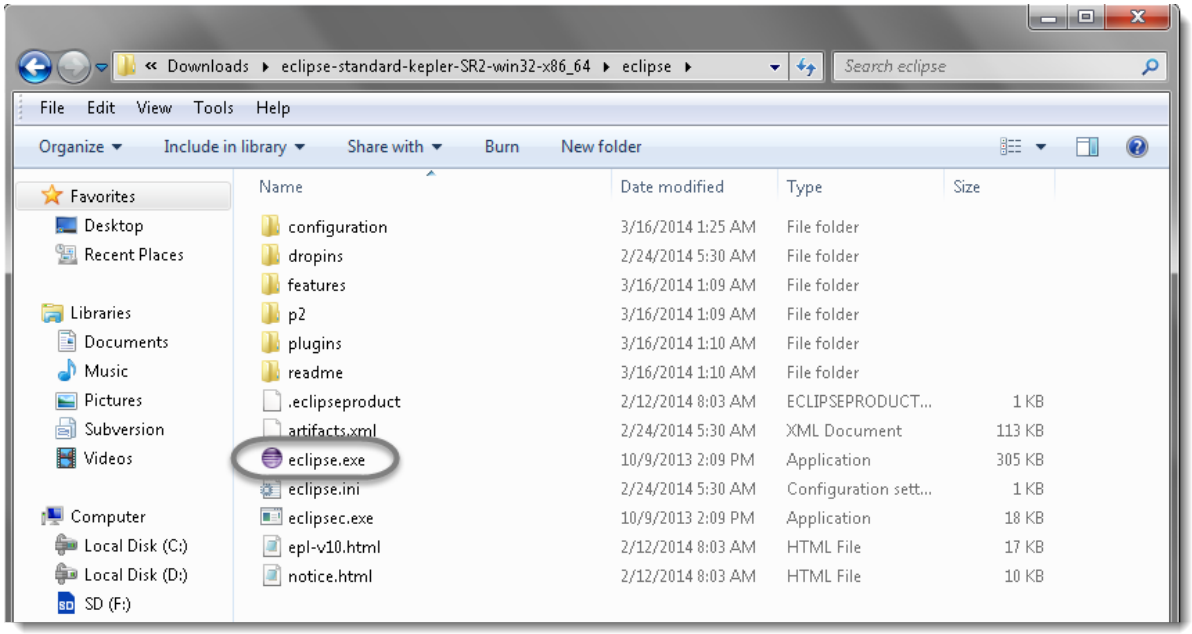
Configure Eclipse with Selenium Webdriver

To configure Eclipse with Selenium webdriver, we need to launch the **Eclipse IDE**, create a **Workspace**, create a **Project**, create a **Package**, create a **Class** and add **External libraries** to the project.

1) Select WorkSpace on Eclipse start up

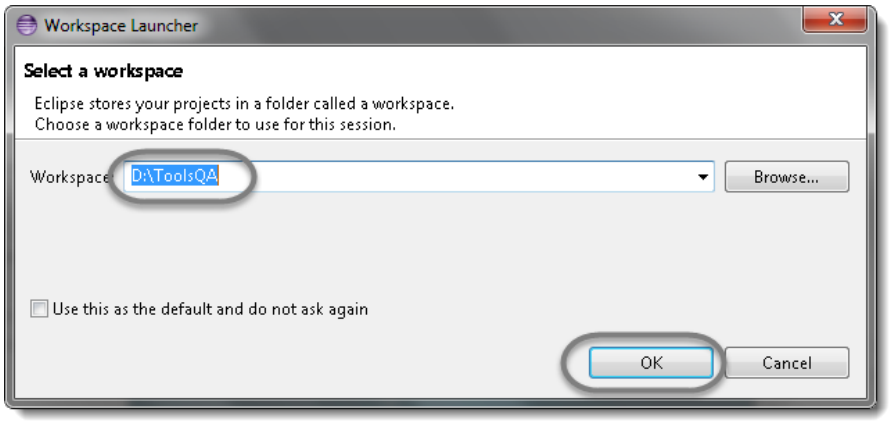
a) **Double click** on 'eclipse.exe' to start eclipse. First time when you start eclipse, it will ask you to select your workspace where your work will be stored as shown in below image.

