**GUIDELINE SET UP WINDOWS ENVIRONEMENT FOR JAVA SELENIUM AUTOMATION TEST WITH ECLIPSE**

**Step #1 – Install Java**

A Java development kit which includes the JRE (Java Runtime Environment) is required to write and run Java programs. JRE is a child of JDK, which comes along with JDK installation. Even for running applications that need the dependency of Java, one needs to install JDK. One such application is Eclipse IDE. [Download Java](https://www.oracle.com/technetwork/java/javase/downloads/index.html), install it and set the environment path. Once the path is set, one can verify the installation by typing java -version on the command prompt, which provides the details of the installed java version. Refer to the following image for better understanding.

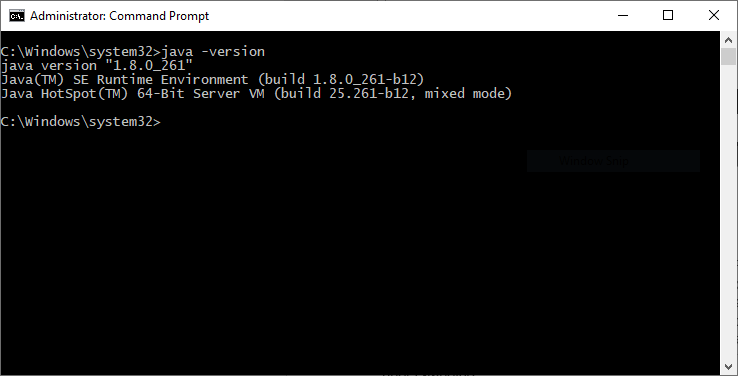


Figure 1. Check Java version to ensure Java was installed successfully

### ****Step #2 – Install Eclipse****

Eclipse is a platform for Java developers to write their code and run it. [Download Eclipse](https://www.eclipse.org/downloads/) from their official page. Based on the OS, one can go with the required option. Once downloaded, do the installation. Once completed, one can see the eclipse.exe in the eclipse folder:

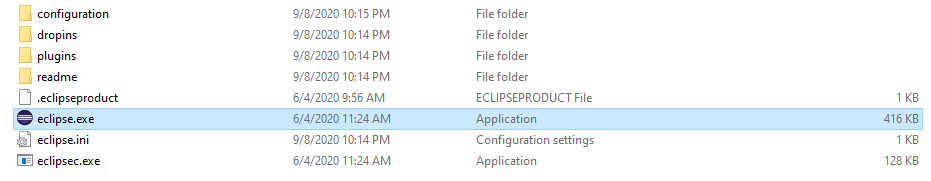


Figure 2. Java installation folder

### ****Step #3 – Selenium Client and WebDriver Language Bindings****

[Selenium Webdriver](https://www.browserstack.com/guide/selenium-webdriver-tutorial) supports multiple languages, and each language has its client driver. As we are using Selenium with Java, we need to have ‘Selenium Java Client Driver’. One can download the client driver from the [official Selenium website](https://docs.seleniumhq.org/download/). One can see the multiple language client drivers provided by Selenium:

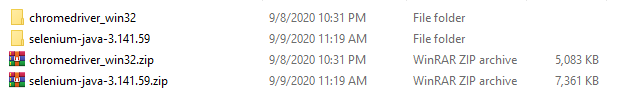


Figure 3 Downloaded and extracted these components



Figure 4. This application is used to control Google Chrome

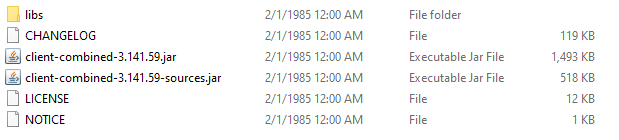


Figure 5. Necessary external JAR file for automation

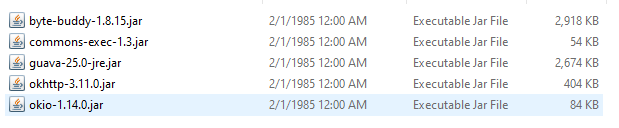


Figure 6. Necessary external JAR file for automation

### ****Step #4 – Configuring Selenium WebDriver with Eclipse****

This is a vital step of starting with Selenium. To configure Eclipse with Selenium Webdriver client,

