11. Demonstrate how Apache POI is configured in Selenium.

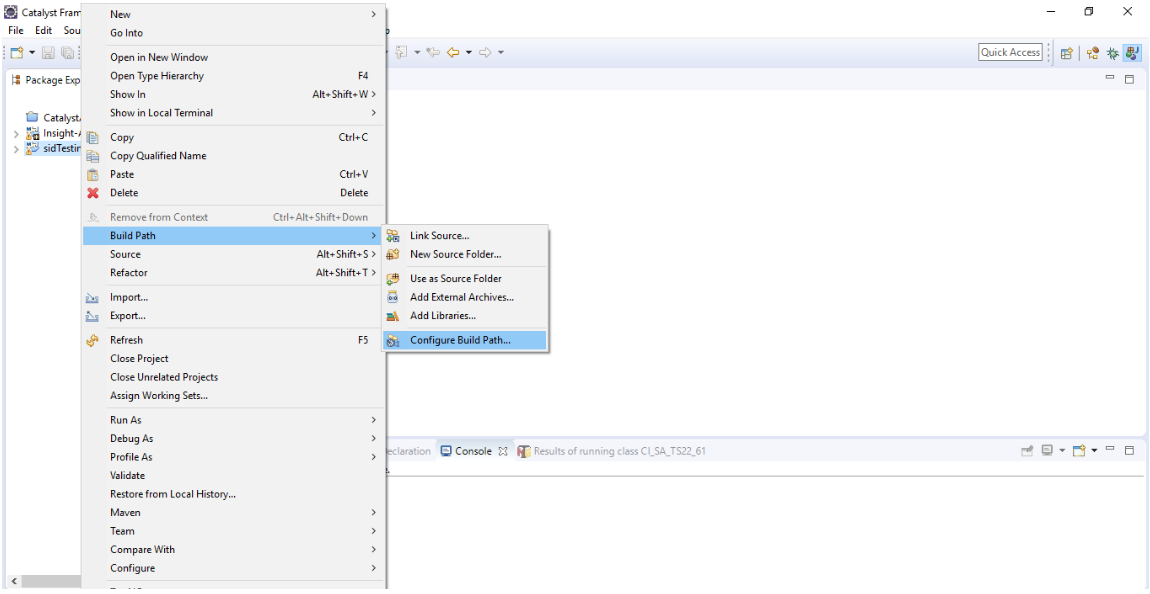
Step 1: Adding external jar files to the Selenium Project in Eclipse

a. Apache Poi is already installed in your Practice lab. Refer to QA to QE lab guide for Phase 2 for more information about its directory details.

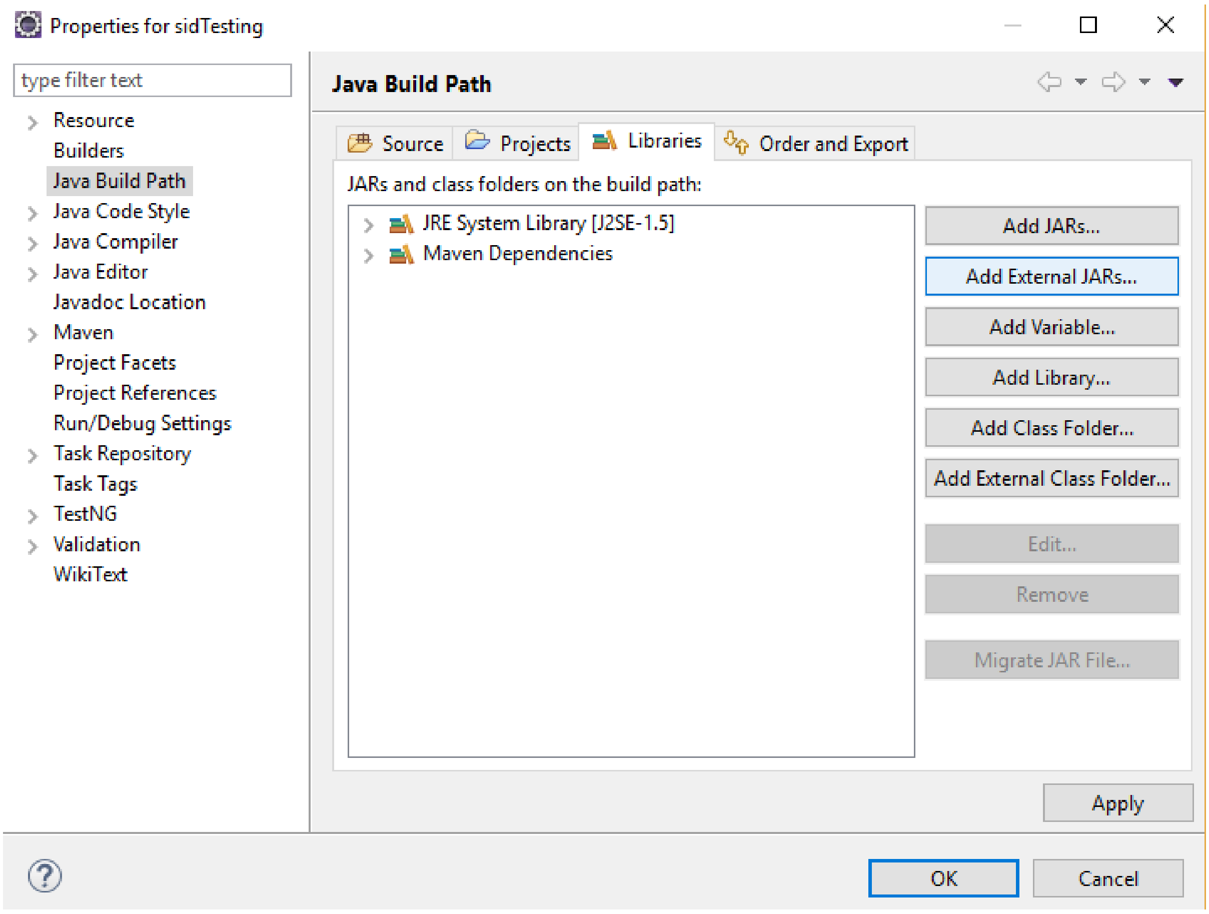
b. Go to Eclipse.