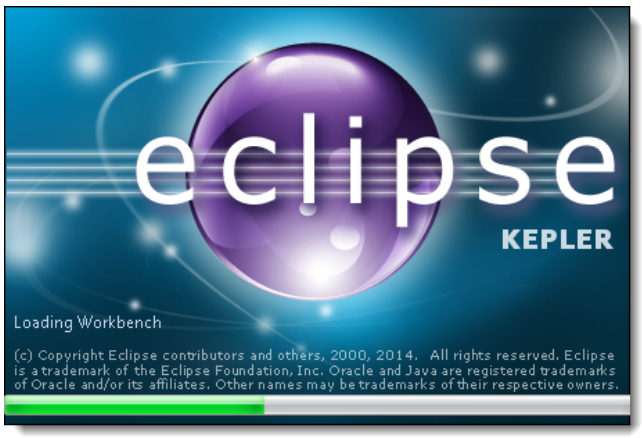


b) Create a "**working directory**" for all of your projects. Think of it like "**My Documents**" in the Windows operating system. It's a folder which contains a lot of your documents, but there's nothing to prevent you from creating another folder called "My Other Documents" (for instance) to house other documents.

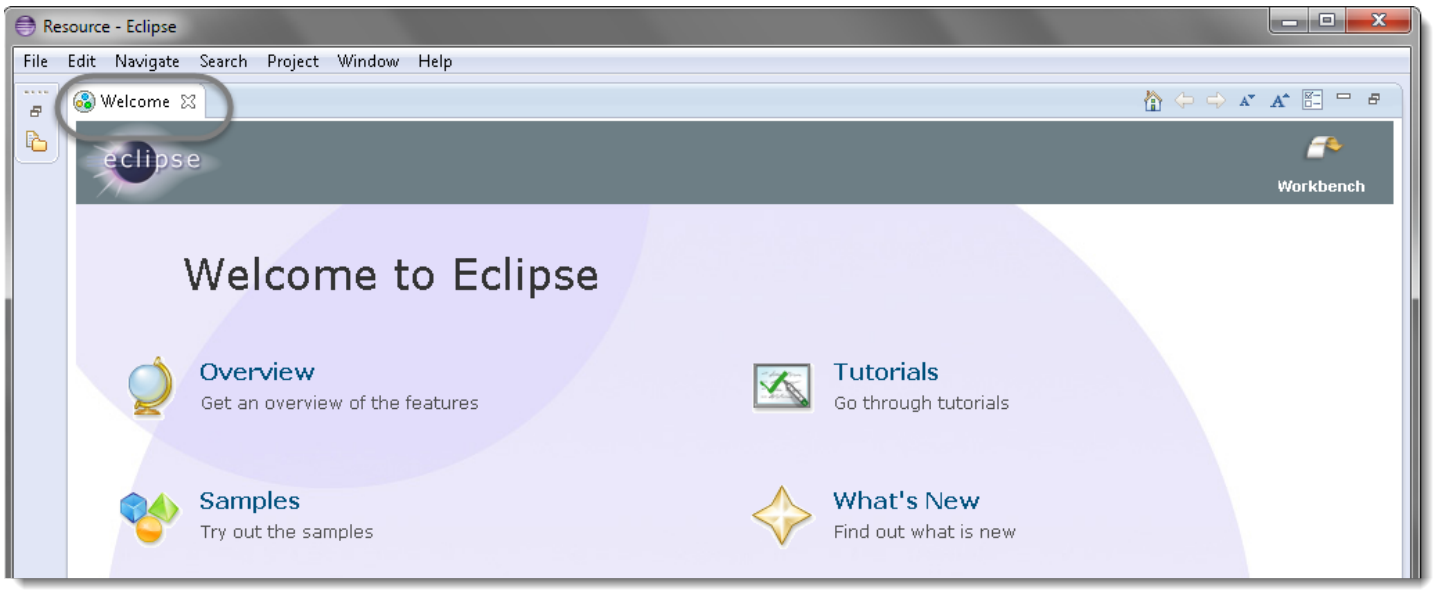
Typically you only need one workspace, and you can think of it as your "My Documents" for Java code. If you wanted to, you could have more than one, but chances are you won't have a use for more. I like to choose my own workplace location and will place all my **ToolsQA** tutorial projects under it.

You can change it later on from '**Switch Workspace**' under '**File**' menu of eclipse. After selecting workspace folder, Eclipse will be open.





c) You may see the window like this, this is the **Welcome** window for Eclipse. You may **close** this window.

