

Configure Selenium-Java Webdriver in Eclipse

Welcome to the Selenium-Java series. This is the first tutorial in this series. In this tutorial we are going to study about selenium-webdriver overview plus how to download and configure selenium-java jars in eclipse from scratch!!

What you will Learn:

Selenium Webdriver overview
Selenium Webdriver Architecture
Java API document
Limitations of WebDriver
Create Java project in eclipse
First Java class
Add Selenium java jars in eclipse

Selenium Webdriver overview:

Selenium is a browser automation tool which interacts with browser and automate end to end tests of a web application. It is a suite of tools: Selenium IDE (for record and playback), Selenium grid (to execute tests in parallel in different browsers/machines) and Selenium webdriver. We will be using selenium webdriver for automation.

1. Selenium WebDriver is a browser automation framework that accepts commands and sends them to a browser. It is implemented through a browser-specific (chrome, Mozilla, IE) driver. It controls the browser by directly communicating with it. Selenium WebDriver supports Java, C#, PHP, Python, Perl, Ruby.
2. Operation System Support – Windows, Mac OS, Linux, Solaris
3. Browser Support – Mozilla Firefox, Internet Explorer, Google Chrome, Safari, Opera, Android, iOS, HtmlUnit

Selenium Webdriver Architecture:

See figure 1. There are 4 components of Selenium architecture:

- 1) **Selenium client library/Language bindings**:- Selenium Developers have developed language bindings to allow Selenium to support multiple languages libraries such as Java, Ruby, Python, etc.
- 2) **JSON Wire Protocol over HTTP**:- JSON stands for JavaScript Object Notation. It is used to transfer data between HTTP server and a client on the web. Each BrowserDriver (such as FirefoxDriver, ChromeDriver etc.,) has its own HTTP server.
- 3) **Browser Drivers**:- Each browser has its own separate browser driver. Browser drivers communicate with respective browser. When a browser driver receives any command then that command will be executed on the respective browser and the response will go back in the form of HTTP response.
- 4) **Real Browsers**:- Selenium supports multiple browsers such as Firefox, Chrome, IE, Safari etc.

