# MAVEN

## Definition:

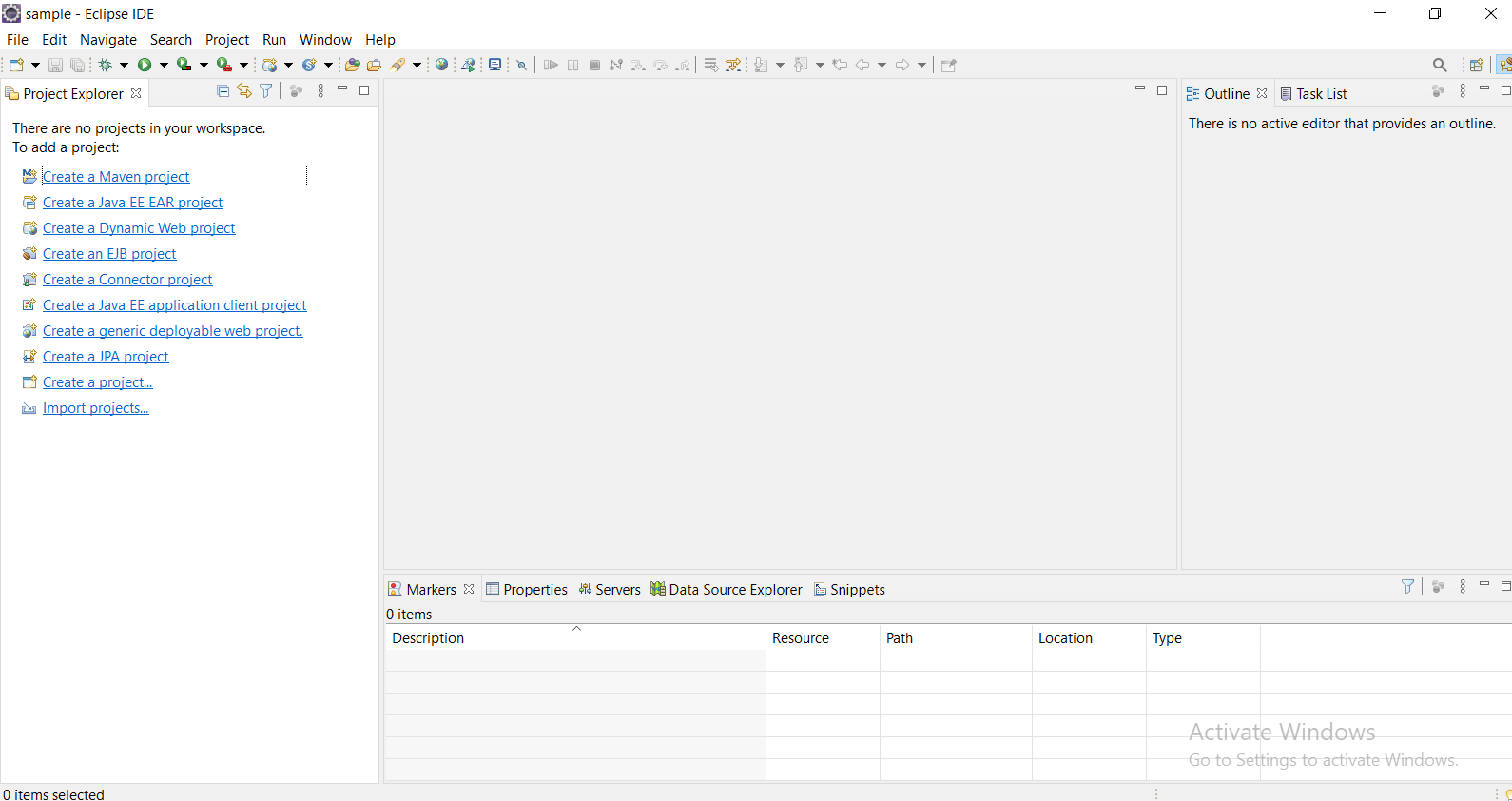
Maven is a popular open-source build tool developed by the Apache Group to build, publish, and deploy several projects at once for better [project management](https://www.simplilearn.com/project-management-skills-article). The tool provides allows developers to build and document the lifecycle framework.

