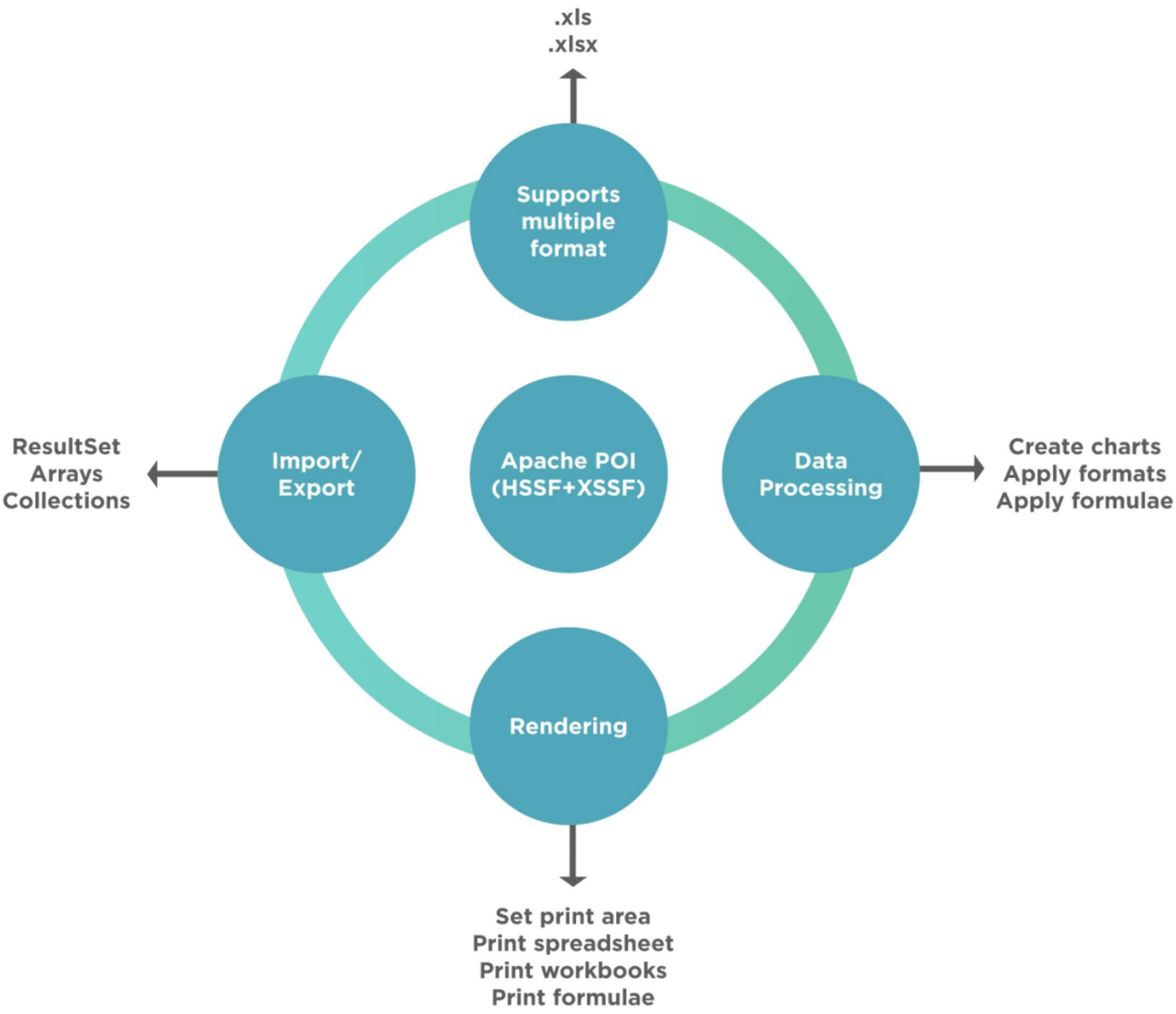
In test automation, the test data plays an important role. The test automation framework developers maintain the test data in various formats and ***Microsoft Excel*** is one of the easiest and most used ways of storing and maintaining test data. Even the manual QAs keep their test data in Excel files. Now to access this test data in the automation frameworks, ***Java*** provides various libraries and ***Apache POI*** is one of the most used among them. Subsequently, in this article, we will kickstart our journey of using the *Apache POI* by understanding how to download apache poi and use the same in an automation framework, by covering the details under the following topics:

- *What is Apache POI used for?*
- *How to download Apache POI?*
- *How to install POI libraries?*
    - *And, how to configure POI libraries in Eclipse?*

What is Apache POI used for?

Apache POI is an open-source library developed and distributed by *Apache Foundation.* Moreover, it is mainly used to create, read, and edit ***Microsoft Office*** files, majorly Excel files in Java programs. Moreover, it is distributed as a JAR, which provides various methods to manipulate Microsoft Excel files. The image below shows details of various formats and actions that Apache POI supports:



The older versions of *Apache POI* support binary file formats such as ***doc, xls, ppt*** etc whereas, from version 3.5 onwards, *Apache POI* supports *OOXML* file formats such as ***docx, xlsx, pptx*** etc. Additionally, the table below gives a brief summary of various components provided by *Apache POI:*

Component	***Explanation***	***Details***
POIFS	Poor Obfuscation Implementation File System.	This component provides the capability to read various files.
HSSF	Horrible Spreadsheet Format	This component is used to read/write an older format of Excel(*xls*).
XSSF	XML Spreadsheet Format	This component is used to read/write a new format of Excel(*xlsx*).
HPSF	Horrible Property Set Format	This component is used to extract the ***"property sets"*** of various types of MS- Office files.
HWPF	Horrible Word Processor Format	This component reads/writes an older format of Word(*doc*).
XWPF	XML Word Processor Format	This component reads/writes a new format of Word(*docx*).

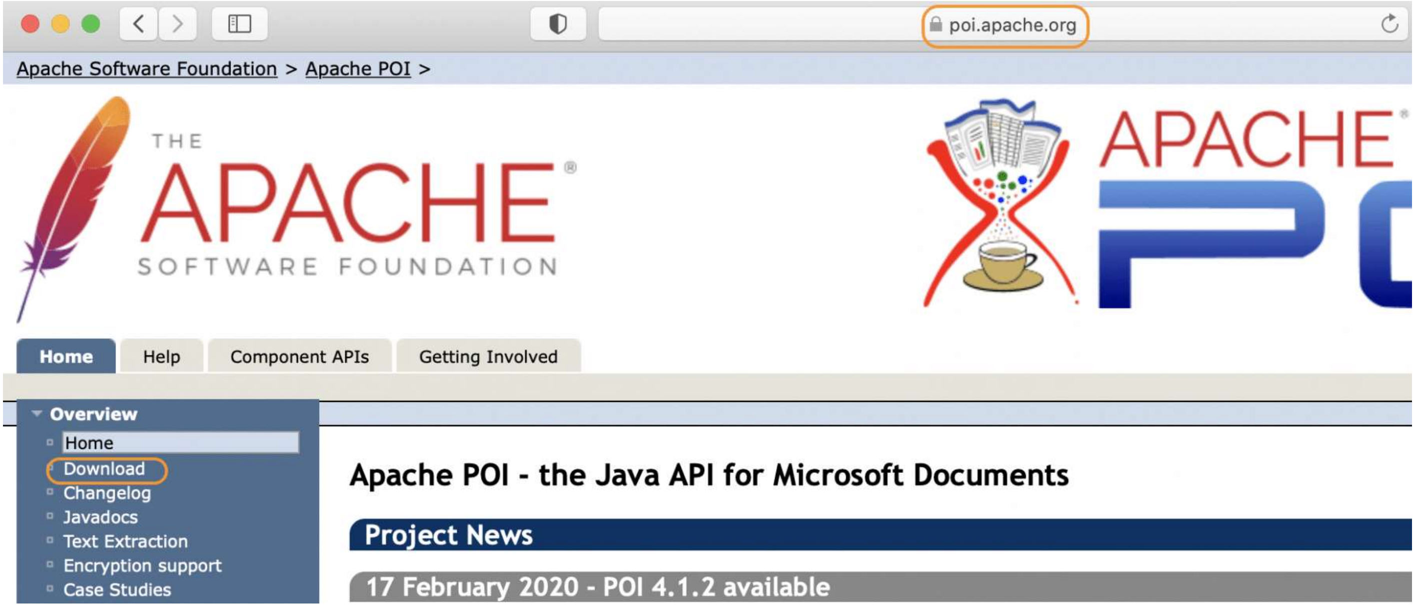
Component	***Explanation***	***Details***
HSLF	Horrible Slide Layout Format	This component reads/writes ***PowerPoint*** presentations.
HDGF	Horrible DiaGram Format	This component reads/writes ***MS-Visio*** files.
HPBF	Horrible PuBlisher Format	This component reads/writes ***MS-Publisher*** files.

