

SELENIUM INSTALLATION AND SETUP HANDBOOK

Saranya Tiruveedhula

<https://github.com/sarupapu7/Selenium-HandBook>

Selenium Installation and Setup Handbook

Step 1: Install Java JDK and set environment variables

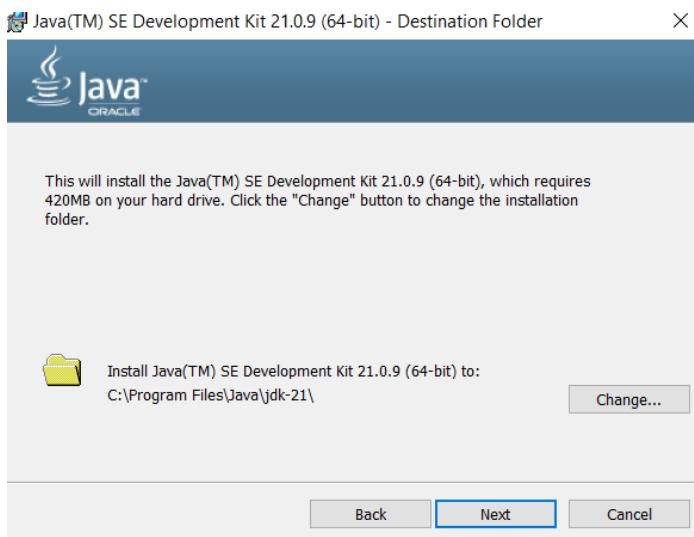
- Go to Oracle official site: <https://www.oracle.com/java/technologies/downloads/>
- Choose JDK 21 and download the installer for your OS

Ex: for Windows, refer below screenshot

The screenshot shows the Oracle Java Downloads page. Under the "Java SE Development Kit 21.0.9 downloads" section, there are three download links:

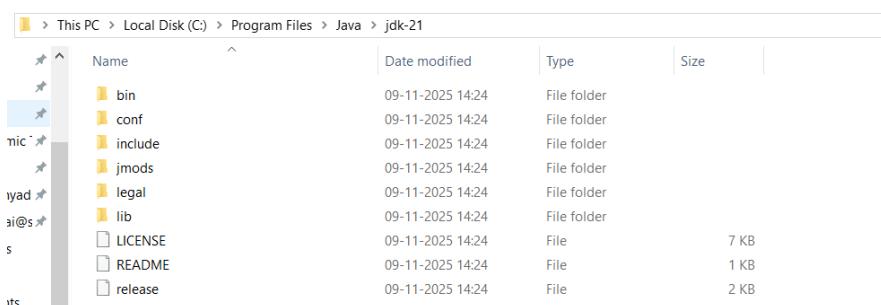
Product/file description	File size	Download
x64 Compressed Archive	186.13 MB	https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.zip (sha256)
x64 Installer	164.45 MB	https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.exe (sha256)
x64 MSI Installer	163.20 MB	https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.msi (sha256)

- Run the installer and follow prompts.