## The Need for Maven:

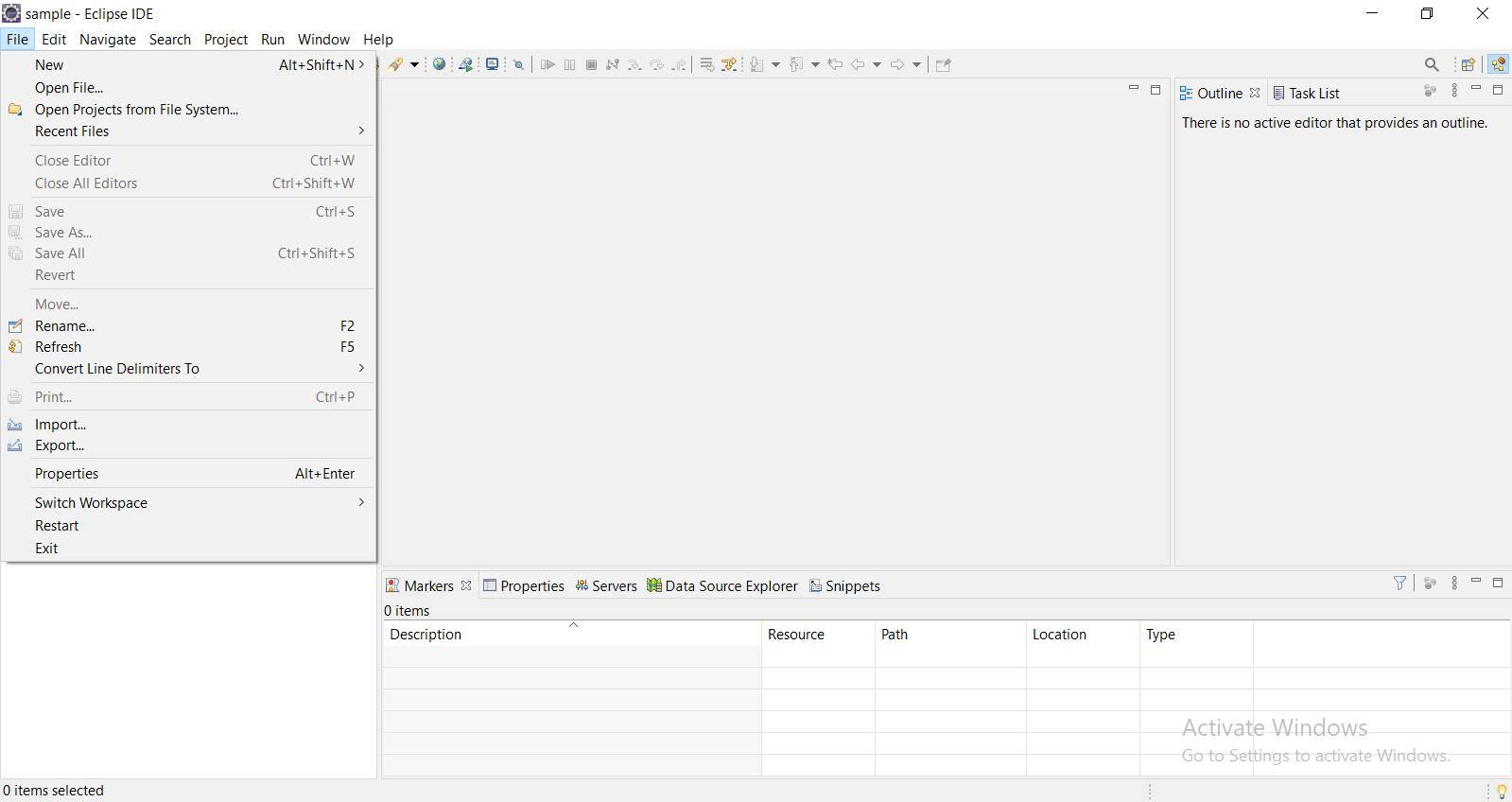
* Maven is chiefly used for Java-based projects, helping to download dependencies, which refers to the libraries or JAR files.
* The tool helps get the right JAR files for each project as there may be different versions of separate packages.