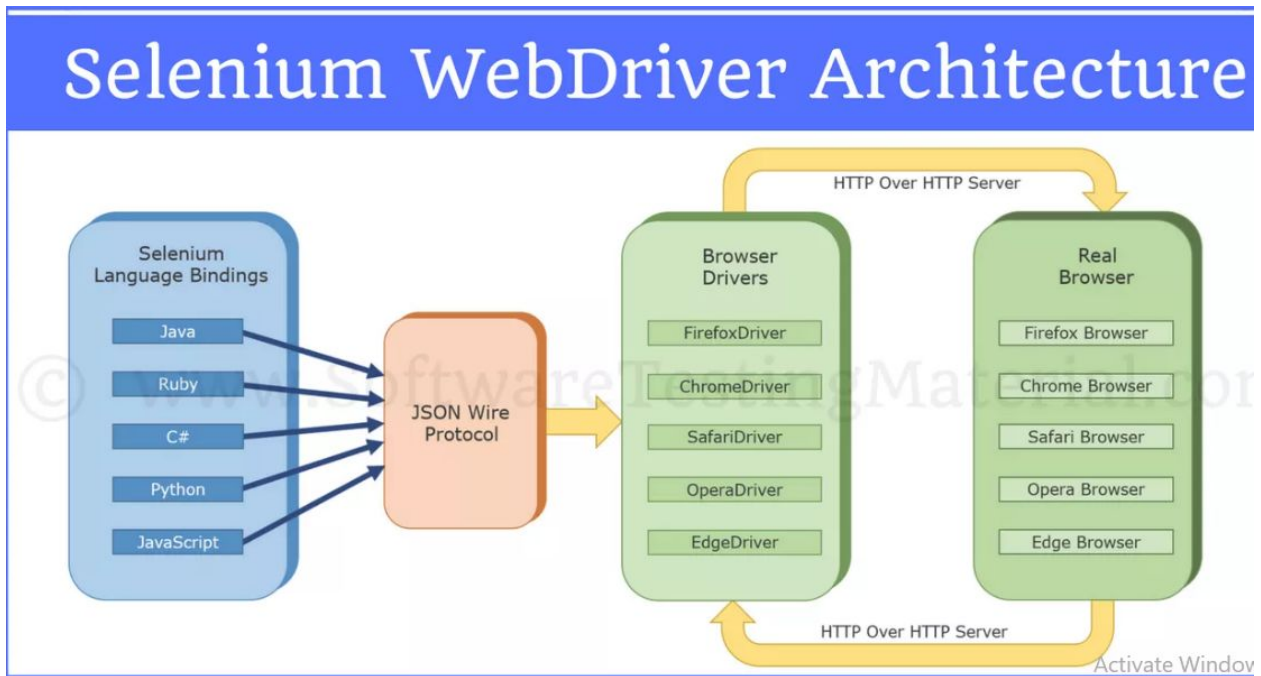


Figure 1

Java API document:

Go to the website <https://www.seleniumhq.org/download/> & click the API docs seen against Java

selenium.dev/downloads/

Selenium Client & WebDriver Language Bindings

In order to create scripts that interact with the Selenium Server (Remote WebDriver) or create local Selenium WebDriver scripts, you need to make use of language-specific client drivers.

While language bindings for **other languages exist**, these are the core ones that are supported by the main project hosted on GitHub.

LANGUAGE	VERSION	RELEASE DATE	LINKS
Ruby	3.142.6	October 04, 2019	Download Changelog API Docs
JavaScript	4.0.0-alpha.5	September 08, 2019	Download Changelog API Docs
Java	3.141.59	November 14, 2018	Download Changelog API Docs
Python	3.141.0	November 01, 2018	Download Changelog API Docs
C#	3.14.0	August 02, 2018	Download Changelog API Docs

Figure 2

The below page comes up. On the left hand panel, search for *WebDriver*. Since WebDriver is an interface, it is seen in italics. As can be seen on the right hand panel, all the web browsers (example, chrome driver etc) implement this interface. We will study more about this interface in further tutorials.