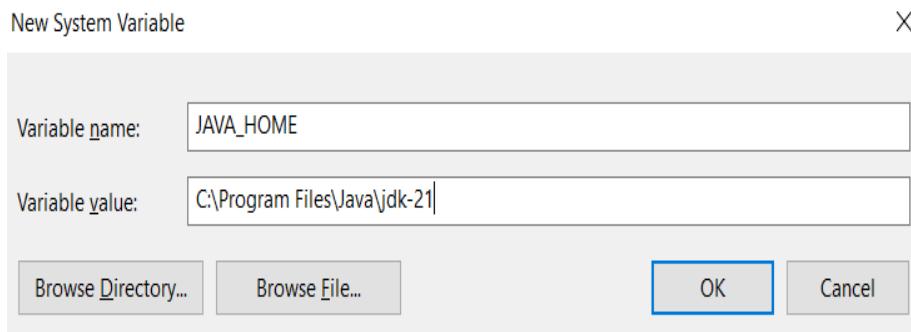
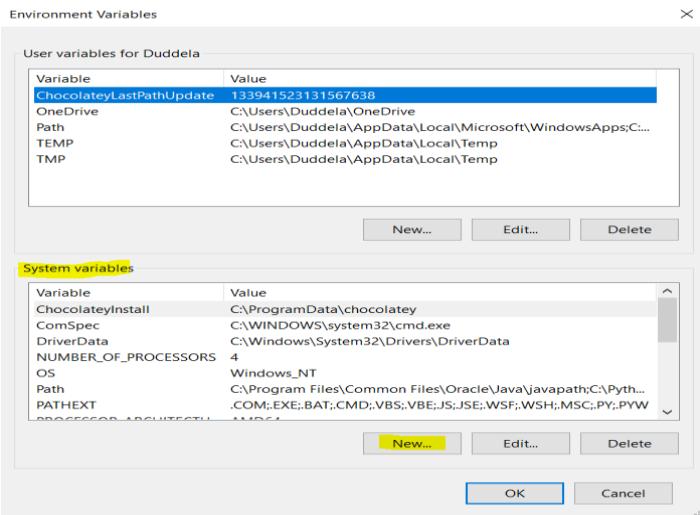


Note: If the installer asks to set JAVA_HOME/add to PATH, choose yes if available. If not, finish install and set manually (steps below).

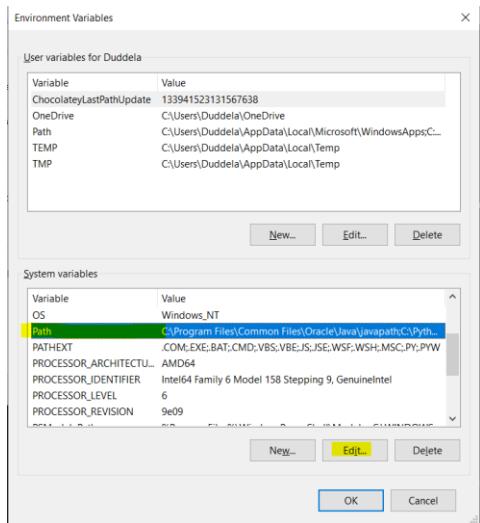
- Use the full path to the JDK installation directory (the folder that contains bin, lib).



- Right-click **This PC** → **Properties** → **Advanced system settings** → **Environment Variables**.
- Under *System variables*:
 - Click **New** → Name: **JAVA_HOME** → Value: **C:\Program Files\Java\jdk-21** (your actual path).
 - Find **Path** → **Edit** → **New** → **%JAVA_HOME%\bin**



- Click **Ok**. It will be added.
- In the same **Environment Variables** window, scroll down in the **System variables** section, select the variable **Path** and Click **Edit**



- Then, click **New** and add this line: %JAVA_HOME%\bin
- Click **OK** → **OK** → **OK** on all windows to close.

✓ This tells Windows to look inside your JDK's bin folder for Java commands.

- Now, verify your configuration by opening a **new Command Prompt** (important! old ones won't pick up new variables) and run below commands one by one.

```
cmd Command Prompt
Microsoft Windows [Version 10.0.19045.6456]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DDudela>echo %JAVA_HOME% ←
C:\Program Files\Java\jdk-21

C:\Users\DDudela>java -version ←
java version "21.0.9" 2025-10-21 LTS
Java(TM) SE Runtime Environment (build 21.0.9+7-LTS-338)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.9+7-LTS-338, mixed mode, sharing)

C:\Users\DDudela>javac -version ←
javac 21.0.9

C:\Users\DDudela>
```