Let's now quickly see, how we can download *Apache POI *libraries:

How to download Apache POI?

The first step in the process of storing and accessing the test data in Excel files is to download the *Apache POI* library. Consequently, follow the steps as mentioned below to download the *Apache POI* library:

. First, navigate to the ***Apache POI*** webpage. After that, click on the ***Download*** link in the left menu. Moreover, it is as highlighted below:

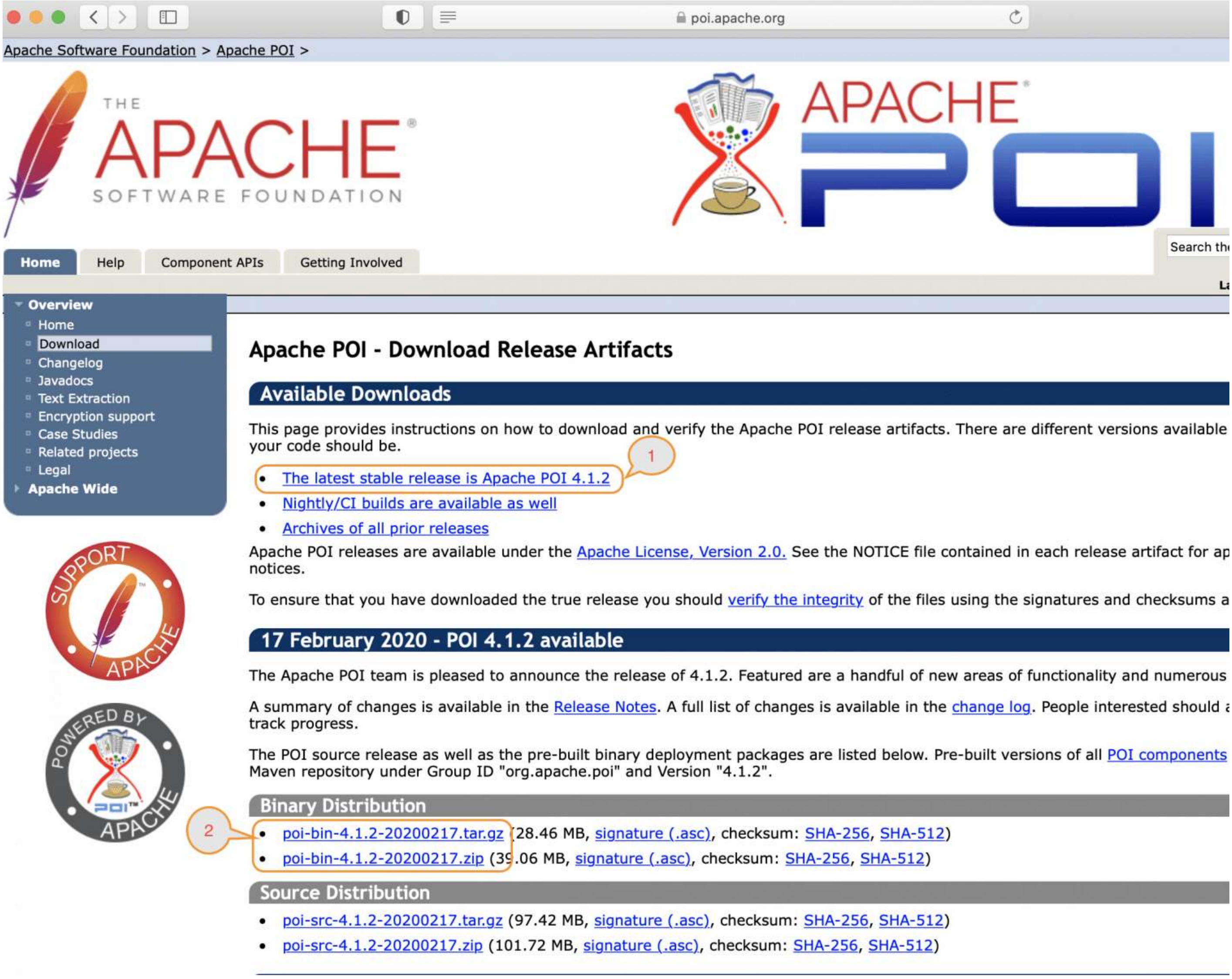


Apache POI - the Java API for Microsoft Documents

Project News

17 February 2020 - POI 4.1.2 available

. Secondly, clicking on the Download link will navigate to the page showing the latest release of ***Apache POI.*** Additionally, it is as highlighted below:



Apache POI - Download Release Artifacts

Available Downloads

This page provides instructions on how to download and verify the Apache POI release artifacts. There are different versions available your code should be.

- The latest stable release is Apache POI 4.1.2
- Nightly/CI builds are available as well
- Archives of all prior releases

Apache POI releases are available under the Apache License, Version 2.0. See the NOTICE file contained in each release artifact for ap notices.

To ensure that you have downloaded the true release you should verify the integrity of the files using the signatures and checksums a

17 February 2020 - POI 4.1.2 available

The Apache POI team is pleased to announce the release of 4.1.2. Featured are a handful of new areas of functionality and numerous

A summary of changes is available in the Release Notes. A full list of changes is available in the change log. People interested should a track progress.

The POI source release as well as the pre-built binary deployment packages are listed below. Pre-built versions of all POI components Maven repository under Group ID "org.apache.poi" and Version "4.1.2".

