
STANDARD OPERATING PROCEDURE

Web Automation Framework Setup

- 1. Purpose:** The purpose of this SOP is to establish consistent guidelines for the effective implementation, execution, and maintenance of the automation testing framework within Apptad Technology Pvt. Ltd. (INDIA).
- 2. Scope:** This SOP applies to the Quality Assurance (QA) department of Apptad Technology Pvt. Ltd. (in India) and outlines the standardized procedures for implementing and managing the automation testing framework within the organization. This includes all team members authorized to execute, maintain, and monitor automated test cases and ensure compliance with testing standards.

Document detail	
Reference Number	SOP-AT-01
Version	0.2
Effective date	05-NOV-2024
Prepared by	Yash Gautam
Reviewed by	Ankur Kumar
Approved by	

Contents

1. Install Java	3
2. Install Eclipse	6
3. Install TestNG.....	7
4. Setup Maven Project.....	8
5. Add POM Dependencies.....	10
6. Create Project Structure	12
7. GitHub link for sample project	14

Install Java

1. **Download Java:** Download the latest stable version of java for windows by following below steps.

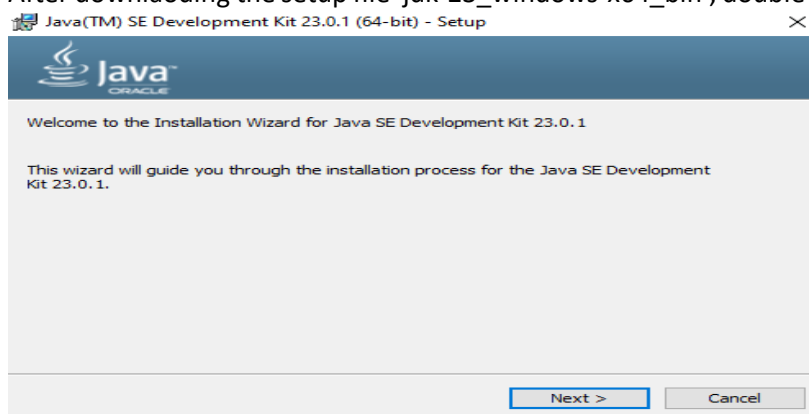
1. Visit the Official Oracle Website- <https://www.oracle.com/java/technologies/downloads/>

2. Click on Windows option and click on download link for installing x64 installer.

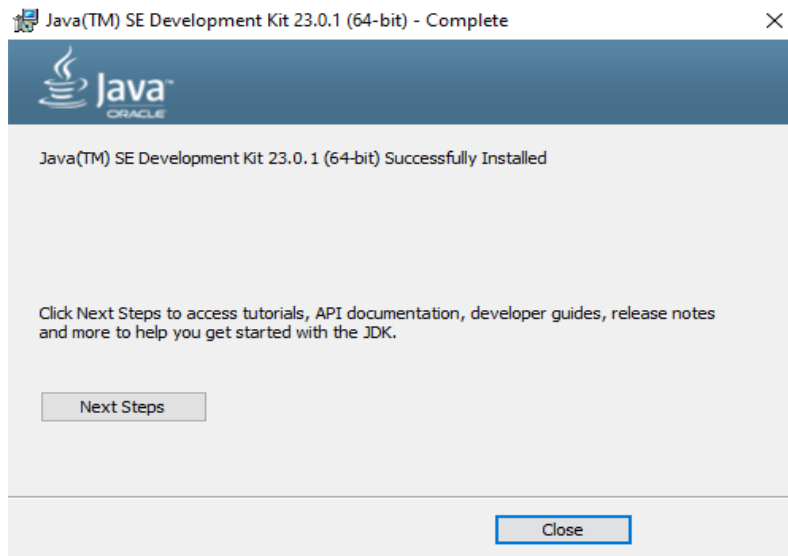
The screenshot shows the Oracle Java Downloads page for JDK 23 on Windows. The browser address bar shows the URL: [oracle.com/java/technologies/downloads/#jdk23-windows](https://www.oracle.com/java/technologies/downloads/#jdk23-windows). The page header includes the Oracle logo and navigation links: Products, Industries, Resources, Customers, Partners, Developers, Company. Below the header, there are tabs for Tools and resources, Java downloads (selected), and Java archive. The main content area states: "JDK 21 is the latest Long-Term Support (LTS) release of the Java SE Platform. Earlier JDK versions are available below." Below this, there are tabs for JDK 23, JDK 21, GraalVM for JDK 23, and GraalVM for JDK 21. The "JDK 23" tab is selected, showing "JDK Development Kit 23.0.1 downloads". A note states: "JDK 23 binaries are free to use in production and free to redistribute, at no cost, under the Oracle No-Fee Terms and Conditions (NFTC). JDK 23 will receive updates under these terms, until March 2025, when it will be superseded by JDK 24." Below this, there are tabs for Linux, macOS, and Windows (selected). A table lists the available download options:

Product/file description	File size	Download
x64 Compressed Archive	228.70 MB	https://download.oracle.com/java/23/latest/jdk-23_windows-x64_bin.zip (sha256)
x64 Installer	205.21 MB	https://download.oracle.com/java/23/latest/jdk-23_windows-x64_bin.exe (sha256)
x64 MSI Installer	203.96 MB	https://download.oracle.com/java/23/latest/jdk-23_windows-x64_bin.msi (sha256)

3. After downloading the setup file 'jdk-23_windows-x64_bin', double click on the setup file.

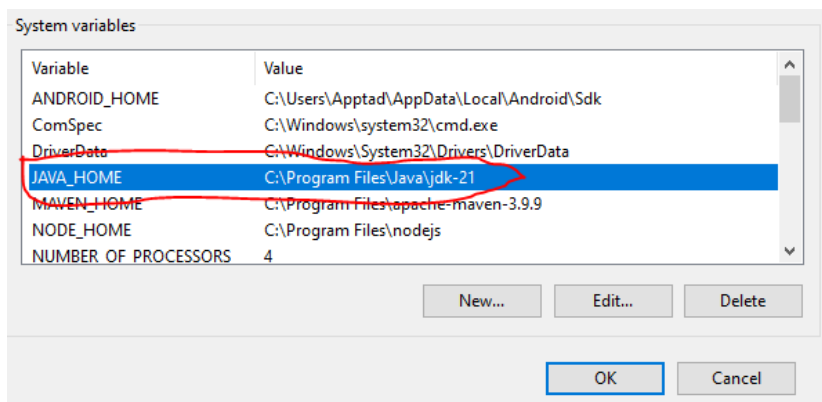


4. Click on Next button and select default installation location and after installation is completed, click on Close button.