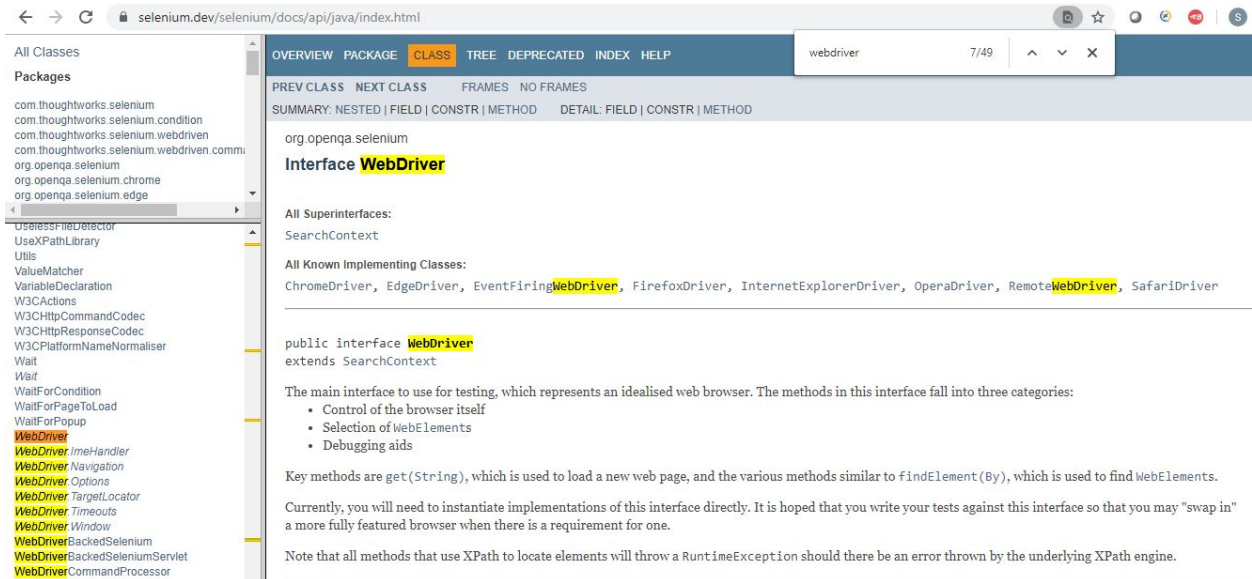


Figure 3

Limitations of WebDriver:

WebDriver Cannot Readily Support New Browsers. Remember that WebDriver operates on the OS level. Also, remember that different browsers communicate with the OS in different ways. If a new browser comes out, it may have a different process of communicating with the OS as compared to other browsers. So, you have to give the WebDriver team quite some time to figure that new process out before they can implement it on the next WebDriver release. However, it is up to the WebDriver's team of developers to decide if they should support the new browser or not.

Create Java project in eclipse:

File >New >Java project

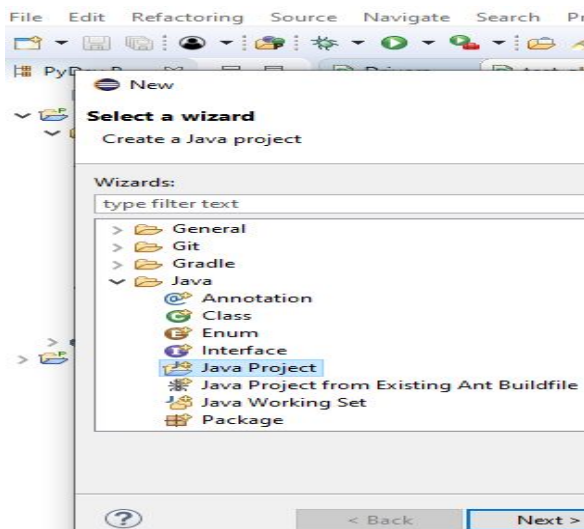


Figure 4

Enter java project name

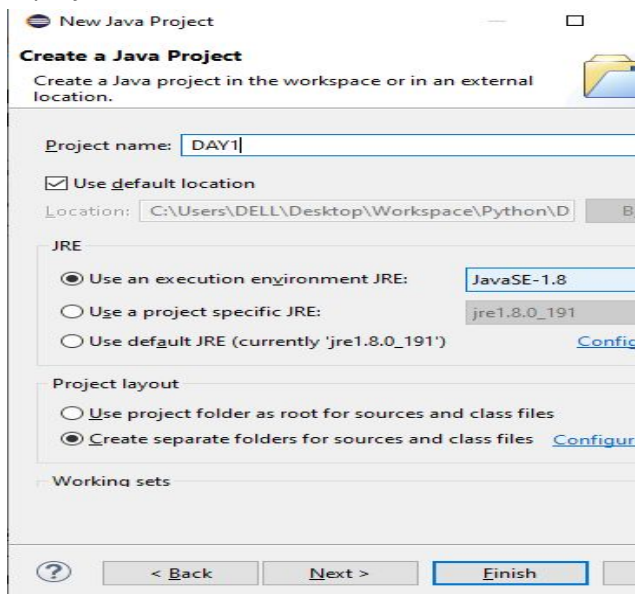


Figure 5

Next, you see that, DAY1 project would be created that will contain src folder

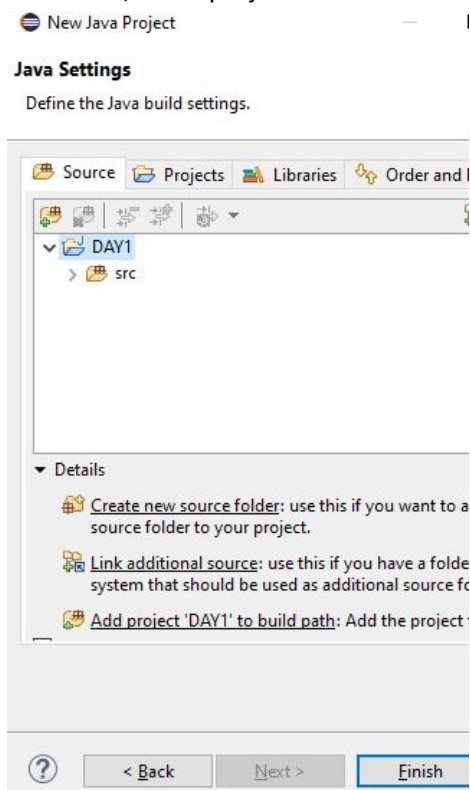


Figure 6

Finish. Now, right click on DAY1 project

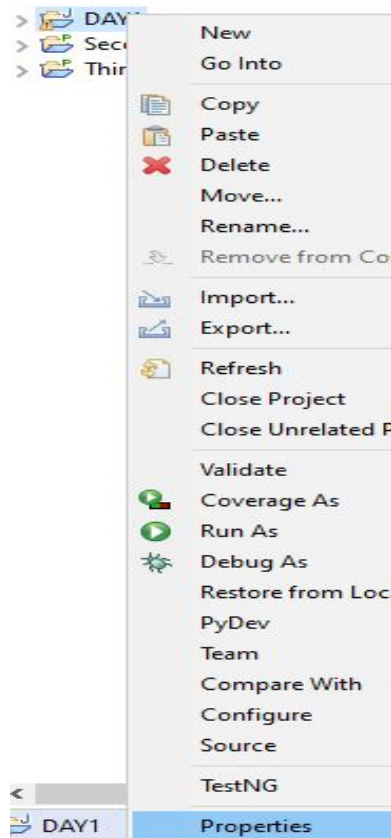


Figure 7

Click 'Properties' > click 'Java Build Path' > click 'Libraries' tab. You will see that java libraries are already added, see below

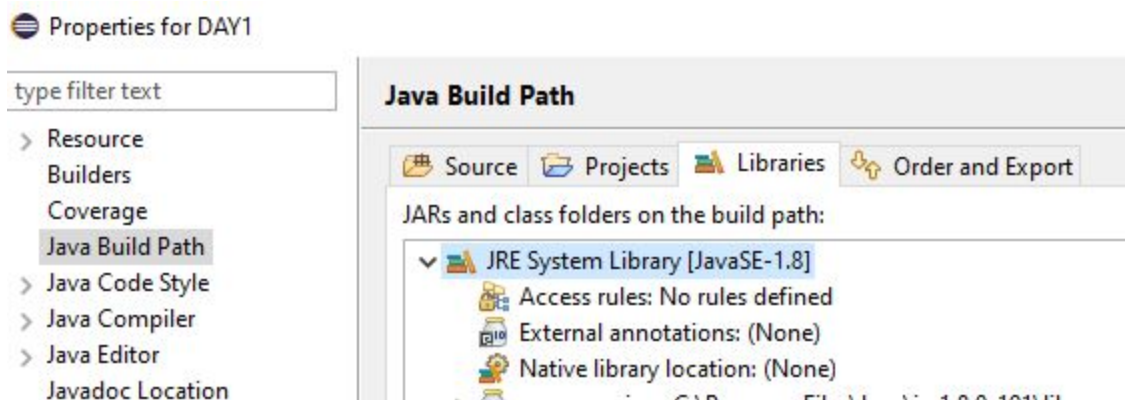


Figure 8

First Java class:

Create first java class having the main method. To do that, expand DAY1 folder & right click 'src' folder > click 'New' > click 'Other', see below