Java is fully configured 😊

Step 2: Install and configure Maven

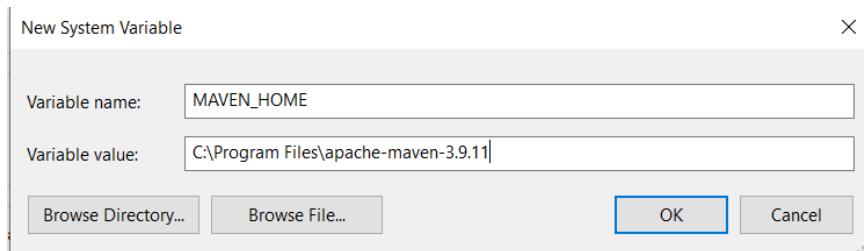
- Download Maven ZIP: <https://maven.apache.org/download.cgi>

The screenshot shows the Apache Maven download page. The left sidebar has a 'Downloads' section selected. Under 'Downloads', there are links for Java Development Kit (JDK), Memory, Disk, and Operating System requirements. Below these, the 'Files' section lists download links for different Maven distributions:

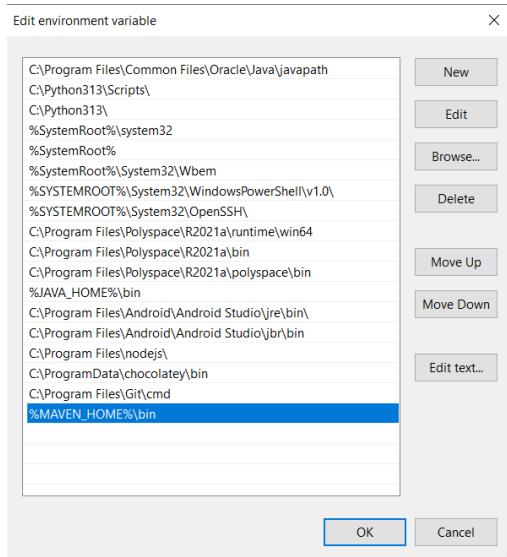
	Link	Checksums	Signature
Binary tar.gz archive	apache-maven-3.9.11-bin.tar.gz	apache-maven-3.9.11-bin.tar.gz.sha512	apache-maven-3.9.11-bin.tar.gz.asc
Binary zip archive	apache-maven-3.9.11-bin.zip	apache-maven-3.9.11-bin.zip.sha512	apache-maven-3.9.11-bin.zip.asc
Source tar.gz archive	apache-maven-3.9.11-src.tar.gz	apache-maven-3.9.11-src.tar.gz.sha512	apache-maven-3.9.11-src.tar.gz.asc
Source zip archive	apache-maven-3.9.11-src.zip	apache-maven-3.9.11-src.zip.sha512	apache-maven-3.9.11-src.zip.asc

At the bottom of the 'Files' section, there are two links: '3.9.11 Release Notes and Release Reference Documentation' and 'Latest source code from source repository'.

- Extract it to a location, e.g.: C:\Program Files\Apache\maven
- Set Environment variables. In **System Variables**, add a new variable:
 - **Variable name:** MAVEN_HOME
 - **Variable value:** C:\Program Files\Apache\maven



- Add this to the **Path** variable: %MAVEN_HOME%\bin



- Click OK → OK → OK
- Open a new Command Prompt and verify: "mvn -v"

```

Command Prompt
microsoft Windows [Version 10.0.19045.6456]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Duddela>mvn -v
Apache Maven 3.9.11 (3e54c93a704957b63ee3494413a2b544fd3d825b)
Maven home: C:\Program Files\apache-maven-3.9.11
Java version: 21.0.9, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-21
Default locale: en_IN, platform encoding: UTF-8
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

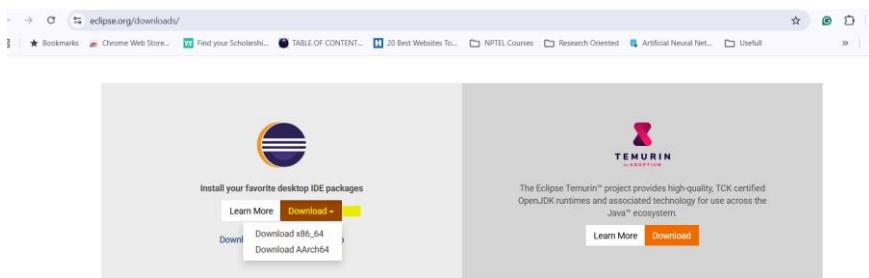
C:\Users\Duddela>

```

If you see maven version and java version, maven is installed successfully!

