SELENIUM TEST CASE IN ECLIPSE

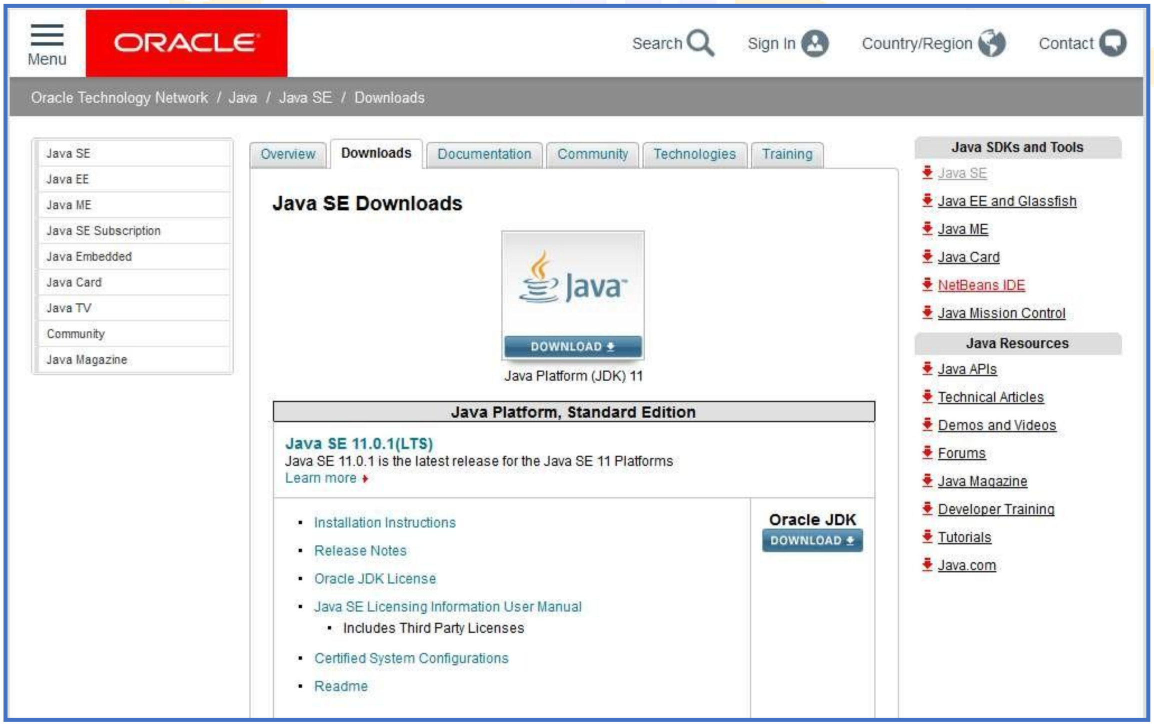
This documentation is divided into three parts:

1. **Java JDK installation**
2. **Eclipse installation**
3. **Selenium Test Case**

1. Install Java JDK

Step 1: Download java jdk.

https://www.oracle.com/technetwork/java/javase/downloads/index.html



Java SE Downloads

Java Platform (JDK) 11

Java Platform, Standard Edition

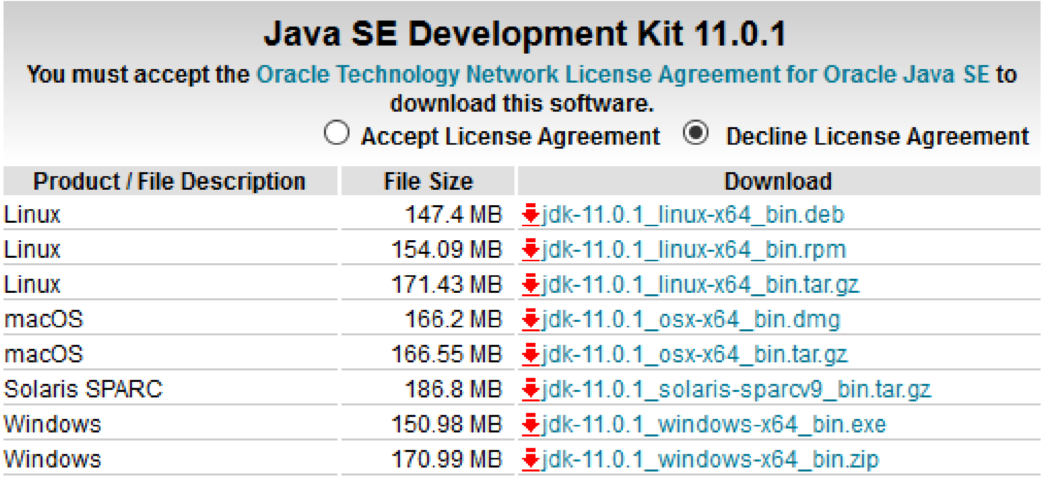
Java SE 11.0.1(LTS)

Java SE 11.0.1 is the latest release for the Java SE 11 Platforms

Learn more

- Installation Instructions
- Release Notes
- Oracle JDK License
- Java SE Licensing Information User Manual
  - Includes Third Party Licenses
- Certified System Configurations
- Readme

Then click on the oracle Jdk Download button, you will land on a page as shown below. Accept the License Agreement. Download the .exe file.



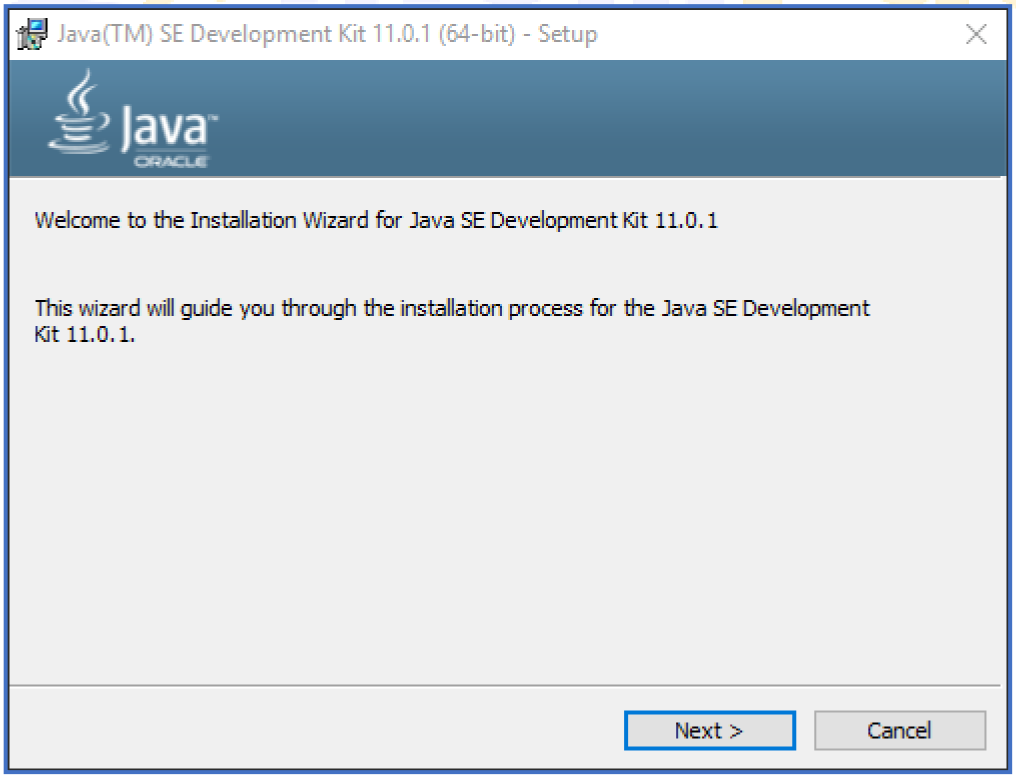
Java SE Development Kit 11.0.1

You must accept the Oracle Technology Network License Agreement for Oracle Java SE to download this software.

○ Accept License Agreement ● Decline License Agreement

Product / File Description	File Size	Download
Linux	147.4 MB	jdk-11.0.1_linux-x64_bin.deb
Linux	154.09 MB	jdk-11.0.1_linux-x64_bin.rpm
Linux	171.43 MB	jdk-11.0.1_linux-x64_bin.tar.gz
macOS	166.2 MB	jdk-11.0.1_osx-x64_bin.dmg
macOS	166.55 MB	jdk-11.0.1_osx-x64_bin.tar.gz
Solaris SPARC	186.8 MB	jdk-11.0.1_solaris-sparcv9_bin.tar.gz
Windows	150.98 MB	jdk-11.0.1_windows-x64_bin.exe
Windows	170.99 MB	jdk-11.0.1_windows-x64_bin.zip

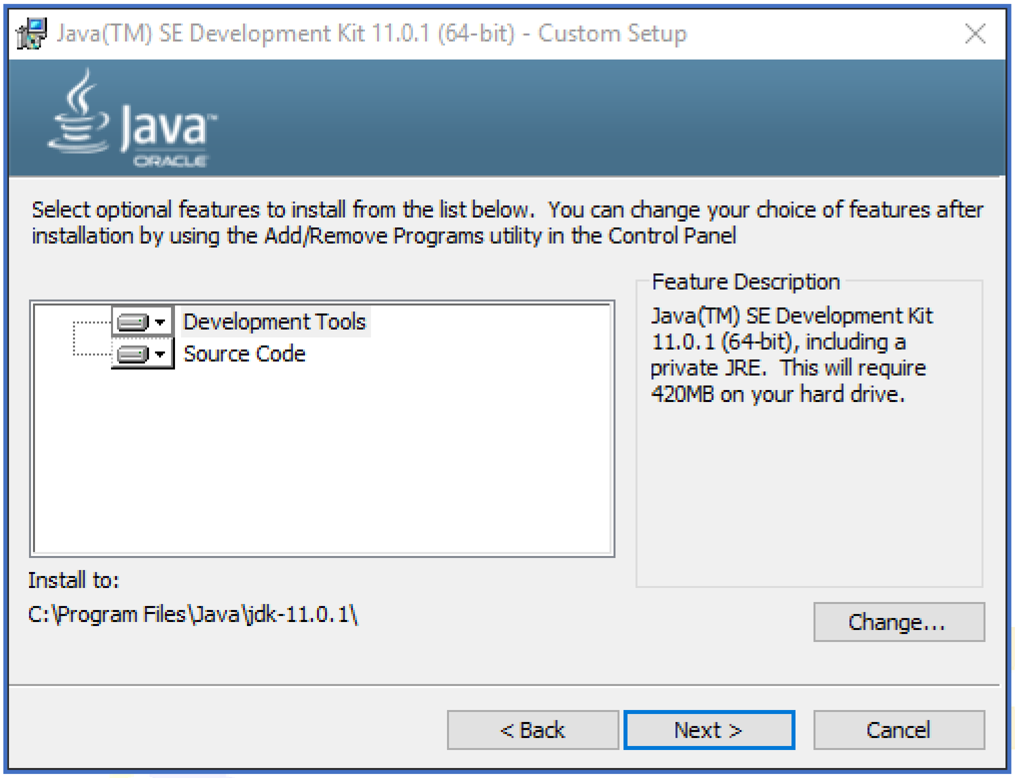
Step 2: Open that downloaded execution file then click on **Next**.



