https://mvnrepository.com/>

* Simply search TestNG maven dependencies. User Maven Repository website to get latest version of dependency

<!-- https://mvnrepository.com/artifact/org.testng/testng -->

<dependency>

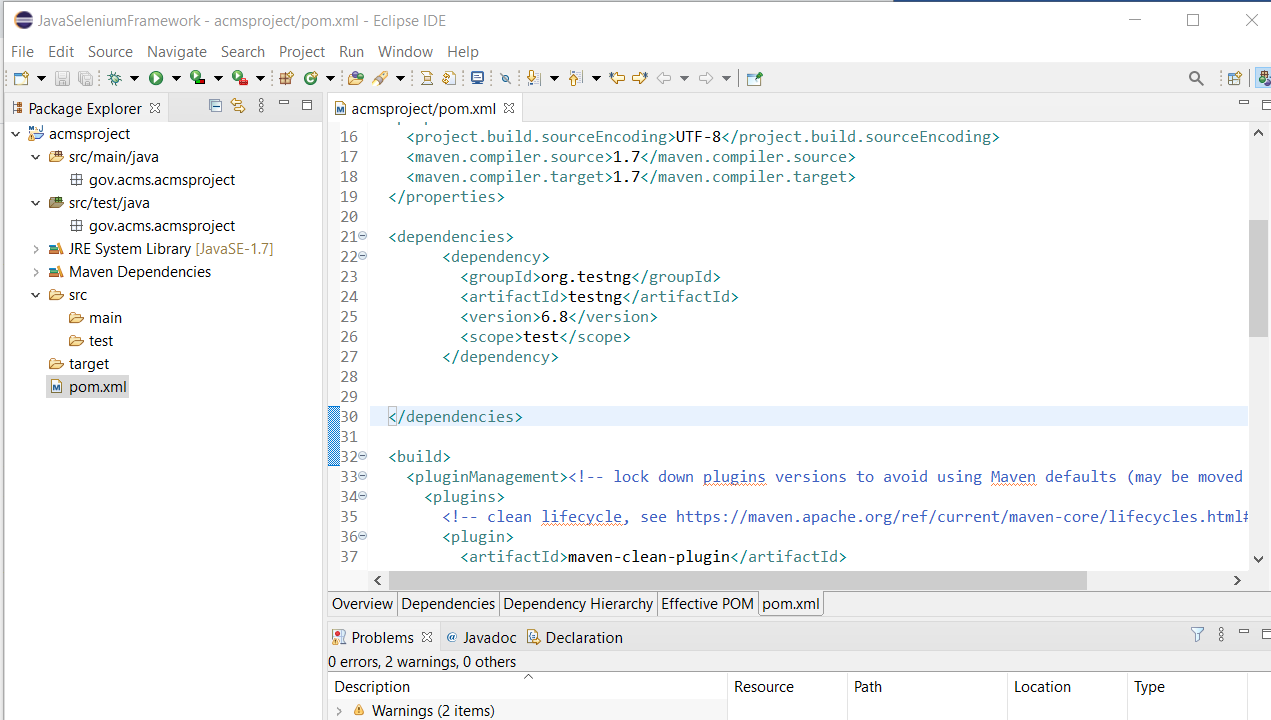
<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>7.4.0</version>

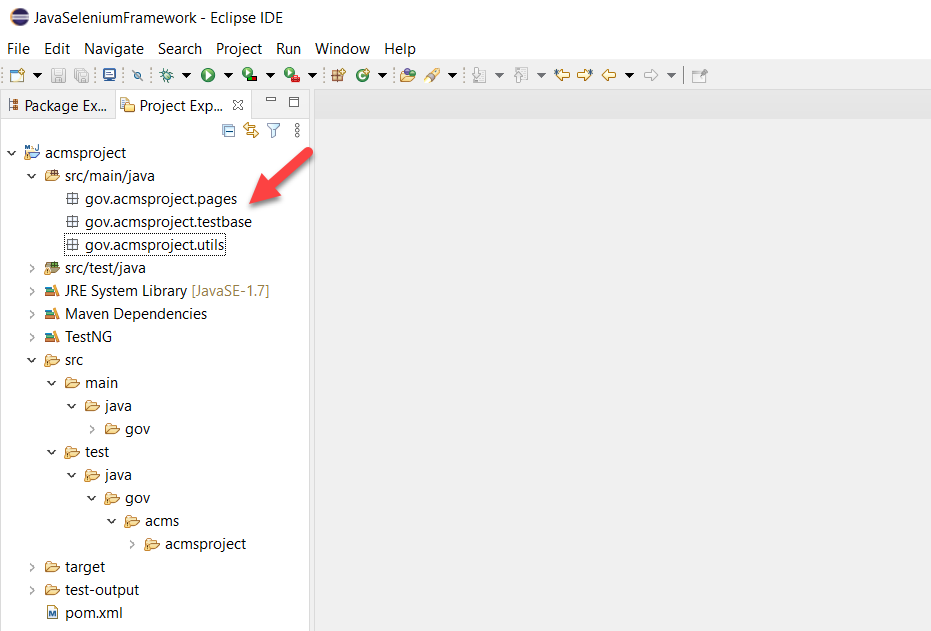
<scope>test</scope>

</dependency>

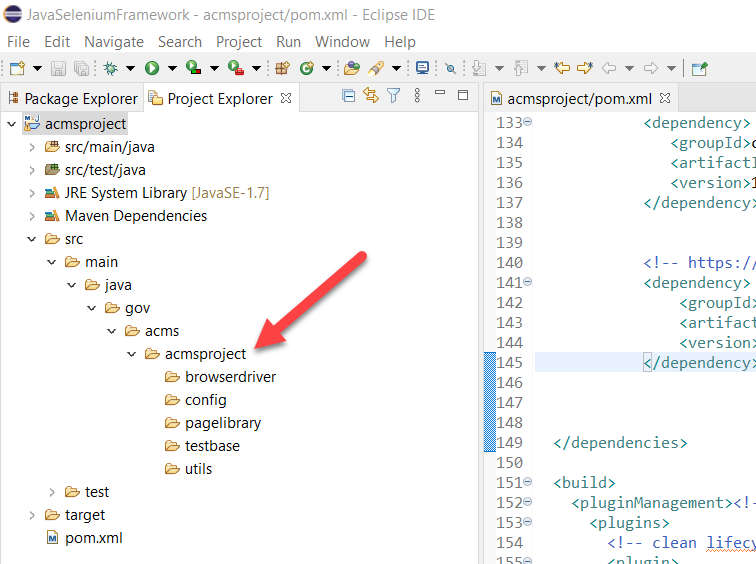


* Add necessary dependencies. Examples

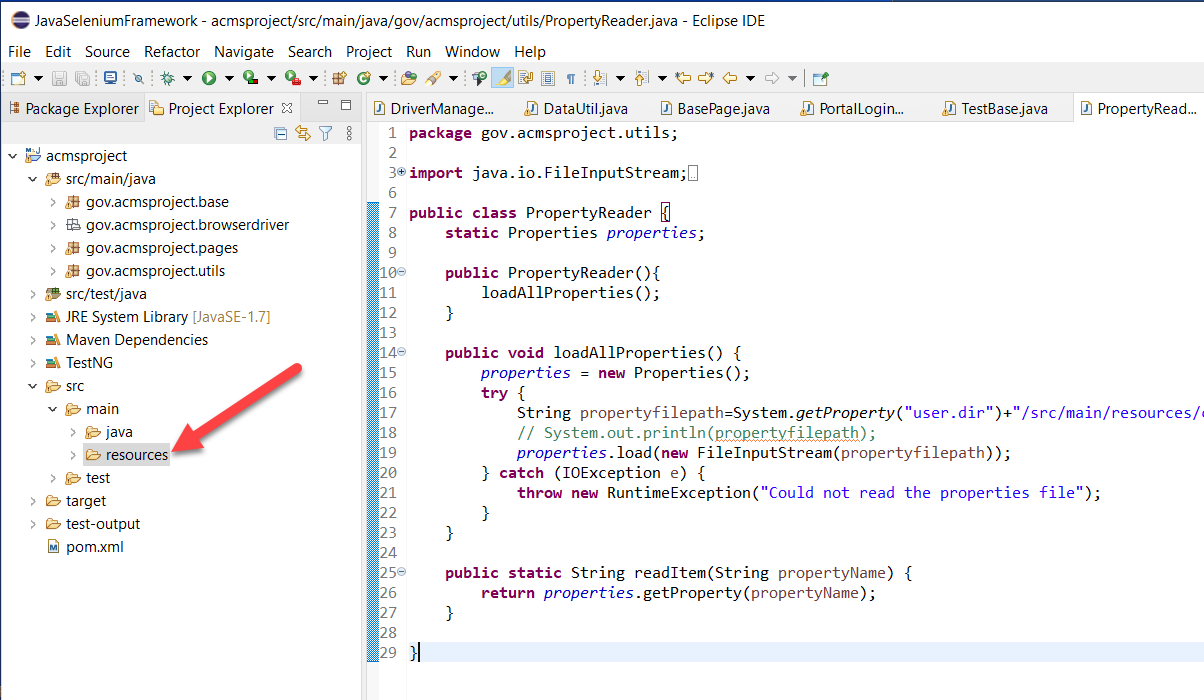
<https://mvnrepository.com/>

* + Selenium Java
  + testng
  + Apache POI -> may need to use older version
  + poi-ooxml -> may need to use older version
  + poi-ooxml-schemas -> may need to use older version
  + commons-io
  + poi-scratchpad --> may need to use older version
  + ooxml-schemas
  + openxml4j
  + webdrivermanager
  + log4j-api
  + log4j-core
  + log4j
  + extentreports
  + extentreports-testng-adapter
  + maven-surefire-report-plugin
* In Main Folder: Create “utils” , “config”, “pagelibrary”, “base”, “browserdriver” package 