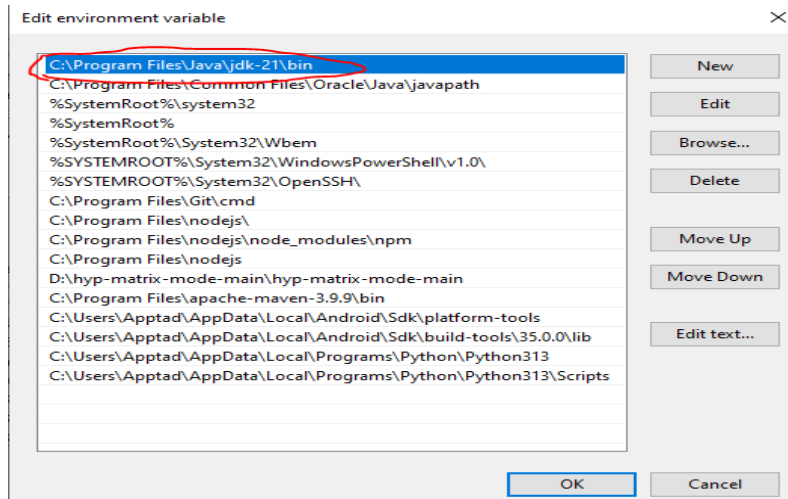


2. Set Environment Variables:

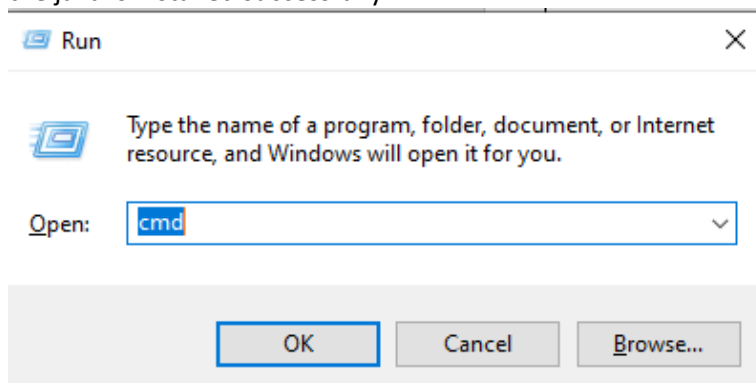
- Add JAVA_HOME to your system environment variables pointing to the JDK installation directory.
- Add %JAVA_HOME%\bin to the PATH variable.



- Edit the Path and add new path up to bin folder of java JDK directory.



- After setting up the environment variable, open the command prompt and type **java -version** command to check if the java is installed successfully.



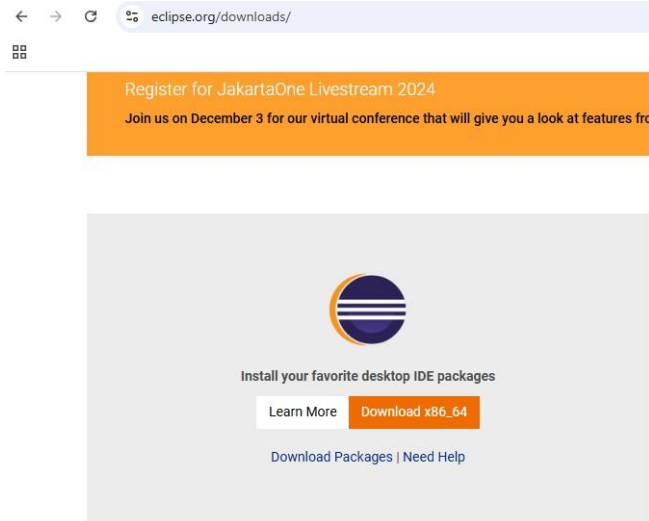
```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19045.5011]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Apptad>java -version
java version "21.0.4" 2024-07-16 LTS
Java(TM) SE Runtime Environment (build 21.0.4+8-LTS-274)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.4+8-LTS-274, mixed mode, sharing)
```