Welcome to the Installation Wizard for Java SE Development Kit 11.0.1

This wizard will guide you through the installation process for the Java SE Development Kit 11.0.1.

Step 3: Select **Developers Tools** and click **Next**.



Java(TM) SE Development Kit 11.0.1 (64-bit) - Custom Setup

Select optional features to install from the list below. You can change your choice of features after installation by using the Add/Remove Programs utility in the Control Panel

- Development Tools
- Source Code

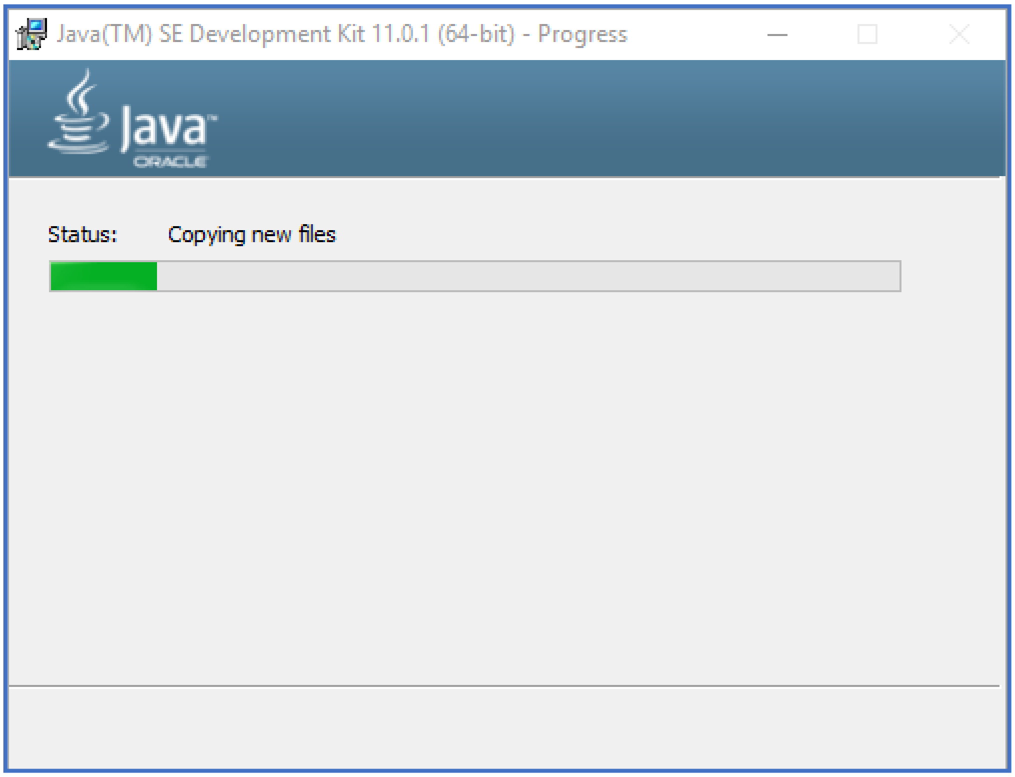
Feature Description

Java(TM) SE Development Kit 11.0.1 (64-bit), including a private JRE. This will require 420MB on your hard drive.

Install to:

C:\Program Files\Java\jdk-11.0.1\

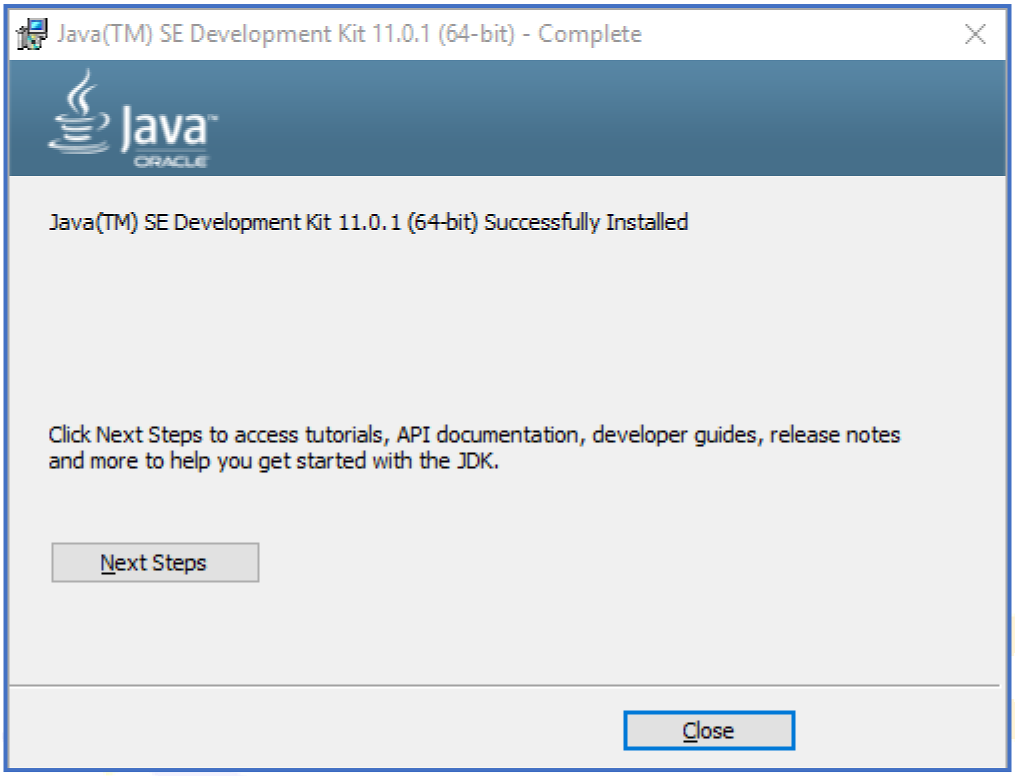
Step 4: It will take a few moments to set up jdk in your system



Java(TM) SE Development Kit 11.0.1 (64-bit) - Progress

Status: Copying new files

Step 5: Click on close to complete the set up.



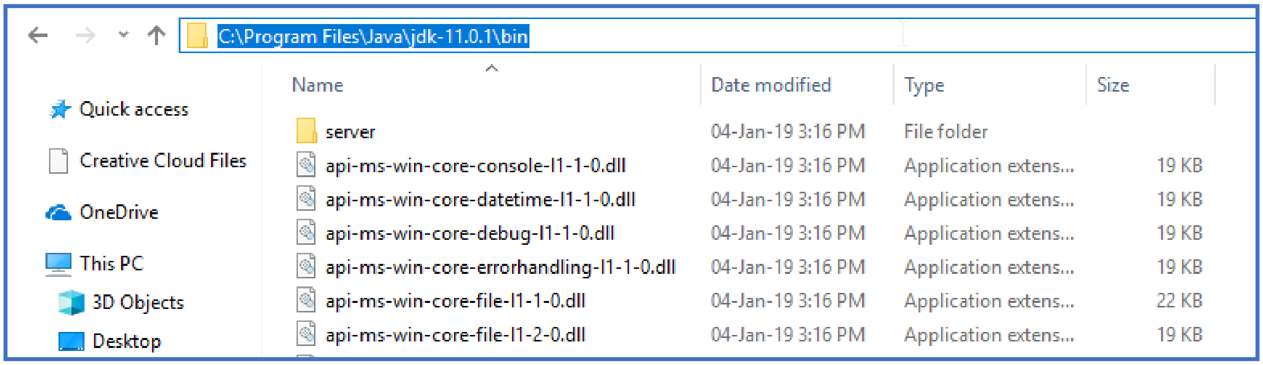
Java(TM) SE Development Kit 11.0.1 (64-bit) - Complete

Java(TM) SE Development Kit 11.0.1 (64-bit) Successfully Installed

Click Next Steps to access tutorials, API documentation, developer guides, release notes and more to help you get started with the JDK.

Before moving ahead, we need **to add the environment variables to the path** as follows.