c. Right-click on your project.

d. Navigate to **Build Path Configure Build Path.**



e. Click on **the Libraries** tab and click on **Add External JARs**.



f. Navigate to the folder where Apache POI jar file is stored.

g. Select the executable JAR files from the lib folder and click on **Open**.

h. Repeat steps e.

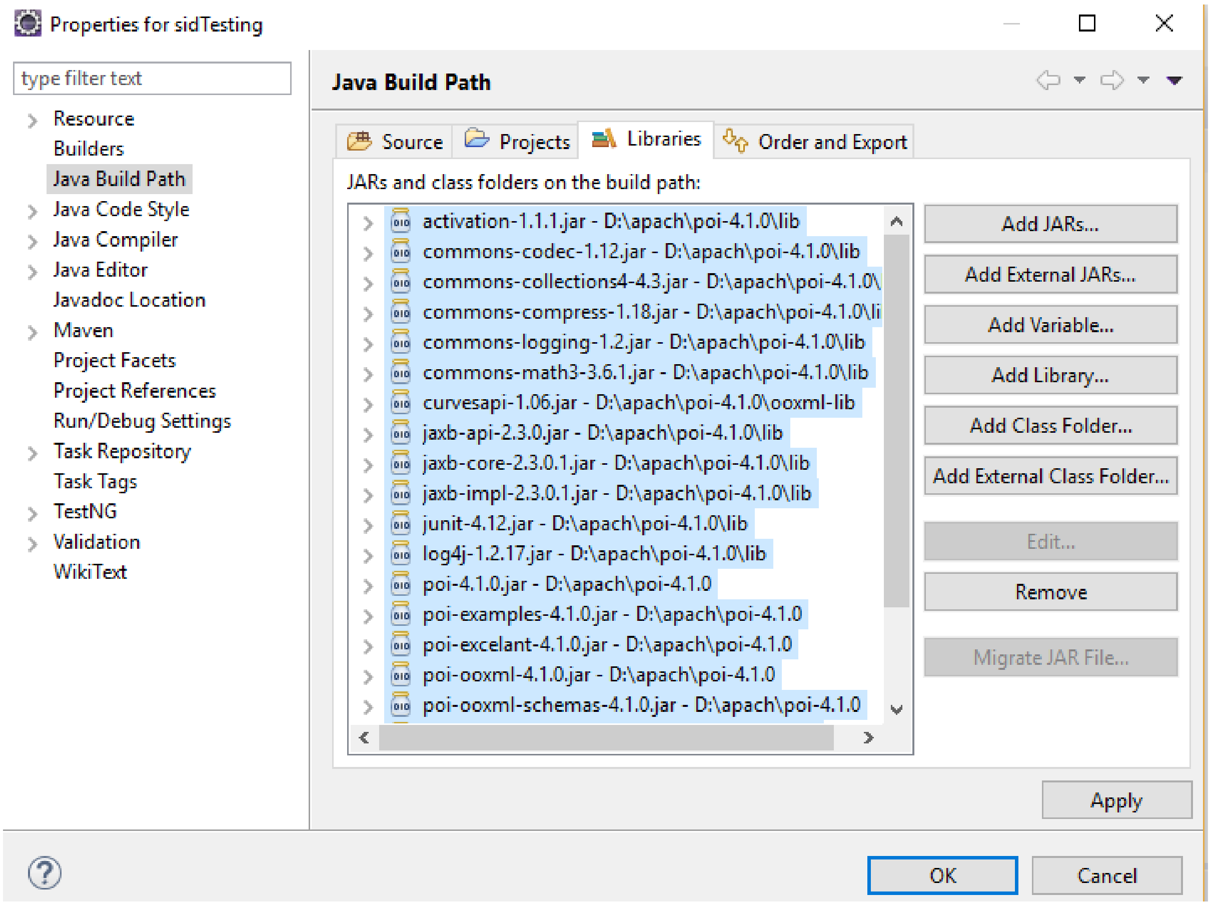
i. Select the executable JAR files from the ooxml-lib folder and click on **Open**.