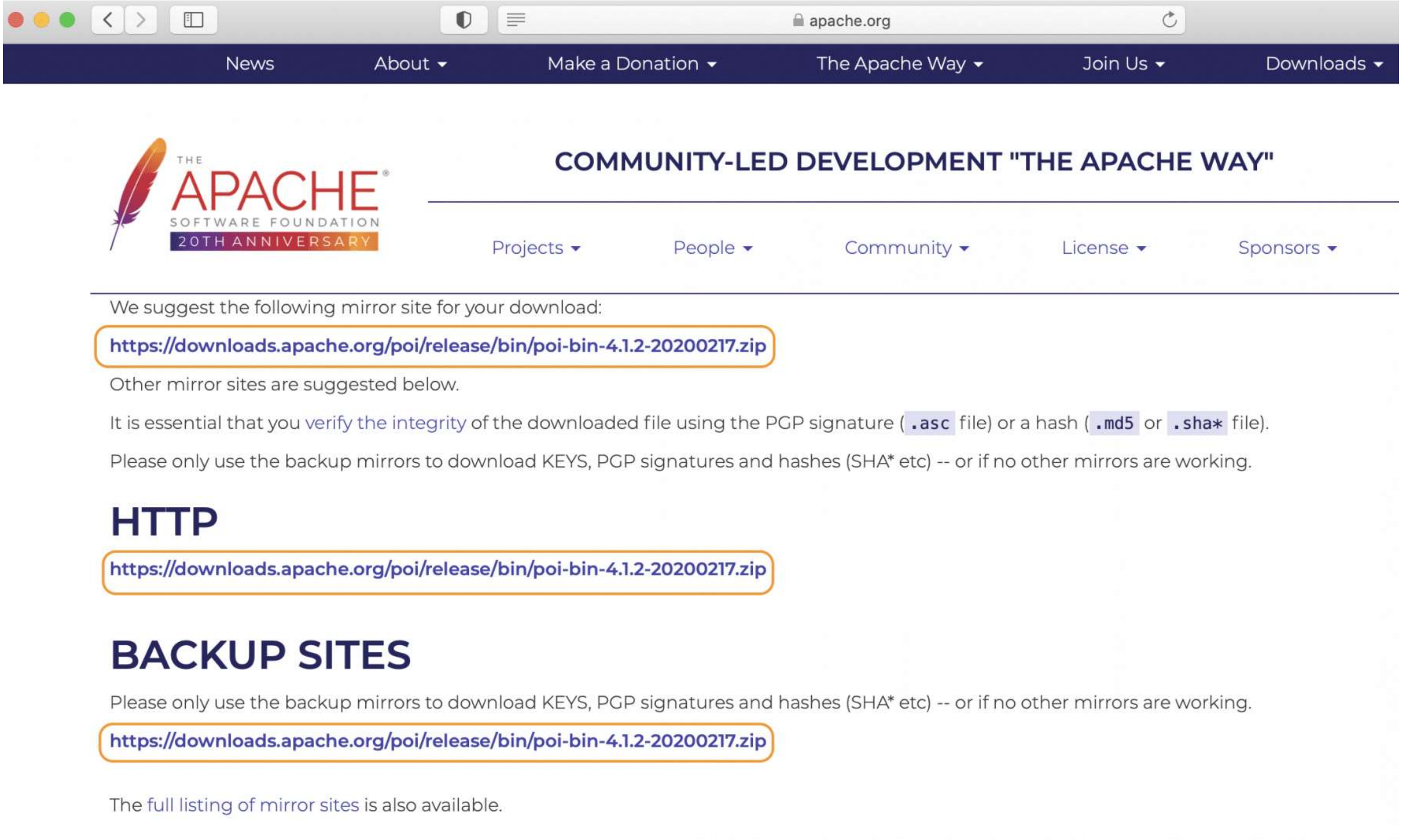
Binary Distribution

- poi-bin-4.1.2-20200217.tar.gz (28.46 MB, signature (.asc), checksum: SHA-256, SHA-512)
- poi-bin-4.1.2-20200217.zip (39.06 MB, signature (.asc), checksum: SHA-256, SHA-512)

Source Distribution

- poi-src-4.1.2-20200217.tar.gz (97.42 MB, signature (.asc), checksum: SHA-256, SHA-512)
- poi-src-4.1.2-20200217.zip (101.72 MB, signature (.asc), checksum: SHA-256, SHA-512)

. Thirdly, you can either click on the ***"Latest Stable Release Link "*** *(as shown by marker 1),* which will scroll the page down to the binaries of *Apache POI (as shown by marker 2),* or can directly scroll down to the section of binaries shown by marker 2. Subsequently, after clicking on the ***"zip "*** file, it will navigate to the page showing various download links as shown below:



COMMUNITY-LED DEVELOPMENT "THE APACHE WAY"

We suggest the following mirror site for your download:

https://downloads.apache.org/poi/release/bin/poi-bin-4.1.2-20200217.zip

Other mirror sites are suggested below.

It is essential that you verify the integrity of the downloaded file using the PGP signature (`.asc` file) or a hash (`.md5` or `.sha*` file).

Please only use the backup mirrors to download KEYS, PGP signatures and hashes (SHA* etc) -- or if no other mirrors are working.

HTTP

https://downloads.apache.org/poi/release/bin/poi-bin-4.1.2-20200217.zip

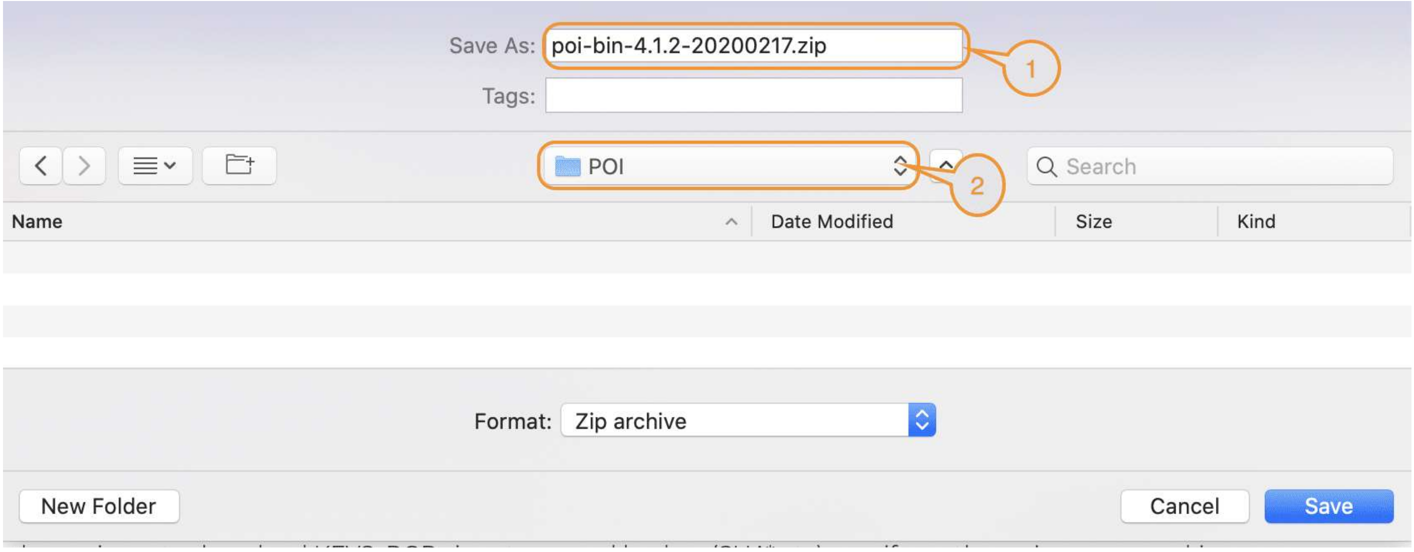
BACKUP SITES

Please only use the backup mirrors to download KEYS, PGP signatures and hashes (SHA* etc) -- or if no other mirrors are working.

https://downloads.apache.org/poi/release/bin/poi-bin-4.1.2-20200217.zip

The full listing of mirror sites is also available.