Step 5: Go to local **C drive** Program Files java **jdk-11.0.1bin**

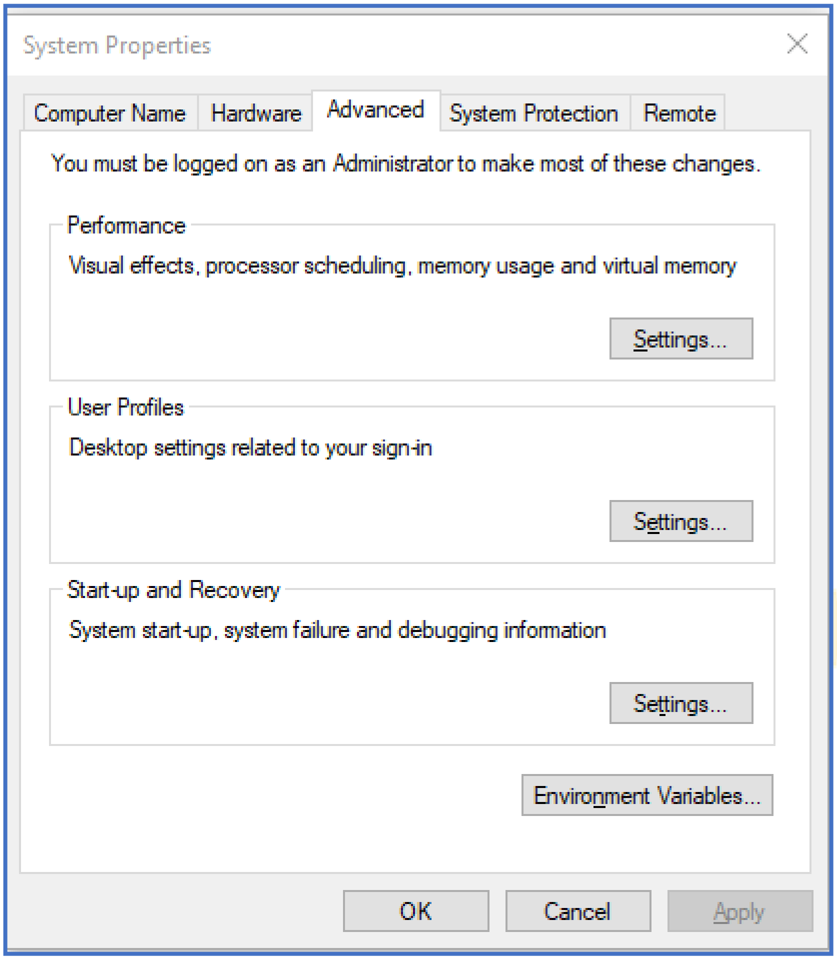


C:\Program Files\Java\jdk-11.0.1\bin

Name	Date modified	Type	Size
server	04-Jan-19 3:16 PM	File folder	
api-ms-win-core-console-l1-1-0.dll	04-Jan-19 3:16 PM	Application extens...	19 KB
api-ms-win-core-datetime-l1-1-0.dll	04-Jan-19 3:16 PM	Application extens...	19 KB
api-ms-win-core-debug-l1-1-0.dll	04-Jan-19 3:16 PM	Application extens...	19 KB
api-ms-win-core-errorhandling-l1-1-0.dll	04-Jan-19 3:16 PM	Application extens...	19 KB
api-ms-win-core-file-l1-1-0.dll	04-Jan-19 3:16 PM	Application extens...	22 KB
api-ms-win-core-file-l1-2-0.dll	04-Jan-19 3:16 PM	Application extens...	19 KB

Step 6: Copy the path as shown above.

Step 7: Go to **Control Panel System and Security System Advanced System Settings**



You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage and virtual memory

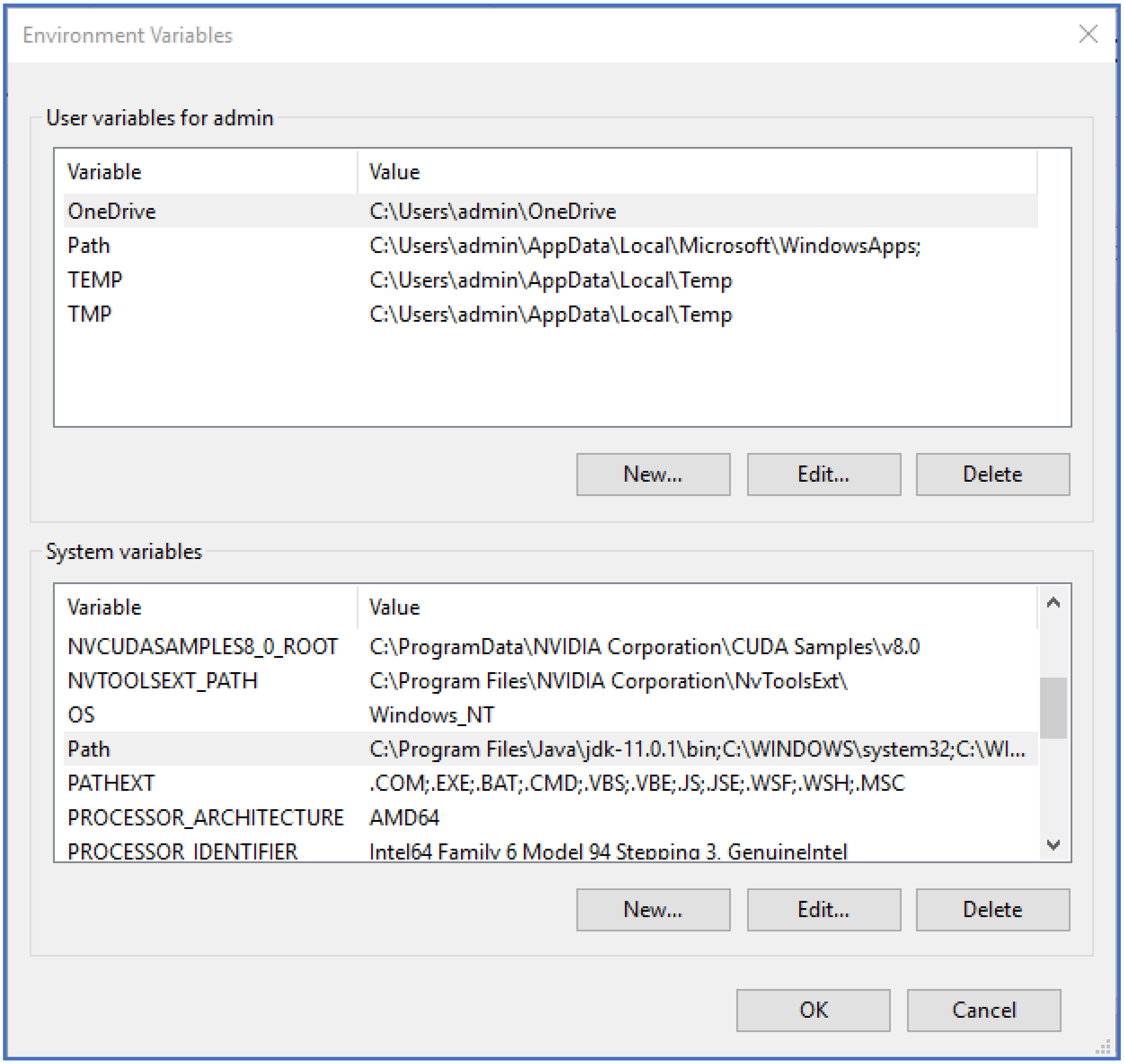
User Profiles

Desktop settings related to your sign-in

Start-up and Recovery

System start-up, system failure and debugging information

Step 8: Click on **Environment Variables.**



Environment Variables

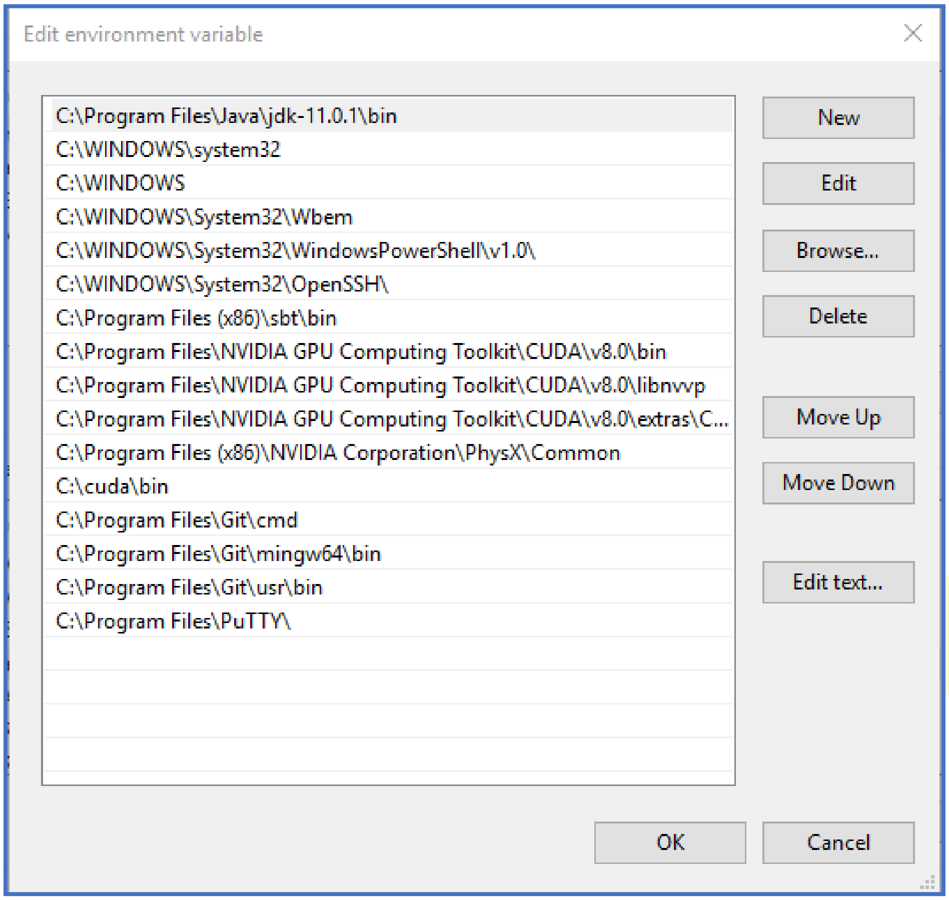
User variables for admin

Variable	Value
OneDrive	C:\Users\admin\OneDrive
Path	C:\Users\admin\AppData\Local\Microsoft\WindowsApps;
TEMP	C:\Users\admin\AppData\Local\Temp
TMP	C:\Users\admin\AppData\Local\Temp

System variables

Variable	Value
NVCUDASAMPLES8_0_ROOT	C:\ProgramData\NVIDIA Corporation\CUDA Samples\v8.0
NVTOOLSEXT_PATH	C:\Program Files\NVIDIA Corporation\NvToolsExt\
OS	Windows_NT
Path	C:\Program Files\Java\jdk-11.0.1\bin;C:\WINDOWS\system32;C:\WI...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64
PROCESSOR_IDENTIFIER	Intel64 Family 6 Model 94 Stepping 3, GenuineIntel

Step 9: Click on Path in the System Variables section. And then paste the previously copied path (***C:\Program Files\Java\jdk-11.0.1\bin***) as shown below.

