

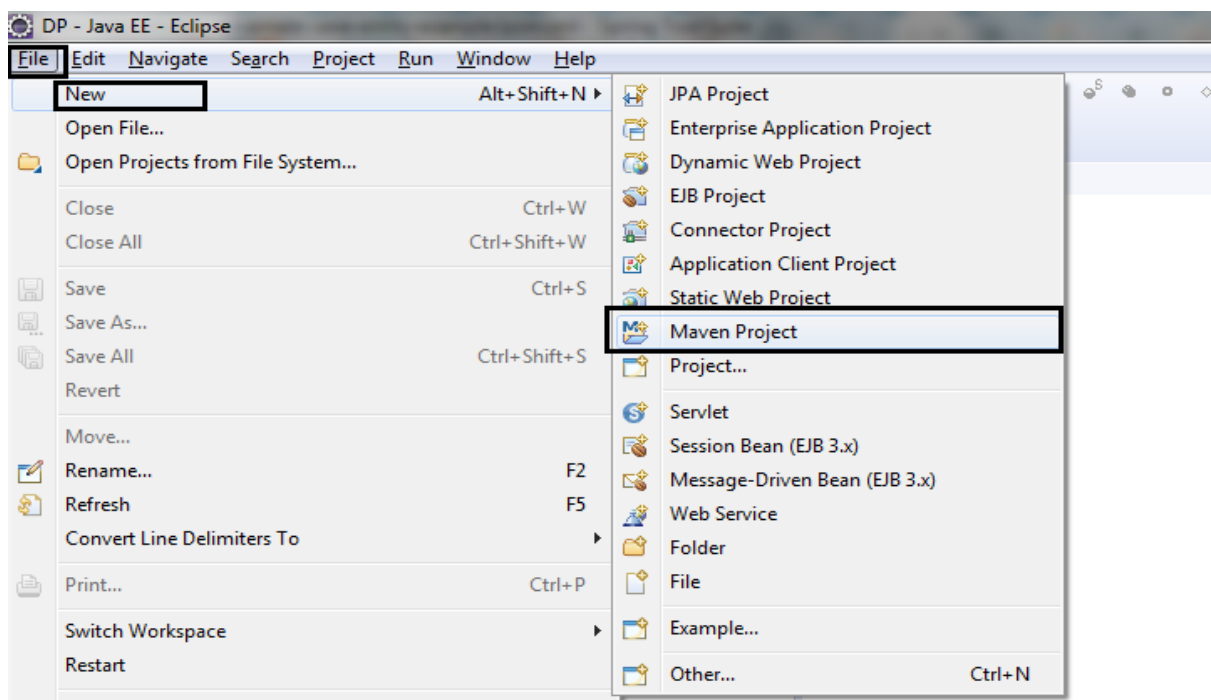
## 13. Integrate Selenium with Maven

### Step 1: Installing Maven in Eclipse

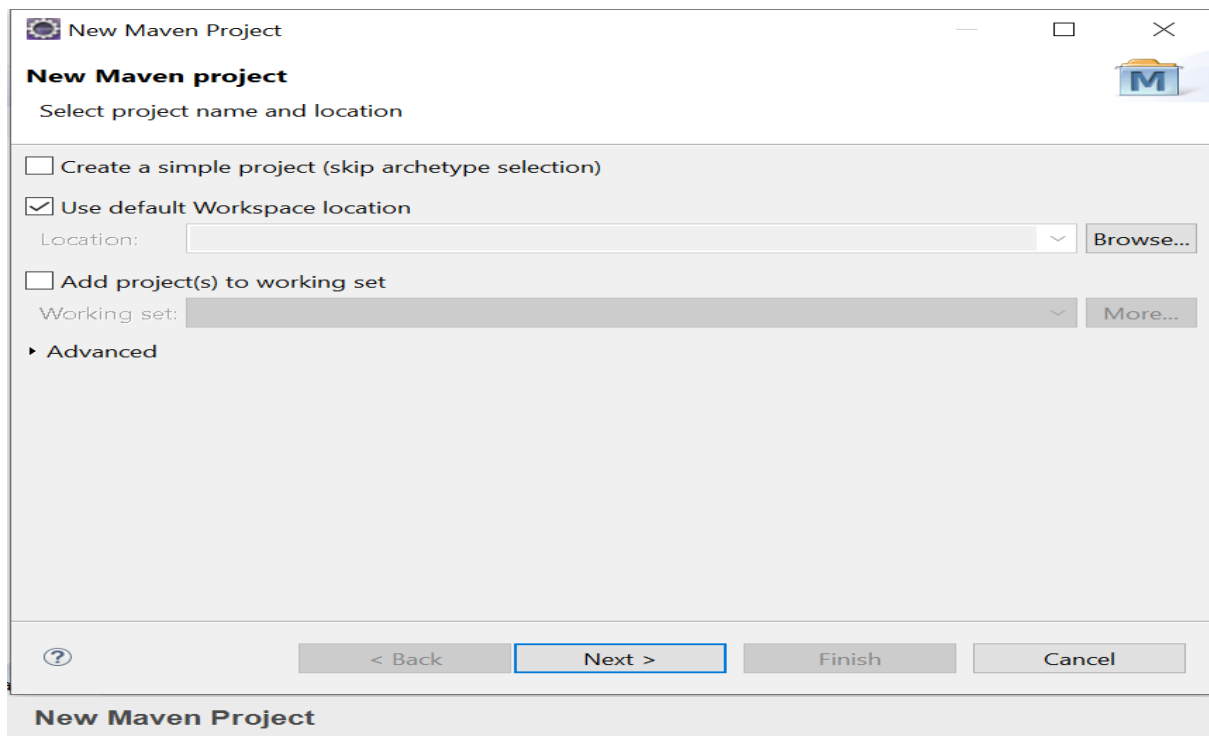
- a. Maven integration for Eclipse is already installed in