## How to Create Maven Project in Eclipse or Spring Tool Suite.

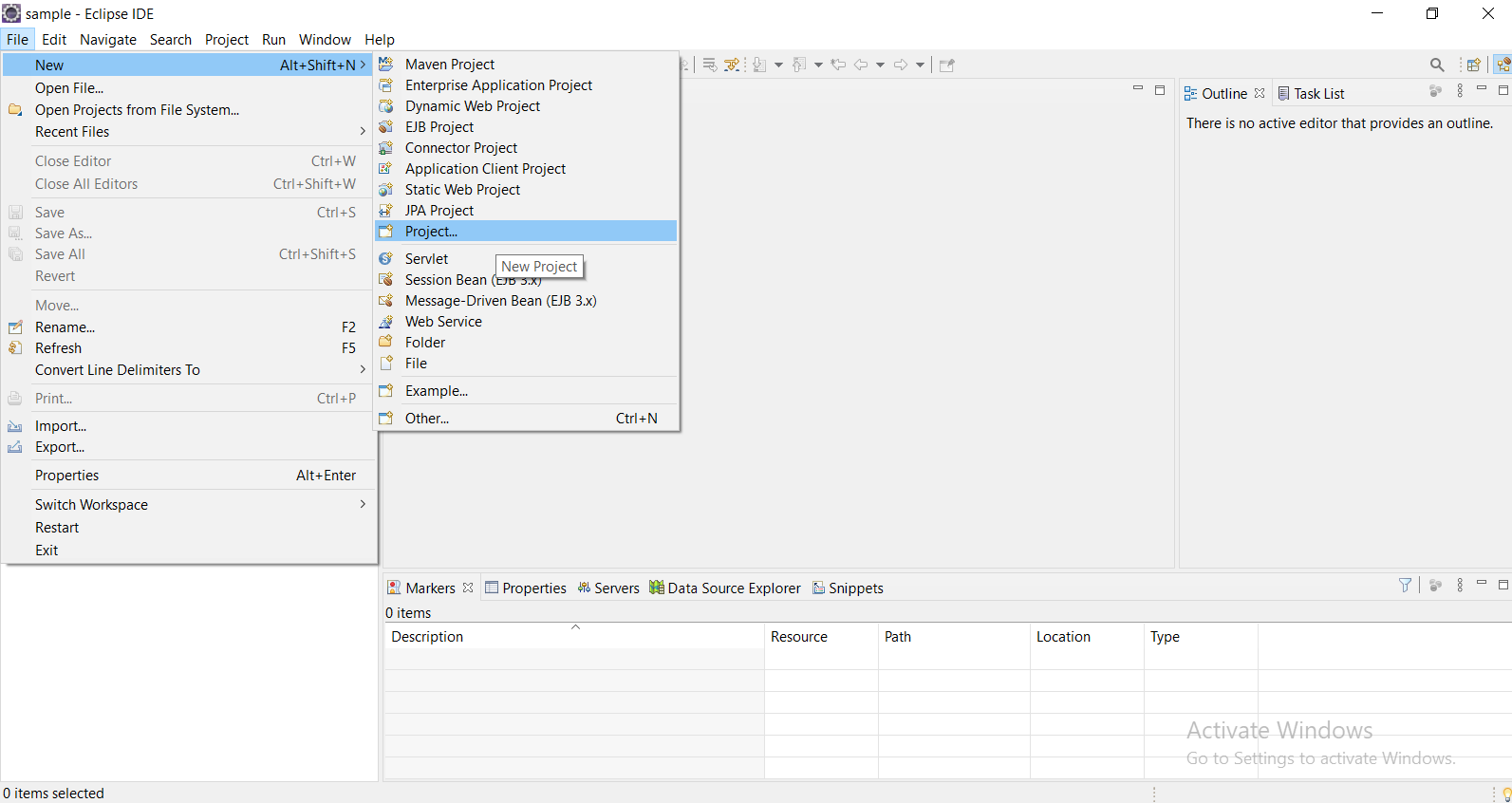
* Step 1: Open Eclipse.
* Step 2: Select a Directory as Workspace. The following Eclipse window screen will be open. Here you can directly click on Create a Maven Project. Or else you can follow the following Steps.



