Table of Contents

Getting TownSend repository from Github

1. You can get TownSend repository with

    ```
    git clone https://github.com/fitohub/TownSend.git
    ```

2. That creates a directory named TownSend, initializes a .git directory inside it, pulls down all the data for that repository, and checks out a working copy of the latest version. If you go into the new TownSend directory that was just created, you will see the project files in there, ready to be worked on or used;

3. Inf you want to clone the repository into a directory named something other than TownSend, you can specify the new directory name as an additional argument:

    ```
    git clone https://github.com/fitohub/TownSend.git cafeTownSend
    ```

    That command does the same thing as the previous one, but the target directory is called "cafeTownSend".

Installing Java

1. To install Java, you first need to download the installer program from Oracle. Visit the Download Java page: https://www.oracle.com/technetwork/java/javase/downloads/index.html. Click on "Download" button;

2. Accept the license agreement. Choose the correct OS corresponding for the specific JDK (in this case jdk-12.0.2_osx-x64_bin.dmg);
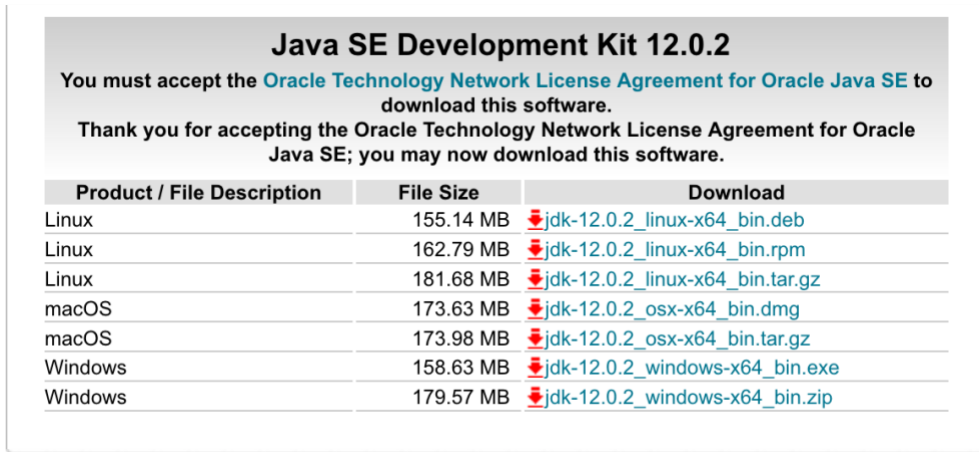
3. The above step will start the downloading of the JDK exe automatically.
4. Once the download is completed, double click the file to begin the installation of JDK. Follow the instructions until Java installation is complete. Click on Close button.
5. Add the M2_HOME environment variable to your ~/.bash_profile, run the following command to open it in text editor:

```
open -e ~/.bash_profile
```

6. Add the following exports to your ~/.bash_profile and save it:

```
export JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk-
12.0.2.jdk/Contents/Home
export PATH=$MAVEN/bin:$JAVA_HOME:$JAVA_HOME/bin:$PATH
```

7. Use source command on ~/.bash_profile which is used to load any functions file into the current shell script or a command prompt:
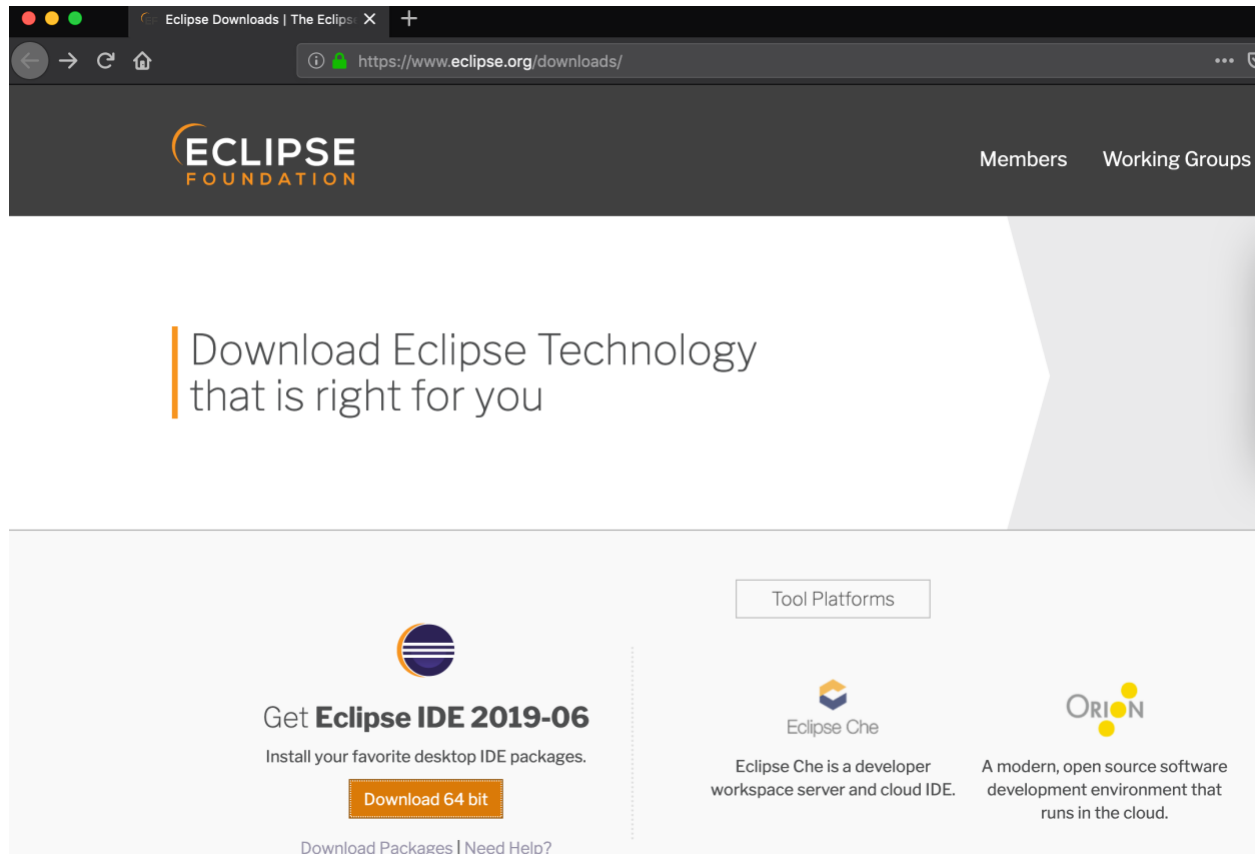
```
source ~/.bash-profile
```