your Practice lab. Refer to QA to QE lab guide for Phase 2 for more information.

### Step 2: Creating a Maven project

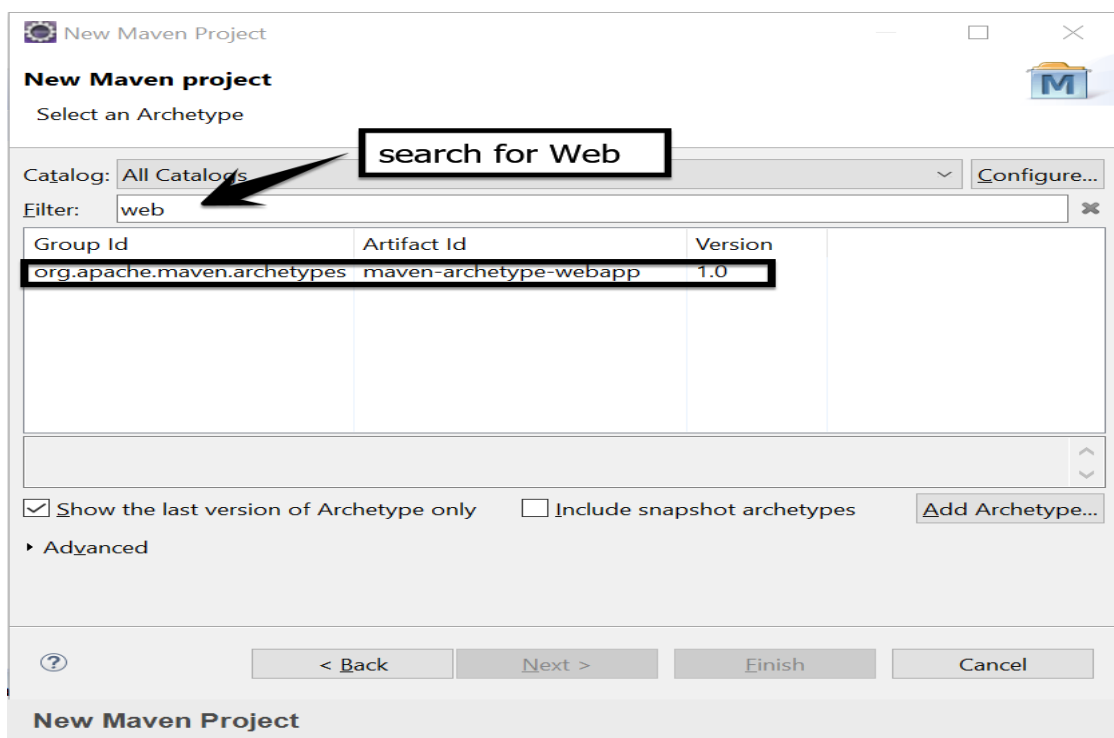
- a. Open Eclipse.
- b. Click on File->New->Maven project.