Project Explorer View:



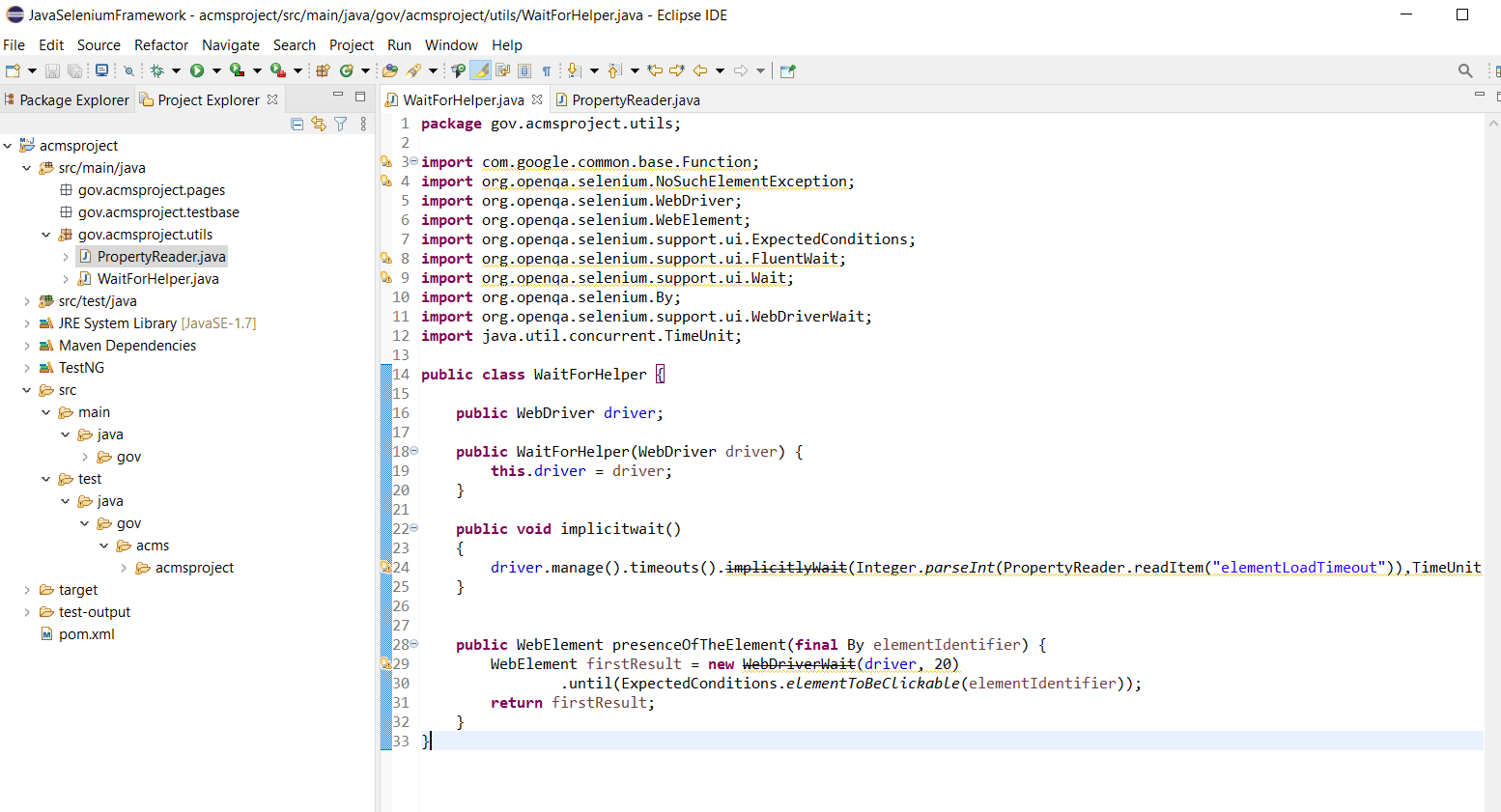
* Create a Resource Folder under Main



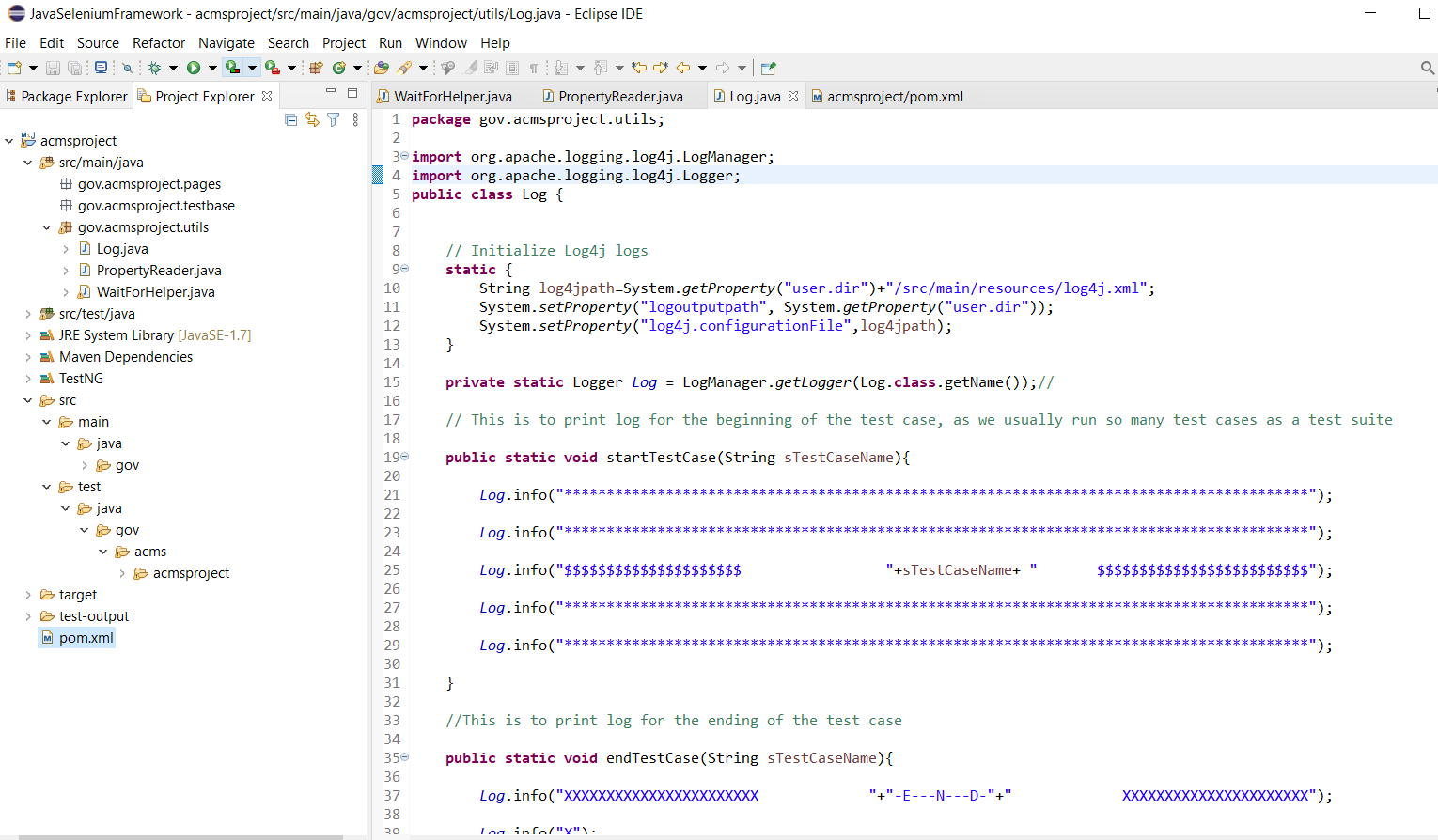
* In “utils” package -> create “PropertyReader” class