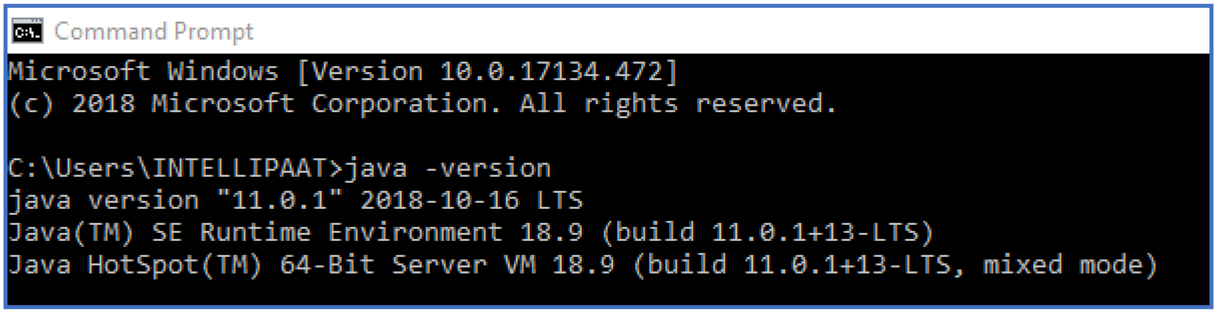


Edit environment variable

```
C:\Program Files\Java\jdk-11.0.1\bin
C:\WINDOWS\system32
C:\WINDOWS
C:\WINDOWS\System32\Wbem
C:\WINDOWS\System32\WindowsPowerShell\v1.0\
C:\WINDOWS\System32\OpenSSH\
C:\Program Files (x86)\sbt\bin
C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v8.0\bin
C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v8.0\libnvvp
C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v8.0\extras\C...
C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common
C:\cuda\bin
C:\Program Files\Git\cmd
C:\Program Files\Git\mingw64\bin
C:\Program Files\Git\usr\bin
C:\Program Files\PuTTY\
```

Step 10: Now open **command prompt** and run the following command.

```
java -version
```



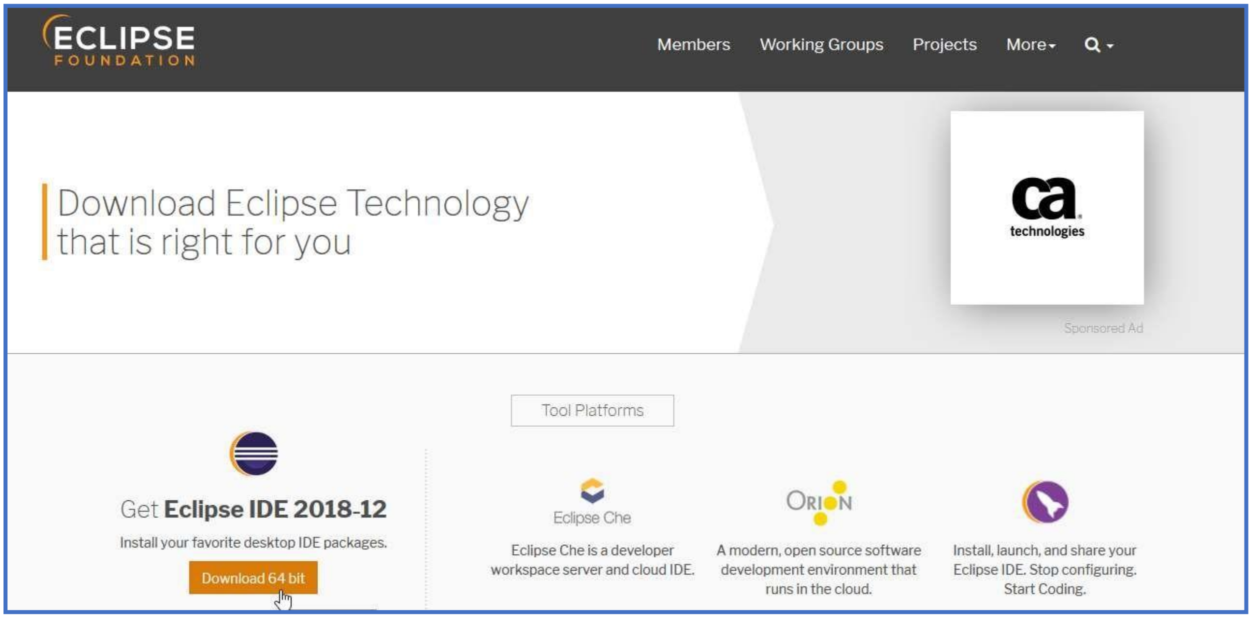
Command Prompt

```
Microsoft Windows [Version 10.0.17134.472]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\INTELLIPAAT>java -version
java version "11.0.1" 2018-10-16 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.1+13-LTS)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.1+13-LTS, mixed mode)
```

2. Eclipse Installation

Step 1: Go to the eclipse download page.

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



Download Eclipse Technology that is right for you

Get **Eclipse IDE 2018-12**

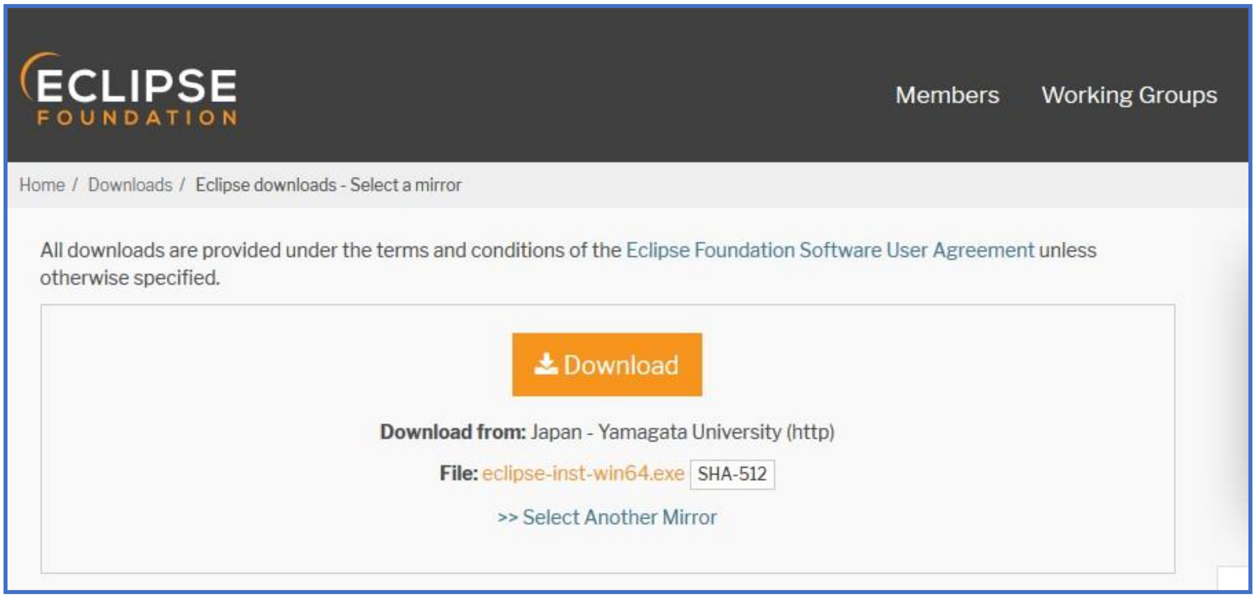
Install your favorite desktop IDE packages.

Eclipse Che is a developer workspace server and cloud IDE.

A modern, open source software development environment that runs in the cloud.

Install, launch, and share your Eclipse IDE. Stop configuring. Start Coding.

Step 2: Click on the download 64-bit link. You should land on a page as shown.



Home / Downloads / Eclipse downloads - Select a mirror

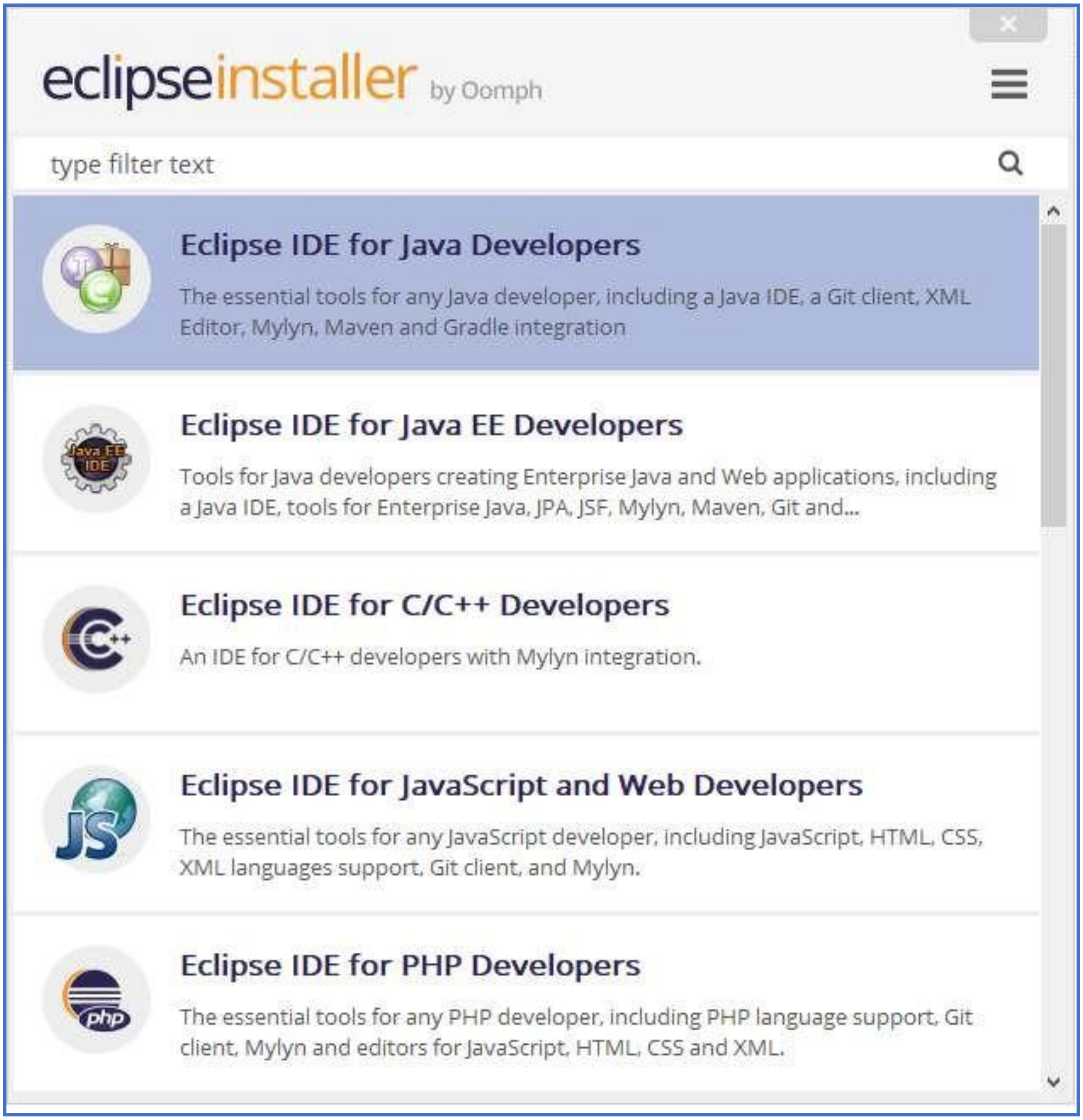
All downloads are provided under the terms and conditions of the Eclipse Foundation Software User Agreement unless otherwise specified.