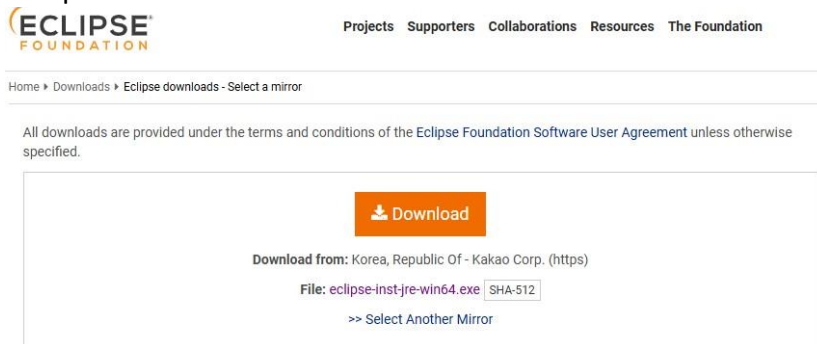
Install Eclipse

1. Download and install eclipse IDE for windows by following below steps,

1. Visit the eclipse official site - <https://www.eclipse.org/downloads/>



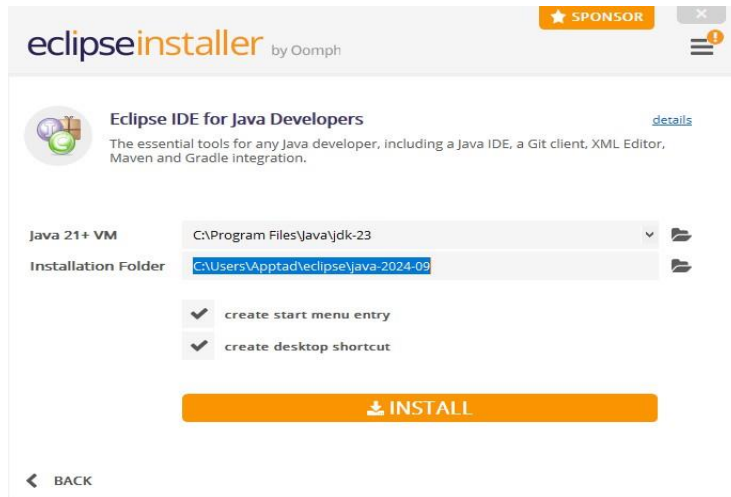
2. Click on Download x86_64 button and follow the steps and download eclipse-inst-jre-win64.exe setup file.



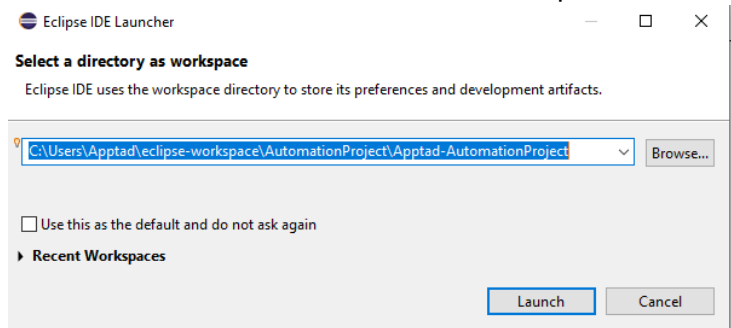
3. After successfully downloading the setup file, double click on the setup file to install.



4. Select Eclipse IDE for Java Developers option and select the default installation location and click on Install button.