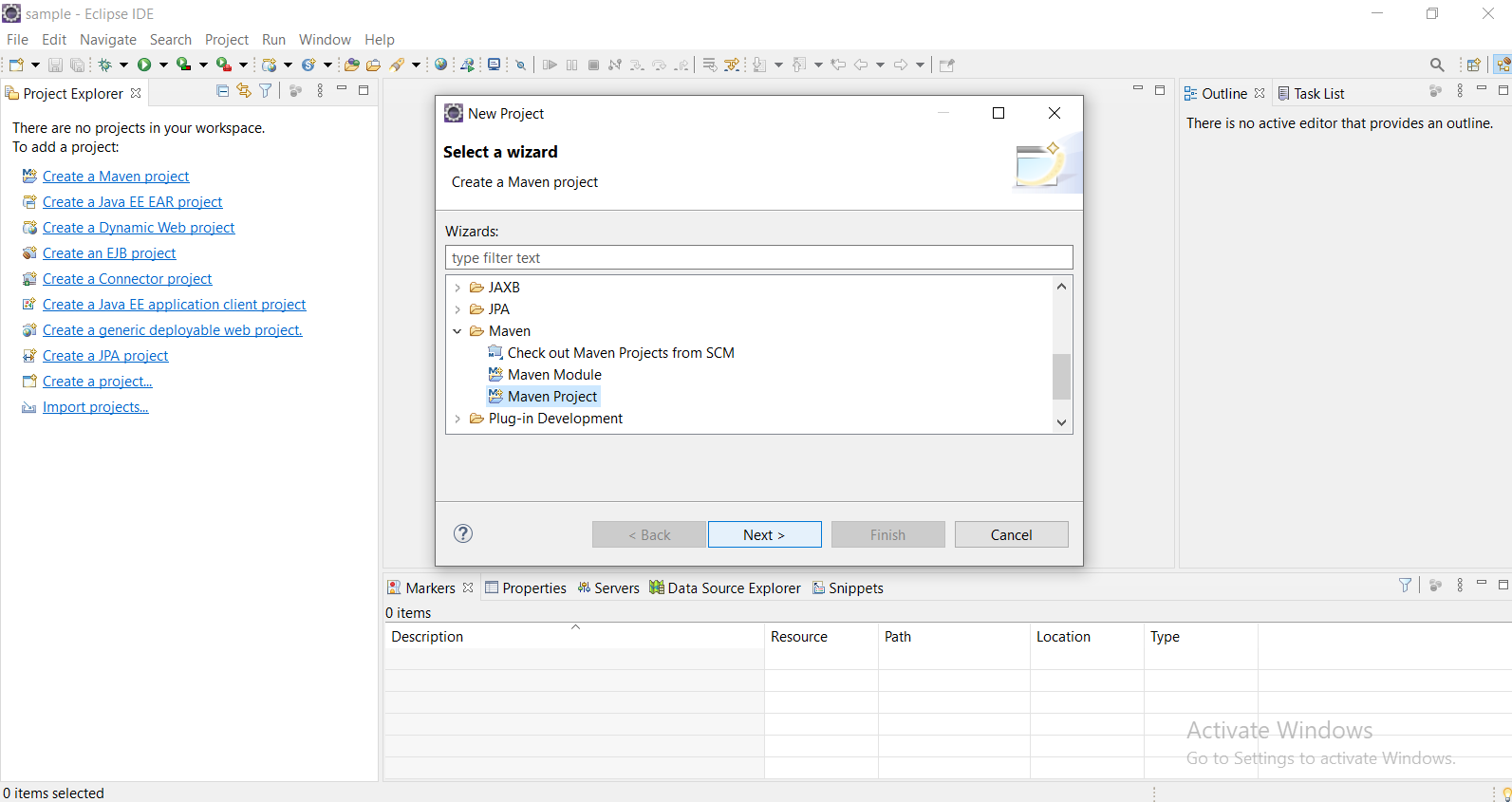
* Step 3: Go to the File Option.