Figure 9

The below window comes up. Expand 'Java' & select 'Class', see below

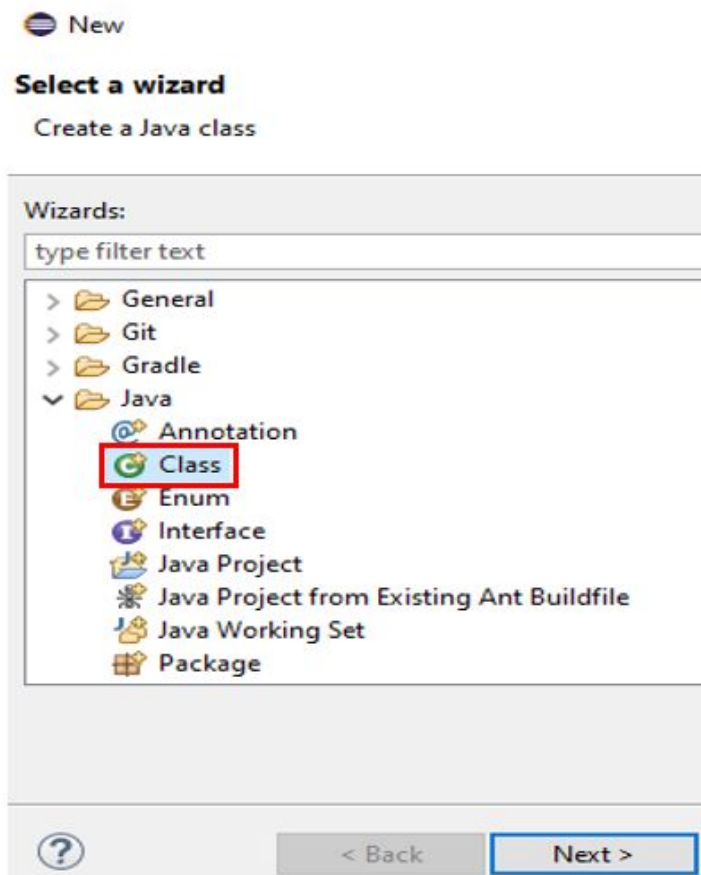


Figure 10

Click 'Next', below window comes up. Enter class name 'Demo' & select the checkbox 'public static void main(String[] args)'

Java Class

Create a new Java class.

Source folder: DAY1/src

Package: test

☐ Enclosing type:

Name: Demo

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

☒ public static void main(String[] args)
☐ Constructors from superclass
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

< Back Next > Finish

Figure 11

Click Finish. Write the print statement `System.out.println("Welcome!");` as seen below

```
1
2 public class Demo {
3
4     public static void main(String[] args) {
5         System.out.println('Welcome!');
6     }
7 }
8
9
10
```