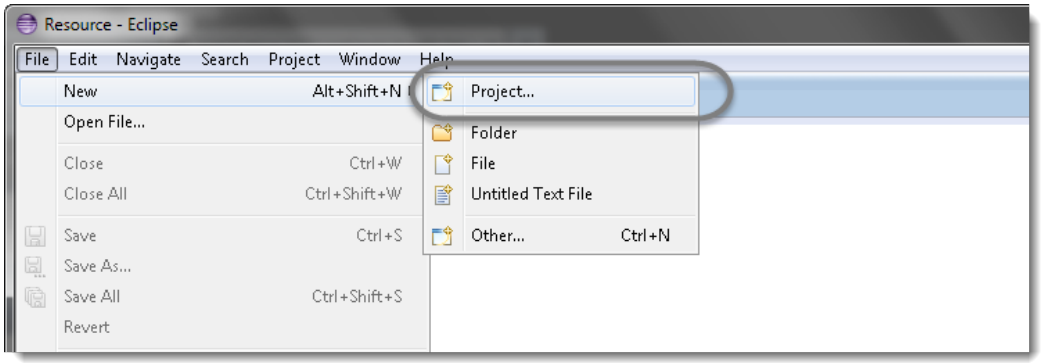


2) Create a new Project

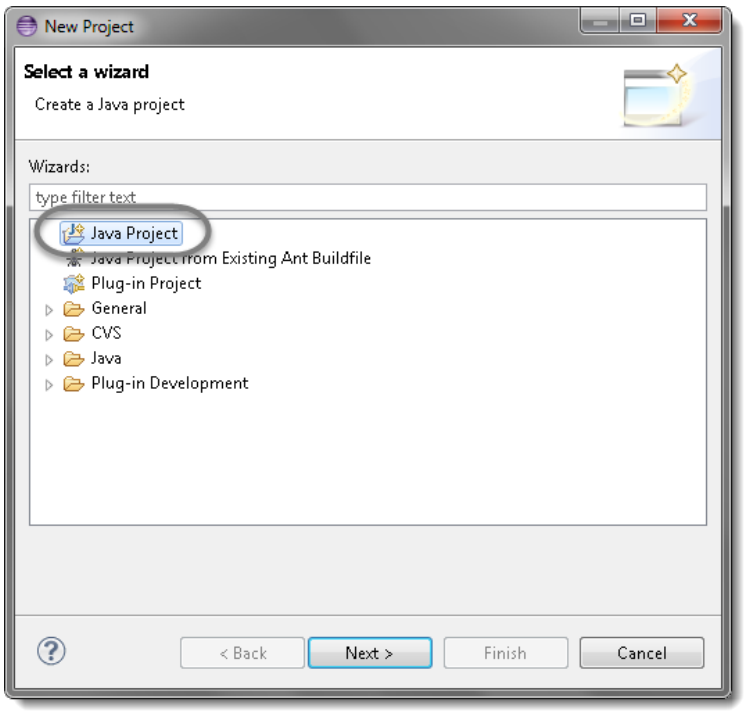
Projects: A collection of related code. Generally speaking, each project encompasses one independent program. Each programming assignment you do will typically require its own project.

Once you've established your workspace, you'll want to create a project and begin writing code. In Eclipse, projects are the next-smallest functional unit after workspaces, but where you might have only one workspace, you will usually have several projects inside one workspace.

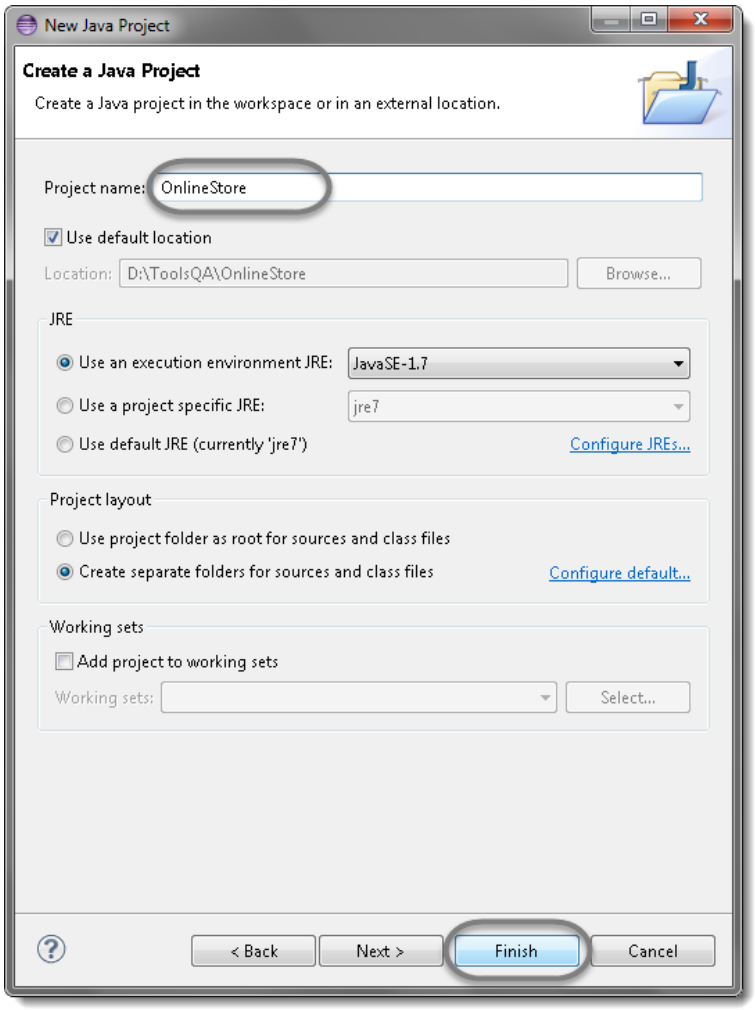
a) Create new Java Project from **File** > **New** > **Project** .



b) Select **Java Project** and click **Next**.



c) Give your Project name '**OnlineStore**' as shown in below given figures. Click on **Finish** button.