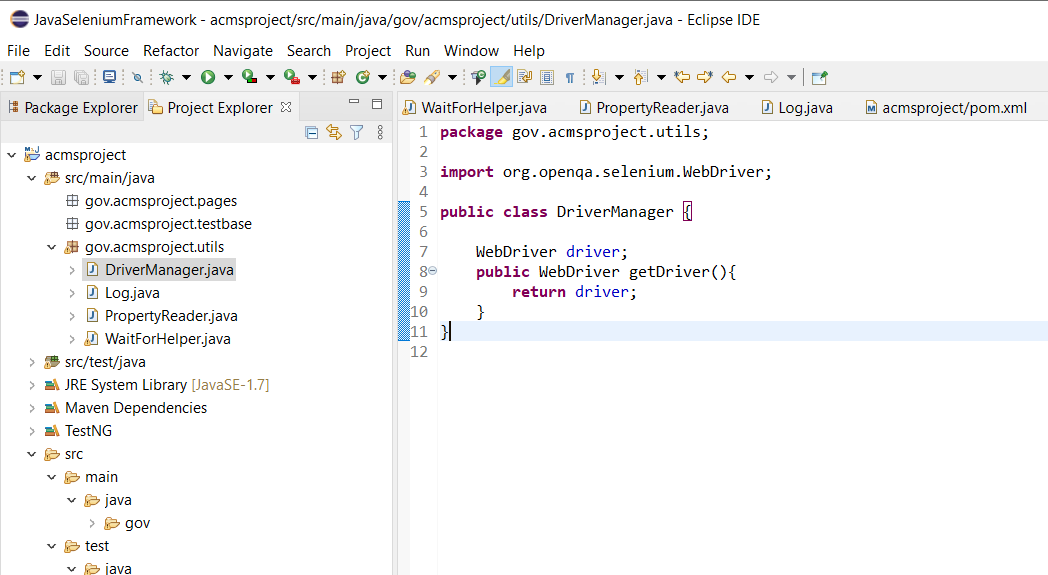
* In “utils” package -> create “WaitForHelper” class



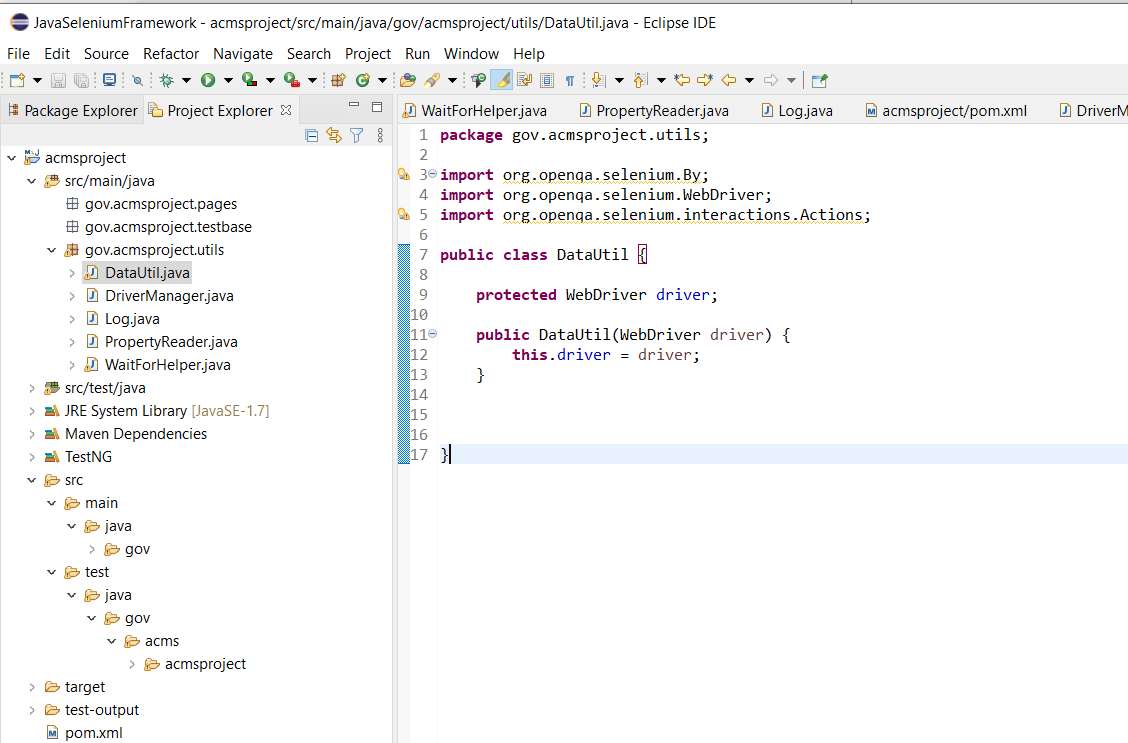
* In “utils” package -> create “Log” class



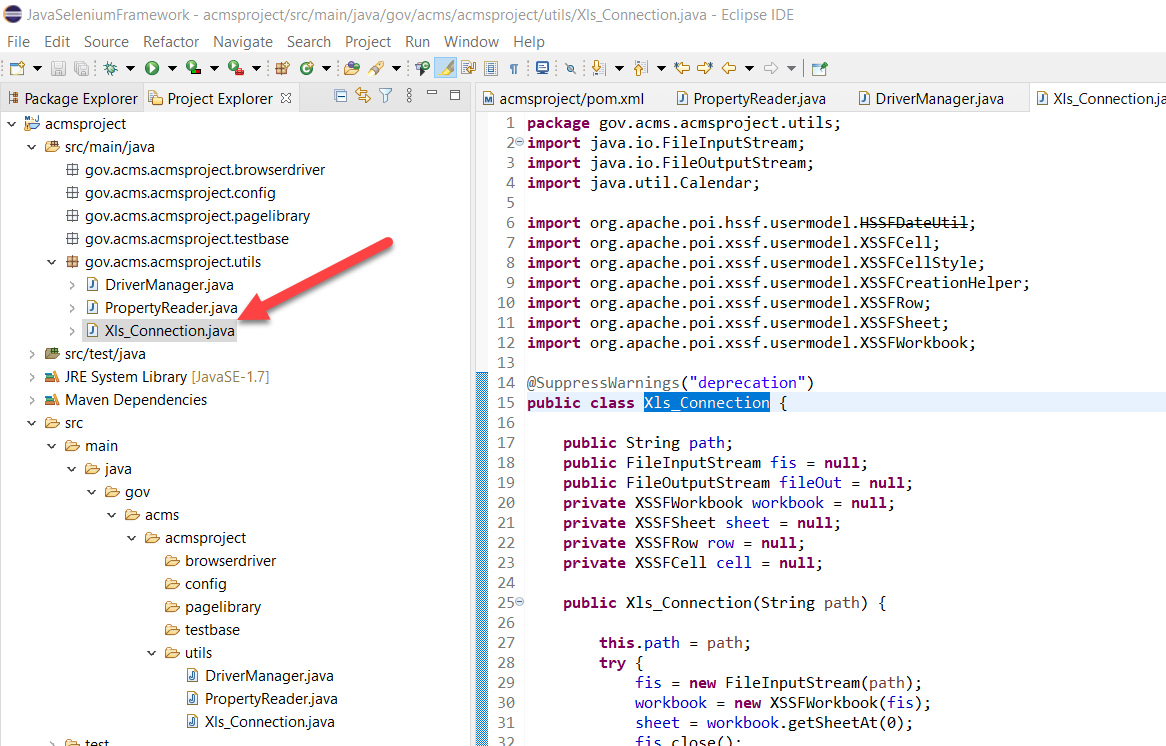
* In “utils” package -> create “DriverManager” class