Download from: Japan - Yamagata University (http)

File: eclipse-inst-win64.exe

>> Select Another Mirror

Step 3: Once the download is finished launch the installer.



Eclipse IDE for Java Developers

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven and Gradle integration

Eclipse IDE for Java EE Developers

Tools for Java developers creating Enterprise Java and Web applications, including a Java IDE, tools for Enterprise Java, JPA, JSF, Mylyn, Maven, Git and...

Eclipse IDE for C/C++ Developers

An IDE for C/C++ developers with Mylyn integration.

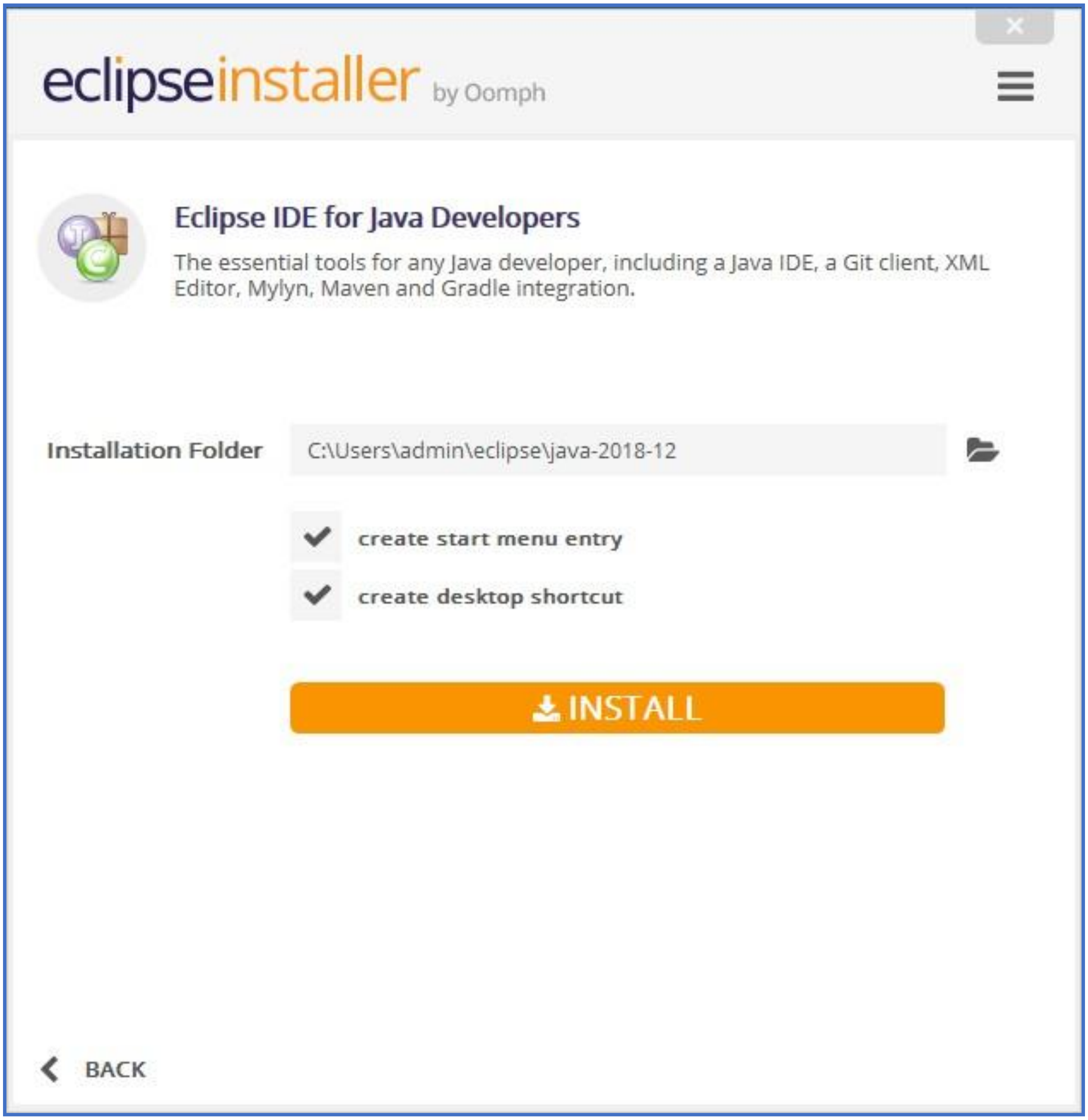
Eclipse IDE for JavaScript and Web Developers

The essential tools for any JavaScript developer, including JavaScript, HTML, CSS, XML languages support, Git client, and Mylyn.

Eclipse IDE for PHP Developers

The essential tools for any PHP developer, including PHP language support, Git client, Mylyn and editors for JavaScript, HTML, CSS and XML.

Step 4: Click on **Eclipse IDE for Java Developers.**



Eclipse IDE for Java Developers

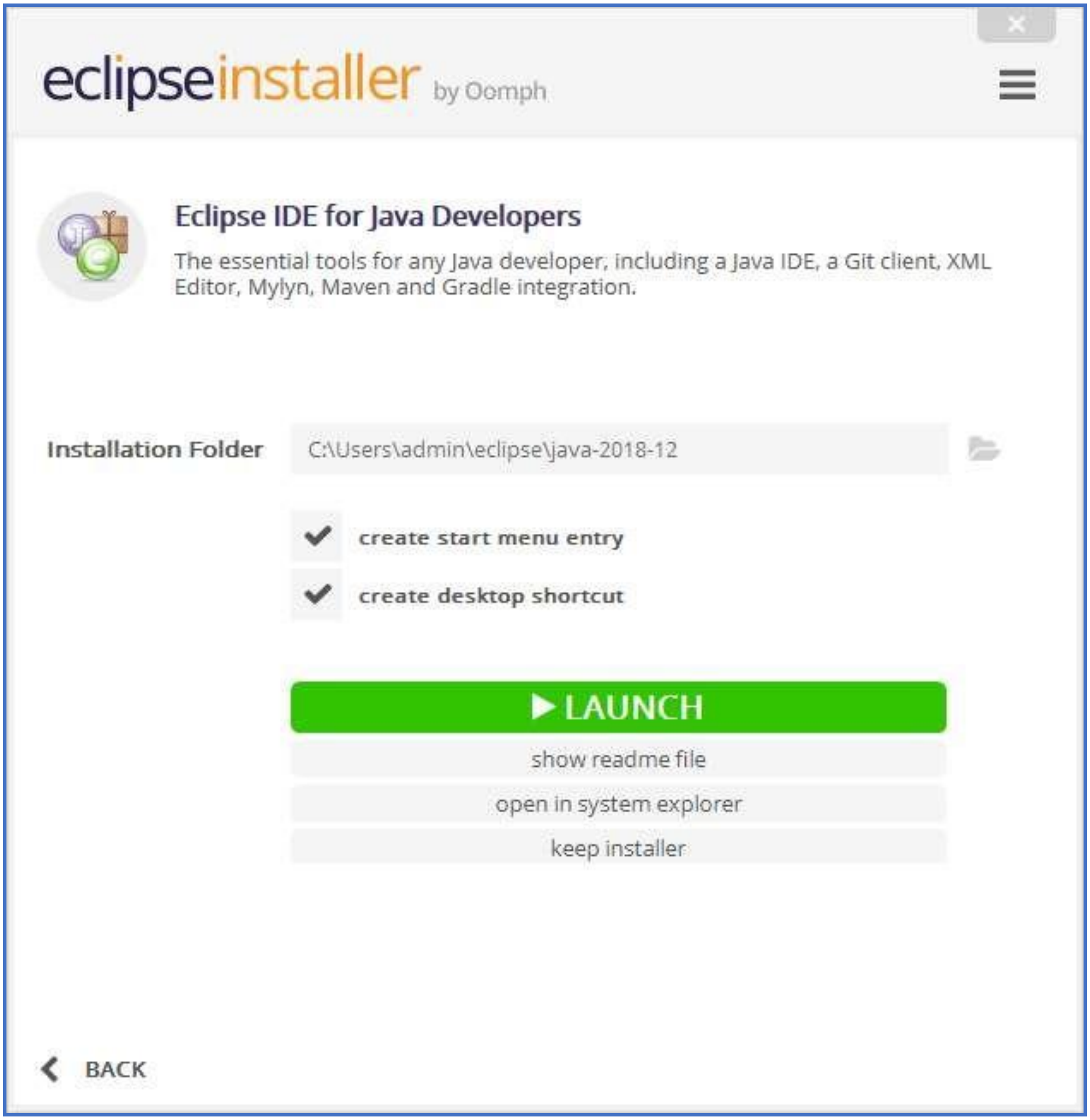
The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven and Gradle integration.

Installation Folder: C:\Users\admin\eclipse\java-2018-12

☑ create start menu entry

☑ create desktop shortcut

Step 5: Once done, click on LAUNCH.



Eclipse IDE for Java Developers

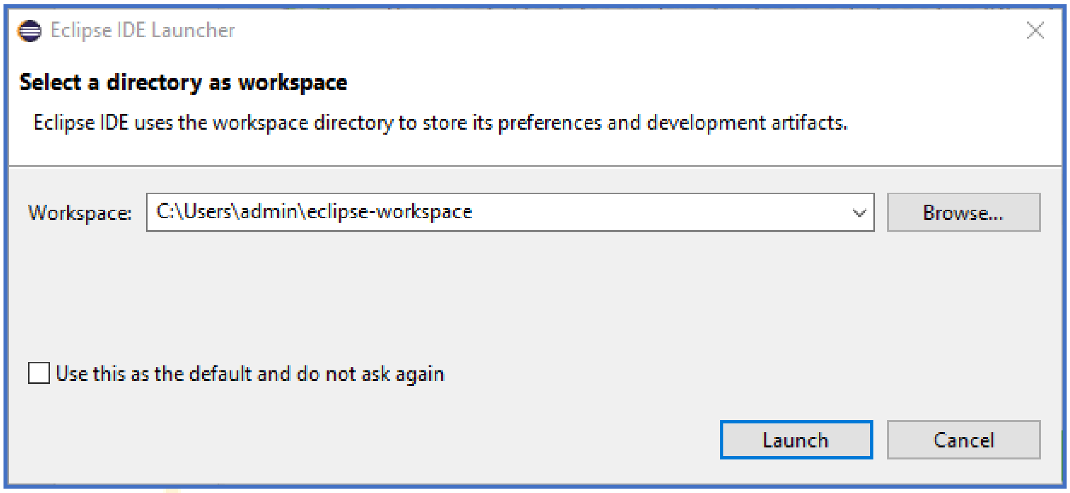
The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven and Gradle integration.