Note: I am naming this as OnlineStore, as Toolsqa.com gives you the opportunity to practice automation on Live demo site. This demo website is a complete ecommerce website where a customer can register and purchase electronic items.

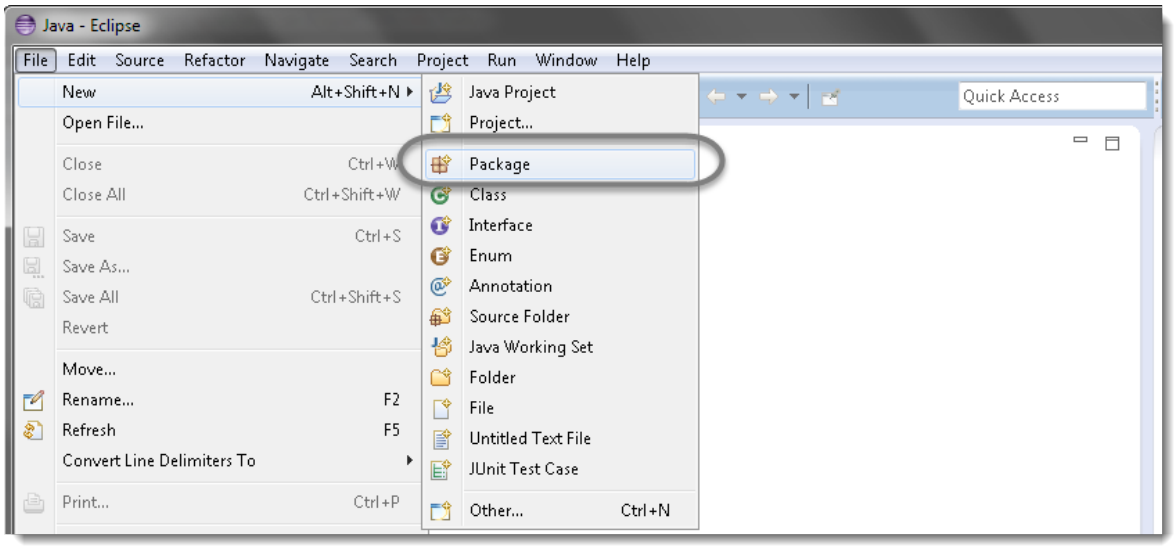
Website : www.onlinestore.toolsqa.com

d) You may or may not see this message but if in case you get any, check **Remember my decision** and click on **Yes**.

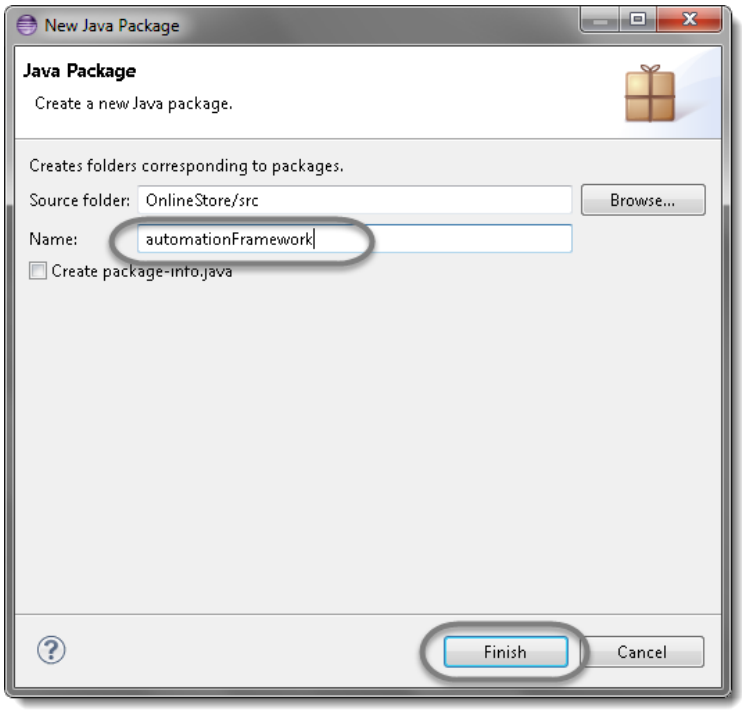
Now your new created project 'OnlineStore' will display in eclipse project explorer.