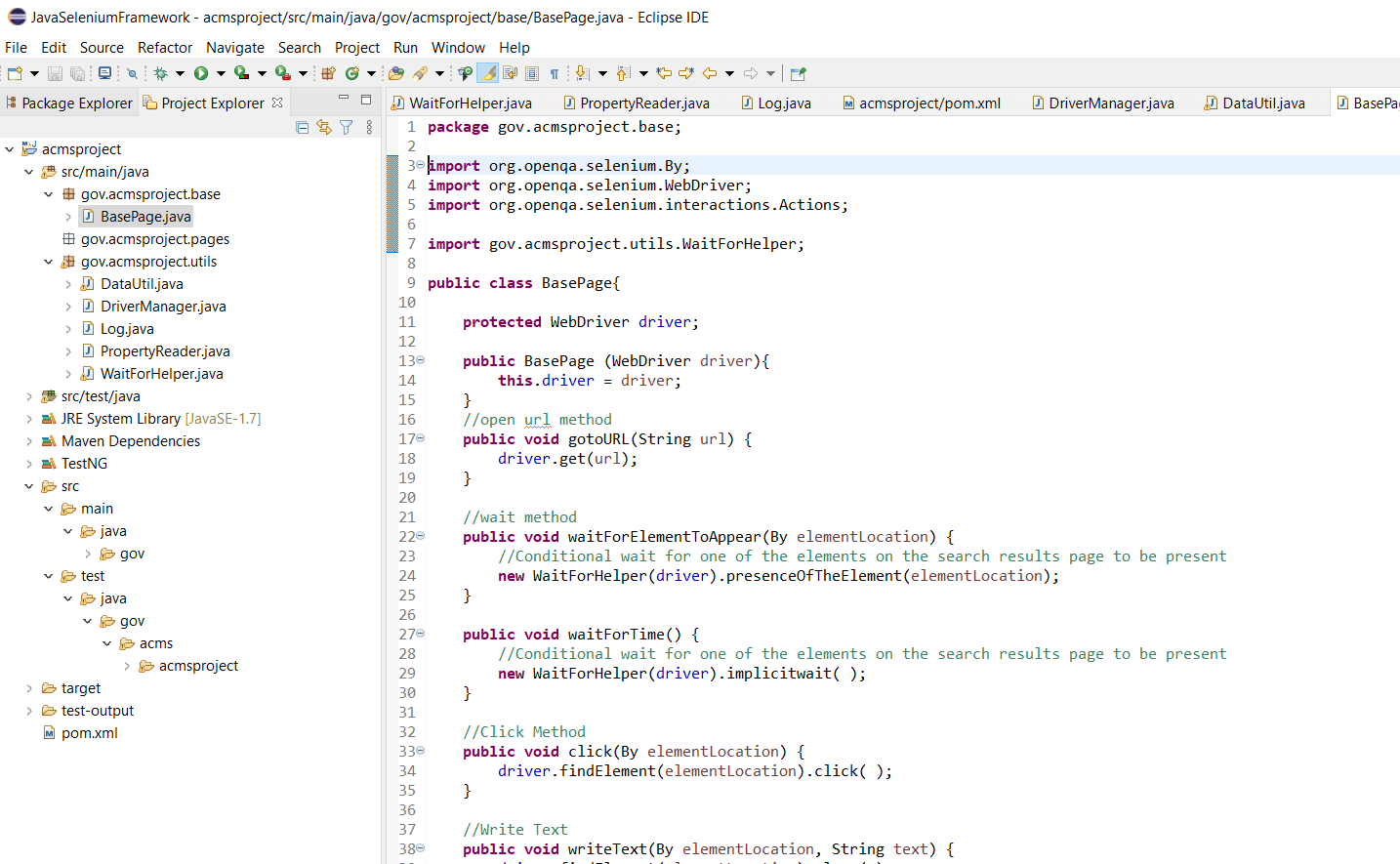
* In “utils” package -> create “DataUtil” class



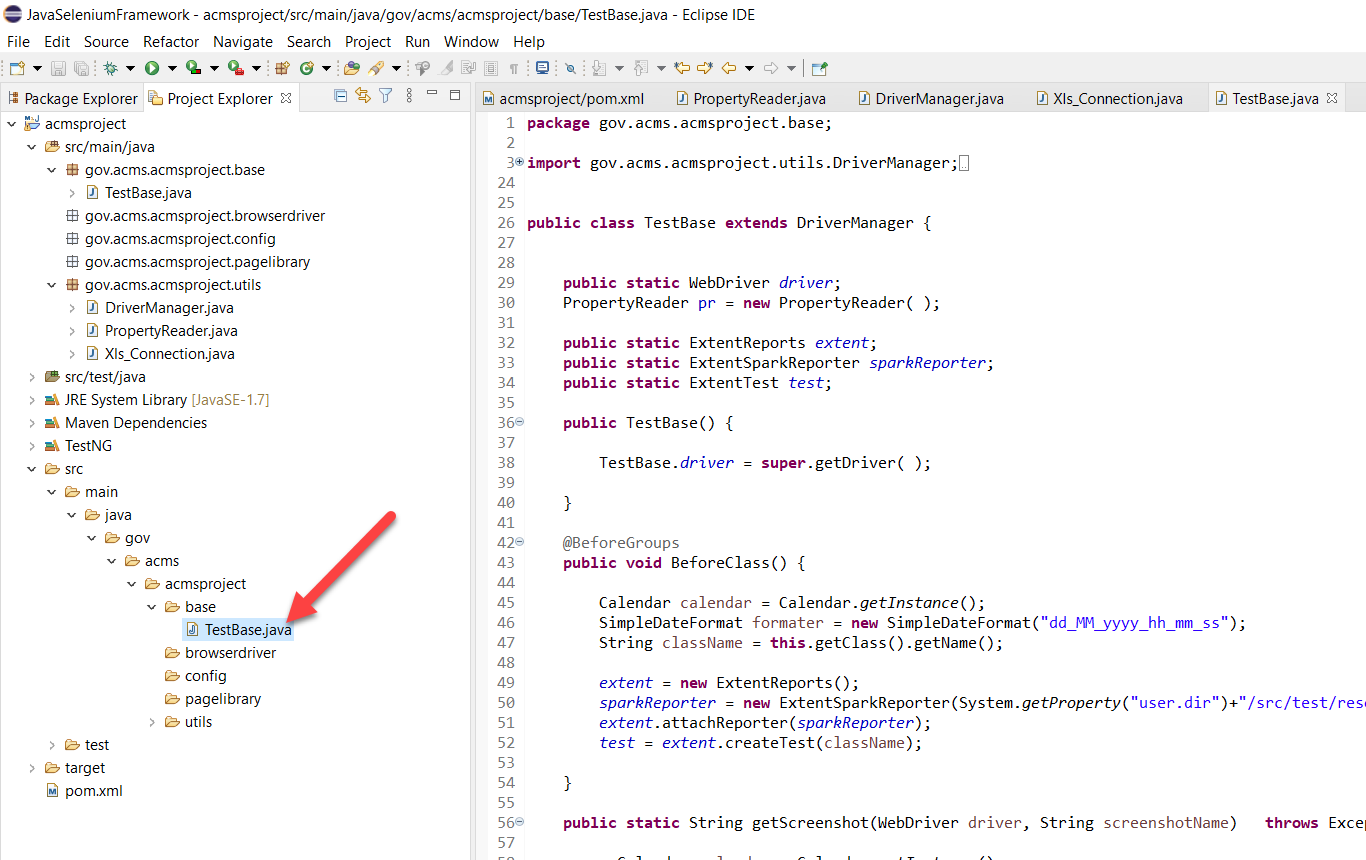
* In “utils” package -> create “Xls\_Connection” class



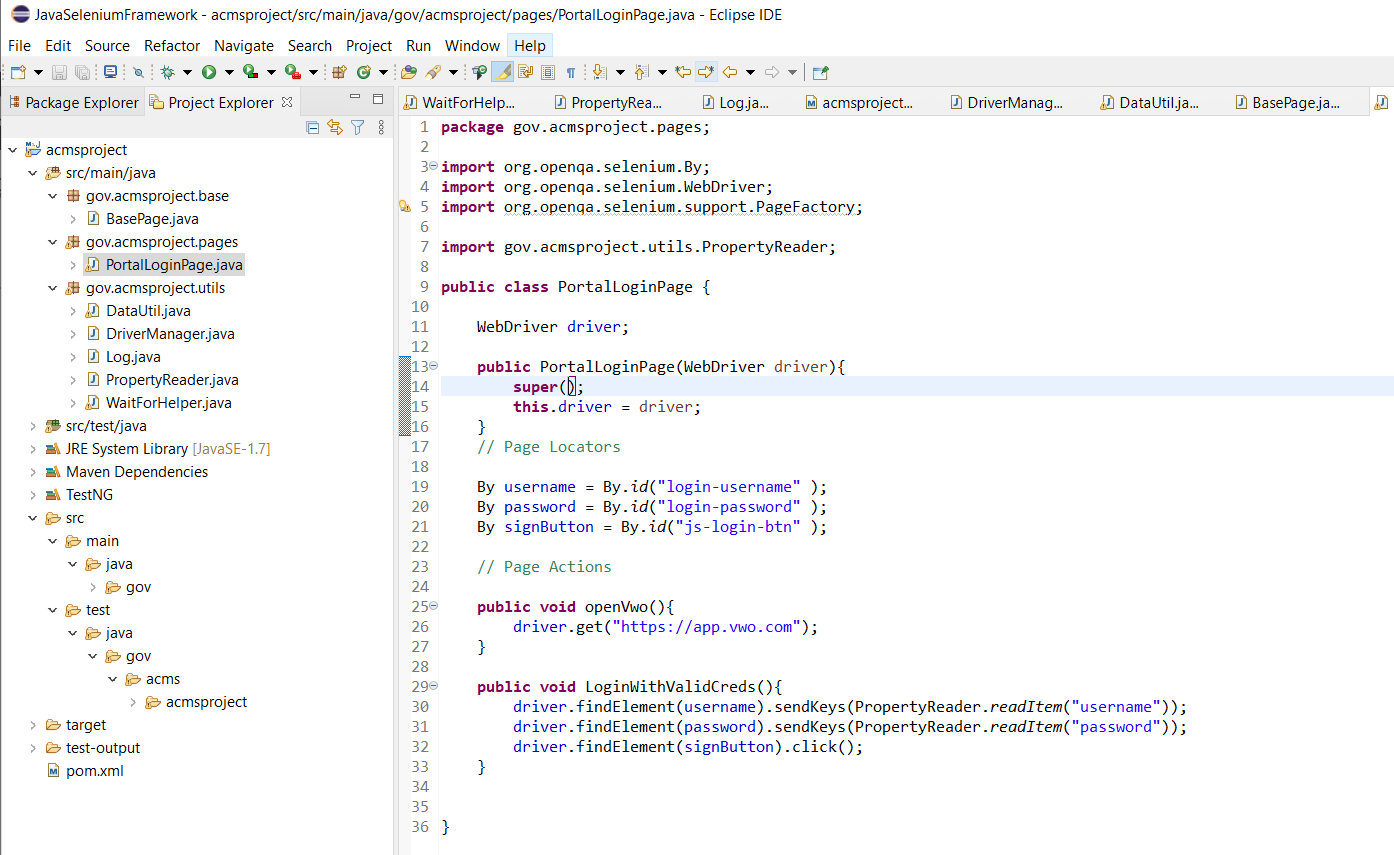
* In “base” package -> create “BasePage” class