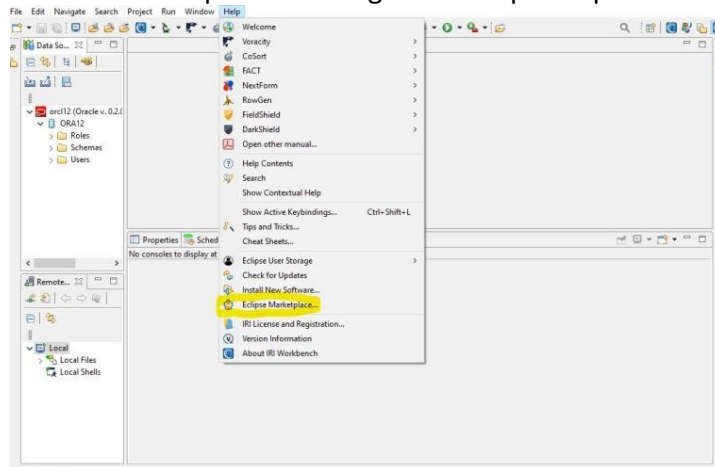


5. After successful installation launch the eclipse and select workspace directory.

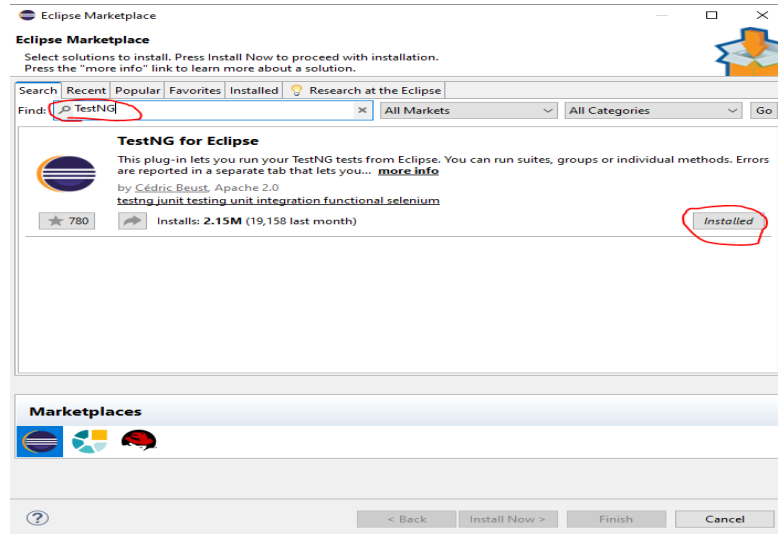


Install TestNG

1. Install TestNG by following below steps,
 1. Launch the eclipse and navigate to Help->Eclipse Marketplace... menu.

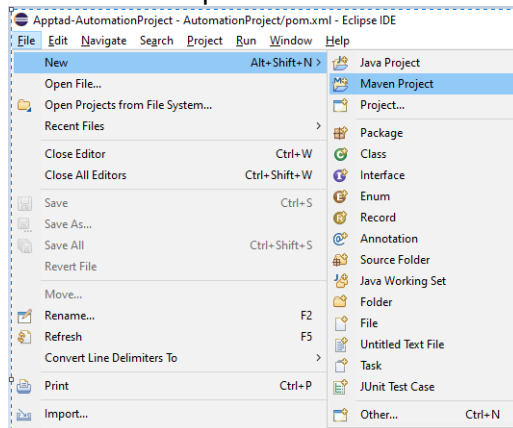


2. Search TestNG and click on install button and follow the steps to install.

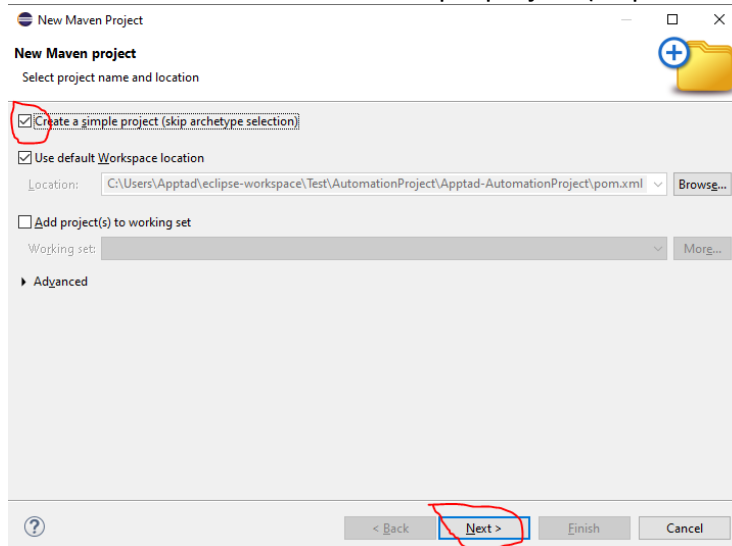


Setup Maven Project

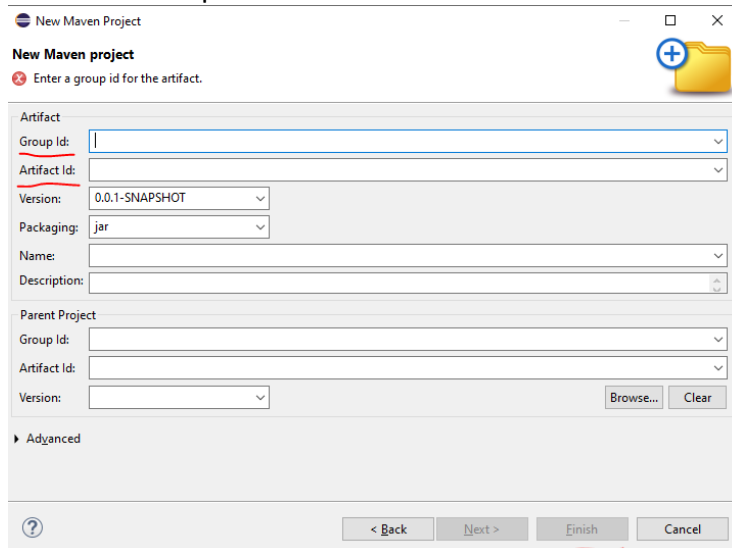
1. Launch the eclipse and click on File->New->Maven Project menu.