Installation Folder: C:\Users\admin\eclipse\java-2018-12

☑ create start menu entry

☑ create desktop shortcut

Step 6: Click on Launch.



Eclipse IDE Launcher

Select a directory as workspace

Eclipse IDE uses the workspace directory to store its preferences and development artifacts.

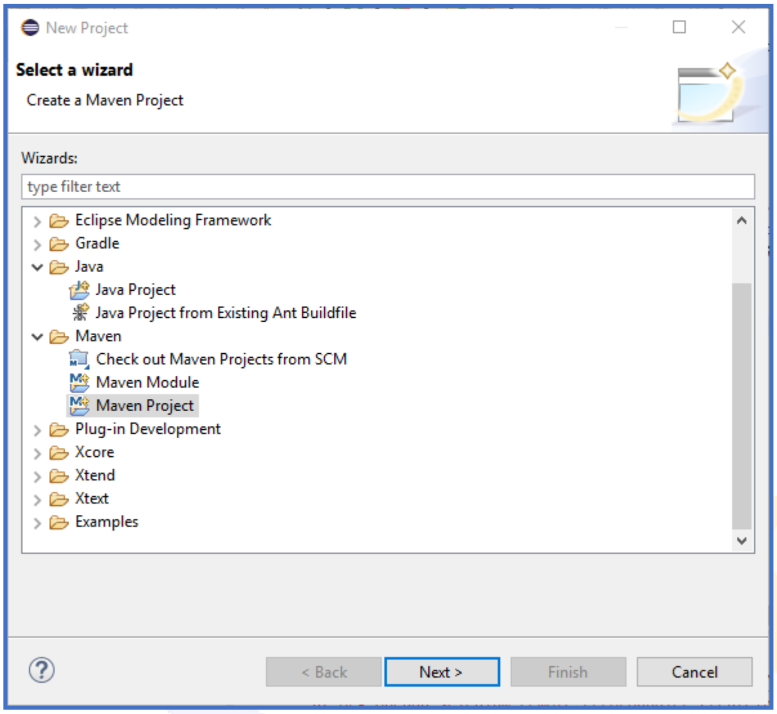
Workspace: C:\Users\admin\eclipse-workspace

□ Use this as the default and do not ask again

Now that we have successfully set up Java and Eclipse in our environment, we will be Performing a Selenium Test Case using Maven.

3. Selenium Test Case:

Step 1: Start a new Maven Project.



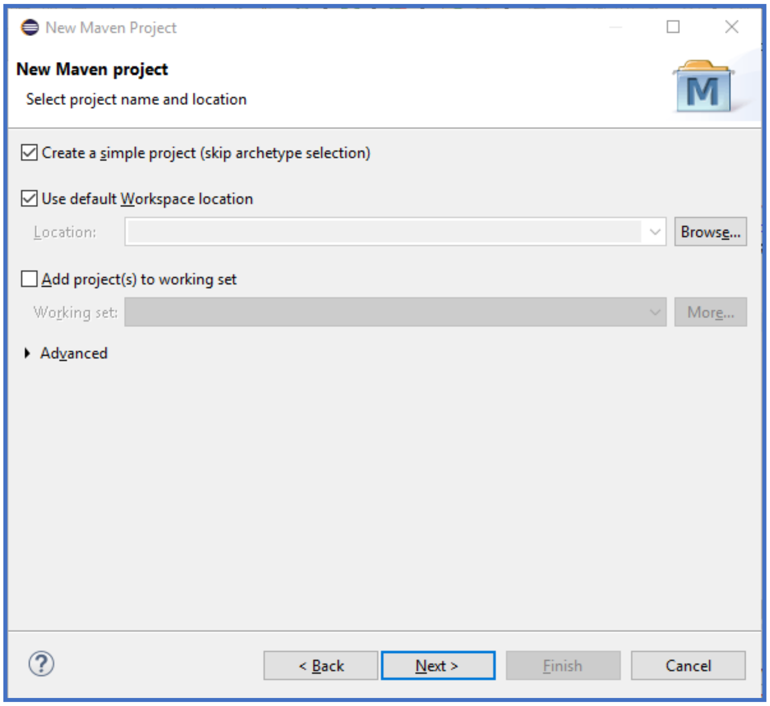
New Project

Select a wizard

Create a Maven Project

Wizards:

Step 2: Check the **Create a Simple Project** box. And click **Next.**



New Maven Project

New Maven project

Select project name and location

☑ Create a simple project (skip archetype selection)

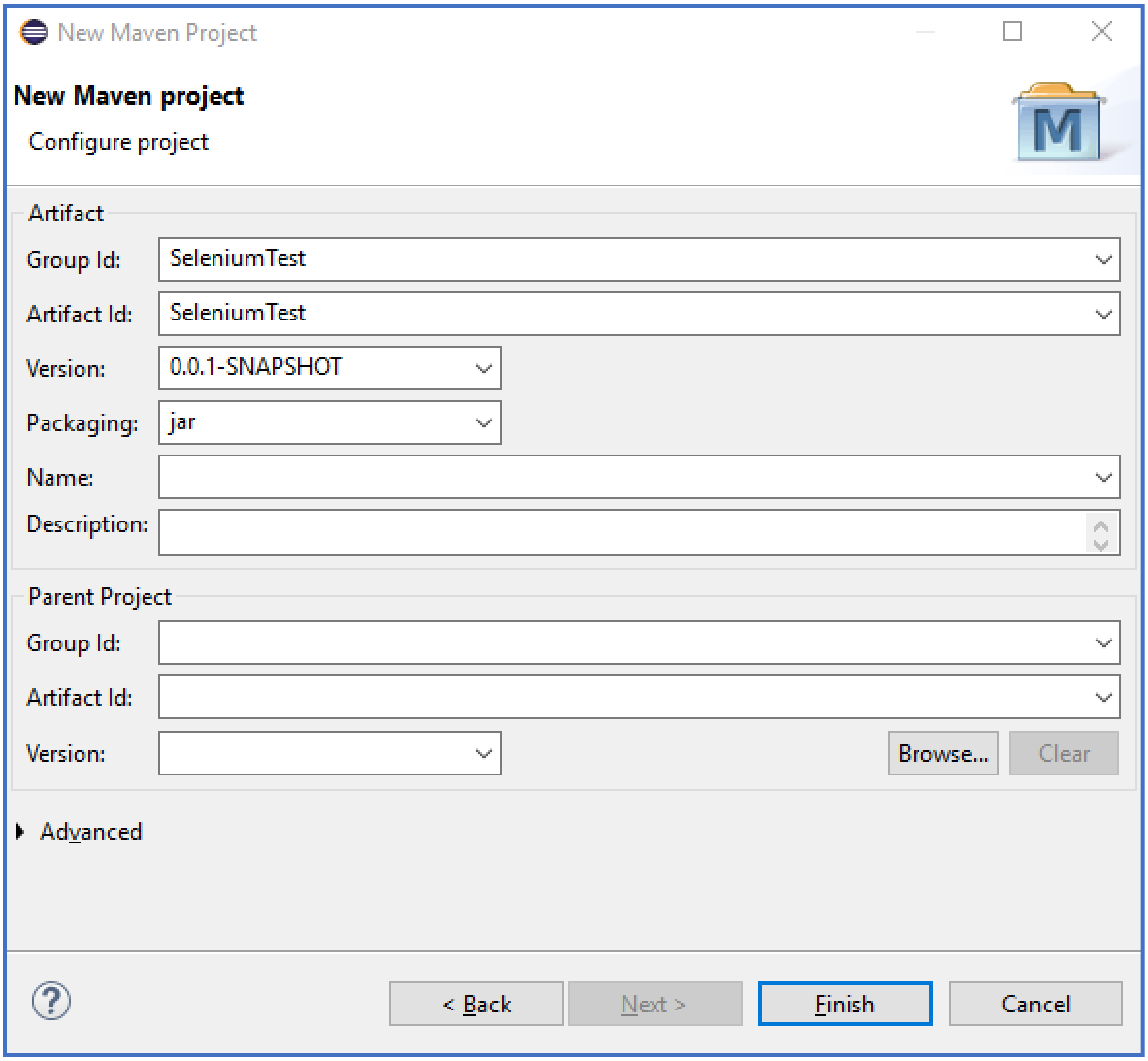
☑ Use default Workspace location

Location:

□ Add project(s) to working set

Working set:

Step 3: Enter **Group Id** and **Artifact Id.** And click on **Finish.**



New Maven Project

New Maven project

Configure project

Artifact

Group Id: SeleniumTest

Artifact Id: SeleniumTest

Version: 0.0.1-SNAPSHOT

Packaging: jar

Name:

Description:

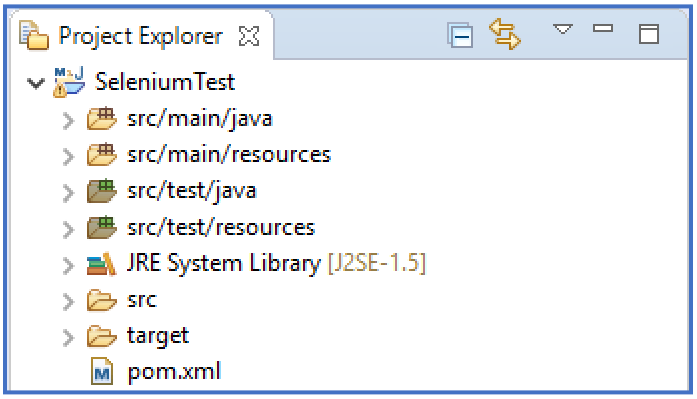
Parent Project

Group Id:

Artifact Id:

Version:

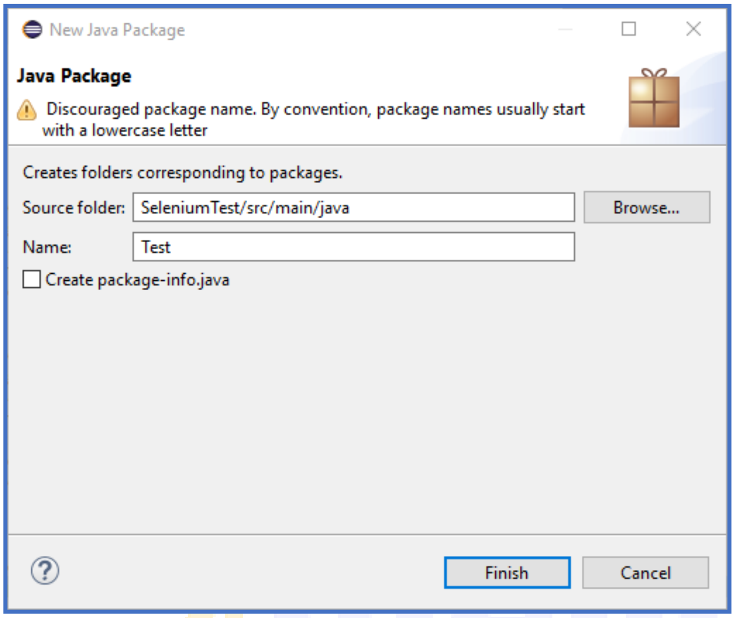
Step 4: Your project should appear in the **Project Explorer** section as shown below.