1. Double click on the eclipse.exe file to launch it
2. Create a workspace

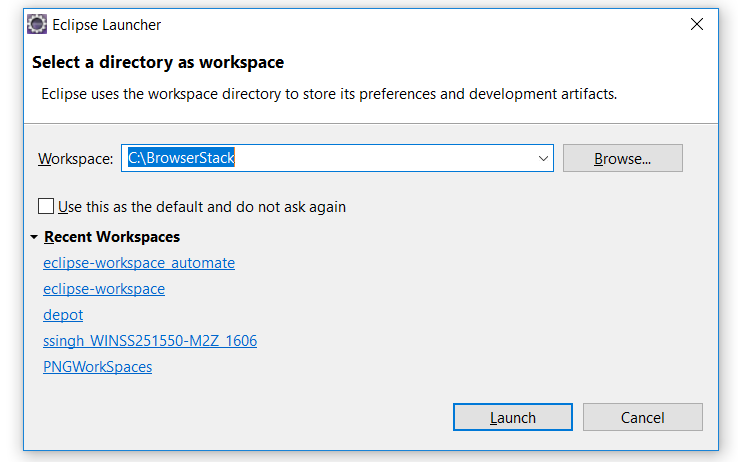


Figure 7. Create Java project in Eclipse

Think of it just like any other folder, which stores all the scripts in one place. One can choose to create as many workspaces as required. Click on Launch to launch the workspace.

Create a new java project by clicking on File-> New-> Java Project and name the project