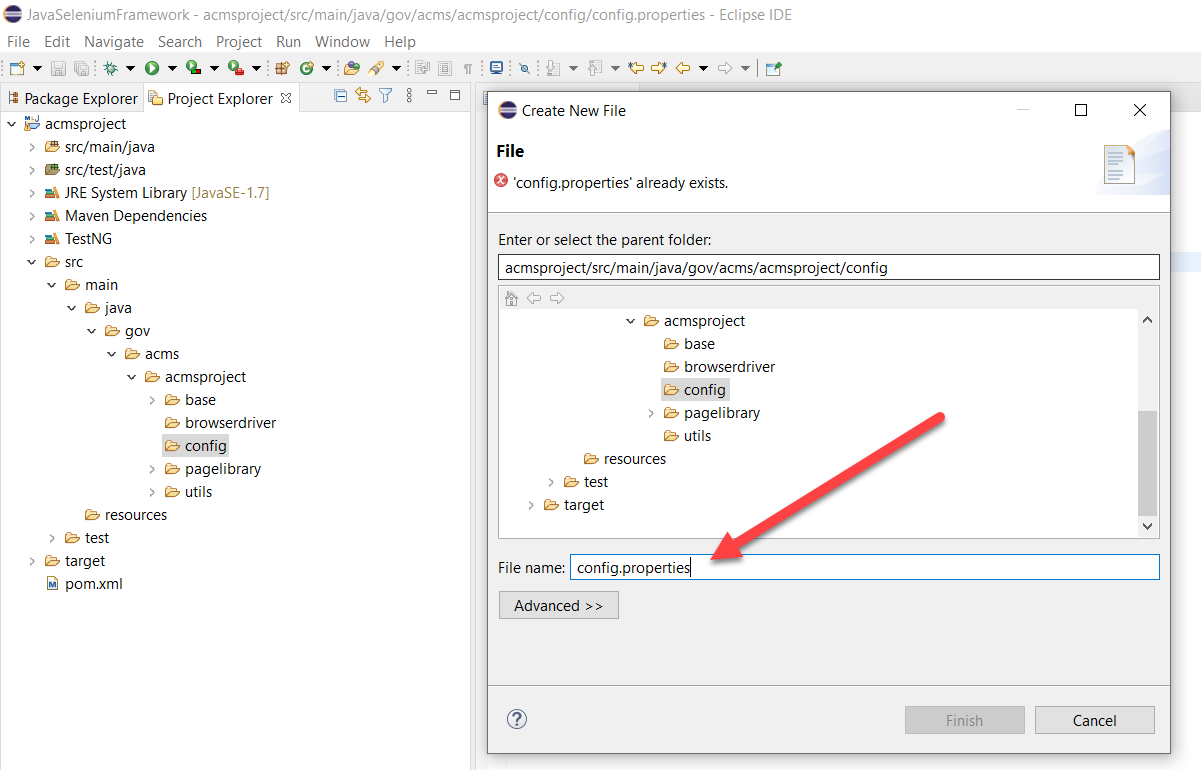
* In “base” package -> create “TestBase” class and extends DriverManager class



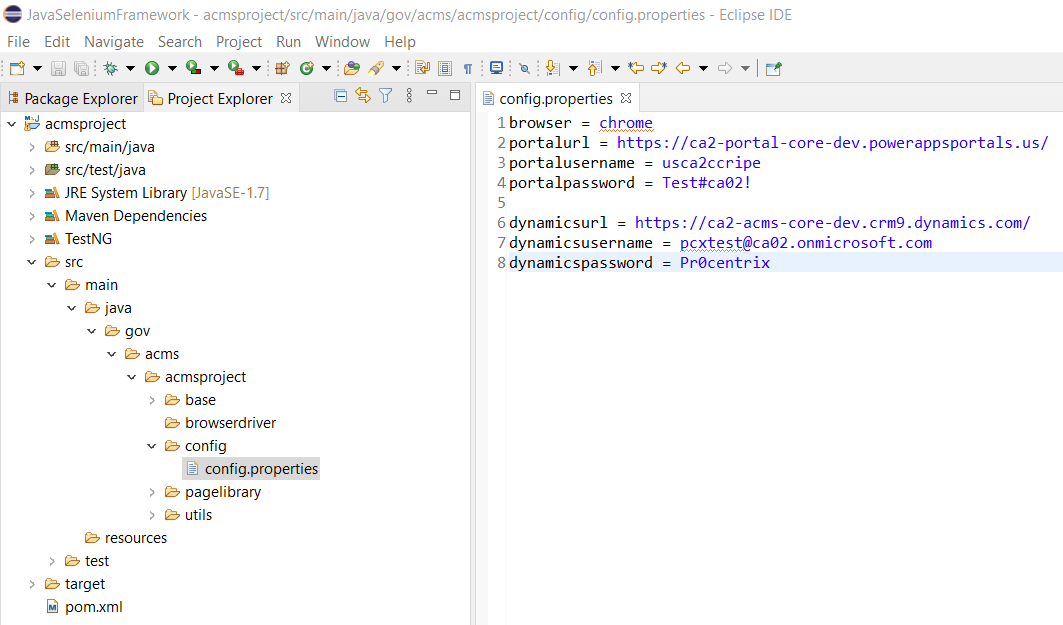
* In “pagelibrary” package -> create “PortalLoginPage” class and extends BasePage



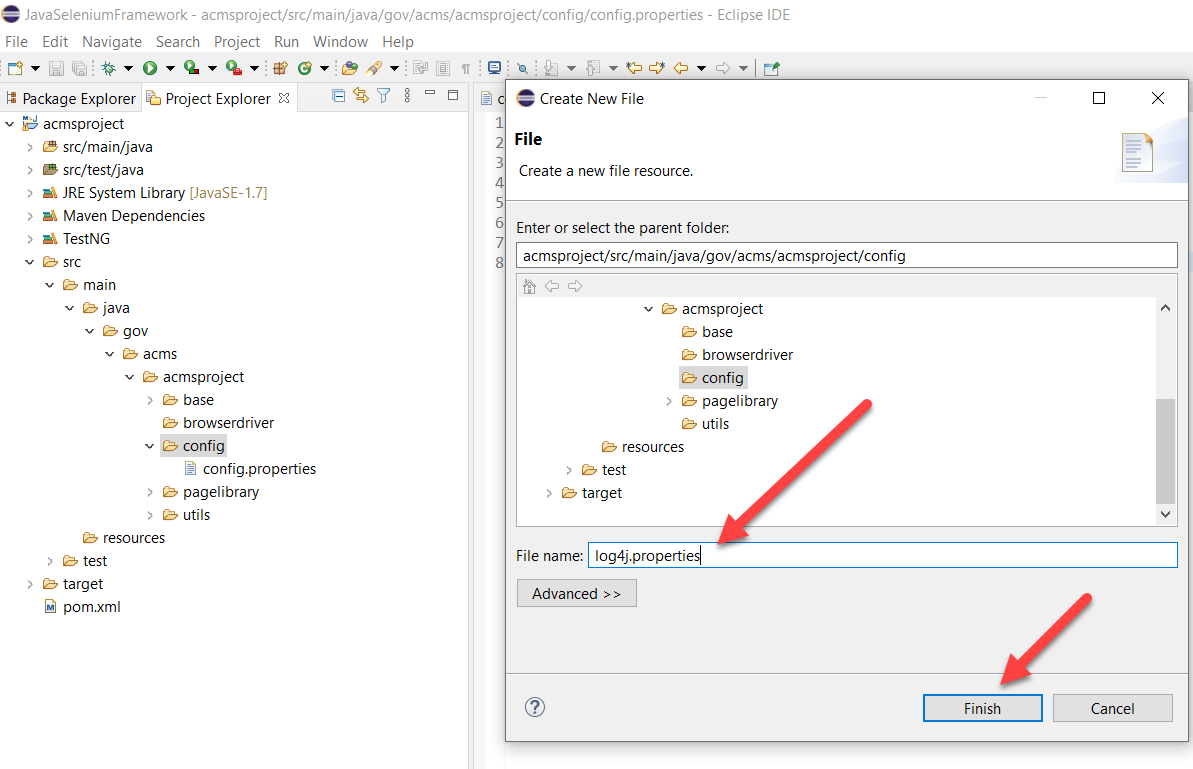
* Config folder under main **Create a “config.properties” file ->Right click on resources folder -> new -> file -> enter file name as “config.properties” -> click Finish**