Figure 12

Right click on the program > Run As > Java Application, see below

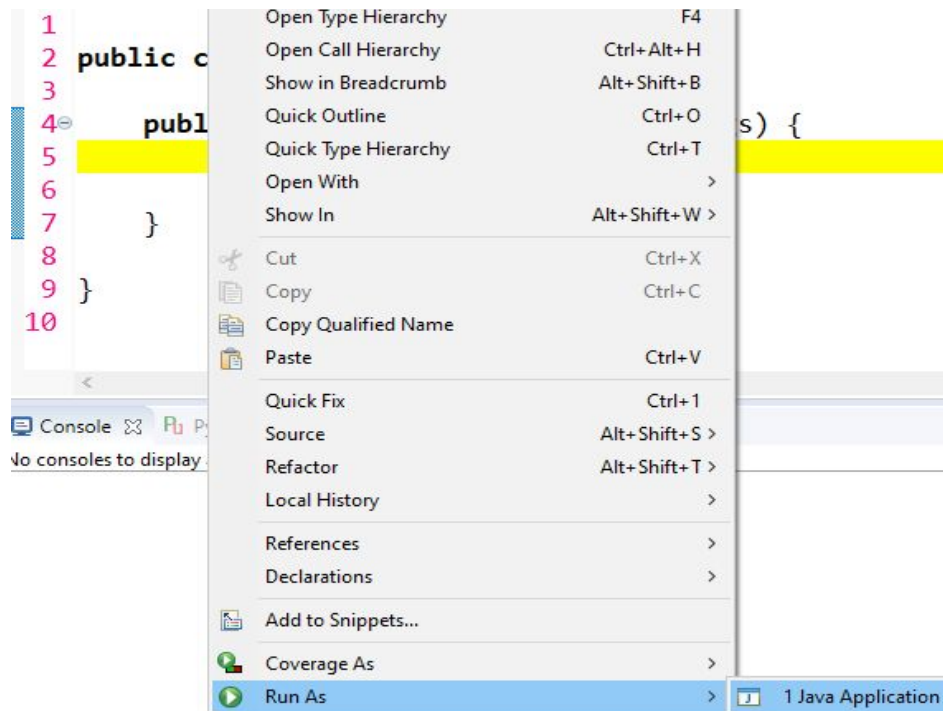


Figure 13

You would see 'Welcome!' gets printed in console.

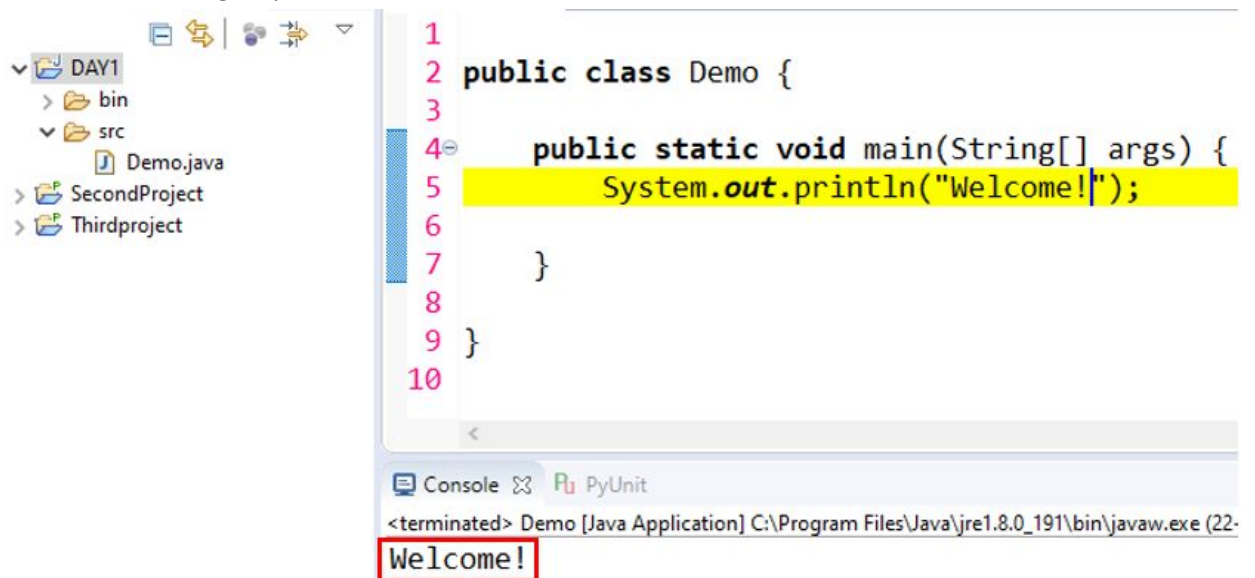


Figure 14

Add Selenium Java jars in eclipse:

Now this java project DAY1 that we have created does NOT have any knowledge about selenium. So let us add that knowledge by adding selenium jars. Selenium is not a tool. It's a jar file package. Go to official website <https://selenium.dev/> > Downloads > you will see the current version of selenium-java jars is 3.141.59

selenium.dev/downloads/			
LANGUAGE	VERSION	RELEASE DATE	LINKS
Ruby	3.142.6	October 04, 2019	Download Change
JavaScript	4.0.0-alpha.5	September 08, 2019	Download Change
Java	3.141.59	November 14, 2018	Download Change