3) Create a new Package

a) Right click on Project name '**OnlineStore**' and select **New** > **Package**.



b) Give your Package name '**automationFramework**' and click on **Finish** button.

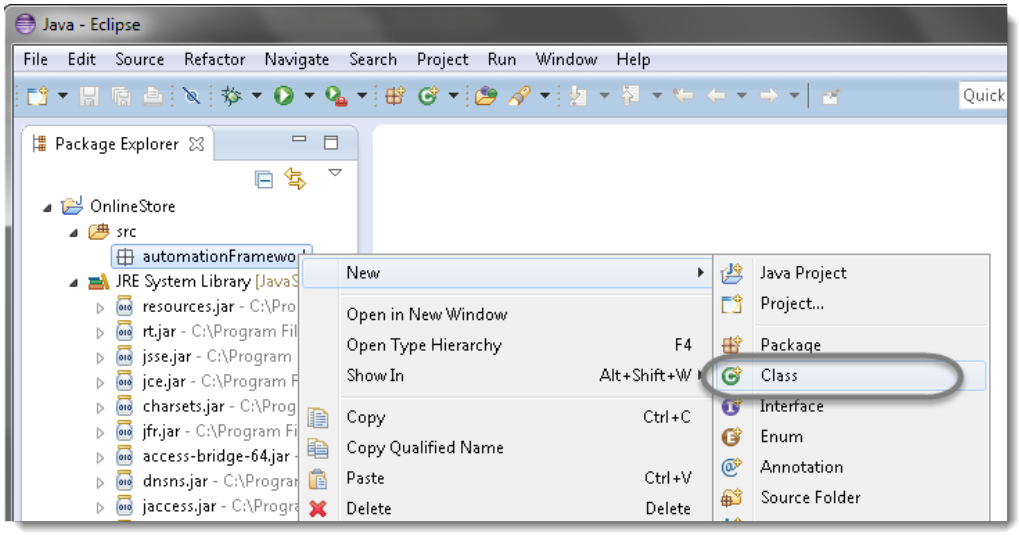


Now you can see a new package with name 'automationFramework' under project name 'OnlineStore'.

4) Create a new Class

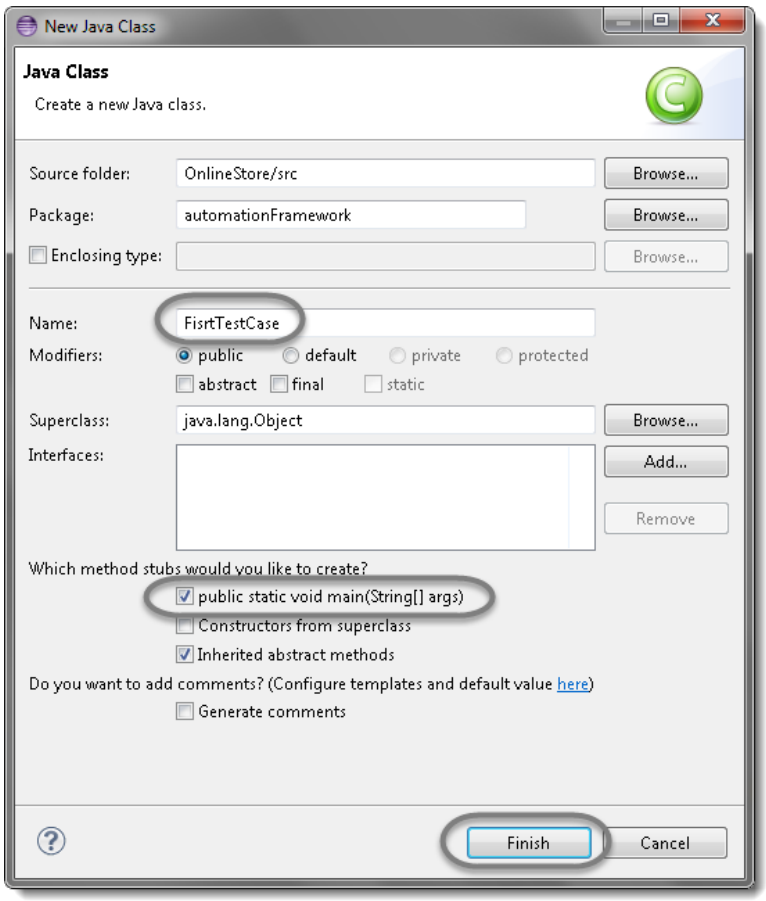
Now that you have a project set up, you're going to want start writing some new classes.

a) Right click on Package '**automationFramework**' and select **New** > **Class.**