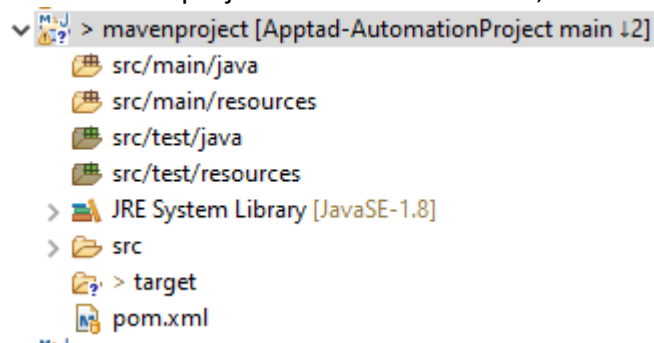
2. Check the checkbox 'Create a simple project (Skip archetype selection)'.



3. Enter the Group ID and Artifact ID and click on Finish button.



4. You will see a project structure like below,



Add POM Dependencies

Below are the dependencies in pom.xml file under <dependency> tag which are necessary to create a base structure for automation framework.

Maven Repository:

Selenium: Use the latest version or specific version depending upon the project requirement.

<https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java>

Home » org.seleniumhq.selenium » selenium-java » 4.26.0



Selenium Java » 4.26.0

Selenium provides support for the automation of web browsers. It provides extensions to emulate user interaction with browsers, a distribution server for scaling browser allocation, and the infrastructure for implementations of the W3C WebDriver specification.

License	Apache 2.0
Categories	Web Testing
Tags	quality selenium testing web
HomePage	https://selenium.dev/
Date	Oct 30, 2024
Files	pom (4 KB) jar (545 bytes) View All
Repositories	Central
Ranking	#291 in MvnRepository (See Top Artifacts) #1 in Web Testing
Used By	1,818 artifacts

Maven [Gradle](#) [Gradle \(Short\)](#) [Gradle \(Kotlin\)](#) [SBT](#) [Ivy](#) [Grape](#) [Leiningen](#) [Buildr](#)

```
<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
<dependency>
  <groupId>org.seleniumhq.selenium</groupId>
  <artifactId>selenium-java</artifactId>
  <version>4.26.0</version>
</dependency>
```

☒ Include comment with link to declaration

```
mavenproject/pom.xml X
https://maven.apache.org/xsd/maven-4.0.0.xsd (xsi:schemaLocation)
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
5   <modelVersion>4.0.0</modelVersion>
6   <groupId>com</groupId>
7   <artifactId>mavenproject</artifactId>
8   <version>0.0.1-SNAPSHOT</version>
9   <dependencies>
10    <!--
11    https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
12    <dependency>
13      <groupId>org.seleniumhq.selenium</groupId>
14      <artifactId>selenium-java</artifactId>
15      <version>4.26.0</version>
16    </dependency>
17  </dependencies>
</project>
```

TestNG: Ignore this dependency if you have followed **Install TestNG** step.

<https://mvnrepository.com/artifact/org.testng/testng>

WebDriverManger:

<https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager>

WebDriverManager » 5.9.2
Automated driver management and other helper features for Selenium WebDriver in Java

License: [Apache 2.0](#)

Categories: [Web Browser Automation](#)

Tags: [github](#) [browser](#) [webdriver](#) [automation](#) [web](#)

HomePage: <https://bonigarcia.dev/webdrivermanager/>

Date: Jul 25, 2024

Files: [pom \(18 KB\)](#) [jar \(877 KB\)](#) [View All](#)

Repositories: [Central](#)

Ranking: #1059 in MvnRepository (See Top Artifacts)
#1 in Web Browser Automation

Used By: 511 artifacts

Vulnerabilities: Vulnerabilities from dependencies:
[CVE-2023-6378](#)