Figure 15

Click 'Download' link > unzip jars

> TRAINING > Software >				
Name	Date modified	Type	Size	
selenium-java-3.141.59	22-11-2019 21:09	File folder		
selenium-java-3.141.59	22-11-2019 21:08	WinRAR ZIP archive	7,361 KB	

Figure 16

Go inside selenium-java-3.141.59 folder. You will see few jars outside 'libs' folder

> TRAINING > Software > selenium-java-3.141.59 >			
Name	Date modified	Type	
libs	01-02-1985 00:00	File folder	
CHANGELOG	01-02-1985 00:00	File	
client-combined-3.141.59	01-02-1985 00:00	Executable Jar File	
client-combined-3.141.59-sources	01-02-1985 00:00	Executable Jar File	

Figure 17

Go inside 'libs' folder. You will see few jars inside 'libs' folder

> TRAINING > Software > selenium-java-3.141.59 > libs			
Name	Date modified	Type	
byte-buddy-1.8.15	01-02-1985 00:00	Executable Jar	
commons-exec-1.3	01-02-1985 00:00	Executable Jar	
guava-25.0-jre	01-02-1985 00:00	Executable Jar	
okhttp-3.11.0	01-02-1985 00:00	Executable Jar	
okio-1.14.0	01-02-1985 00:00	Executable Jar	

Figure 18

Go to java build path (Right click on DAY1 project > Click 'Properties' > click 'Java Build Path' > click 'Libraries' tab)

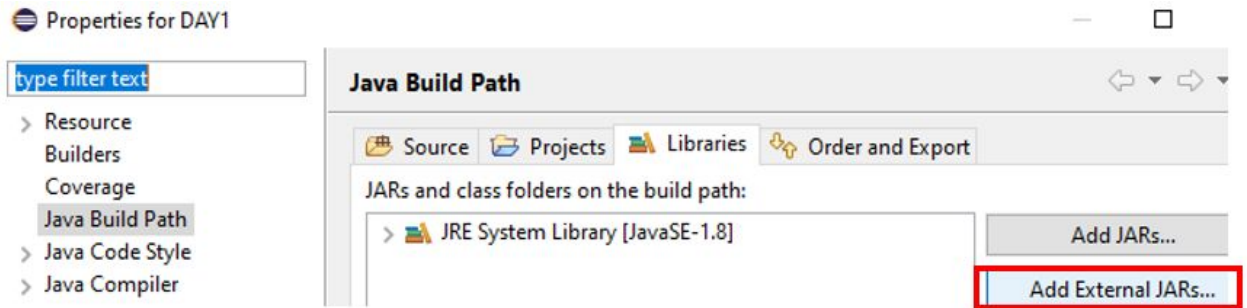


Figure 19

Click 'Add External jars', the below window comes up. Browse to the location in your machine where you have downloaded the selenium jars, see below. Select all the jars outside 'libs' folder