8. Run the following command to verify Maven version:

```
java -version
```

9. If your Mac OS system displays information on the Java version installed, this means Java is successfully installed on your Mac.

Installing Eclipse

1. To download Eclipse IDE, copy the link https://www.eclipse.org/downloads/ into your browser and hit Enter
2. Get Eclipse IDE 2019-06.

3. Upon clicking on Download 64-bit button, you would be redirected to a download page where you can choose the server nearest to you so that you can download faster. Simply click download button at the left panel as shown below. Your download will start automatically.
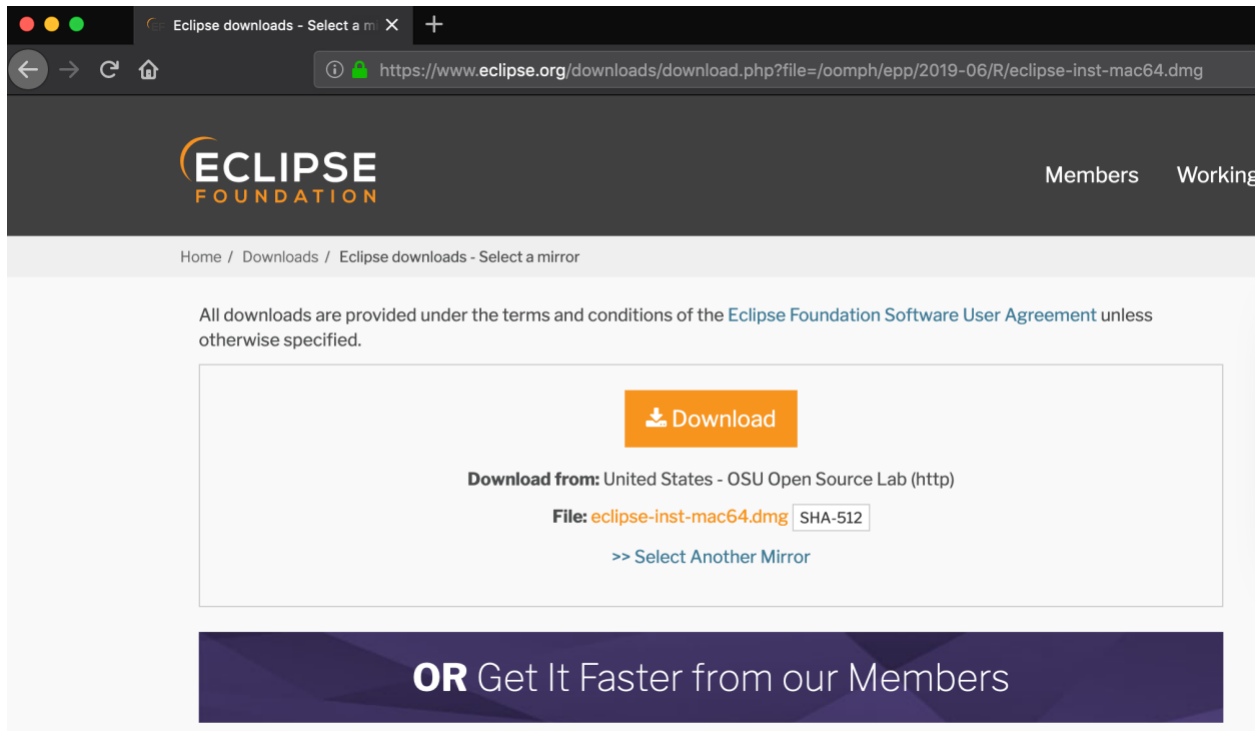
4. Once download is done, go to Downloads folder and you will find "eclipse-inst-mac64.dmg" (the file name may be different based on the version you download but it should be a .dmg file starting with prefix "eclipse");
5. Double click the .dmg file to make its content available (name will show up in the Finder sidebar), usually a window opens showing the content as well;
6. Drag the application from the .dmg window into /Applications to install (may need an administrator password);
7. Wait for the copy process to finish;
8. Eject the .dmg (by clicking the eject button in the Sidebar);
9. Delete the .dmg from Downloads folder;
10. Open eclipse.
11. Open project TownSend cloned with git from Github repository. You can see the structure below:

▼ 🗂 TownSend
  ▶ 📚 JRE System Library [JavaSE-1.8]
  ▼ 🗄 src
    ▼ ▦ appModules
      ▶ 🔷 CreateAction.java
      ▶ 🔷 DeleteAction.java
      ▶ 🔷 ReadAction.java
      ▶ 🔷 SignInAction.java
      ▶ 🔷 SignOutAction.java
      ▶ 🔷 UpdateAction.java
    ▼ 📁 framework
      ▶ 🔷 BasePage.java
    ▼ ▦ pageObjects
      ▶ 🔷 EmployeePage.java
      ▶ 🔷 HomePage.java
      ▶ 🔷 LoginPage.java
    ▼ ▦ testData
        📊 testData.xlsx
    ▼ ▦ utility
      ▶ 🔷 Constants.java
      ▶ 🔷 ExcelUtils.java
      ▶ 🔷 Log.java
        📄 testng.xml
  ▶ 📚 Referenced Libraries
  ▶ 📚 TestNG
  ▶ 📚 Maven Dependencies
  ▶ 📂 bin
  ▶ 📂 target
  ▶ 📂 test-output
    📄 log4j.xml
    📄 logfile.log
    📄 pom.xml
    📄 README.md

Installing Maven

1. Download Apache Maven bin.tar.gz file from http://maven.apache.org/download.cgi. Extract/unzip the distribution archive file into /Applications/apache-maven-3.5.4 (version can vary) by the following command:

   ```
   $ tar -xvf apache-maven-3.5.4-bin.tar.gz -C /Applications/
   ```

2. Add the M2_HOME environment variable to your ~/.bash_profile, run the following command to open it in text editor:

   ```
   open -e ~/.bash_profile
   ```

3. Add the following exports to your ~/.bash_profile and save it:

   ```
   export M2_HOME=/Applications/apache-maven-3.4.5
   export PATH=$PATH:$M2_HOME/bin
   ```

4. Use source command on ~/.bash_profile which is used to load any functions file into the current shell script or a command prompt:

   ```
   source ~/.bash-profile
   ```
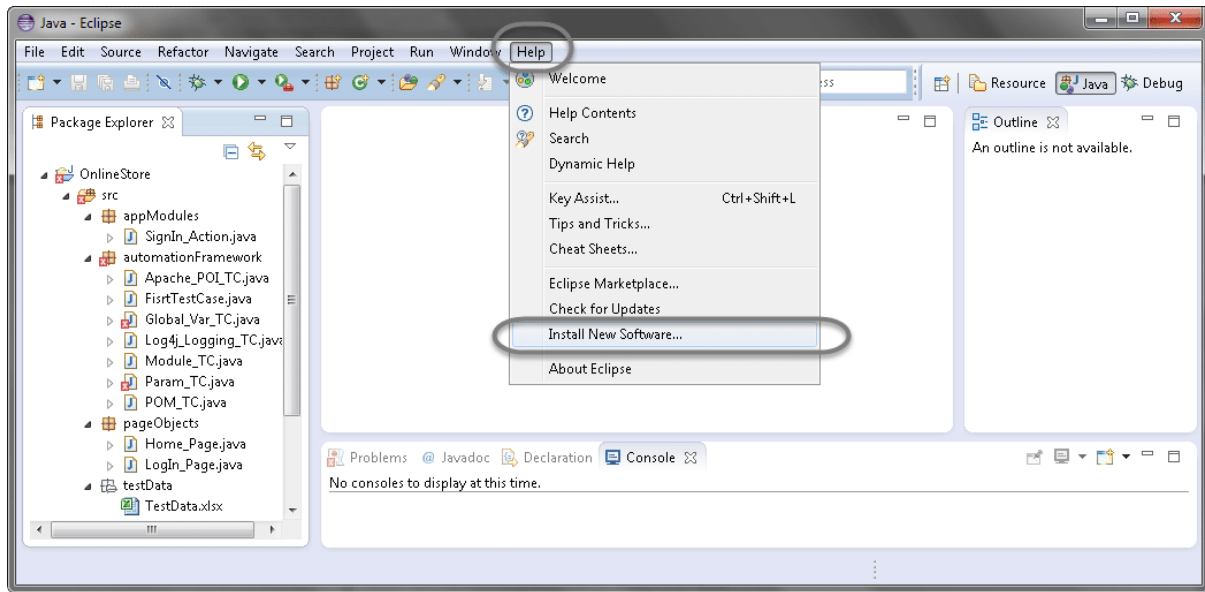
5. Run the following command to verify Maven version:

   ```
   mvn -version
   ```

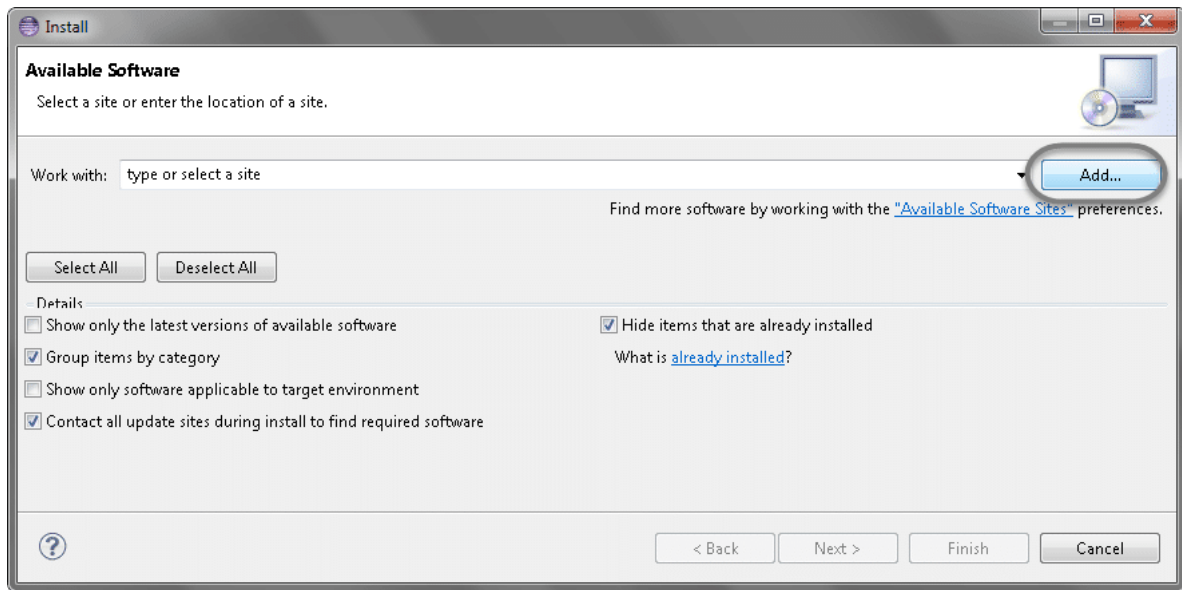6. If your Mac OS system displays information on the Apache Maven version installed, this means Maven is successfully installed on your Mac.