Project Explorer

- SeleniumTest
  - src/main/java
  - src/main/resources
  - src/test/java
  - src/test/resources
  - JRE System Library [J2SE-1.5]
  - src
  - target
  - pom.xml

Step 5: Right click on **src/main/java** and create a **java package.** Enter a **Package Name (Test** in this case**).** Then click on **Finish.**



New Java Package

Java Package

⚠ Discouraged package name. By convention, package names usually start with a lowercase letter

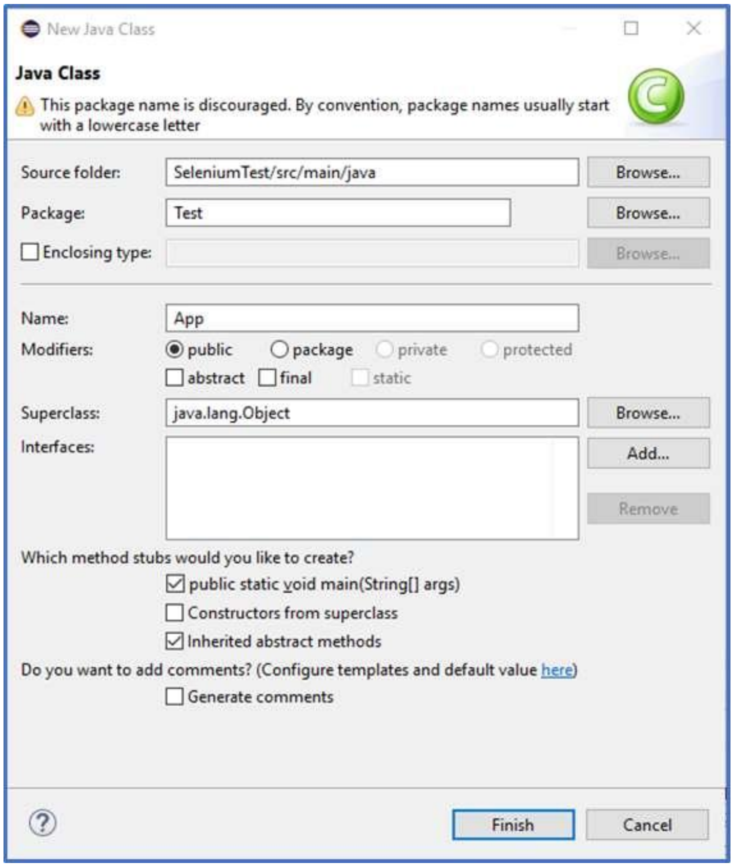
Creates folders corresponding to packages.

Source folder: SeleniumTest/src/main/java

Name: Test

□ Create package-info.java

Step 6: Now create a **Java Class** by right clicking on **test package**, name the class as **App.** Check the **Public static void main(String[] args)** box**.** And click on **Finish.**



New Java Class

Java Class

⚠ This package name is discouraged. By convention, package names usually start with a lowercase letter

Source folder: SeleniumTest/src/main/java

Package: Test

□ Enclosing type:

Name: App

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)

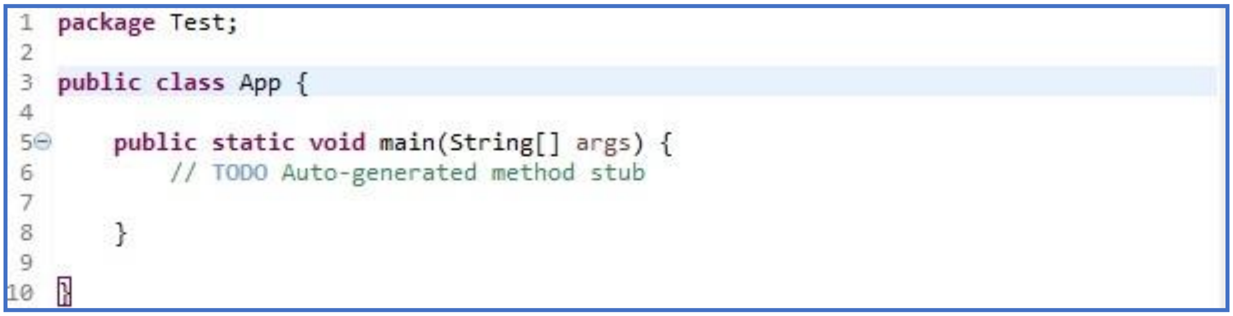
□ Constructors from superclass

☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

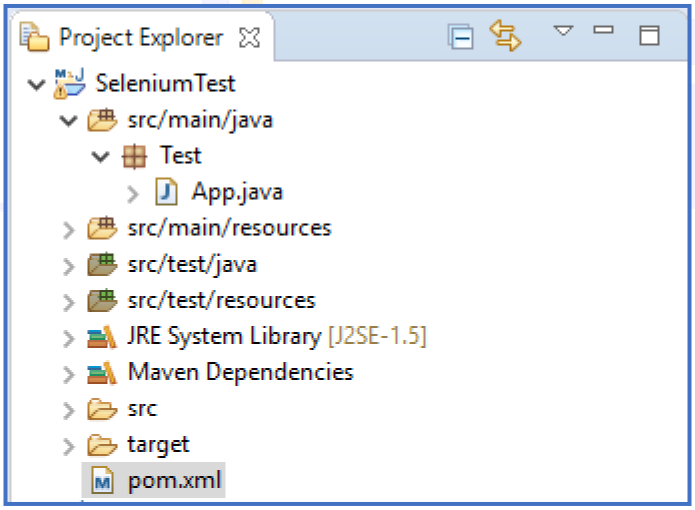
□ Generate comments

Step 7: Now, your eclipse workspace should look like this.

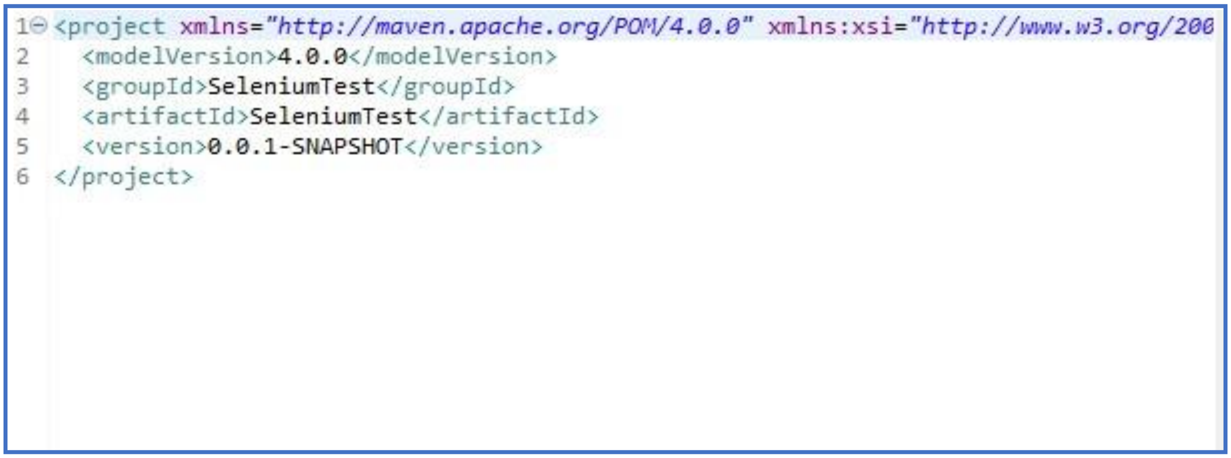


```
package Test;

public class App {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

    }

}
```

Before moving on to the scripting part we need to **configure the Maven Dependencies to perform Selenium in Eclipse**. We will be adding the **Maven Dependencies** to the **Pom.xml** file under **target folder**.