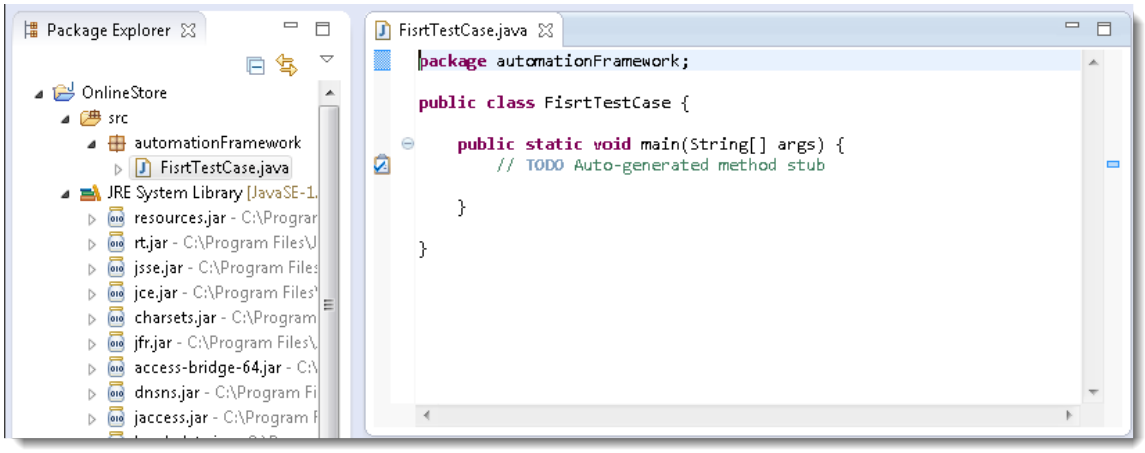


b) Give your Class name '**FirstTestCase**', check the option '**public static void main**' and click on **Finish** button. This will bring up totally a sweet class creation window.

Note: In case of not creating class for Main test case, please do not click 'public static void main'. We need to select it only in case of writing test cases which we are going to execute and from where we call other classes. For functional classes, POM classes or any other classes we dont need this to be checked.



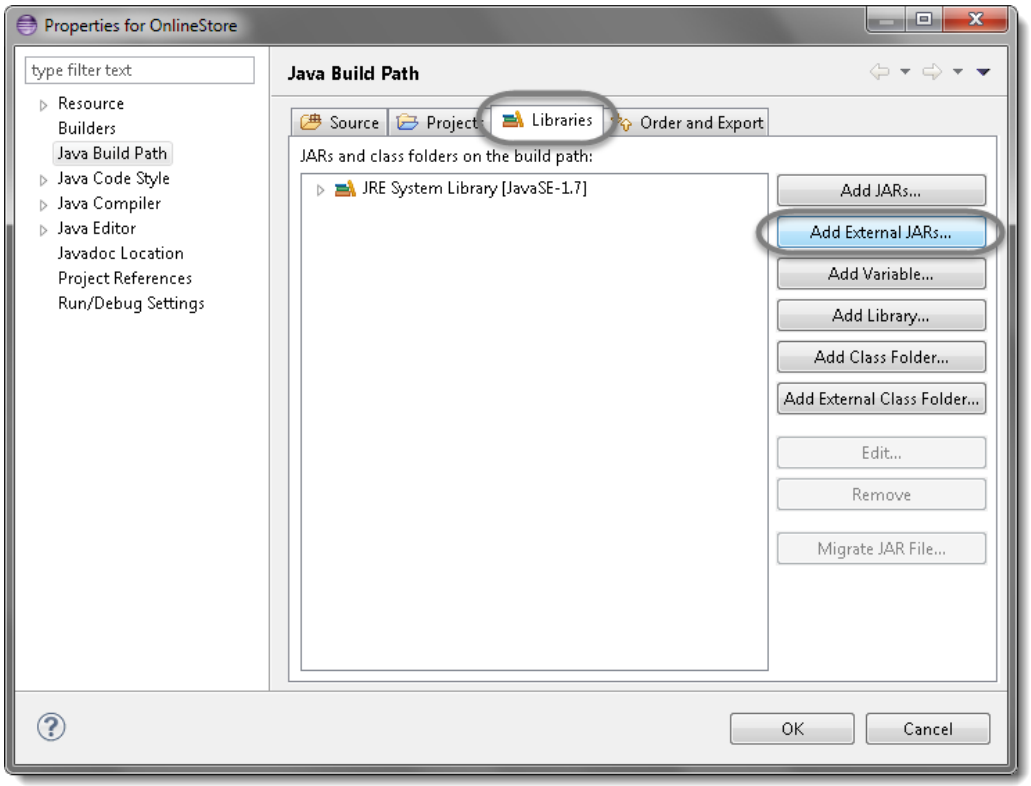
c) Now your Eclipse window will looks like bellow.