Step 3: Install Eclipse IDE for Java and point to JDK

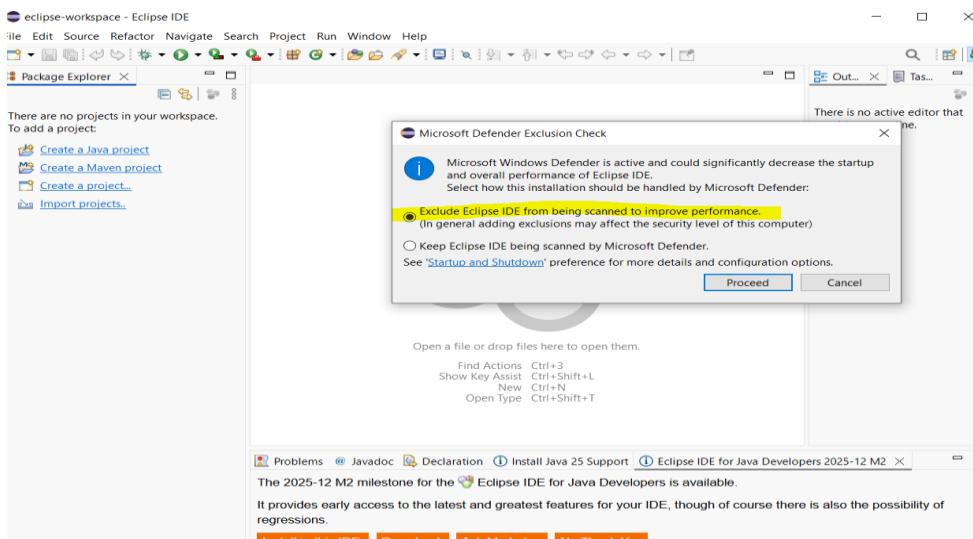
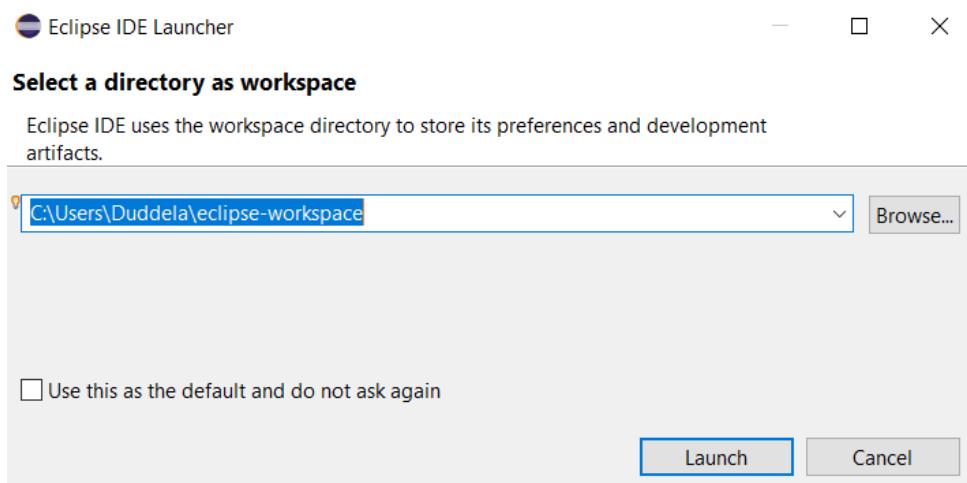
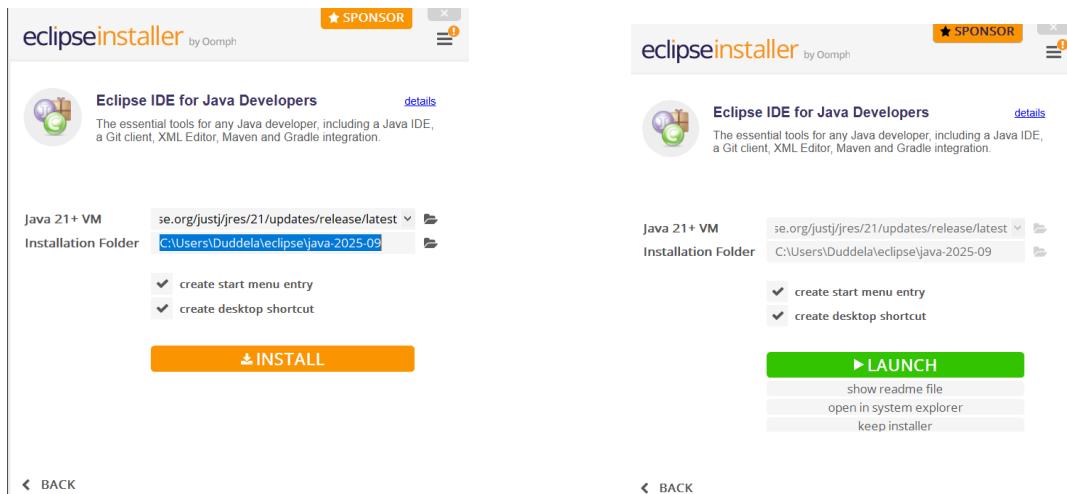
- Go to <https://www.eclipse.org/downloads/> and download