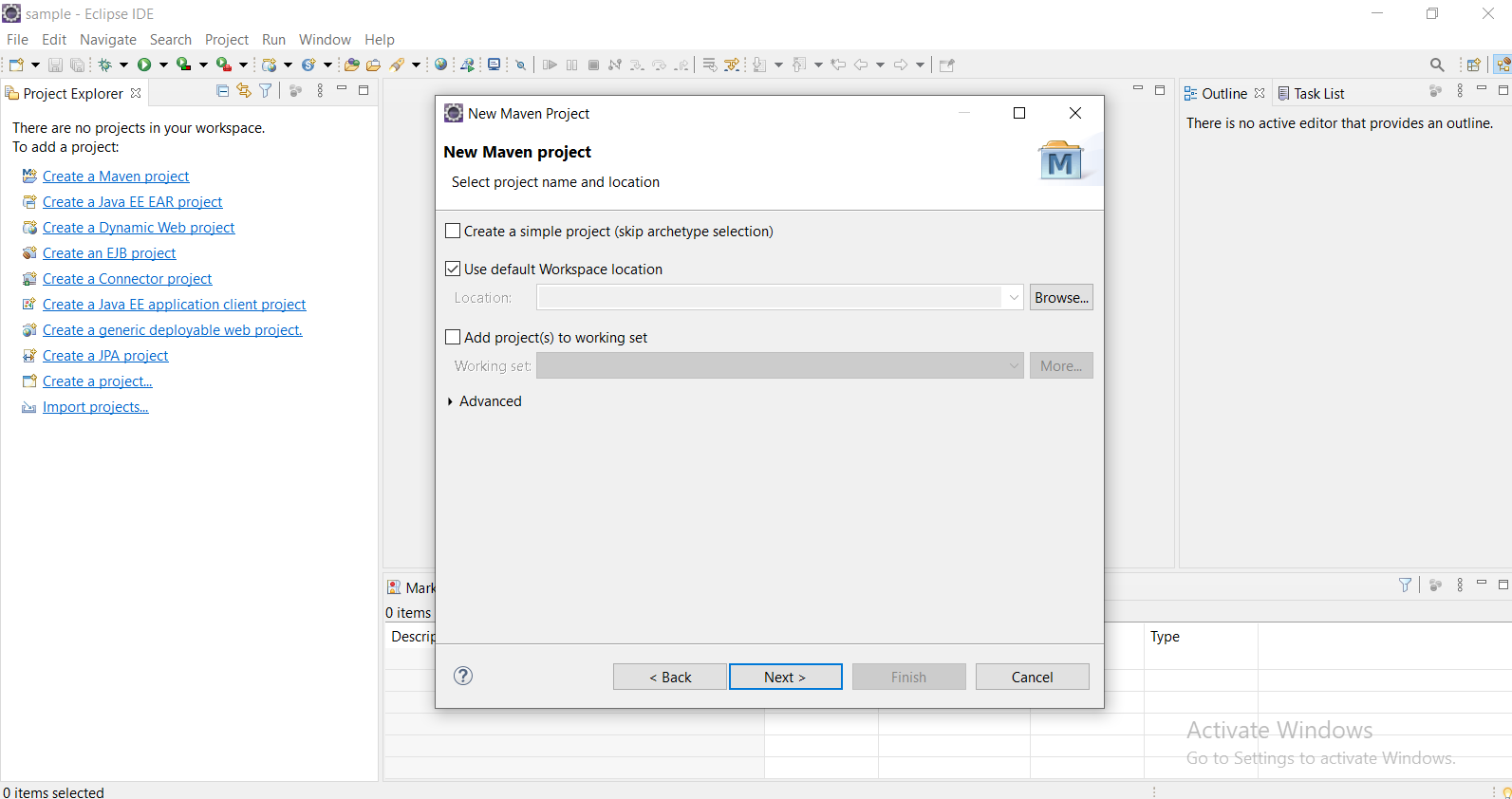
* Step 4: Select New and Select the Project Option. Or else After Select New you can also Select Maven Project. Here It is done through Project Option.



* Step 5: After Clicking on Project Option. New Project window will be open there you need to select Maven folder then inside Maven folder you need to click on Maven Project and finally click on Next button.



* Step 6: After Clicking on Next button. A New Maven Project Window will be Open. Click on Create a simple project (skip archetype selection) checkbox and Next button.



* Step 7: After Clicking on Next button, again a New Maven Project window will be open but with new categories. Here it will ask to enter Group Id, Artifact Id, Version, Packaging, Name, Description.
* Step 7.1: so Let’s understand first what are all these Identifiers (Group Id, Artifact Id, Version, Packaging, Name, Description).

Group Id: It Specifies the id of the project. A Unique base name of the company (or) group that created the project. It uniquely identifies your project across all projects.

You can enter it as “com.companyName.projectGroup” .