. After that, when you click on any of the highlighted links, it will download a zip. Additionally, you can save it in any folder of your choice as shown below:



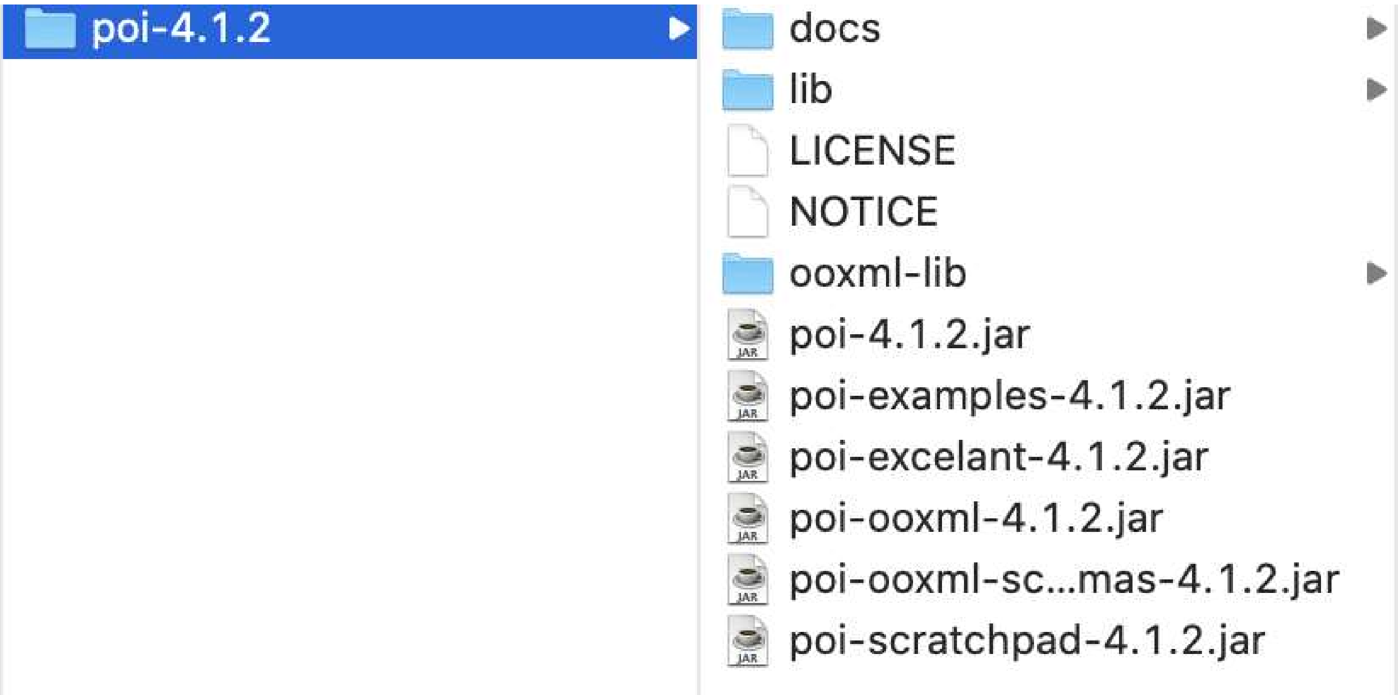
Save As: poi-bin-4.1.2-20200217.zip

Tags:

POI

Format: Zip archive

. Fifthly, once you unzip the file, it will show the contents as below:



These are various *JAR* files that provide the classes and methods that we use for the manipulation of various *MS-Office* file types. Subsequently, let's see how we can install these *JARs* in our projects and use them for the manipulation of various supported file types.

How to install POI libraries?

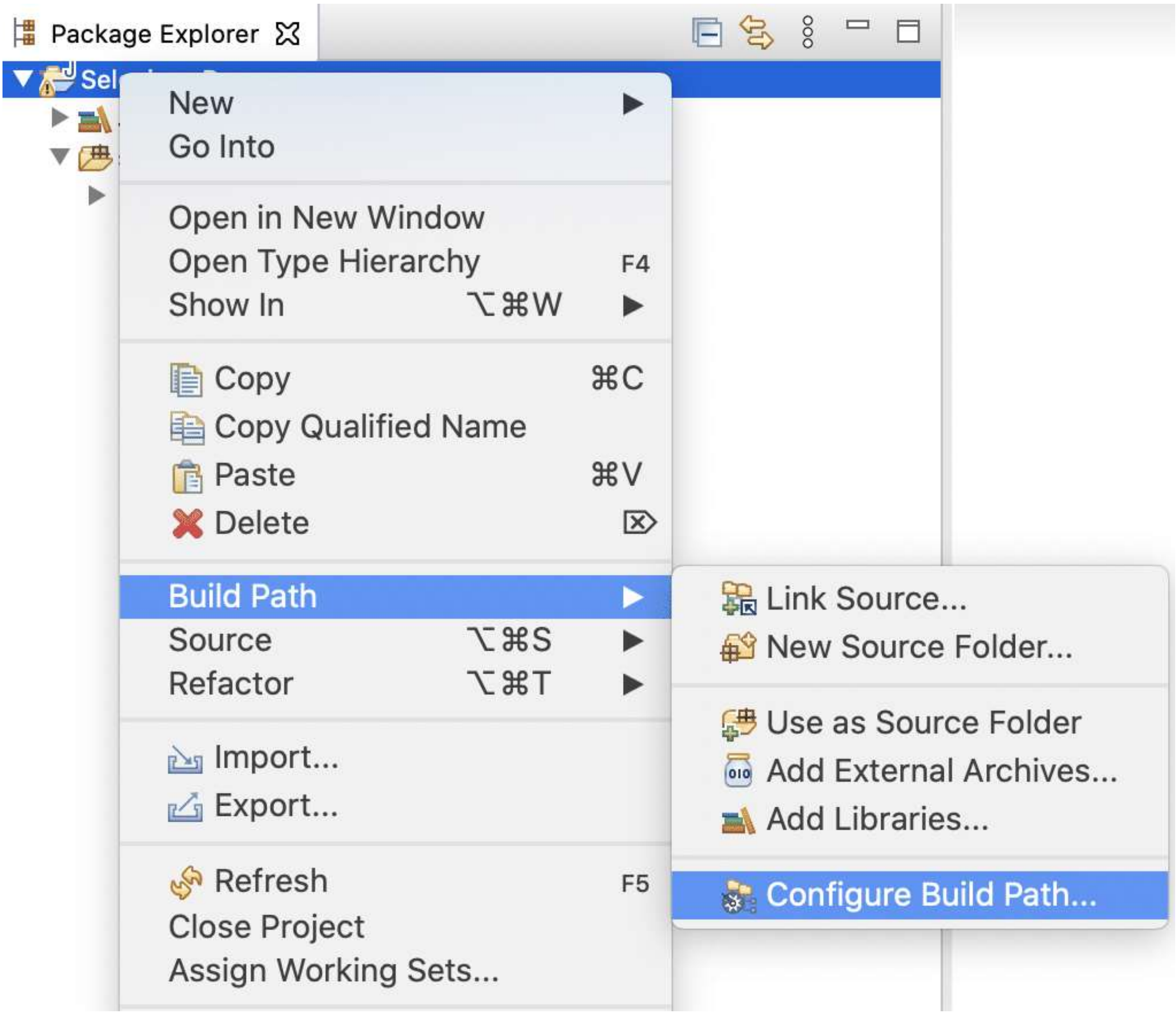
As we understood in the above sections that all the *Apache POI libraries* are available as *JARs.* Now to access the functionalities of *POI*, these *JARs* should be available in the build path of your application/framework. Additionally, we are majorly using *Eclipse* as *IDE* in our articles. Consequently, let's quickly see how we can install the *Apache POI JARs* in the build path of a project in *Eclipse :*

How to configure POI libraries in Eclipse?