j. Repeat the steps e.

k. Select the remaining executable JAR files from the folder and click on **Open**.

l. The following snapshot shows that all the JARs are included in the Library list.

m. Click on **Apply** and then on the **OK** button.



Java Build Path

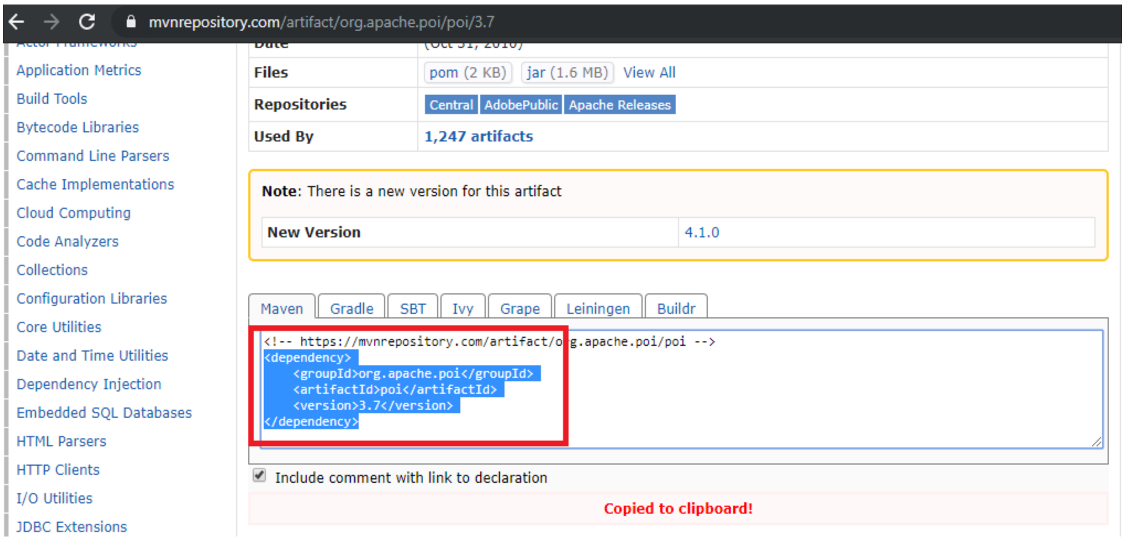
JARs and class folders on the build path:

- activation-1.1.1.jar - D:\apach\poi-4.1.0\lib
- commons-codec-1.12.jar - D:\apach\poi-4.1.0\lib
- commons-collections4-4.3.jar - D:\apach\poi-4.1.0\
- commons-compress-1.18.jar - D:\apach\poi-4.1.0\li
- commons-logging-1.2.jar - D:\apach\poi-4.1.0\lib
- commons-math3-3.6.1.jar - D:\apach\poi-4.1.0\lib
- curvesapi-1.06.jar - D:\apach\poi-4.1.0\ooxml-lib
- jaxb-api-2.3.0.jar - D:\apach\poi-4.1.0\lib
- jaxb-core-2.3.0.1.jar - D:\apach\poi-4.1.0\lib
- jaxb-impl-2.3.0.1.jar - D:\apach\poi-4.1.0\lib
- junit-4.12.jar - D:\apach\poi-4.1.0\lib
- log4j-1.2.17.jar - D:\apach\poi-4.1.0\lib
- poi-4.1.0.jar - D:\apach\poi-4.1.0
- poi-examples-4.1.0.jar - D:\apach\poi-4.1.0
- poi-excelant-4.1.0.jar - D:\apach\poi-4.1.0
- poi-ooxml-4.1.0.jar - D:\apach\poi-4.1.0
- poi-ooxml-schemas-4.1.0.jar - D:\apach\poi-4.1.0