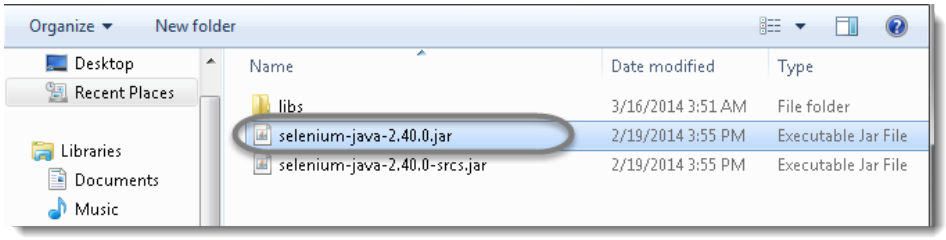
5) Add External Jars to Java build path

Now you need to add selenium webdriver's jar files in to Java build path.

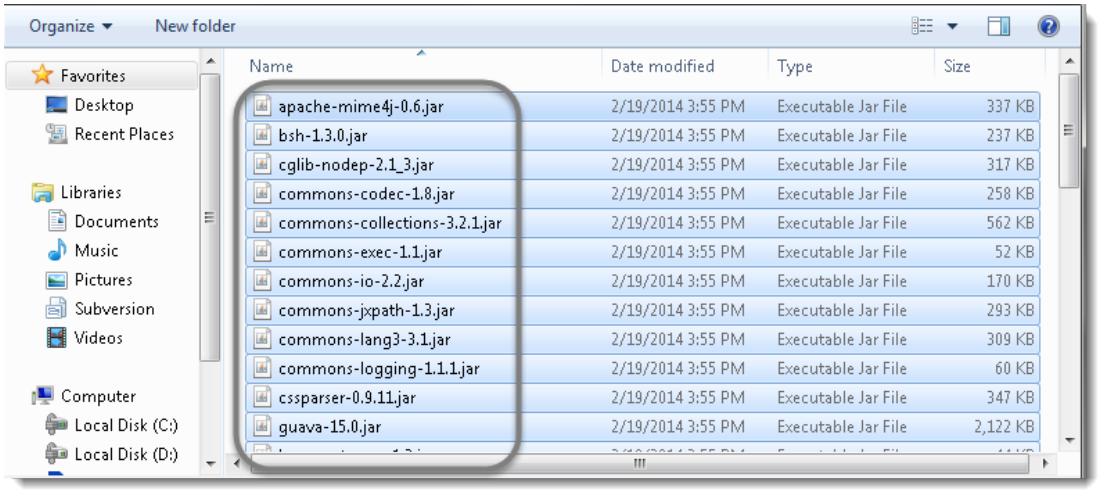
a) Right click on Project '**OnlineStore**' > **Select Properties** > **Java build path.** Then navigate to **Libraries** tab and click **Add External JARs**.



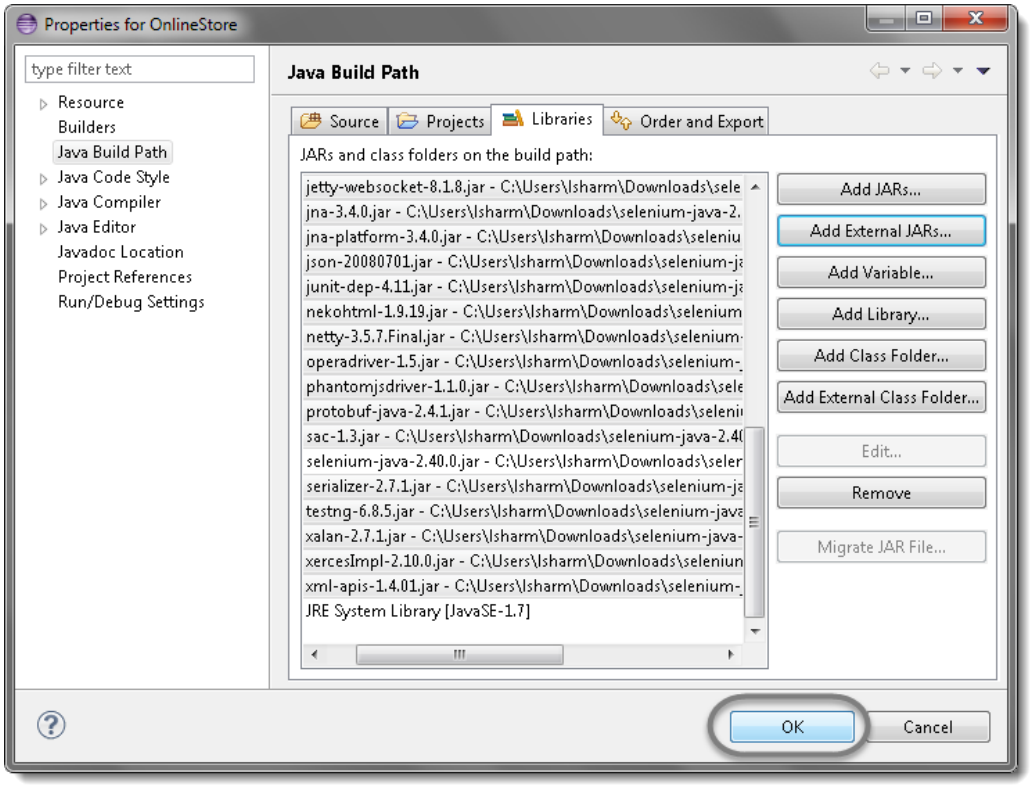
b) Add Selenium Java jar, you may add the source file too.