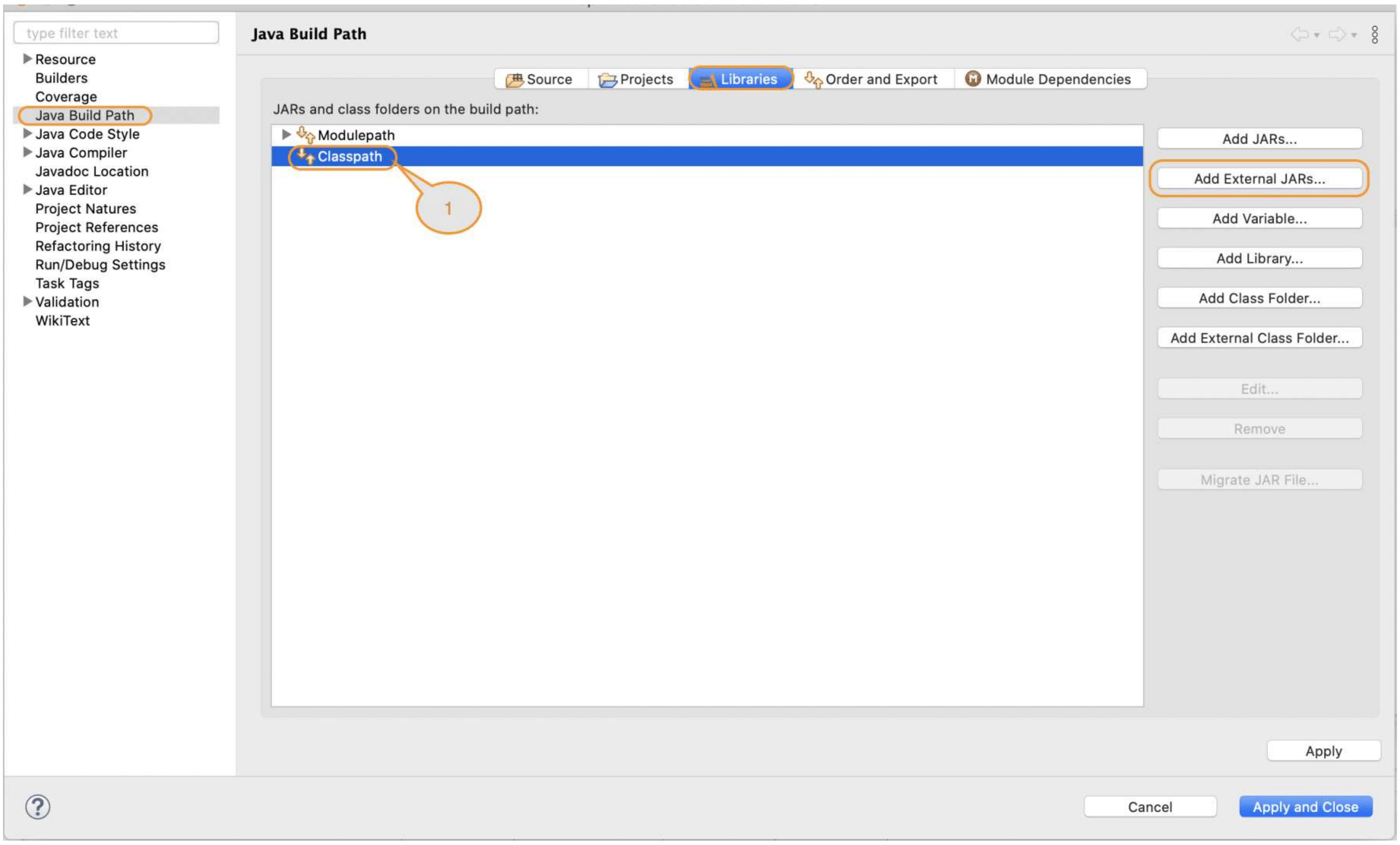
Follow the steps as mentioned below to add the *POI JARs* in a project in *Eclipse:*

. Firstly, suppose you have created a *JAVA* project in *Eclipse,* as per the steps mentioned in the article ***"Configure Selenium WebDriver with Eclipse".***

. After that, right -click on the project in *Eclipse. Subsequently,* select ***Build Path >> Configure Build Path*** as shown below:-

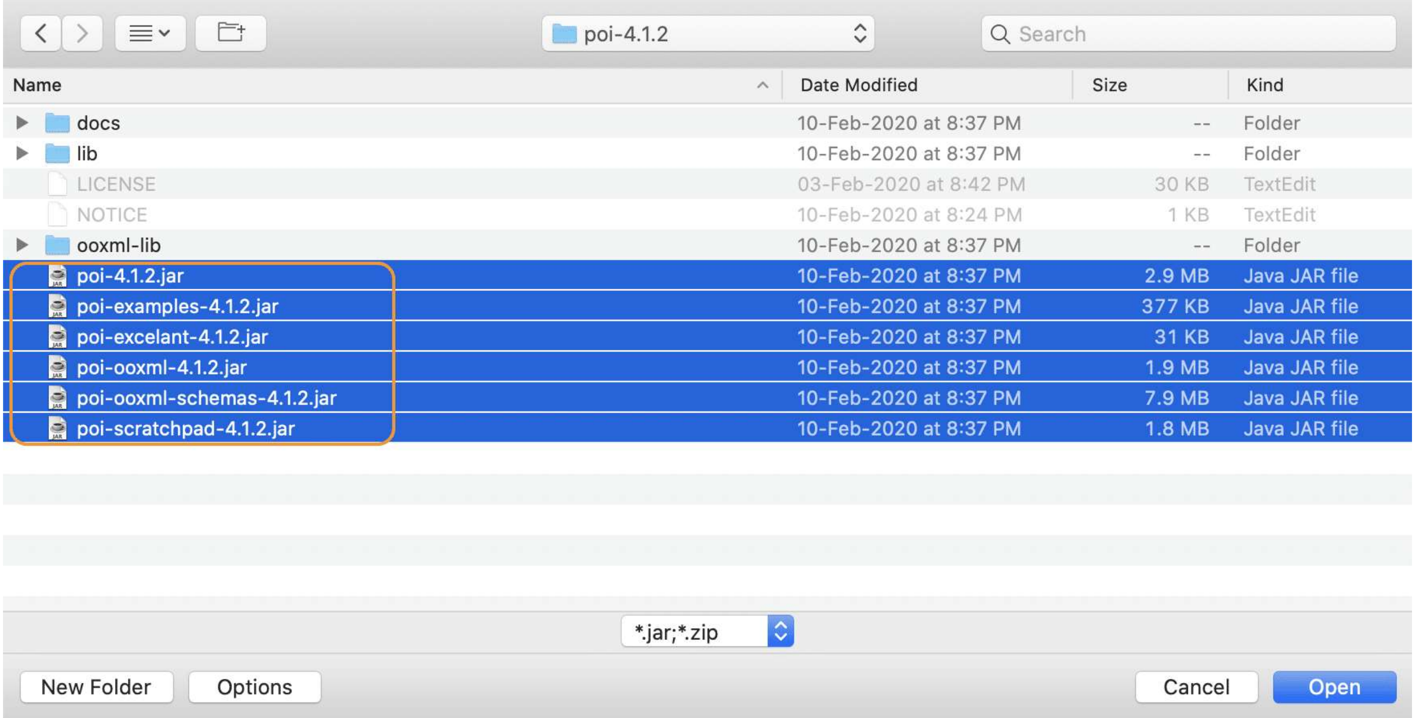


. Thirdly, it will open the ***"Properties "*** of the project. After that, select the ***Libraries*** tab. Finally, click on the ***Add External JARs*** as highlighted below.