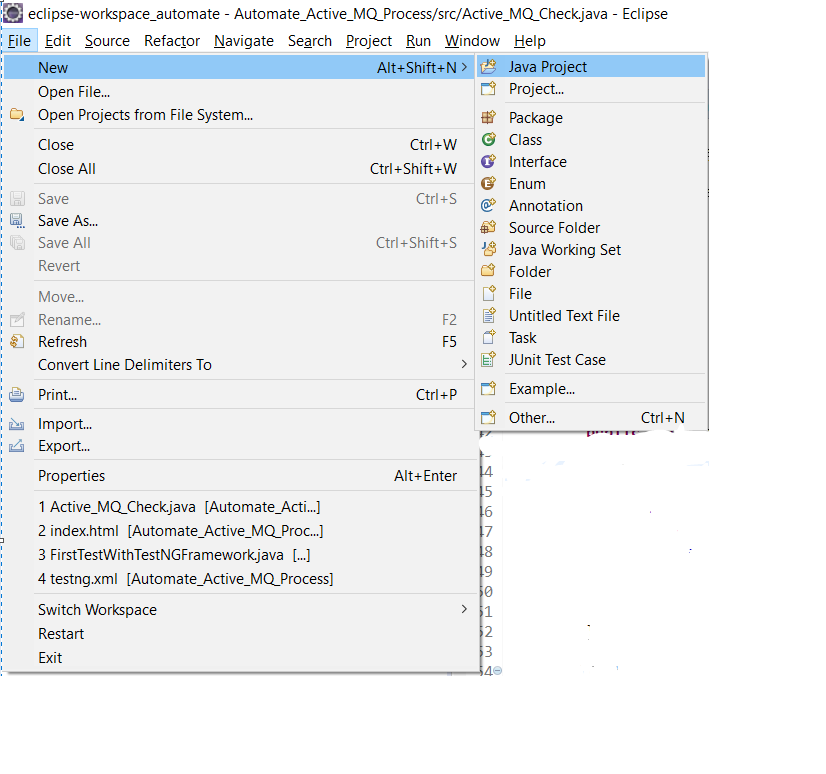


Figure 8. Menu to create new Java project in Eclipse

Create a package under this project, by right-clicking on the ‘src’ folder

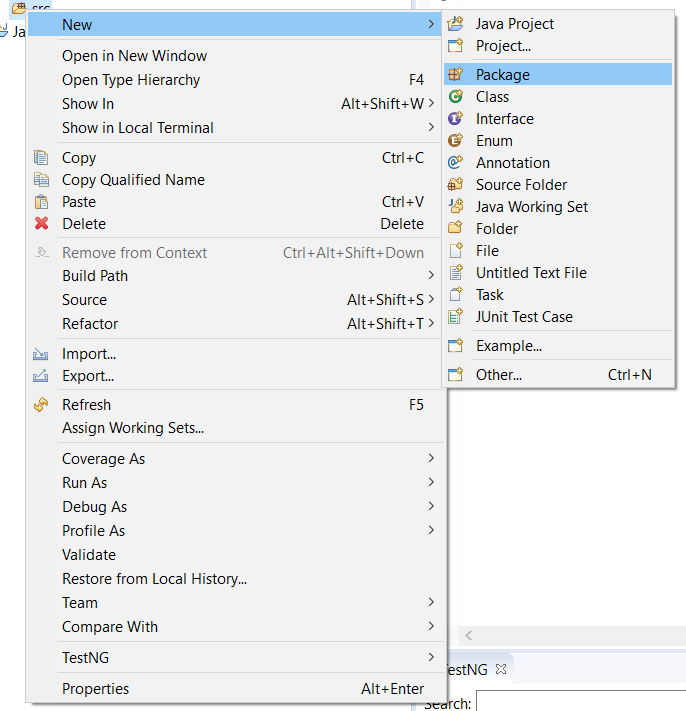


Figure 9. Add new package

Once the package is created, right-click on the package and create a class

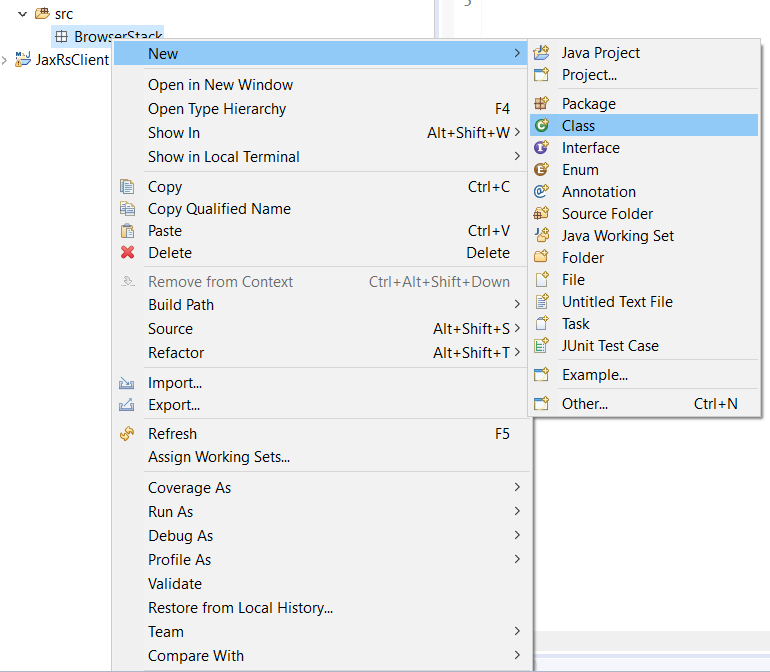


Figure 10. Add a class into source code

Once the class is created, go ahead with adding the Selenium Jars to the project.