Maven [Gradle](#) [Gradle \(Short\)](#) [Gradle \(Kotlin\)](#) [SBT](#) [Ivy](#) [Grape](#) [Leiningen](#) [Buildr](#)

```
<!-- https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager -->
<dependency>
  <groupId>io.github.bonigarcia</groupId>
  <artifactId>webdrivermanager</artifactId>
  <version>5.9.2</version>
</dependency>
```

☒ Include comment with link to declaration

mavenproject/pom.xml X

```
https://maven.apache.org/xsd/maven-4.0.0.xsd (xsi:schemaLocation)
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
5   <modelVersion>4.0.0</modelVersion>
6   <groupId>com</groupId>
7   <artifactId>mavenproject</artifactId>
8   <version>0.0.1-SNAPSHOT</version>
9   <dependencies>
10    <!--
11    https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
12    <dependency>
13      <groupId>org.seleniumhq.selenium</groupId>
14      <artifactId>selenium-java</artifactId>
15      <version>4.26.0</version>
16    </dependency>
17    <!--
18    https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager -->
19    <dependency>
20      <groupId>io.github.bonigarcia</groupId>
21      <artifactId>webdrivermanager</artifactId>
22      <version>5.9.2</version>
23    </dependency>
24  </dependencies>
25 </project>
```

Apache POI:

<https://mvnrepository.com/artifact/org.apache.poi/poi>

Apache xmlbeans:

<https://mvnrepository.com/artifact/org.apache.xmlbeans/xmlbeans>

Apache commons:

<https://mvnrepository.com/artifact/org.apache.commons/commons-collections4>

commons io:

<https://mvnrepository.com/artifact/commons-io/commons-io>

Log4j:

<https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-core>

ExtentReports:

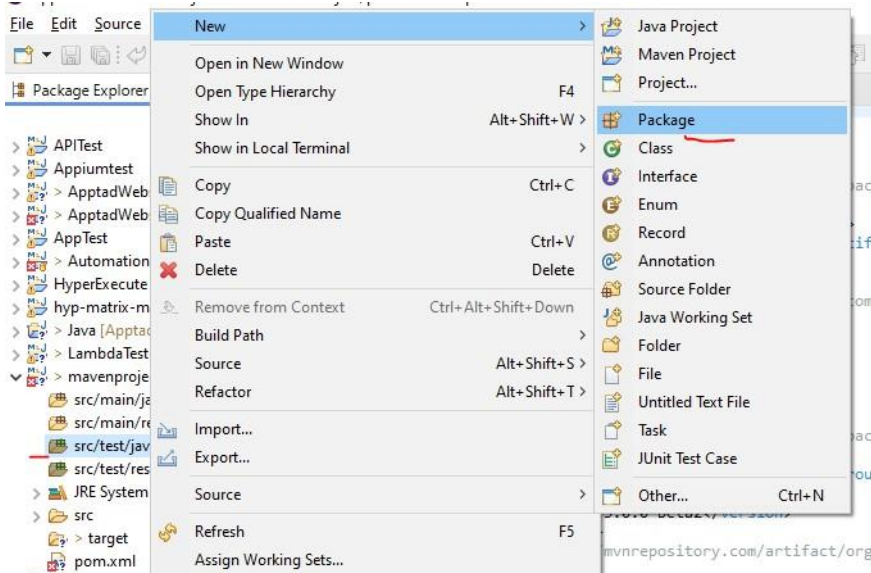
<https://mvnrepository.com/artifact/com.aventstack/extentreports>

Create Project Structure

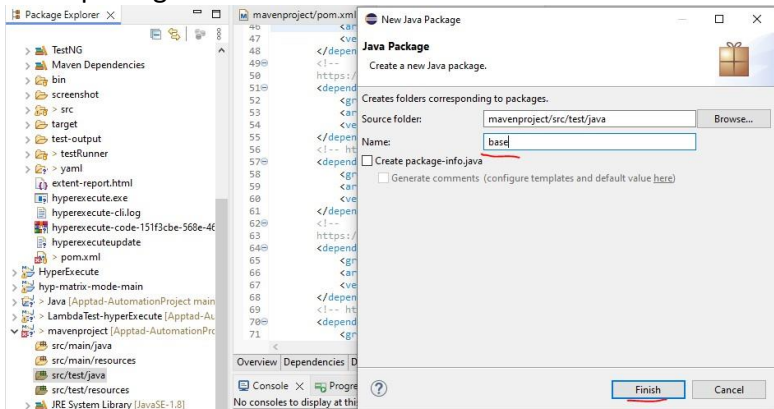
This is a basic folder/project structure for setting up the framework. It may be different depending upon the project need and requirements.