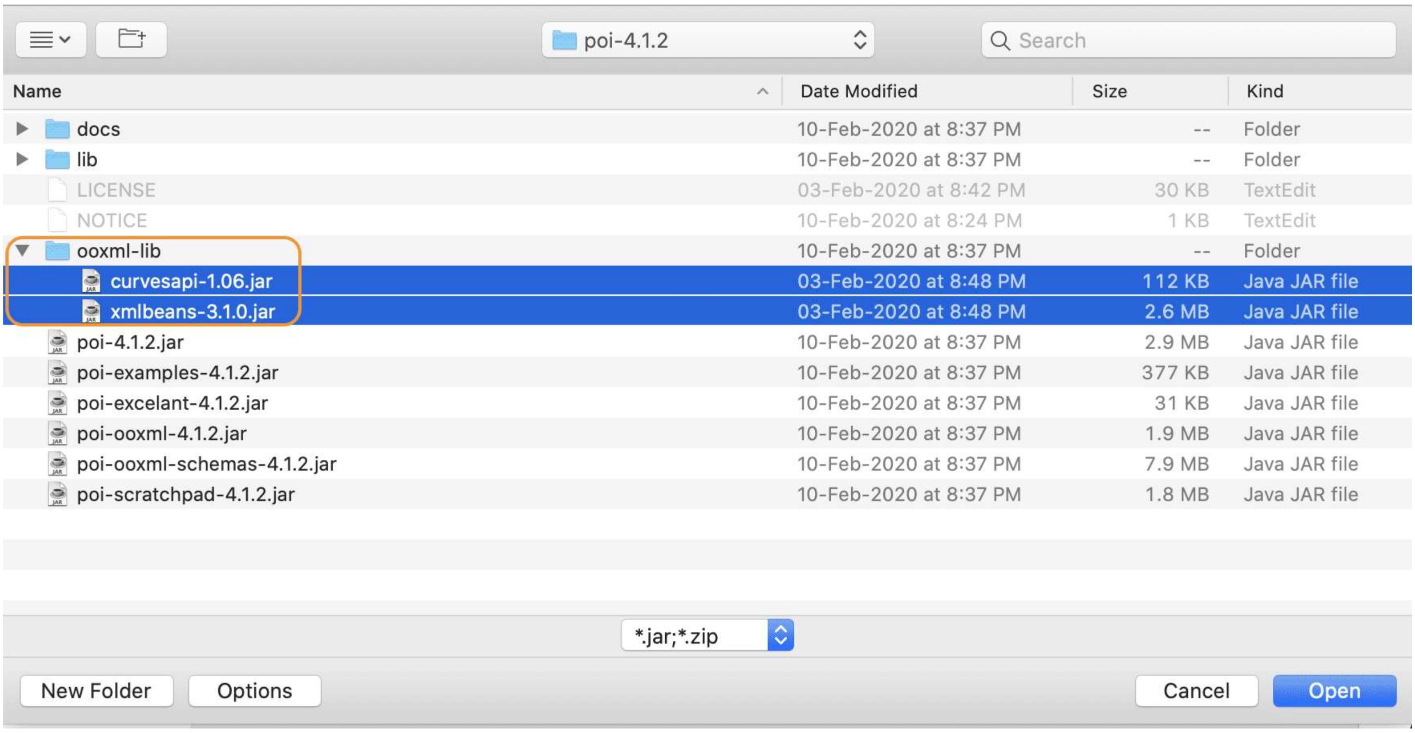
Note: *Classpath(as highlighted by marker 1) should be selected when adding the External JARs.*

. Fourthly, select the *JARs* in the parent folder of the unzipped *POI* files**.** Subsequently, click on the ***Open*** button to include them in the *Eclipse* project:



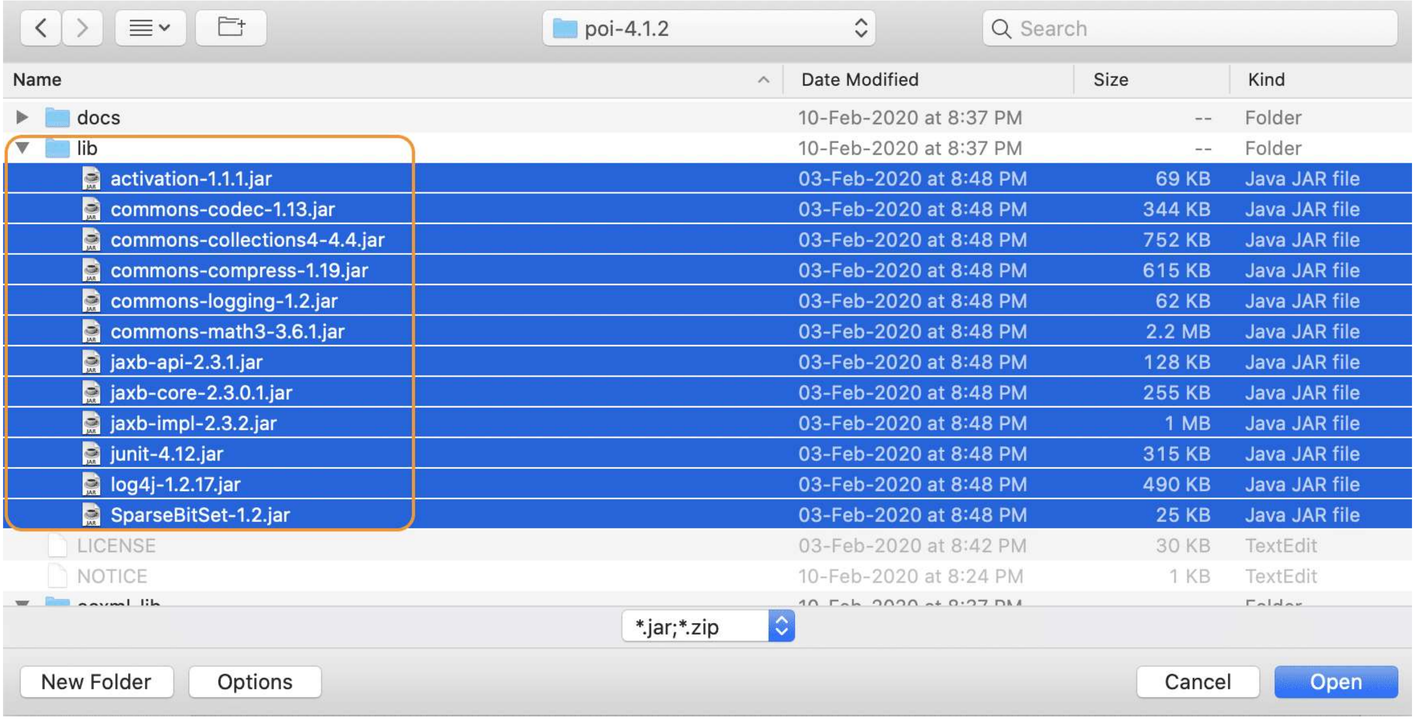
Name	Date Modified	Size	Kind
docs	10-Feb-2020 at 8:37 PM	--	Folder
lib	10-Feb-2020 at 8:37 PM	--	Folder
LICENSE	03-Feb-2020 at 8:42 PM	30 KB	TextEdit
NOTICE	10-Feb-2020 at 8:24 PM	1 KB	TextEdit
ooxml-lib	10-Feb-2020 at 8:37 PM	--	Folder
poi-4.1.2.jar	10-Feb-2020 at 8:37 PM	2.9 MB	Java JAR file
poi-examples-4.1.2.jar	10-Feb-2020 at 8:37 PM	377 KB	Java JAR file
poi-excelant-4.1.2.jar	10-Feb-2020 at 8:37 PM	31 KB	Java JAR file
poi-ooxml-4.1.2.jar	10-Feb-2020 at 8:37 PM	1.9 MB	Java JAR file
poi-ooxml-schemas-4.1.2.jar	10-Feb-2020 at 8:37 PM	7.9 MB	Java JAR file
poi-scratchpad-4.1.2.jar	10-Feb-2020 at 8:37 PM	1.8 MB	Java JAR file