Installing TestNG

1. Launch the Eclipse IDE and from Help menu, click "Install New Software":

2. You will see a dialog window, click "Add" button:

3. Type name as you wish, lets take "TestNG" and type http://beust.com/eclipse/ as location. Click OK:
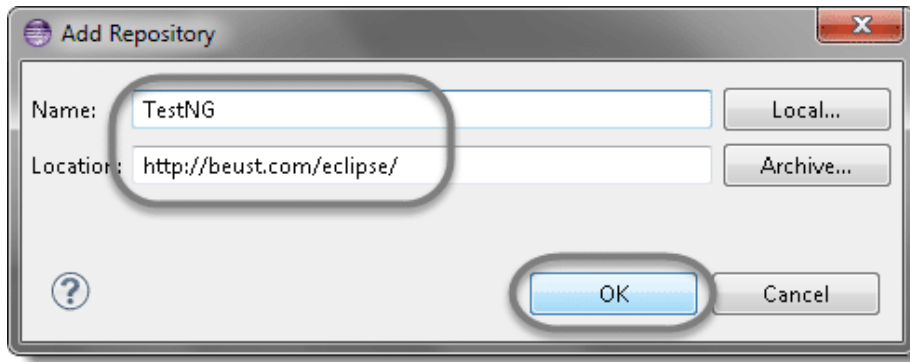
4. You come back to the previous window but this time you must see TestNG option in the available software. Just click "TestNG" and press "Next" button:
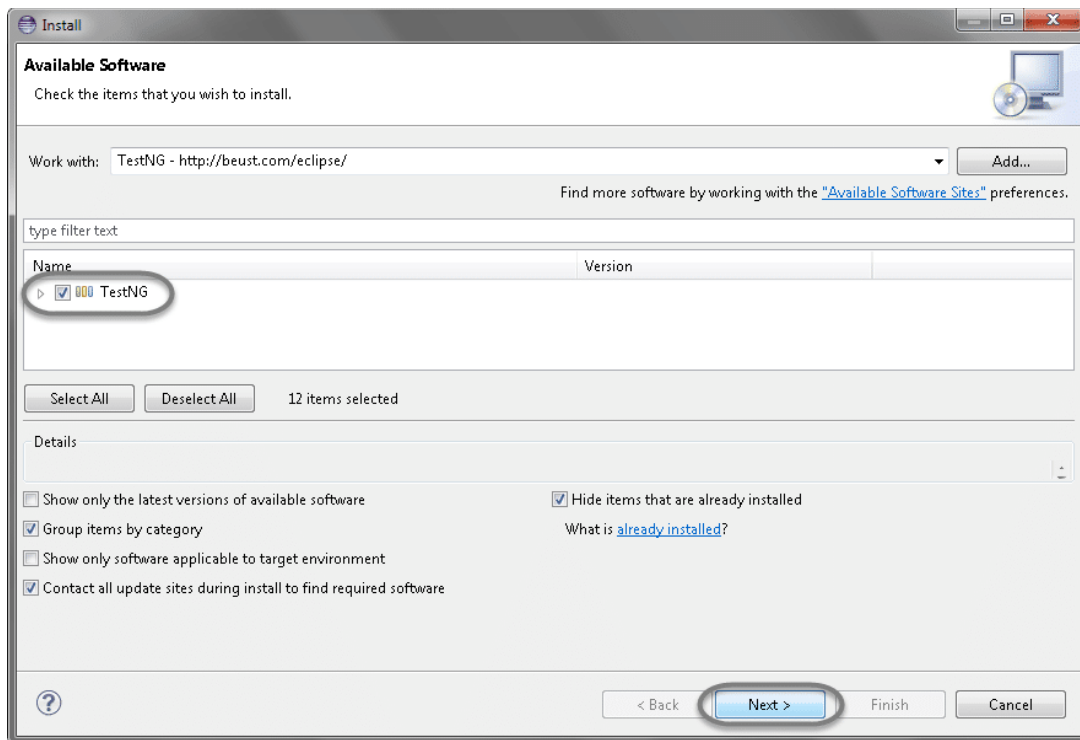
5. Click "I accept the terms of the license agreement" then click Finish.
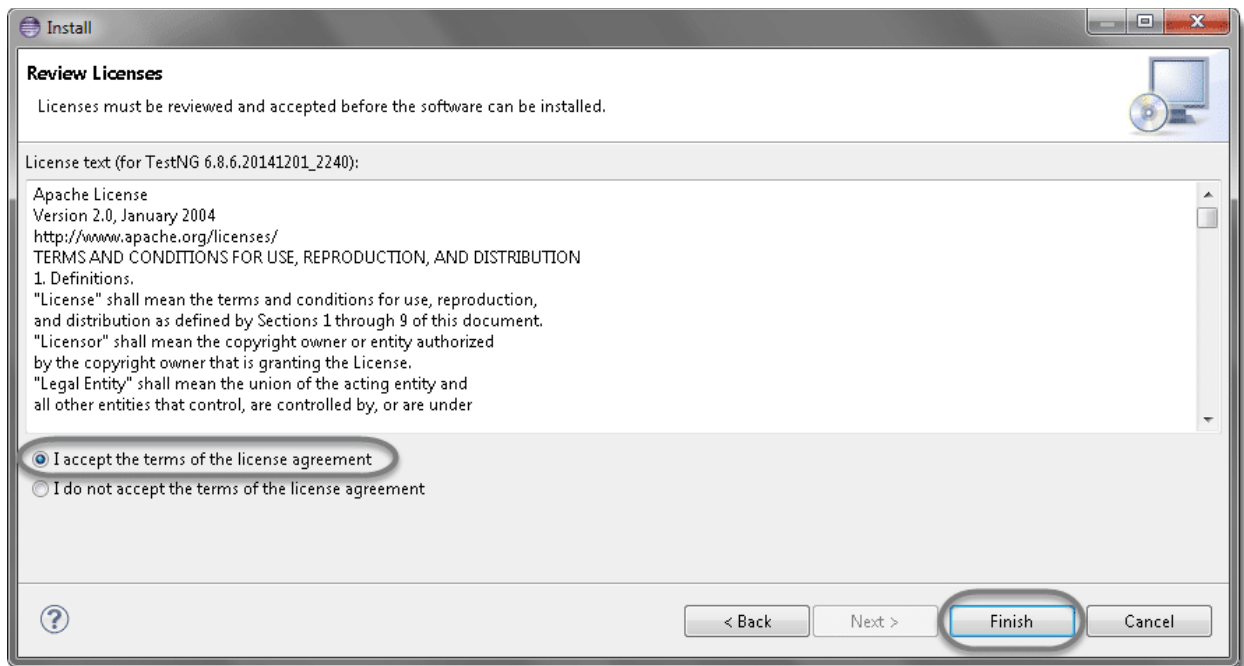