- Install and open it – Then, Choose Eclipse IDE for Java Developers



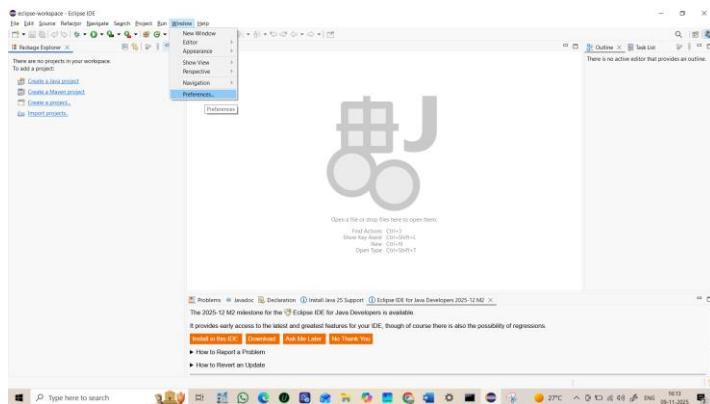
- Select the default location, click install and then launch.



➤ Verify JDK integration

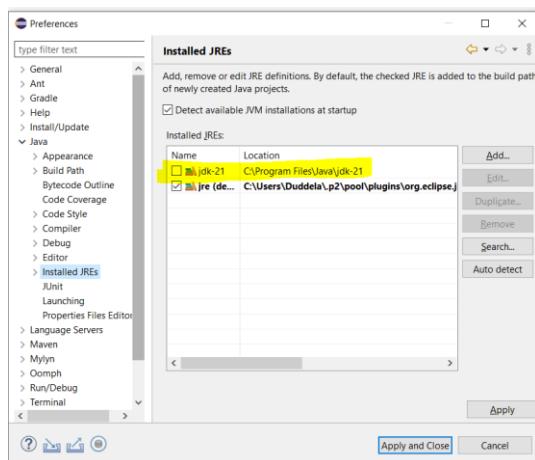
Once Eclipse opens:

1. Go to **Window → Preferences**



2. Expand Java → Installed JREs

3. Make sure you see:



4. If not:

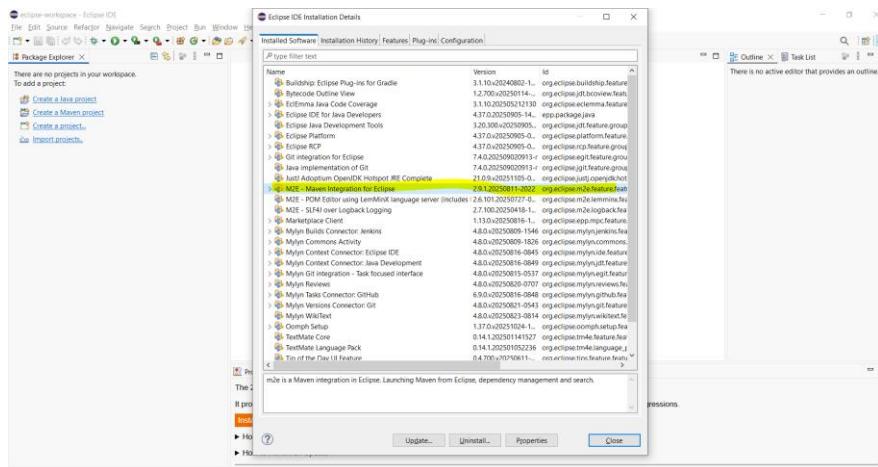
- Click Add → Standard VM
- Click Directory
- Select your JDK folder (C:\Program Files\Java\jdk-21)
- Click Finish → Apply and Close

Now Eclipse knows where your JDK is!

➤ Verify Maven integration

Eclipse for Java Developers already includes Maven by default. You can confirm:

1. Click Help → About Eclipse IDE
2. Click Installation Details → Installed Software
3. You should see “Maven Integration for Eclipse (m2e)”



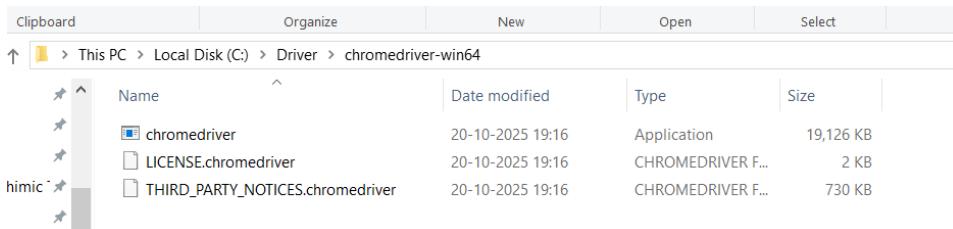
Step 4: Download ChromeDriver

- Selenium cannot open chrome browser, unless ChromeDriver is installed.
- Check your chrome version (Chrome://settings/help)
- Download matching ChromeDriver
- Extract and place it in C and add path in code.

Ex. For chrome browser of version **141.0.7390.123**

👉 <https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/141.0.7390.123/win64/chromedriver-win64.zip>

After extracting, the single file should look like below



Step 5: Download Selenium Java JARs

- Download link : <https://www.selenium.dev/downloads/>
- Click below link to download latest Selenium java zip.

The screenshot shows the Selenium official website at selenium.dev/downloads/. It features a navigation bar with links like About, Downloads, Documentation, Projects, Support, Blog, and English. Below the navigation is a search bar. The main content area displays language bindings for .NET, Ruby, and Java. Each section includes a logo, the word 'Stable', a date (e.g., 4.38.0 (October 25, 2025)), a Changelog link, and an API Docs link.

Selenium Clients and 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.