. Next select the *JARs* under the ***ooxml-lib*** folder in the unzipped *POI* folder**.** Moreover, it is as highlighted below:



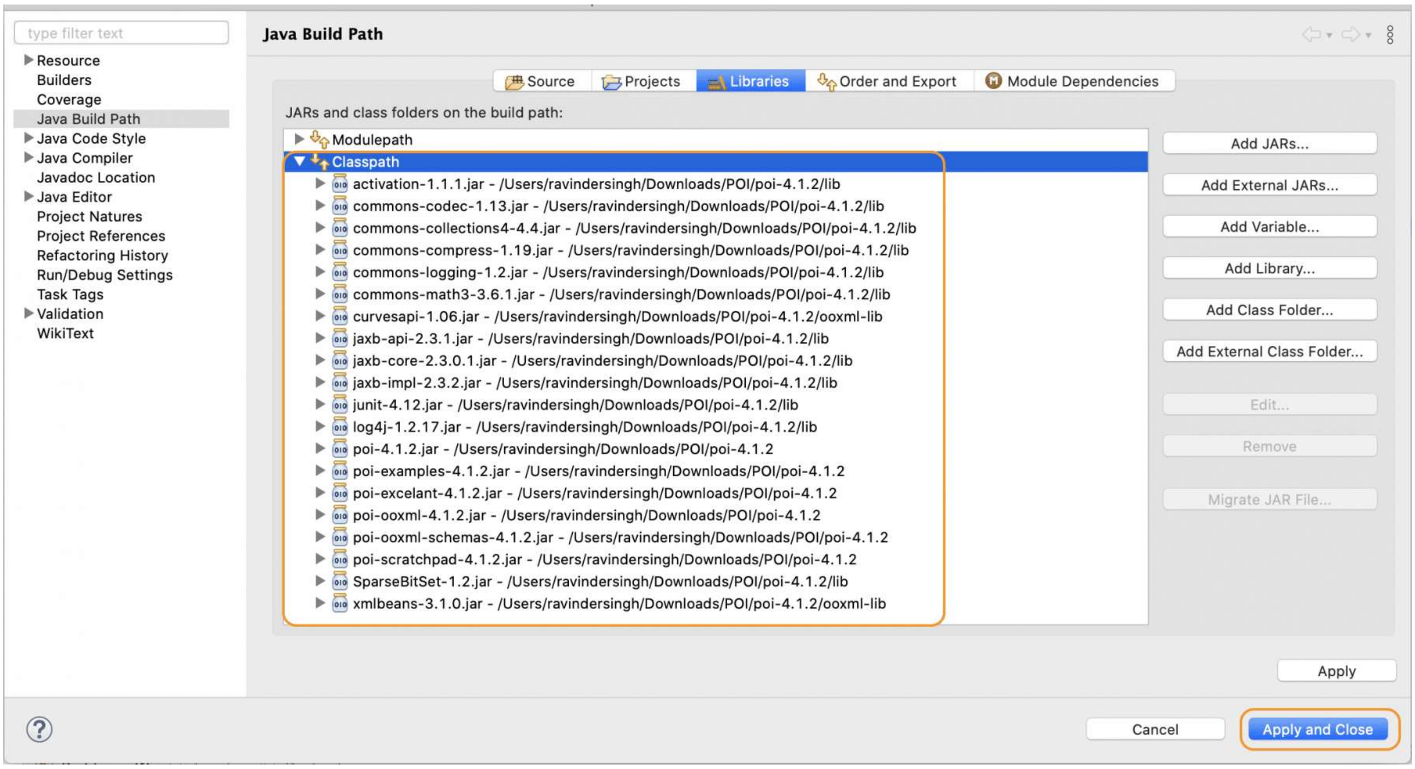
Name	Date Modified	Size	Kind
docs	10-Feb-2020 at 8:37 PM	--	Folder
lib	10-Feb-2020 at 8:37 PM	--	Folder
LICENSE	03-Feb-2020 at 8:42 PM	30 KB	TextEdit
NOTICE	10-Feb-2020 at 8:24 PM	1 KB	TextEdit
ooxml-lib	10-Feb-2020 at 8:37 PM	--	Folder
curvesapi-1.06.jar	03-Feb-2020 at 8:48 PM	112 KB	Java JAR file
xmlbeans-3.1.0.jar	03-Feb-2020 at 8:48 PM	2.6 MB	Java JAR file
poi-4.1.2.jar	10-Feb-2020 at 8:37 PM	2.9 MB	Java JAR file
poi-examples-4.1.2.jar	10-Feb-2020 at 8:37 PM	377 KB	Java JAR file
poi-excelant-4.1.2.jar	10-Feb-2020 at 8:37 PM	31 KB	Java JAR file
poi-ooxml-4.1.2.jar	10-Feb-2020 at 8:37 PM	1.9 MB	Java JAR file
poi-ooxml-schemas-4.1.2.jar	10-Feb-2020 at 8:37 PM	7.9 MB	Java JAR file
poi-scratchpad-4.1.2.jar	10-Feb-2020 at 8:37 PM	1.8 MB	Java JAR file

- Sixthly, select the *JARs* under the ***lib*** folder in the unzipped *POI* folder**.** Additionally, it is as highlighted below:



Name	Date Modified	Size	Kind
docs	10-Feb-2020 at 8:37 PM	--	Folder
lib	10-Feb-2020 at 8:37 PM	--	Folder
activation-1.1.1.jar	03-Feb-2020 at 8:48 PM	69 KB	Java JAR file
commons-codec-1.13.jar	03-Feb-2020 at 8:48 PM	344 KB	Java JAR file
commons-collections4-4.4.jar	03-Feb-2020 at 8:48 PM	752 KB	Java JAR file
commons-compress-1.19.jar	03-Feb-2020 at 8:48 PM	615 KB	Java JAR file
commons-logging-1.2.jar	03-Feb-2020 at 8:48 PM	62 KB	Java JAR file
commons-math3-3.6.1.jar	03-Feb-2020 at 8:48 PM	2.2 MB	Java JAR file
jaxb-api-2.3.1.jar	03-Feb-2020 at 8:48 PM	128 KB	Java JAR file
jaxb-core-2.3.0.1.jar	03-Feb-2020 at 8:48 PM	255 KB	Java JAR file
jaxb-impl-2.3.2.jar	03-Feb-2020 at 8:48 PM	1 MB	Java JAR file
junit-4.12.jar	03-Feb-2020 at 8:48 PM	315 KB	Java JAR file
log4j-1.2.17.jar	03-Feb-2020 at 8:48 PM	490 KB	Java JAR file
SparseBitSet-1.2.jar	03-Feb-2020 at 8:48 PM	25 KB	Java JAR file
LICENSE	03-Feb-2020 at 8:42 PM	30 KB	TextEdit
NOTICE	10-Feb-2020 at 8:24 PM	1 KB	TextEdit
ooxml-lib	10-Feb-2020 at 8:37 PM		Folder