Artifact Id: It Specifies the id of the project. A Unique name of the project.

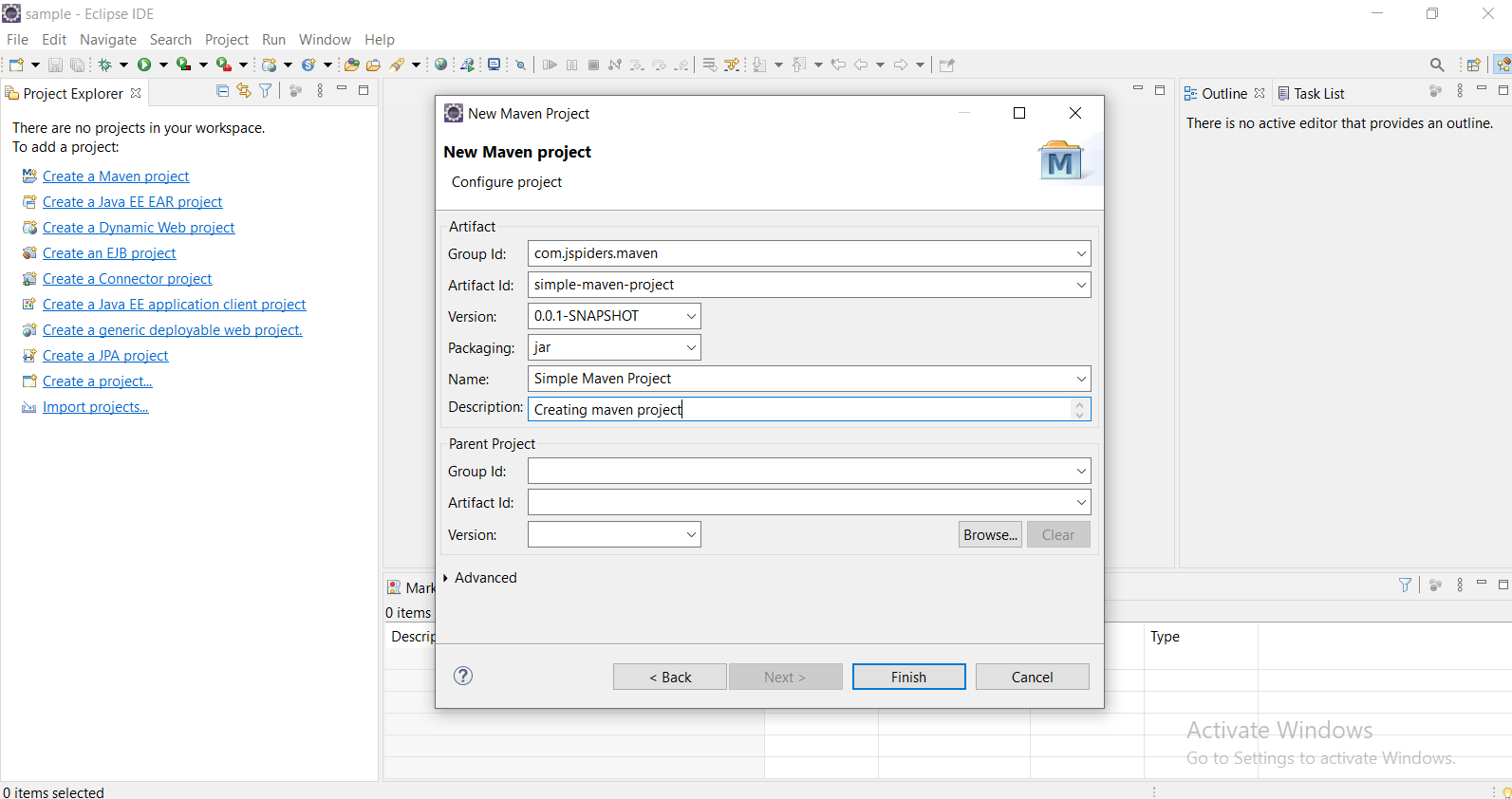
Version: Version of the Project.

Packaging: A packaging method (e.g. JAR/WAR/POM).