- c. Select the default workspace location.
- d. Click on **Next**.



e. Select the Maven archetype as maven-archetype-webapps and click on **Next**.



f. Enter the Group ID and Artifact ID and click on **Finish**.

**New Maven Project**

Specify Archetype parameters

Group Id:

Artifact Id:

Version:

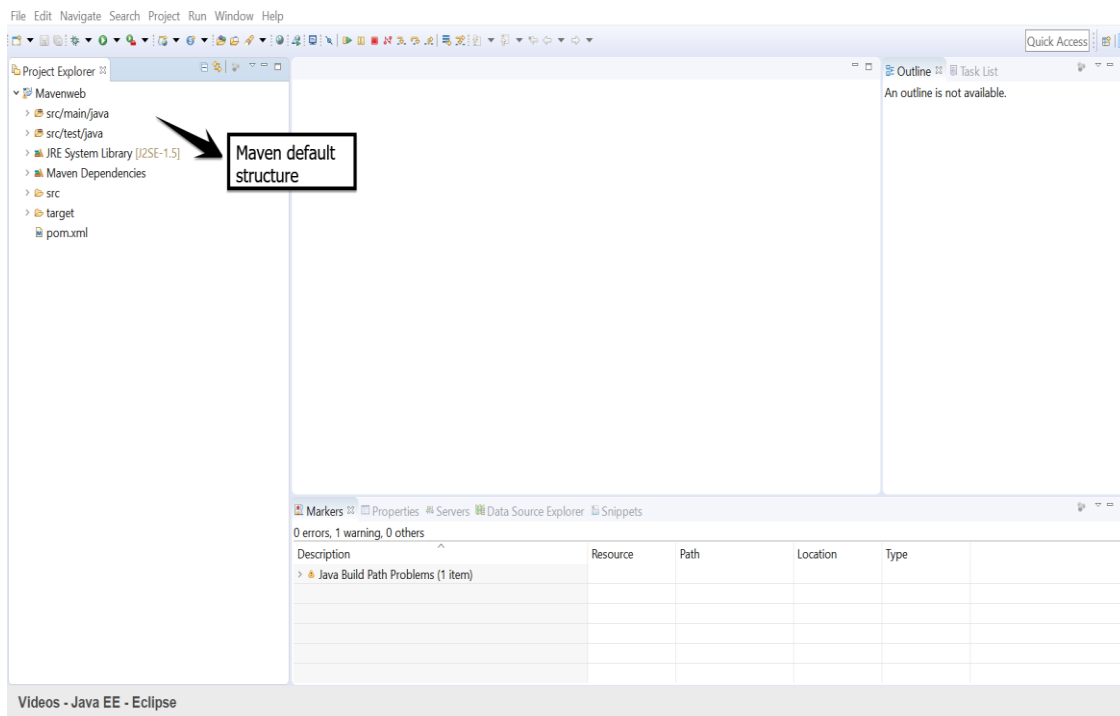
Package:

Properties available from archetype:

Name	Value

► Advanced

g. You are all set. The new Maven structure will look like the below screenshot:



### Step 3: Adding Maven Dependency

- If we are using the Selenium web driver, then we need to specify the Selenium web driver dependencies to our pom.xml.
  - a. Go to Maven Repository:  
<https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java/3.14.0>
  - b. Copy the dependency.



- c. Go to Eclipse.
- d. Open the Pom.xml file from the project.
- e. Paste the dependency in the Pom.xml file:

```
<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
<dependency>
  <groupId>org.seleniumhq.selenium</groupId>
  <artifactId>selenium-java</artifactId>
  <version>3.14.0</version>
</dependency>
```

- If we are using TestNG, then we need to specify the TestNG dependencies to our pom.xml
  - a. Go to the Maven Repository:  
<https://mvnrepository.com/artifact/org.testng/testng/6.8>
  - b. Copy the dependency.