[.NET/C#](#)
Stable: 4.38.0 (October 25, 2025)

[Changelog](#)
[API Docs](#)

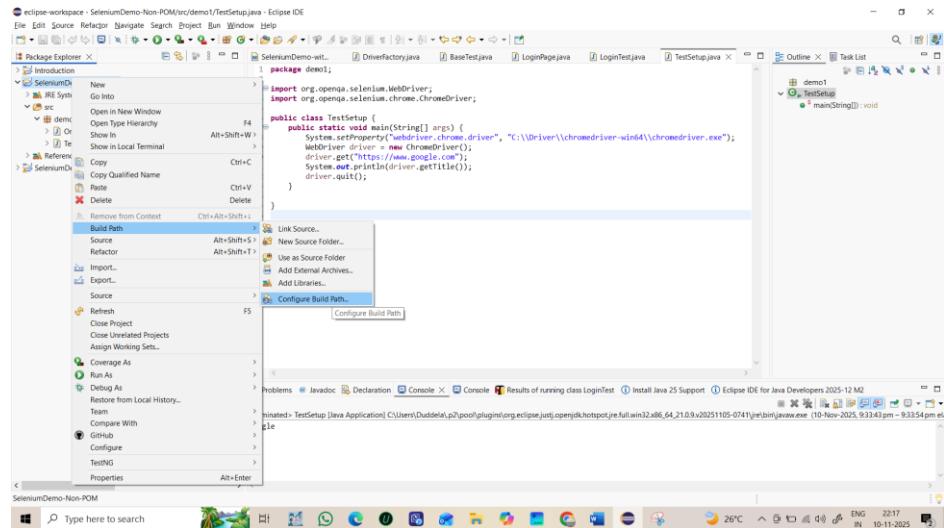


Stable: 4.38.0 (October 25, 2025)
[Changelog](#)

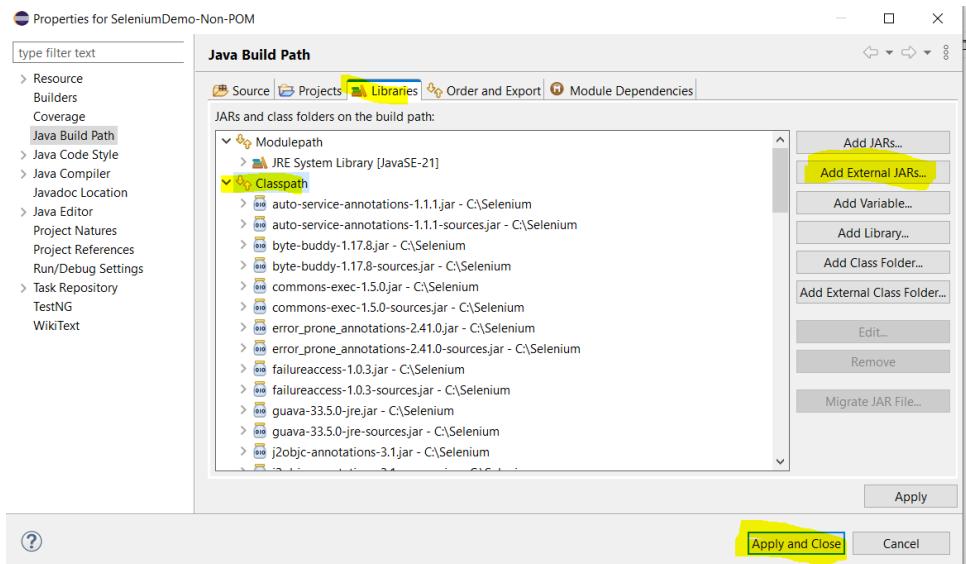


Stable: 4.38.0 (October 25, 2025)
[Changelog](#)
[API Docs](#)

- Extract the ZIP to C drive.
- Create a Java Project. File → New → Java Project → Name: SeleniumDemo-Non-POM
- Go to configure build path.