To add the Selenium Jars, right-click on the project folder and go to Properties:

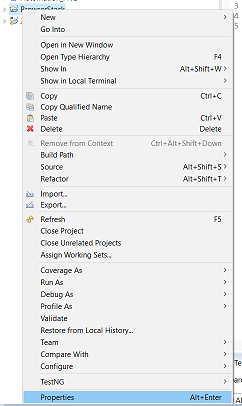


Figure 11. Choose Properties

From the Properties window, navigate to ‘Java Build Path’ and click on ‘Add External JAR’s’

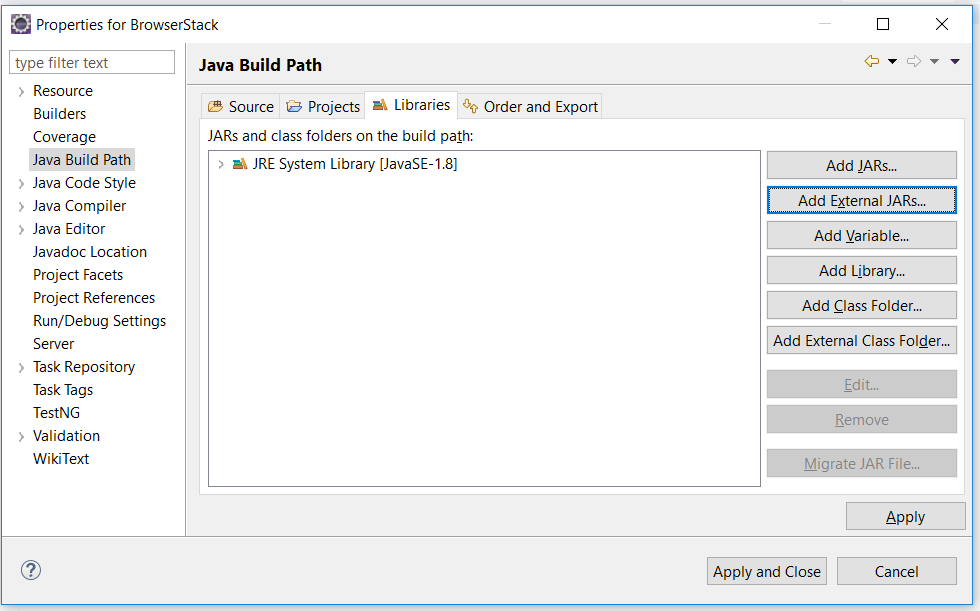


Figure 12. Add external JAR

Add the downloaded Selenium Jars and click on ‘Apply and Close.’ Selenium with Eclipse is configured now. Now Eclipse is ready to execute the first script.