6. You may or may not encounter a Security warning, if in case you do just click OK.
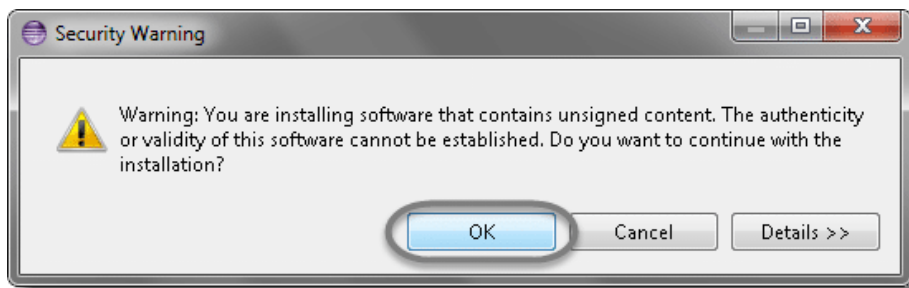
7. Click on "Next" button again on the succeeding dialog box until it prompts you to restart Eclipse.
8. You are all done now, just click on "Yes" button:
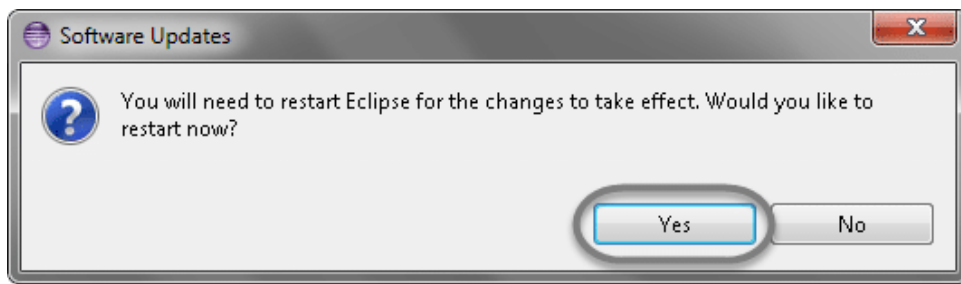
9. Proceed with your workplace:
10. After restart, verify If TestNG was endeed successfully installed. Right click on your project and see if TestNG is displayed in the opened menu:
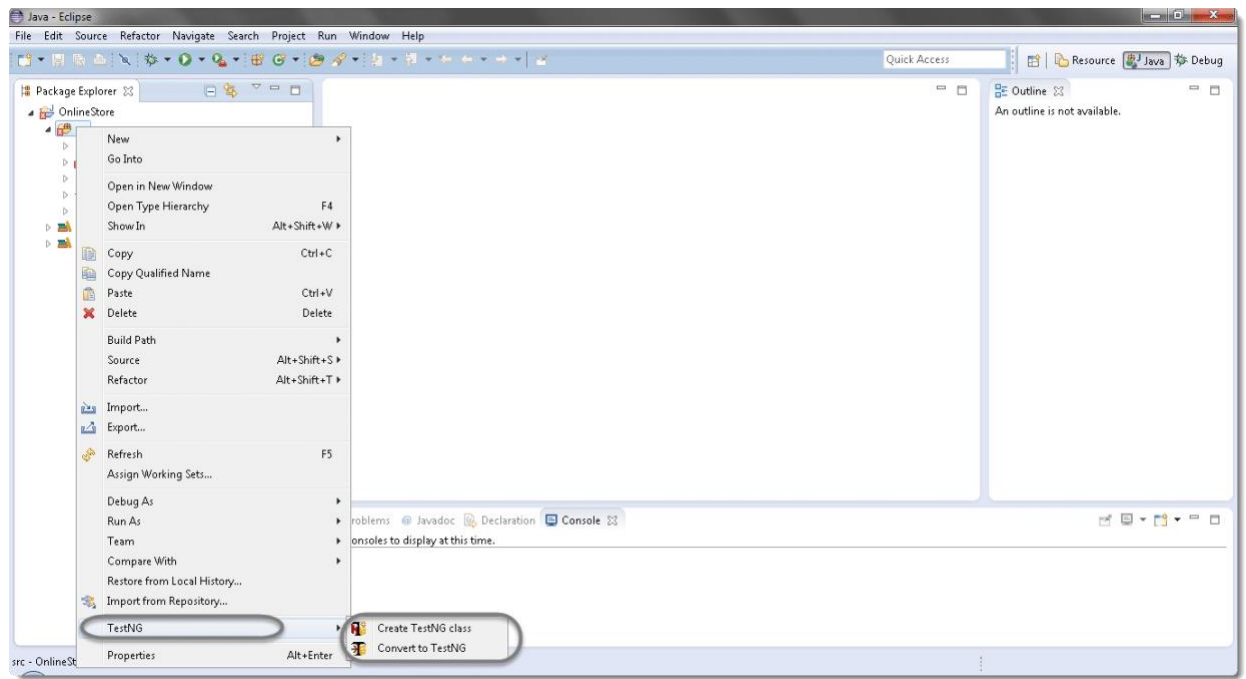
Installing Selenium WebDriver