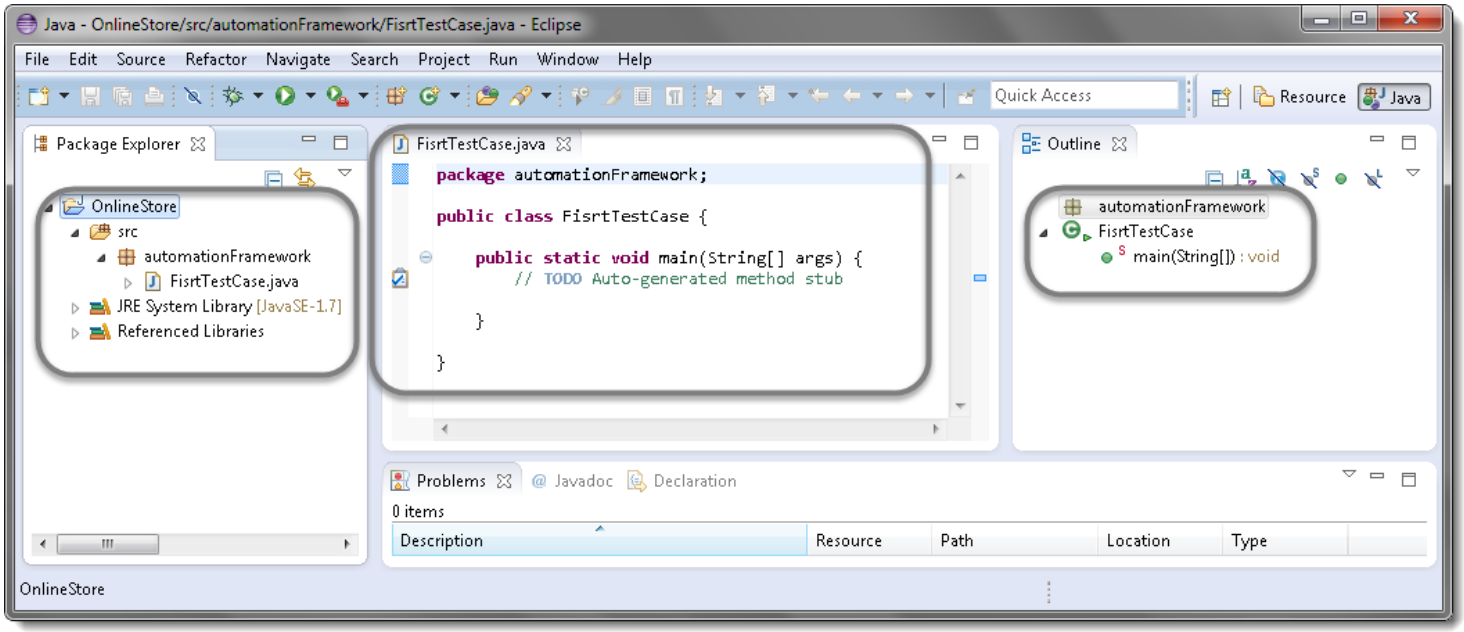
c) Add all the jars from the **libs** folder as well.



d) Click **OK**.



That's all about configuration of WebDriver with eclipse. Now you are ready to write your test script in eclipse and run it in WebDriver.


Java - OnlineStore/src/automationFramework/FisrtTestCase.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Resource
Java
Package Explorer
OnlineStore
src
automationFramework
FisrtTestCase.java
JRE System Library [JavaSE-1.7]
Referenced Libraries
FisrtTestCase.java
package automationFramework;
public class FisrtTestCase {
public static void main(String[] args) {
// TODO Auto-generated method stub
}
}
Outline
automationFramework
FisrtTestCase
main(String[]) : void
Problems
Javadoc
Declaration
0 items
Description
Resource
Path
Location
Type
OnlineStore