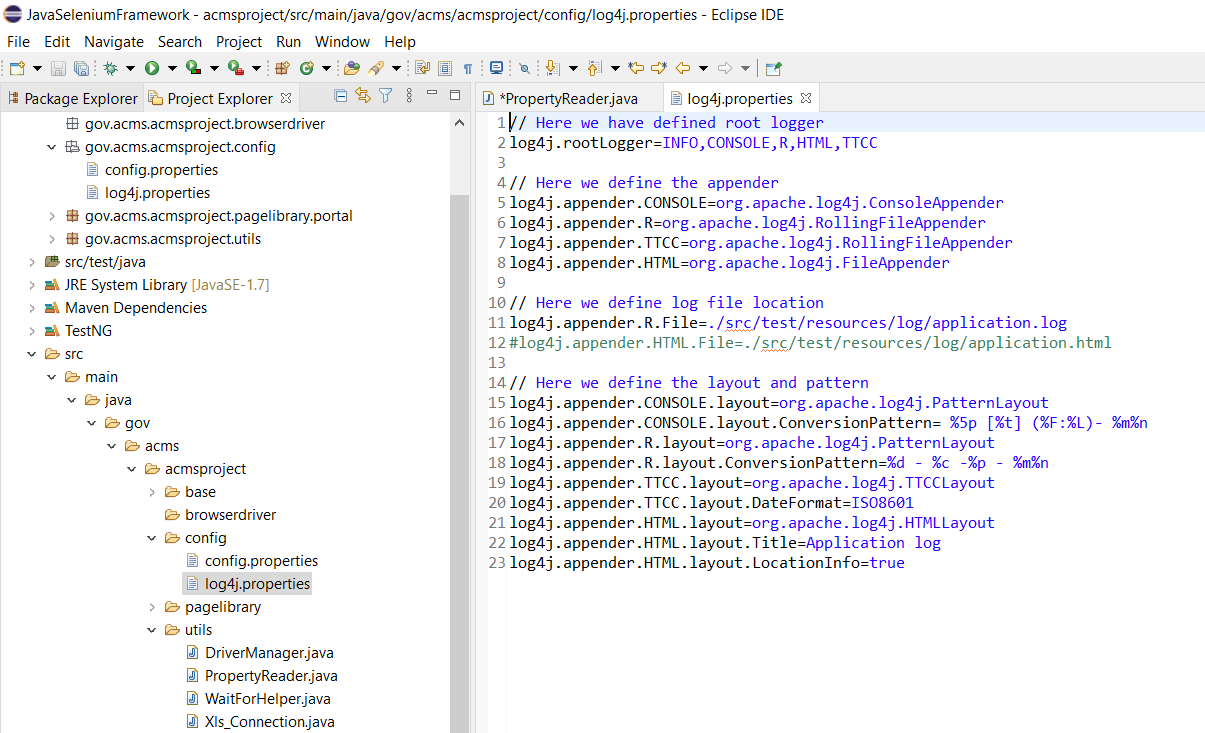
**Enter appropriate browser, url, usename, password information.**



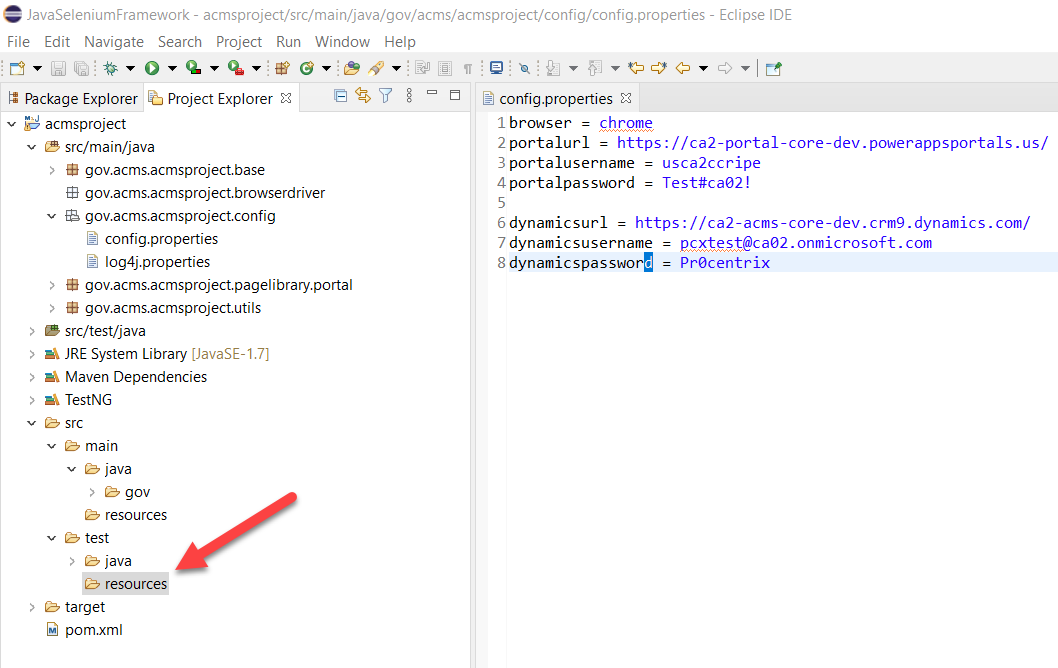
* Config folder under main **Create a “log4j.properties” file ->Right click on resources folder -> new -> file -> enter file name as “log4j.properties” -> click Finish**



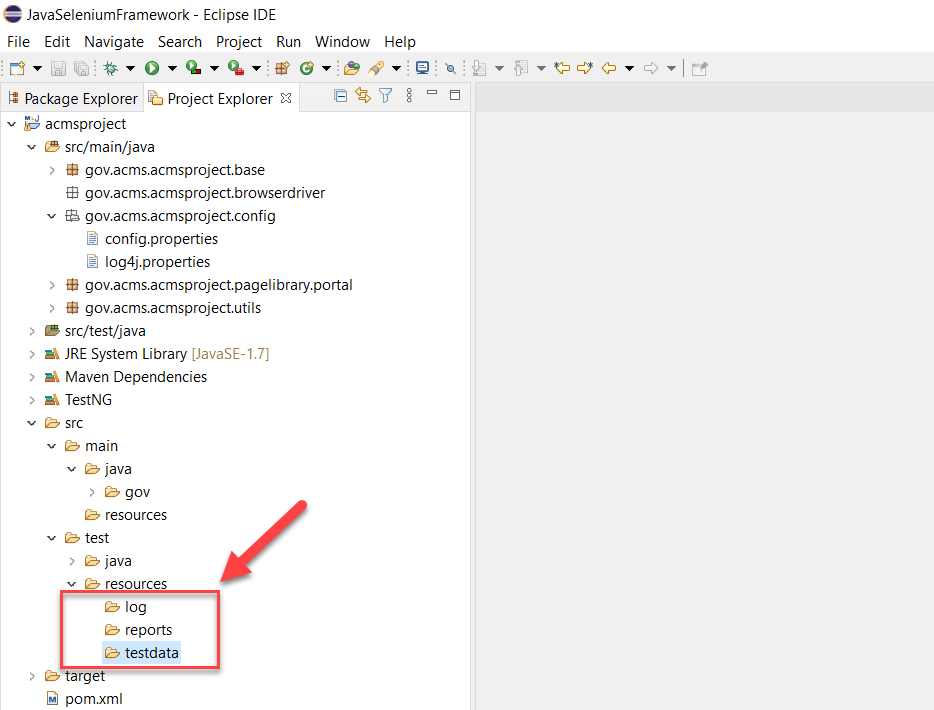
Enter below information.