Step 2: Adding Maven Dependency

a. Go to the Maven Repository: https://mvnrepository.com/artifact/org.apache.poi/poi/3.7

b. Copy the dependency.



Files	pom (2 KB) jar (1.6 MB) View All
Repositories	Central AdobePublic Apache Releases
Used By	1,247 artifacts

Note: There is a new version for this artifact

New Version	4.1.0

```
<!-- https://mvnrepository.com/artifact/org.apache.poi/poi -->
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi</artifactId>
    <version>3.7</version>
</dependency>
```

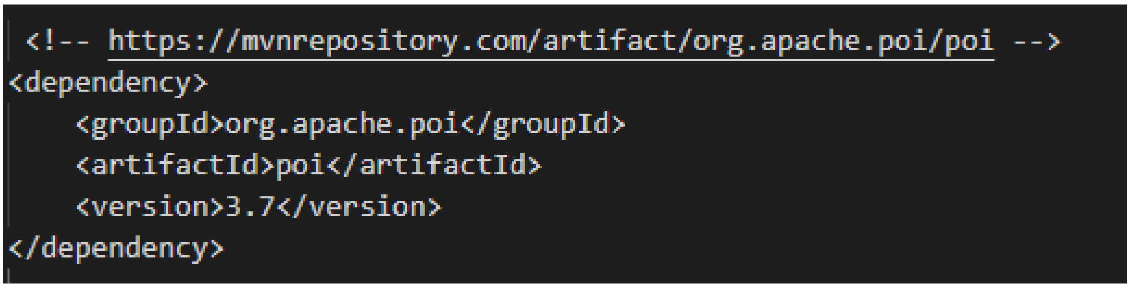
☑ Include comment with link to declaration

Copied to clipboard!

c. Go to Eclipse.

d. Open the Pom.xml file from the project.

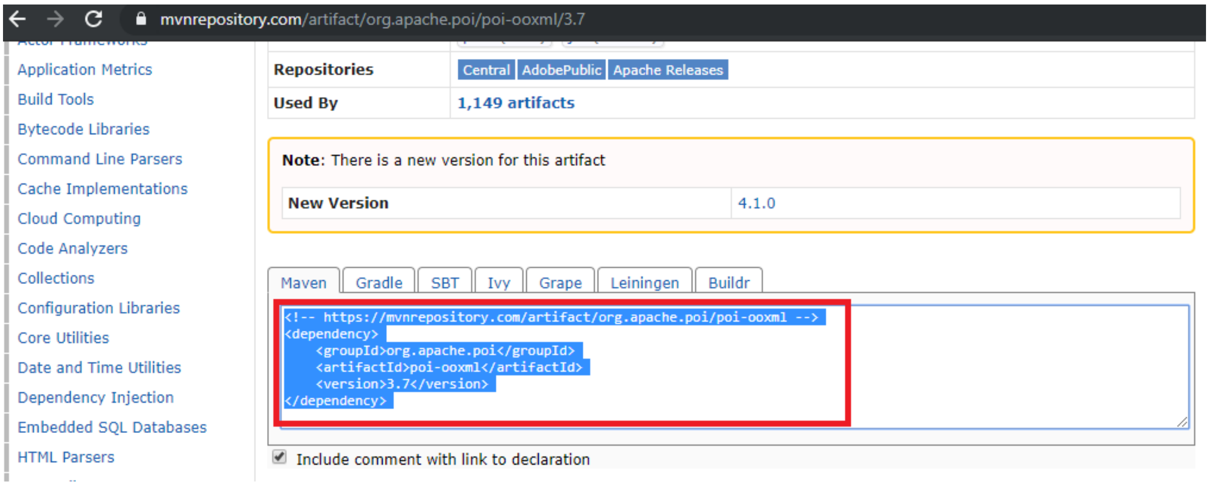
e. Copy the dependency in the Pom.xml file:



```
<!-- https://mvnrepository.com/artifact/org.apache.poi/poi -->
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi</artifactId>
    <version>3.7</version>
</dependency>
```

f. Go to the Maven Repository: https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml/3.7

g. Copy the dependency.