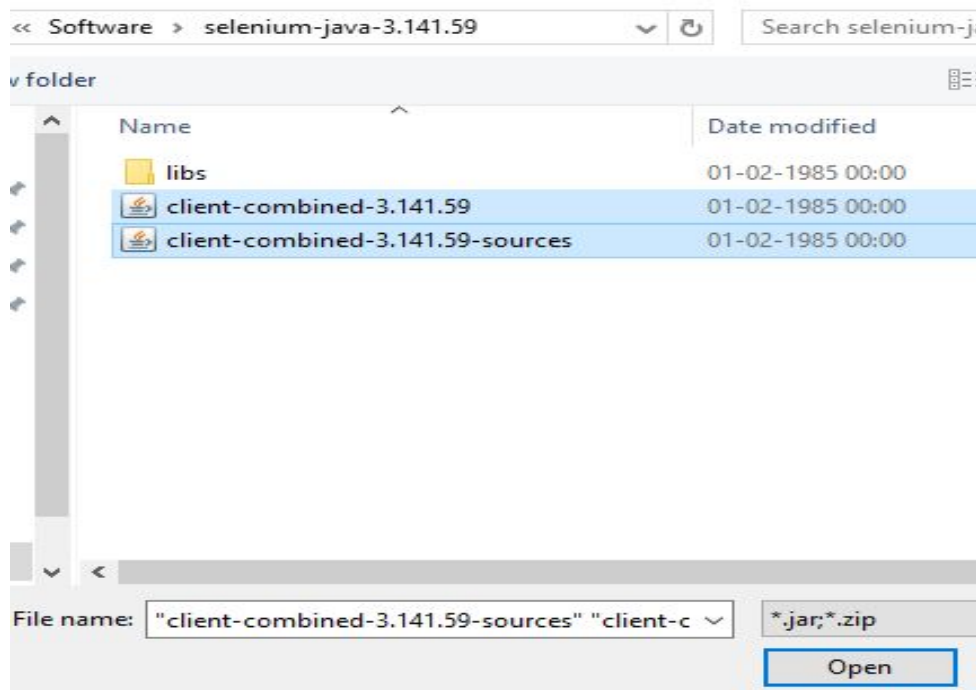


Figure 20

Click 'Open'. The jars should be seen as added in the 'Java Build Path' like seen below



Figure 21

Click 'Add External jars'. Select all the jars inside 'libs' folder like seen below

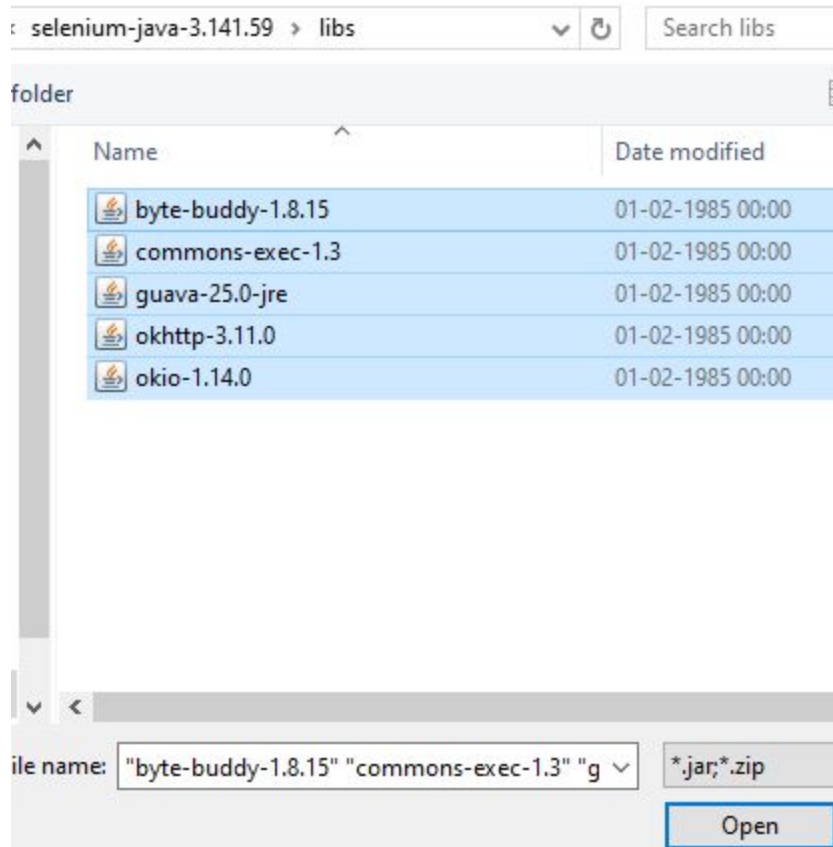


Figure 22

Click 'Open'. So we should see all the jars added in the 'Java Build Path' like seen below (7 jars in this case)