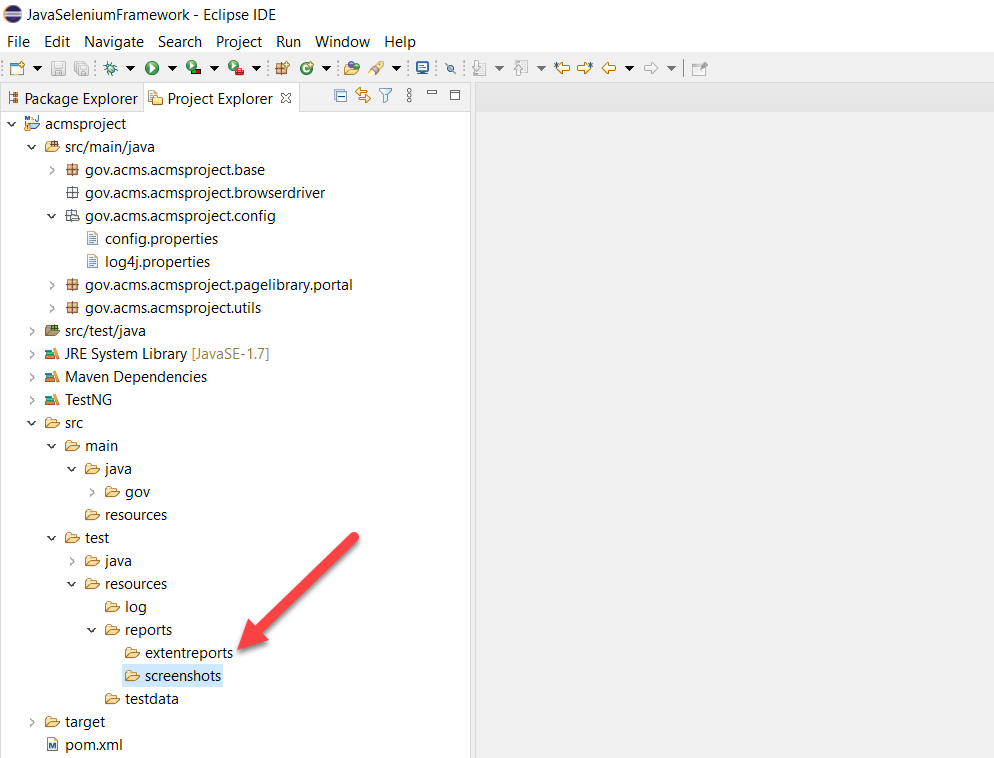
* Create a Resource Folder under Test



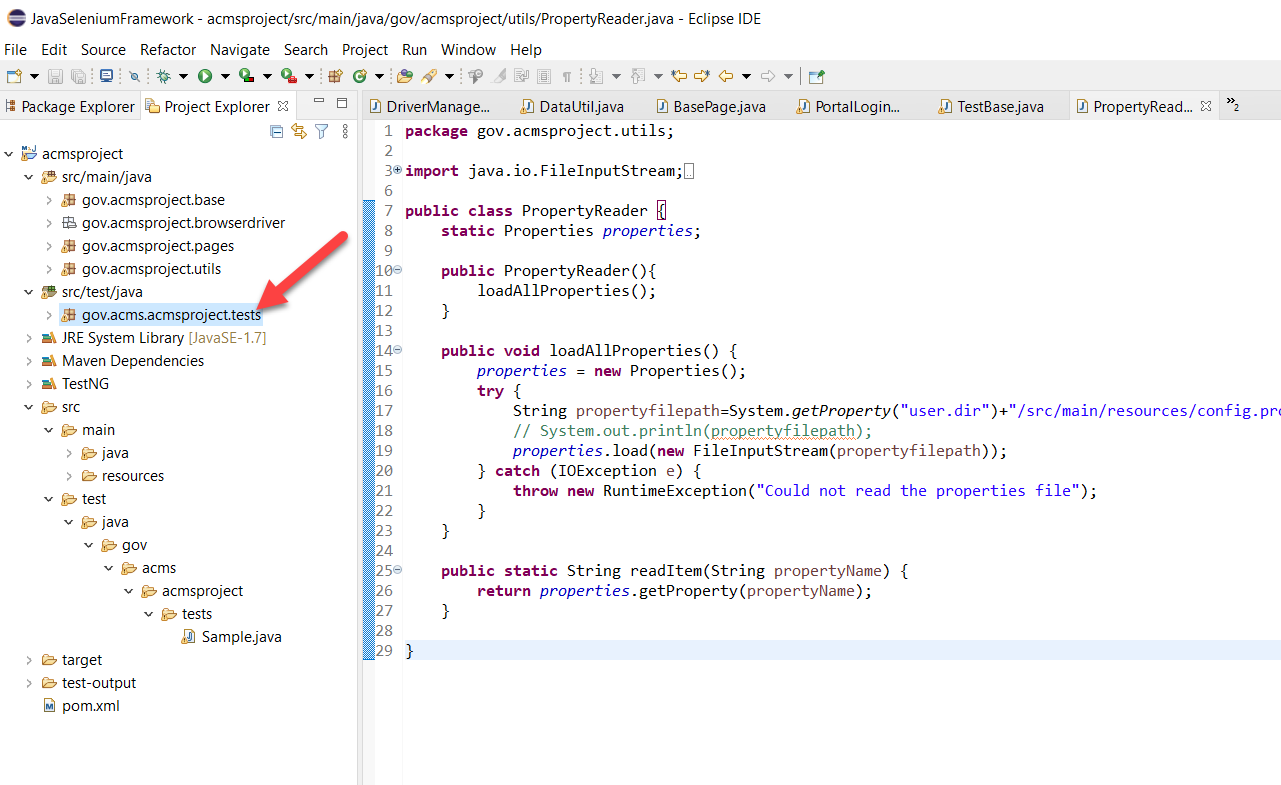
* Create 3 more folder under resources (test) -> log, reports, testdata



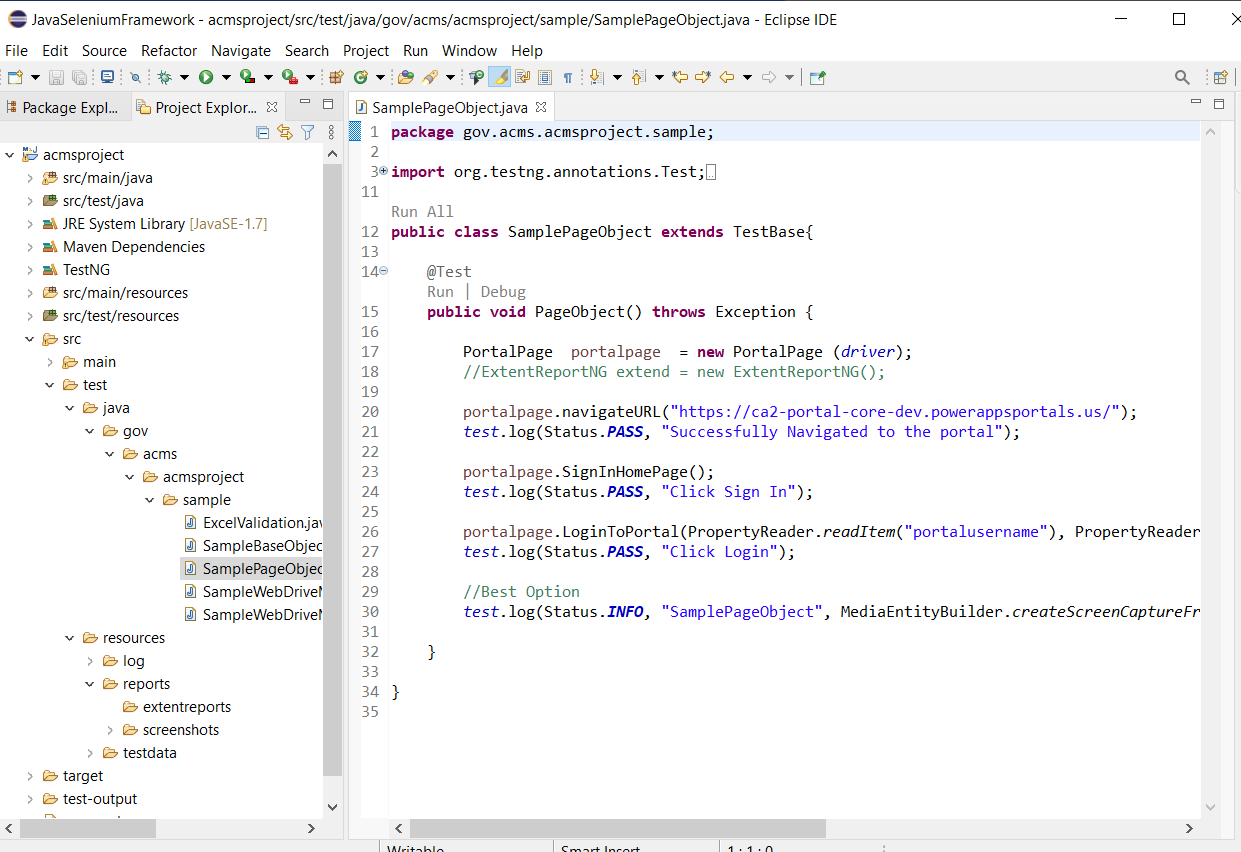
* Create 2 more folders under reports-> extentreports, screenshots