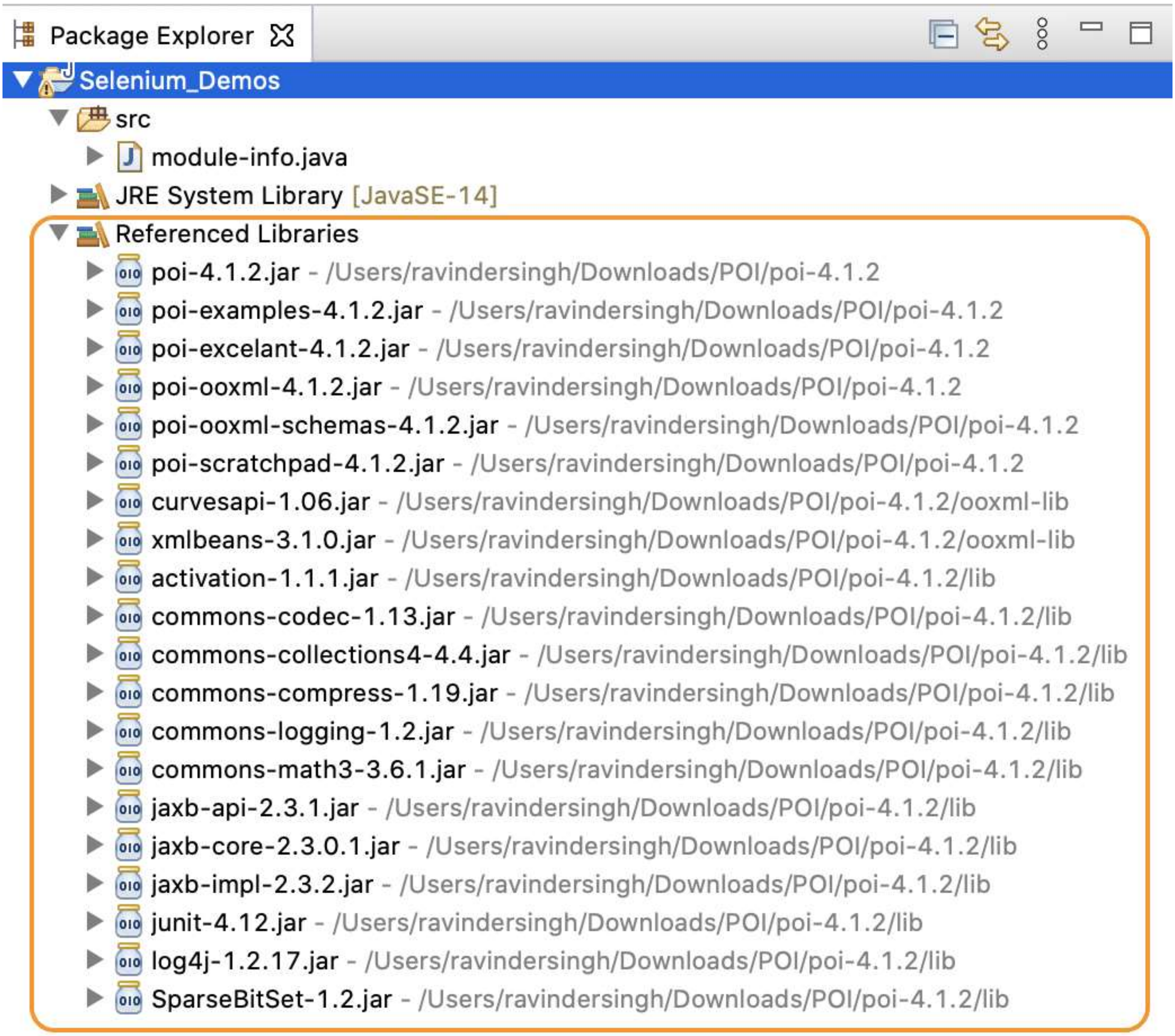
- After that, once all the *POI JARs* add, click on the ***Apply and Close*** button**.** Moreover, it is as highlighted below:



JARs and class folders on the build path:

- Modulepath
- Classpath
  - activation-1.1.1.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/lib
  - commons-codec-1.13.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/lib
  - commons-collections4-4.4.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/lib
  - commons-compress-1.19.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/lib
  - commons-logging-1.2.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/lib
  - commons-math3-3.6.1.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/lib
  - curvesapi-1.06.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/ooxml-lib
  - jaxb-api-2.3.1.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/lib
  - jaxb-core-2.3.0.1.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/lib
  - jaxb-impl-2.3.2.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/lib
  - junit-4.12.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/lib
  - log4j-1.2.17.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/lib
  - poi-4.1.2.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2
  - poi-examples-4.1.2.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2
  - poi-excelant-4.1.2.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2
  - poi-ooxml-4.1.2.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2
  - poi-ooxml-schemas-4.1.2.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2
  - poi-scratchpad-4.1.2.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2
  - SparseBitSet-1.2.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/lib
  - xmlbeans-3.1.0.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/ooxml-lib

. Once all the *POI libraries* successfully install in the *Eclipse* project, they will reflect under the ***Referenced Libraries*** folder in the left pane of the *Eclipse* project structure, as shown below:



```
▼ Selenium_Demos
  ▼ src
    ▶ module-info.java
  ▶ JRE System Library [JavaSE-14]
  ▼ Referenced Libraries
    ▶ poi-4.1.2.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2
    ▶ poi-examples-4.1.2.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2
    ▶ poi-excelant-4.1.2.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2
    ▶ poi-ooxml-4.1.2.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2
    ▶ poi-ooxml-schemas-4.1.2.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2
    ▶ poi-scratchpad-4.1.2.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2
    ▶ curvesapi-1.06.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/ooxml-lib
    ▶ xmlbeans-3.1.0.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/ooxml-lib
    ▶ activation-1.1.1.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/lib
    ▶ commons-codec-1.13.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/lib
    ▶ commons-collections4-4.4.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/lib
    ▶ commons-compress-1.19.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/lib
    ▶ commons-logging-1.2.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/lib
    ▶ commons-math3-3.6.1.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/lib
    ▶ jaxb-api-2.3.1.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/lib
    ▶ jaxb-core-2.3.0.1.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/lib
    ▶ jaxb-impl-2.3.2.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/lib
    ▶ junit-4.12.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/lib
    ▶ log4j-1.2.17.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/lib
    ▶ SparseBitSet-1.2.jar - /Users/ravindersingh/Downloads/POI/poi-4.1.2/lib
```

So, this completes the installation of Apache POI in an Eclipse project. Subsequently, we can now start using the capabilities of these libraries in our *JAVA* Project.

Key Takeaways

Apache POI libraries provide the capabilities to handle various types of MS-Office files.
Additionally, for an automation framework, keeping the test data in a file(eg, Excel file) is one of the common practices and Apache POI makes it very easy to read and write test data to an Excel file.
Lastly, Apache POI libraries are available as a set of JAR files, which we can download and install in an Eclipse project by simply including the JAR files in the project.