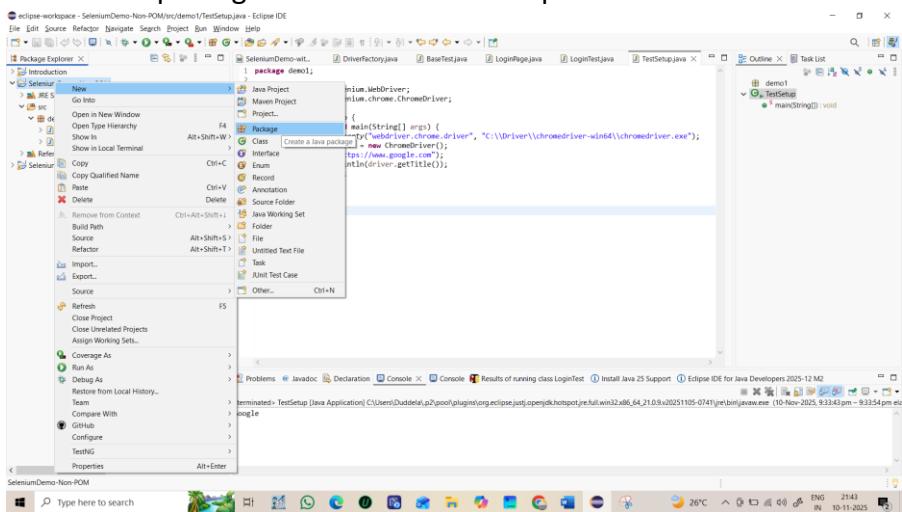


- Under Libraries → select classpath → click Add external JARs → select the location of above extracted ZIP file → click apply and close.



Step 6: Verify Setup

- Create a package under the above java project and name it demo1. Then create a class in the package and name it TestSetup.



- Paste the below code and run as java application.

```

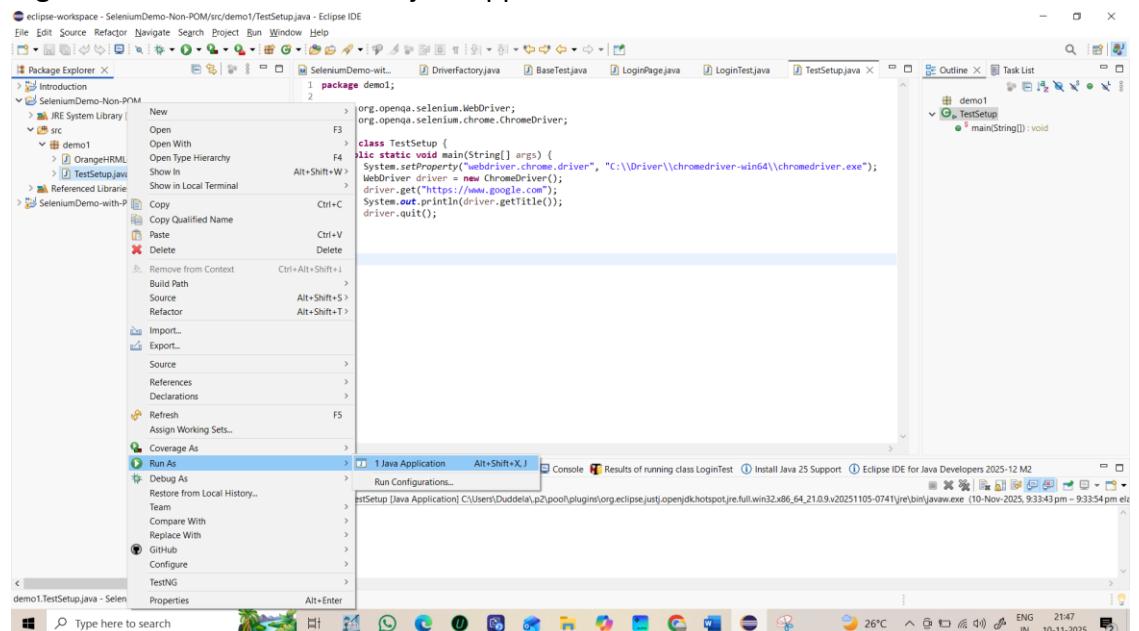
package demo1;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class TestSetup {
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver",
"C:\\\\Driver\\\\chromedriver-win64\\\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.google.com");
        System.out.println(driver.getTitle());
        driver.quit();
    }
}

```

- Right click the class and run as java application



- Expected output is, chrome will be launched and google browser opens and google page title is printed in console.