1. Go to https://docs.seleniumhq.org/download/ and download Java client driver. On that page click on "Download" link of Java client as show in the image below:

Selenium Client & WebDriver Language Bindings

In order to create scripts that interact with the Selenium Server (Selenium RC, Selenium Remote WebDriver) or create local Selenium WebDriver scripts, you need to make use of language-specific client drivers. These languages include both 1.x and 2.x style clients.

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

| Language	Client Version	Release Date			
Java	3.141.59	2018-11-14	Download	Change log	Javadoc
C#	3.14.0	2018-08-02	Download	Change log	API docs
Ruby	3.14.0	2018-08-03	Download	Change log	API docs
Python	3.14.0	2018-08-02	Download	Change log	API docs
Javascript (Node)	4.0.0-alpha.1	2018-01-13	Download	Change log	API docs

2. Save the .zip file to your disk;
3. Once you have downloaded the archive you will need to Extract the zip file, which will create the unzipped Selenium Java folder;
4. Copy the extracted folder into another location of your preference;
5. In Eclipse, right click on TownSend project and select **Properties** -> **Java Build Path**. Then navigate to **Libraries** tab and click **Add External JAR** button;
6. Add all .jar files in your selenium folder, including the ones within your **libs** subdirectory.
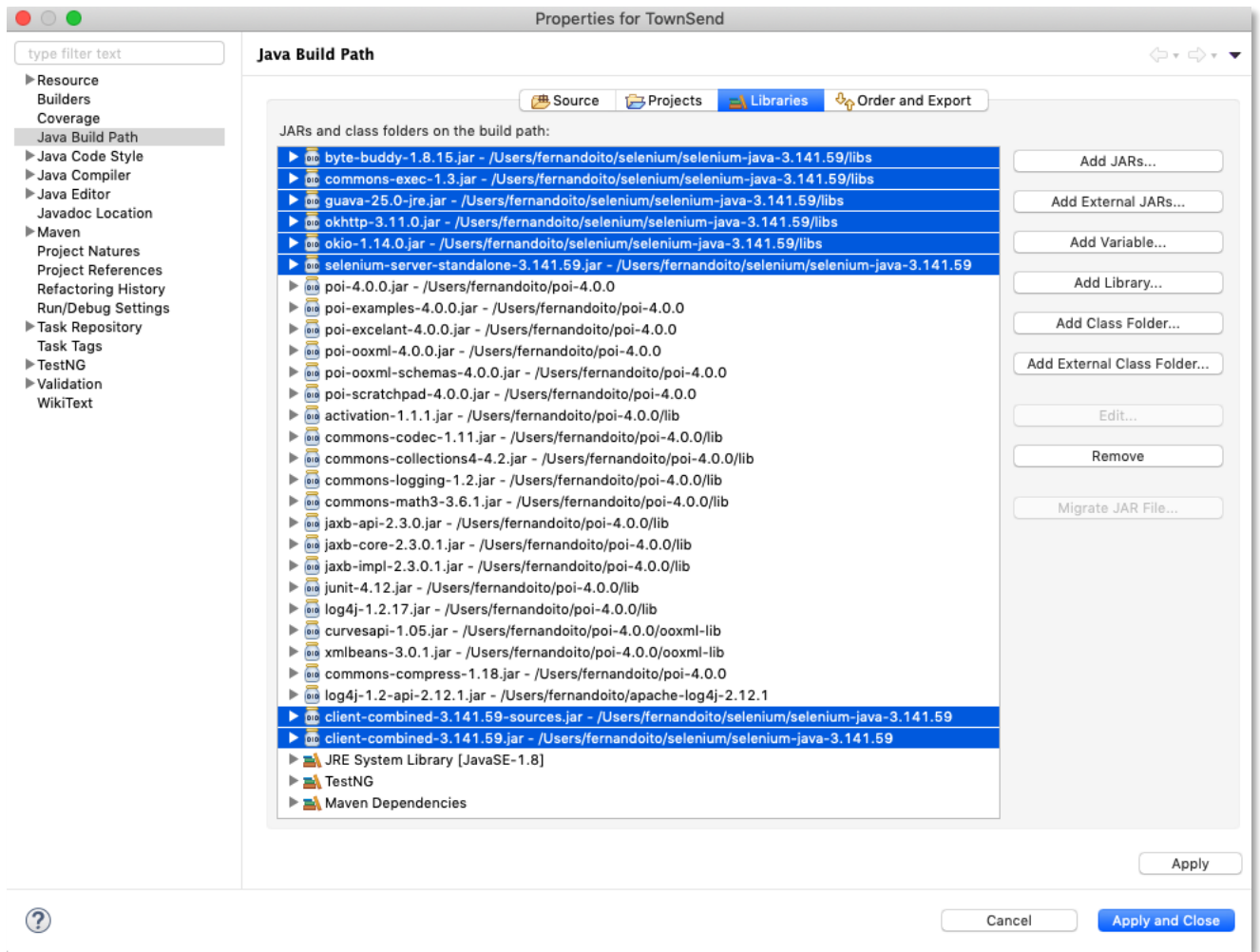
Installing Apache POI