Steps:

1. Right click on **src/test/java** and navigate to new->Package.



2. Enter package name as **base** and click on Finish button.



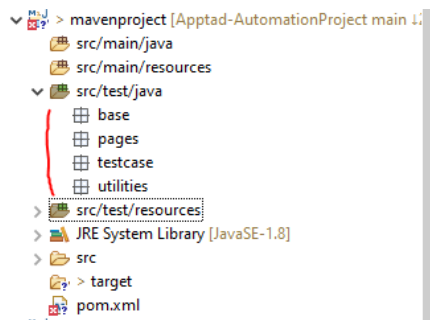
base- The base package will contain the essential classes and configurations required across tests. E.g. drive initialization methods, setup and teardown methods, configuration methods to read external data file locations etc.

3. Similarly, create packages under **src/test/java** as below,

testcase- The testcase package will contain all the automated test cases/scripts.

pages- The pages package will contain or implement Page Object Model (POM) pattern, where each page of the application has a corresponding class. Each page class represents a specific webpage.

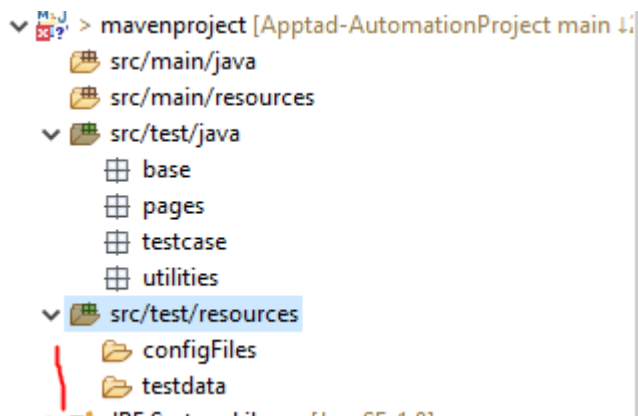
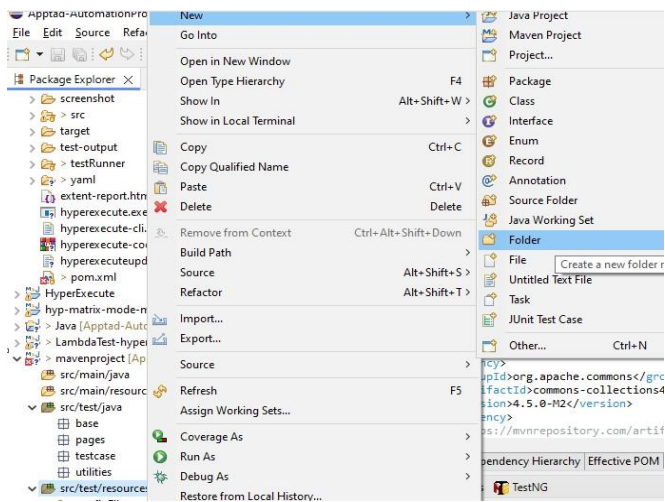
utilities- The utilities package includes utility classes and methods that can be reused across tests. E.g. class which will screenshot for failure test cases, read external file like excel to provide data in test scripts.



4. Under **src/java/resource** folder create two new folders named,

1. **configFiles** – This will have properties files which will have web element locators and test environment data like testURLs, browser etc.

2. **testData**- This will have .xlsx file which will contain all the test data, test case wise which will be provided in test scripts.



GitHub link for sample project-

<https://github.com/ankurA987123/Apptad-AutomationProject.git>