Before adding the dependencies, the Pom.xml file looks like this:



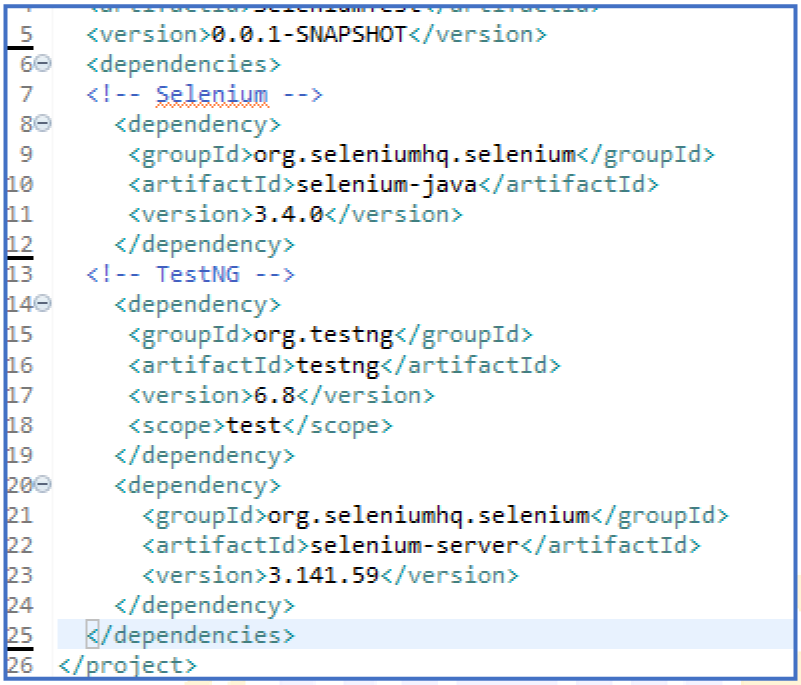
```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/200
  <modelVersion>4.0.0</modelVersion>
  <groupId>SeleniumTest</groupId>
  <artifactId>SeleniumTest</artifactId>
  <version>0.0.1-SNAPSHOT</version>
</project>
```

Step 8: Now go ahead and add the below dependencies to the **pom.xml** file.

```
<dependencies>
 <!-- Selenium -->
 <dependency>
   <groupId>org.seleniumhq.selenium</groupId>
   <artifactId>selenium-java</artifactId>
   <version>3.4.0</version>
 </dependency>
<!-- TestNG -->
 <dependency>
   <groupId>org.testng</groupId>
   <artifactId>testng</artifactId>
   <version>6.8</version>
   <scope>test</scope>
 </dependency>
 <!-- Selenium Server-->
 <dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-server</artifactId>
    <version>3.141.59</version>
  </dependency>
 </dependencies>
```

Add the dependencies in between **<project> </project>** tags.

Refer to the below given image.



```
  <version>0.0.1-SNAPSHOT</version>
  <dependencies>
  <!-- Selenium -->
    <dependency>
     <groupId>org.seleniumhq.selenium</groupId>
     <artifactId>selenium-java</artifactId>
     <version>3.4.0</version>
    </dependency>
  <!-- TestNG -->
    <dependency>
     <groupId>org.testng</groupId>
     <artifactId>testng</artifactId>
     <version>6.8</version>
     <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>org.seleniumhq.selenium</groupId>
      <artifactId>selenium-server</artifactId>
      <version>3.141.59</version>
    </dependency>
  </dependencies>
</project>
```

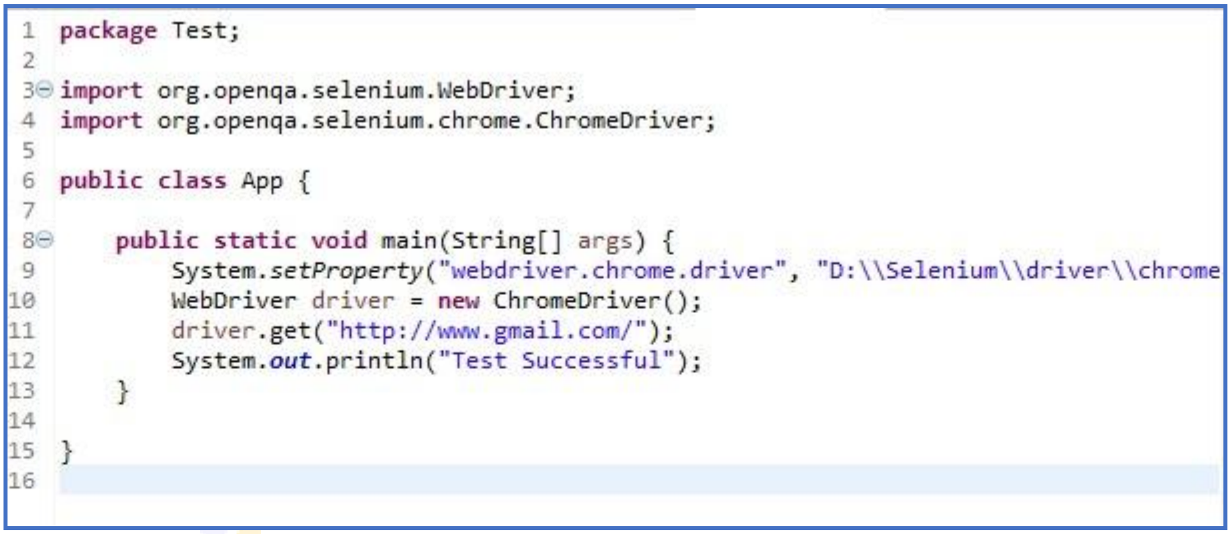
Step 9: Now, add the below given script to the **main() function** to the **App.java** file.

```
System.setProperty("webdriver.chrome.driver", "D:\\Selenium\\driver\\chromedriver.exe");
WebDriver driver = new ChromeDriver();
driver.get("http://www.gmail.com/");
System.out.println("Test Successful");
```

Step 10: To make this run properly we need to import **org.openqa.selenium.WebDriver** and **org.openqa.selenium.chrome.ChromeDriver**

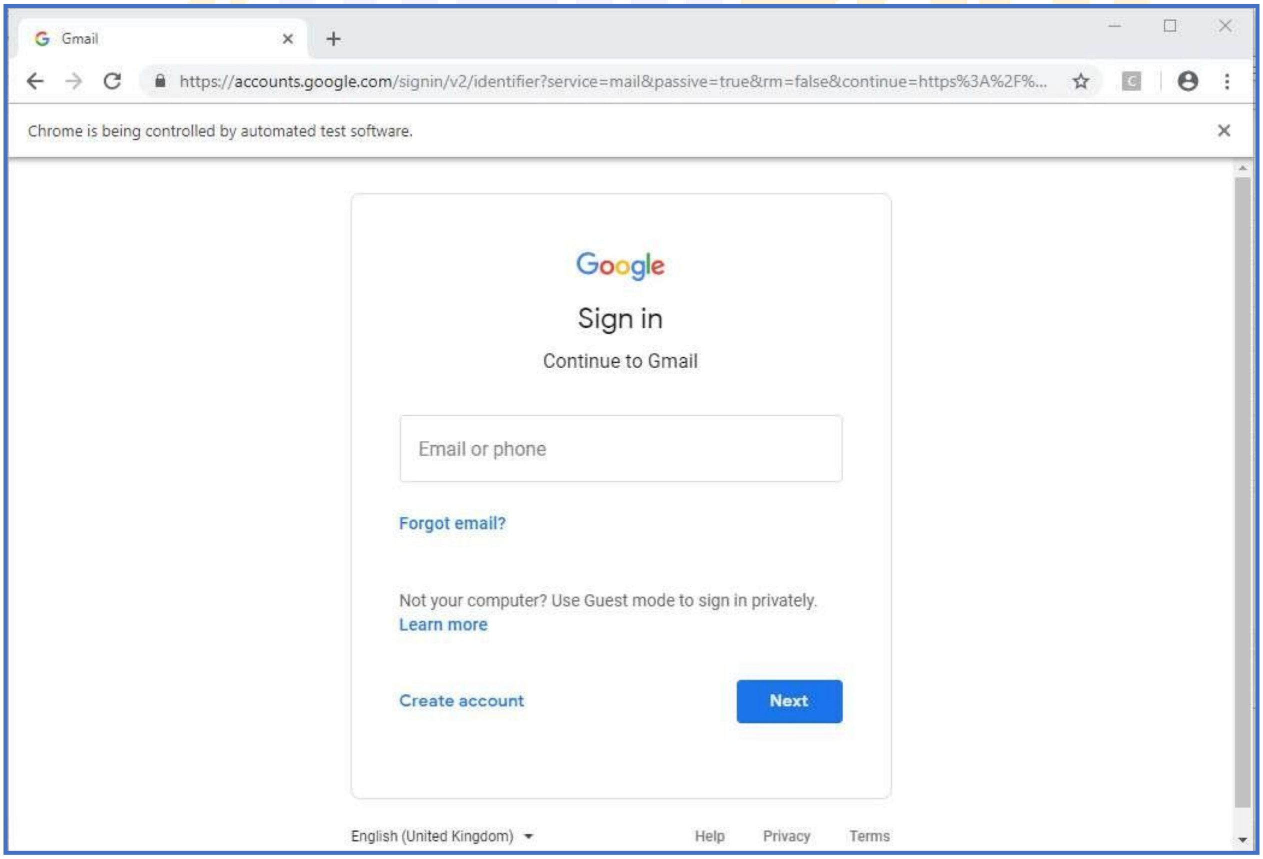
```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
```

It should look like this.



```
package Test;

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

public class App {

    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver", "D:\\Selenium\\driver\\chrome
        WebDriver driver = new ChromeDriver();
        driver.get("http://www.gmail.com/");
        System.out.println("Test Successful");
    }

}

```

Step 11: Now run **App**. Chrome browser window will pop up with Gmail sign in page.

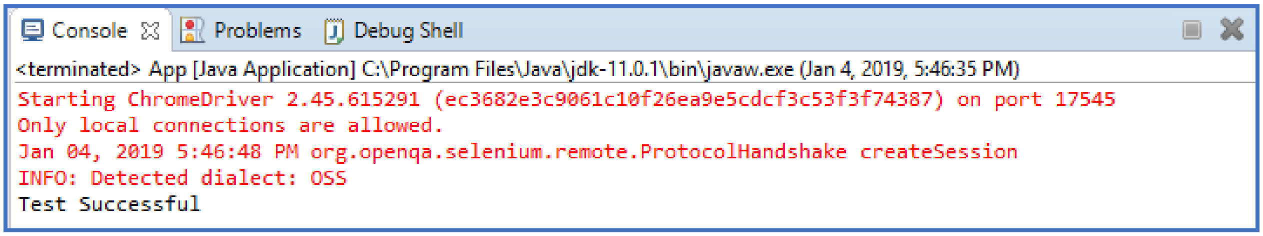


https://accounts.google.com/signin/v2/identifier?service=mail&passive=true&rm=false&continue=https%3A%2F%...

Chrome is being controlled by automated test software.

Not your computer? Use Guest mode to sign in privately.

Step 12: Go to Eclipse again and see the console. You should see the Test Successful message appearing in there.



```
<terminated> App [Java Application] C:\Program Files\Java\jdk-11.0.1\bin\javaw.exe (Jan 4, 2019, 5:46:35 PM)
Starting ChromeDriver 2.45.615291 (ec3682e3c9061c10f26ea9e5cdcf3c53f3f74387) on port 17545
Only local connections are allowed.
Jan 04, 2019 5:46:48 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: OSS
Test Successful
```