1. Go to https://mvnrepository.com/artifact/org.apache.poi/poi;

	Version	Repository	Usages	Date
4.1.x	4.1.0	Central	58	Apr, 2019
4.0.x	4.0.1	Central	92	Dec, 2018
	4.0.0	Central	65	Aug, 2018

2. Click on the latest version in the list (4.1.0):

Home » org.apache.poi » poi » 4.1.0

Apache POI » 4.1.0

Apache POI - Java API To Access Microsoft Format Files

License	Apache 2.0
Categories	Excel Libraries
Organization	Apache Software Foundation
HomePage	http://poi.apache.org/
Date	(Apr 09, 2019)
Files	jar (2.6 MB) View All
Repositories	Central
Used By	1,247 artifacts

Maven | Gradle | SBT | Ivy | Grape | Leiningen | Buildr

```
<!-- https://mvnrepository.com/artifact/org.apache.poi/poi -->
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi</artifactId>
    <version>4.1.0</version>
</dependency>
```

☑ Include comment with link to declaration

3. Click on "jar" link;
4. Save, extract the file and move the folder into the location of your preference;
5. In Eclipse, right click on TownSend project and select **Properties** -> **Java Build Path**. Then navigate to **Libraries** tab and click **Add External JAR** button;
6. Add all .jar files in your selenium folder, including the ones within your **ooxml-lib** and **lib** subdirectories.

Java Build Path

type filter text

▶ Resource
   Builders
   Coverage
   Java Build Path
▶ Java Code Style
▶ Java Compiler
▶ Java Editor
   Javadoc Location
▶ Maven
   Project Natures
   Project References
   Refactoring History
   Run/Debug Settings
▶ Task Repository
   Task Tags
▶ TestNG
▶ Validation
   WikiText

Source | Projects | Libraries | Order and Export

JARs and class folders on the build path:

▶ byte-buddy-1.8.15.jar - /Users/fernandoito/selenium/selenium-java-3.141.59/libs
▶ commons-exec-1.3.jar - /Users/fernandoito/selenium/selenium-java-3.141.59/libs
▶ guava-25.0-jre.jar - /Users/fernandoito/selenium/selenium-java-3.141.59/libs
▶ okhttp-3.11.0.jar - /Users/fernandoito/selenium/selenium-java-3.141.59/libs
▶ okio-1.14.0.jar - /Users/fernandoito/selenium/selenium-java-3.141.59/libs
▶ selenium-server-standalone-3.141.59.jar - /Users/fernandoito/selenium/selenium-java-3.141.59
▶ poi-4.0.0.jar - /Users/fernandoito/poi-4.0.0
▶ poi-examples-4.0.0.jar - /Users/fernandoito/poi-4.0.0
▶ poi-excelant-4.0.0.jar - /Users/fernandoito/poi-4.0.0
▶ poi-ooxml-4.0.0.jar - /Users/fernandoito/poi-4.0.0
▶ poi-ooxml-schemas-4.0.0.jar - /Users/fernandoito/poi-4.0.0
▶ poi-scratchpad-4.0.0.jar - /Users/fernandoito/poi-4.0.0
▶ activation-1.1.1.jar - /Users/fernandoito/poi-4.0.0/lib
▶ commons-codec-1.11.jar - /Users/fernandoito/poi-4.0.0/lib
▶ commons-collections4-4.2.jar - /Users/fernandoito/poi-4.0.0/lib
▶ commons-logging-1.2.jar - /Users/fernandoito/poi-4.0.0/lib
▶ commons-math3-3.6.1.jar - /Users/fernandoito/poi-4.0.0/lib
▶ jaxb-api-2.3.0.jar - /Users/fernandoito/poi-4.0.0/lib
▶ jaxb-core-2.3.0.1.jar - /Users/fernandoito/poi-4.0.0/lib
▶ jaxb-impl-2.3.0.1.jar - /Users/fernandoito/poi-4.0.0/lib
▶ junit-4.12.jar - /Users/fernandoito/poi-4.0.0/lib
▶ log4j-1.2.17.jar - /Users/fernandoito/poi-4.0.0/lib
▶ curvesapi-1.05.jar - /Users/fernandoito/poi-4.0.0/ooxml-lib
▶ xmlbeans-3.0.1.jar - /Users/fernandoito/poi-4.0.0/ooxml-lib
▶ commons-compress-1.18.jar - /Users/fernandoito/poi-4.0.0
▶ log4j-1.2-api-2.12.1.jar - /Users/fernandoito/apache-log4j-2.12.1
▶ client-combined-3.141.59-sources.jar - /Users/fernandoito/selenium/selenium-java-3.141.59
▶ client-combined-3.141.59.jar - /Users/fernandoito/selenium/selenium-java-3.141.59
▶ JRE System Library [JavaSE-1.8]
▶ TestNG
▶ Maven Dependencies

Add JARs...
Add External JARs...
Add Variable...
Add Library...
Add Class Folder...
Add External Class Folder...
Edit...
Remove
Migrate JAR File...

Apply

Cancel | Apply and Close