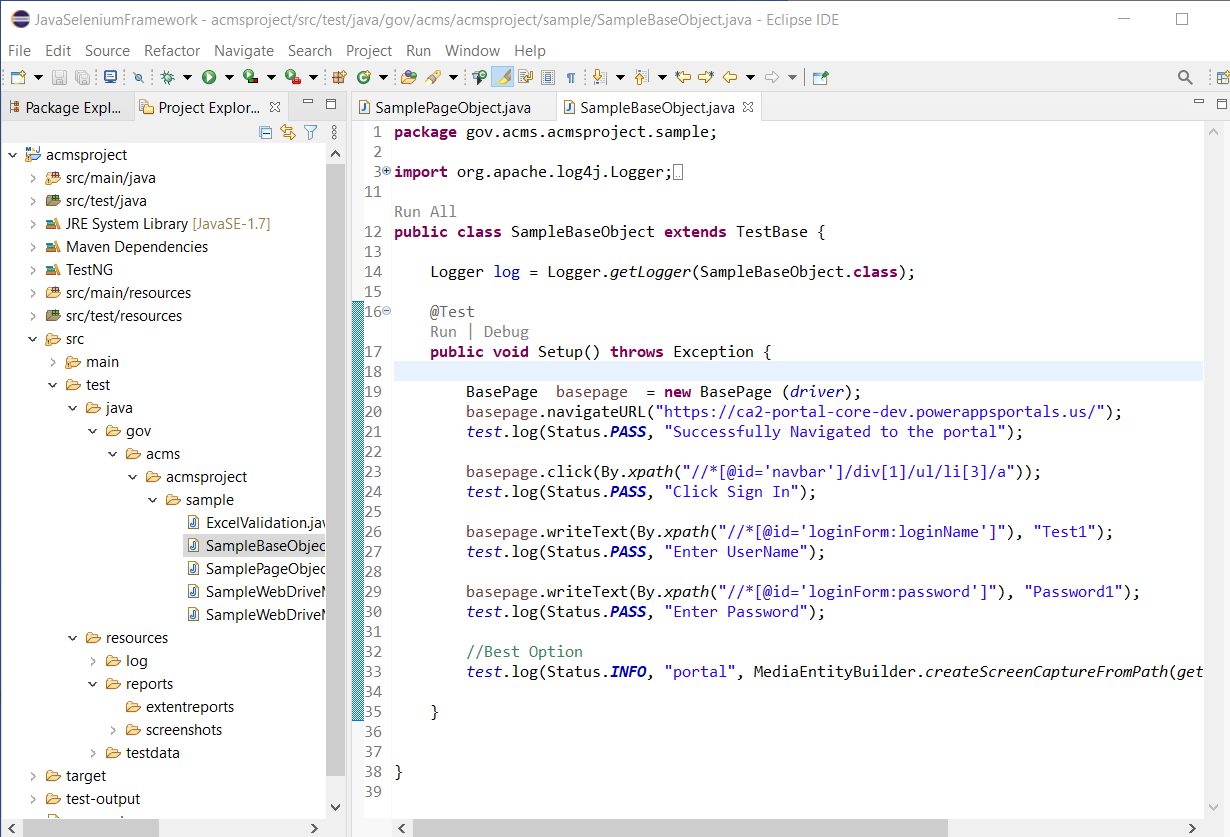
* Create a “tests” package underFolder under Test Folder



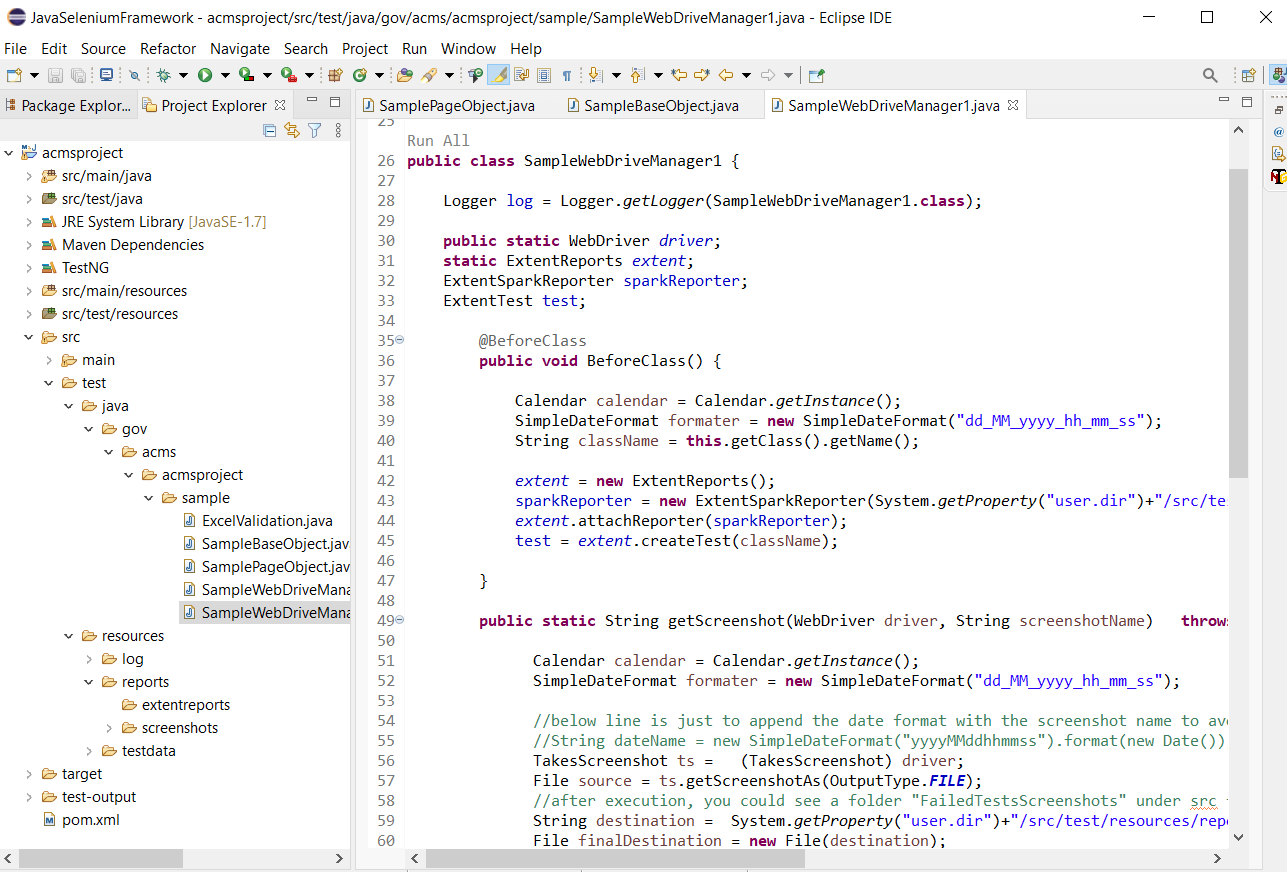
* Create a Test Class in tests package. May need to create sub folder depending on the application and modules
* Sample Test1:



* Sample Test2:



* Sample Test3:



* Pom File



* Base:



* Pagelibrary



* Utils

* Resources (Main)



* Sample Test



<https://www.simplilearn.com/tutorials/maven-tutorial/maven-project-in-eclipse>

<https://github.com/anujkumar21/JWAF>

<https://github.com/apitestingco/selenium-pom/tree/main/src/test/java/com/sdetblueprint>

<https://github.com/prakashnarkhede/Advanced_Selenium_Test_Automation_Framework>

<https://github.com/deepika-rao/Page-Object-Model-POM-Design-With-Selenium/tree/master/src>

<https://www.youtube.com/watch?v=917gq9mn8jA>

<https://www.youtube.com/watch?v=U02Ob9JRH84>

* Extent HTML Reporter and Extend logger reporter. Both of these has been deprecated.
* Now new report is called ExtentSparkReporter

<https://www.youtube.com/watch?v=7FbKUd4VL-0>

<https://www.youtube.com/watch?v=7FbKUd4VL-0>

<https://github.com/extent-framework/extentreports-testng-adapter/blob/master/config/extent.properties>

Create Extend Report:

<https://www.youtube.com/watch?v=aPDF6HfzVck>