Repositories	Central AdobePublic Apache Releases
Used By	1,149 artifacts

Note: There is a new version for this artifact

New Version	4.1.0

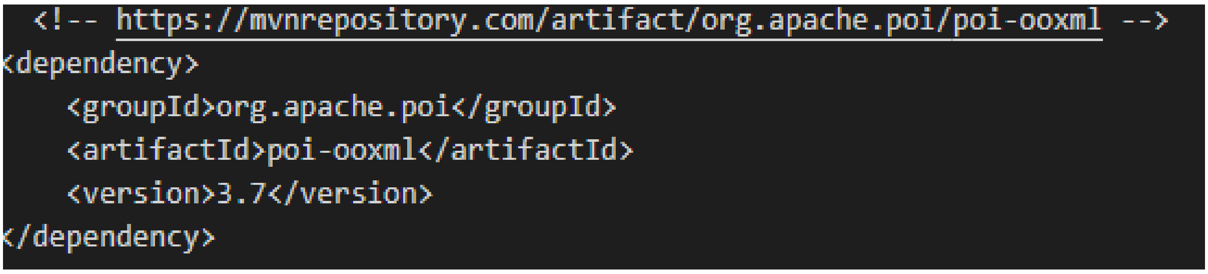
```
<!-- https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml -->
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi-ooxml</artifactId>
    <version>3.7</version>
</dependency>
```

☑ Include comment with link to declaration

h. Go to Eclipse.

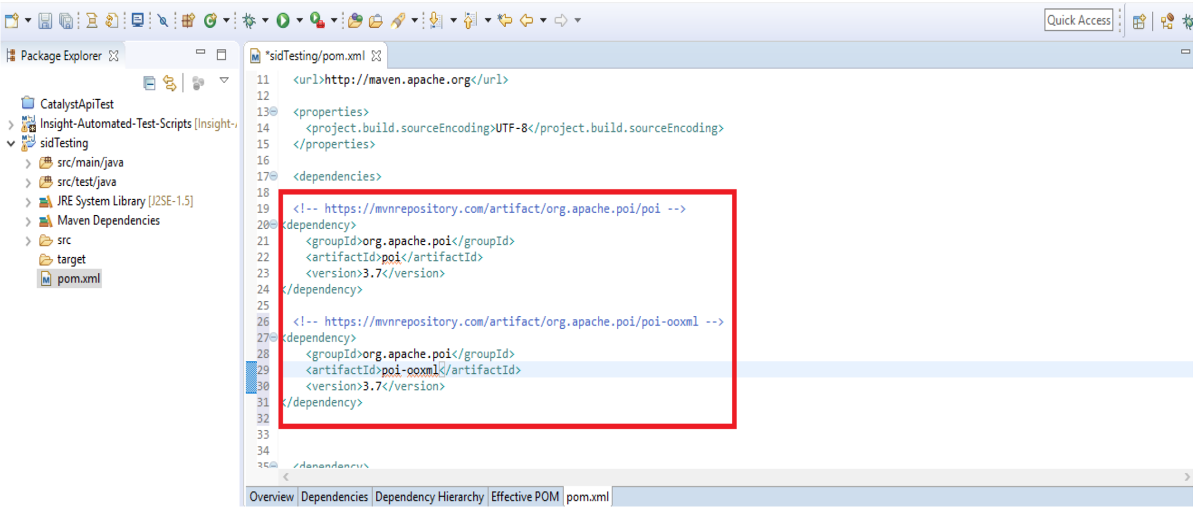
i. Open the Pom.xml file from the project.

j. Copy the dependency in the Pom.xml file.



```
<!-- https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml -->
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi-ooxml</artifactId>
    <version>3.7</version>
</dependency>
```

k. This is how the pom.xml file will look like in Eclipse:



```
  <url>http://maven.apache.org</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>

  <dependencies>

  <!-- https://mvnrepository.com/artifact/org.apache.poi/poi -->
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi</artifactId>
    <version>3.7</version>
</dependency>

  <!-- https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml -->
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi-ooxml</artifactId>
    <version>3.7</version>
</dependency>



  <dependency>
```

Step 3: 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