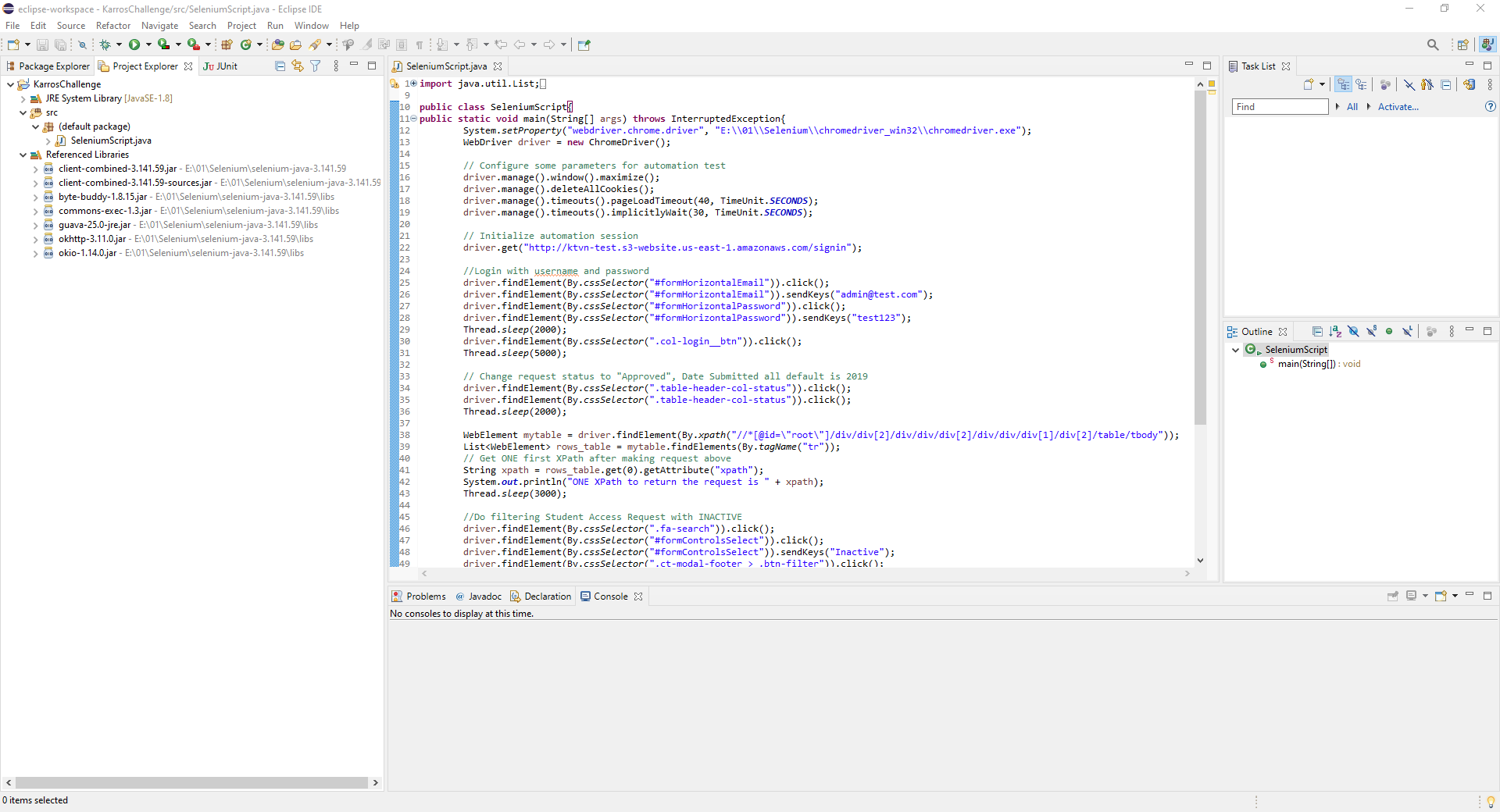


Figure . Sample final configuration for Eclipse Java project

### ****Step #5 – Creating and Running the first test using Selenium and Java****

As the first test, we will write a script to open ‘google.com’ on the Chrome browser. To use Chrome, it is mandatory to have the driver executable. To download the driver executable, visit the [Selenium website](https://www.seleniumhq.org/download/). In the third-party driver browser section, one can download the executable file for a specific browser.

Post downloading, below is the code snippet to run the first test:

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class FirstTestInSelenium {

public static void main(String[] args) {

// TODO Auto-generated method stub

//setting the driver executable

System.setProperty("webdriver.chrome.driver", ".\\Driver\\chromedriver.exe");

//Initiating your chromedriver

WebDriver driver=new ChromeDriver();

//Applied wait time

driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

//maximize window

driver.manage().window().maximize();

//open browser with desried URL

driver.get("https://www.google.com");

//closing the browser

driver.close();

}

}

In the code snippet above, we have used the Selenium keyword driver.get(“URL to open in browser”) to open URL in the desired browser. Other keywords like driver.close help to close the browser window, as a cleanup part.

So, this was a quick start guide to start using Selenium with Java. One should keep in mind the best practices to be incorporated while writing Selenium tests.