← → ↻ <https://mvnrepository.com/artifact/org.testng/testng/6.8>

**Popular Categories**

- Aspect Oriented
- Actor Frameworks
- Application Metrics
- Build Tools
- Bytecode Libraries
- Command Line Parsers
- Cache Implementations
- Cloud Computing
- Code Analyzers
- Collections
- Configuration Libraries
- Core Utilities
- Date and Time Utilities
- Dependency Injection
- Embedded SQL Databases
- HTML Parsers
- HTTP Clients
- I/O Utilities

<b>HomePage</b>	<a href="http://testng.org">http://testng.org</a>
<b>Date</b>	(Sep 10, 2012)
<b>Files</b>	<a href="#">pom (7 KB)</a> <a href="#">jar (794 KB)</a> <a href="#">View All</a>
<b>Repositories</b>	<a href="#">Central</a> <a href="#">IBiblio</a> <a href="#">Sonatype</a>
<b>Used By</b>	<b>7,943 artifacts</b>

**Note:** There is a new version for this artifact

**New Version** [7.0.0-beta7](#)

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

```
<!-- https://mvnrepository.com/artifact/org.testng/testng -->
<dependency>
  <groupId>org.testng</groupId>
  <artifactId>testng</artifactId>
  <version>6.8</version>
  <scope>test</scope>
</dependency>
```

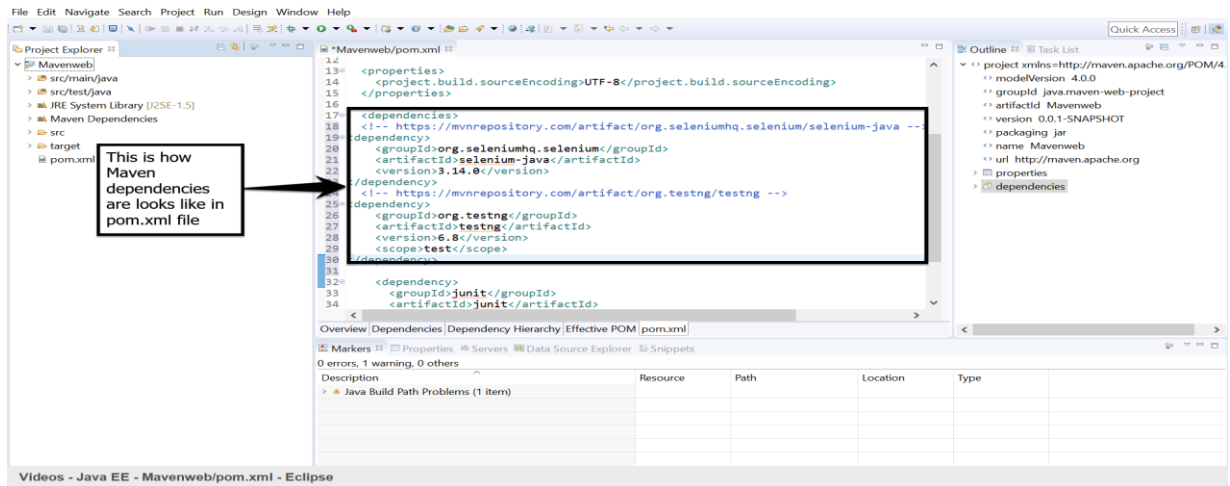
☒ Include comment with link to declaration

Google Chrome Version 74.0.3729.108

- Go to Eclipse.
- Open the Pom.xml file from the project.
- Paste the dependency in the Pom.xml file:

```
<!-- https://mvnrepository.com/artifact/org.testng/testng -->
<dependency>
  <groupId>org.testng</groupId>
  <artifactId>testng</artifactId>
  <version>6.8</version>
  <scope>test</scope>
</dependency>
```

- This is what the Maven dependencies will look like in the pom.xml file.



#### Step 4: Pushing the code to your GitHub repositories

Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add .

Commit the changes using the following command:

git commit . -m "Changes have been committed."

Push the files to the folder you initially created using the following command:

git push -u origin master