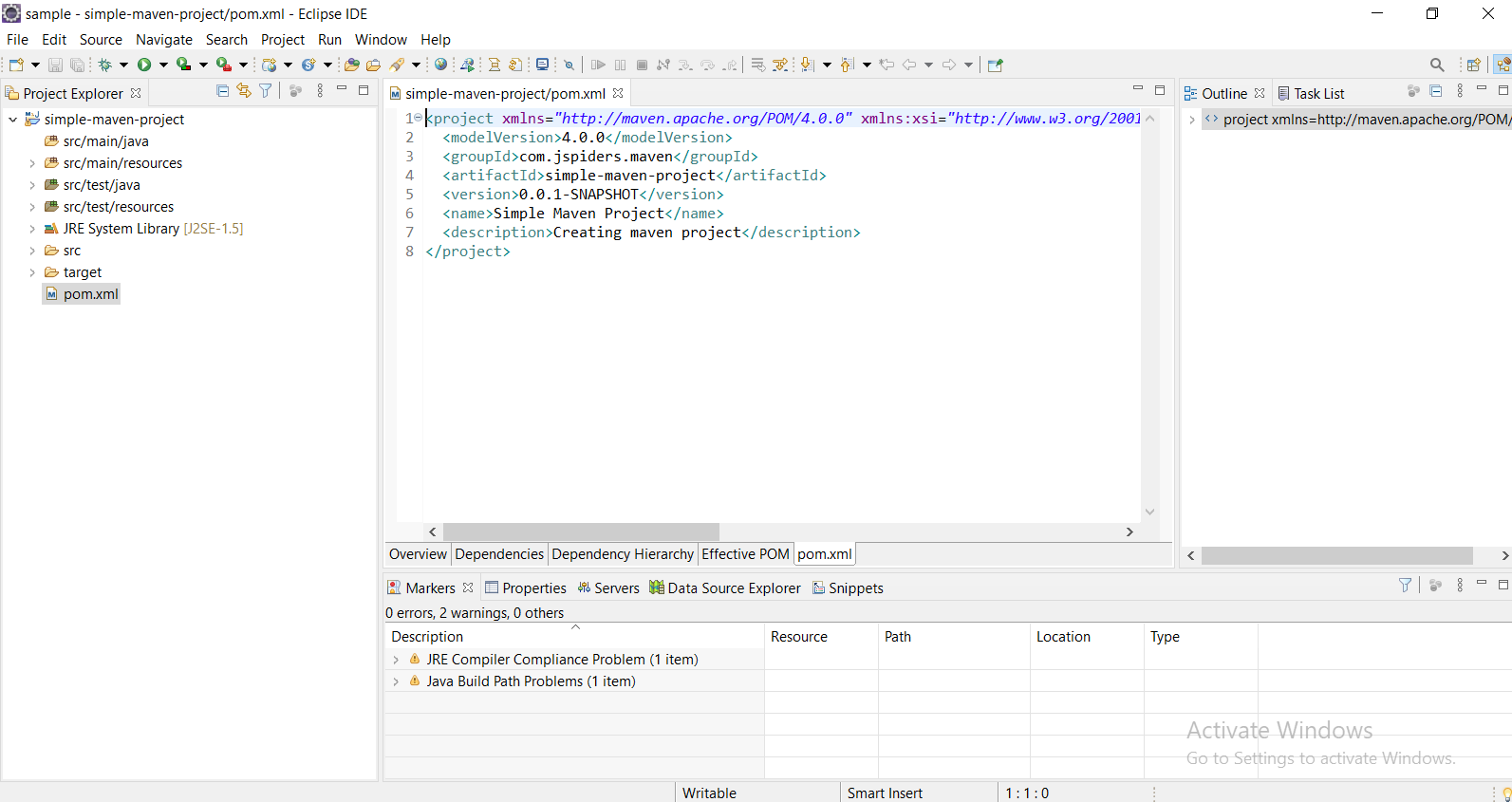
Name: Name for the Maven Project.

Description: You can provide the Description about Project.

After entering all details, click on Finish button.



After clicking the Finish button, the popup wizard will close and the project will get created. It will be visible in the project explorer. The below image shows all the directories and the pom.xml of the newly created project.



## MAVEN Repository:

A **maven repository** is a directory of packaged JAR file with pom.xml file. Maven searches for dependencies in the repositories.

There are 3 types of maven repository:

* Local Repository
* Central Repository
* Remote Repository

Maven searches for the dependencies in the following order:

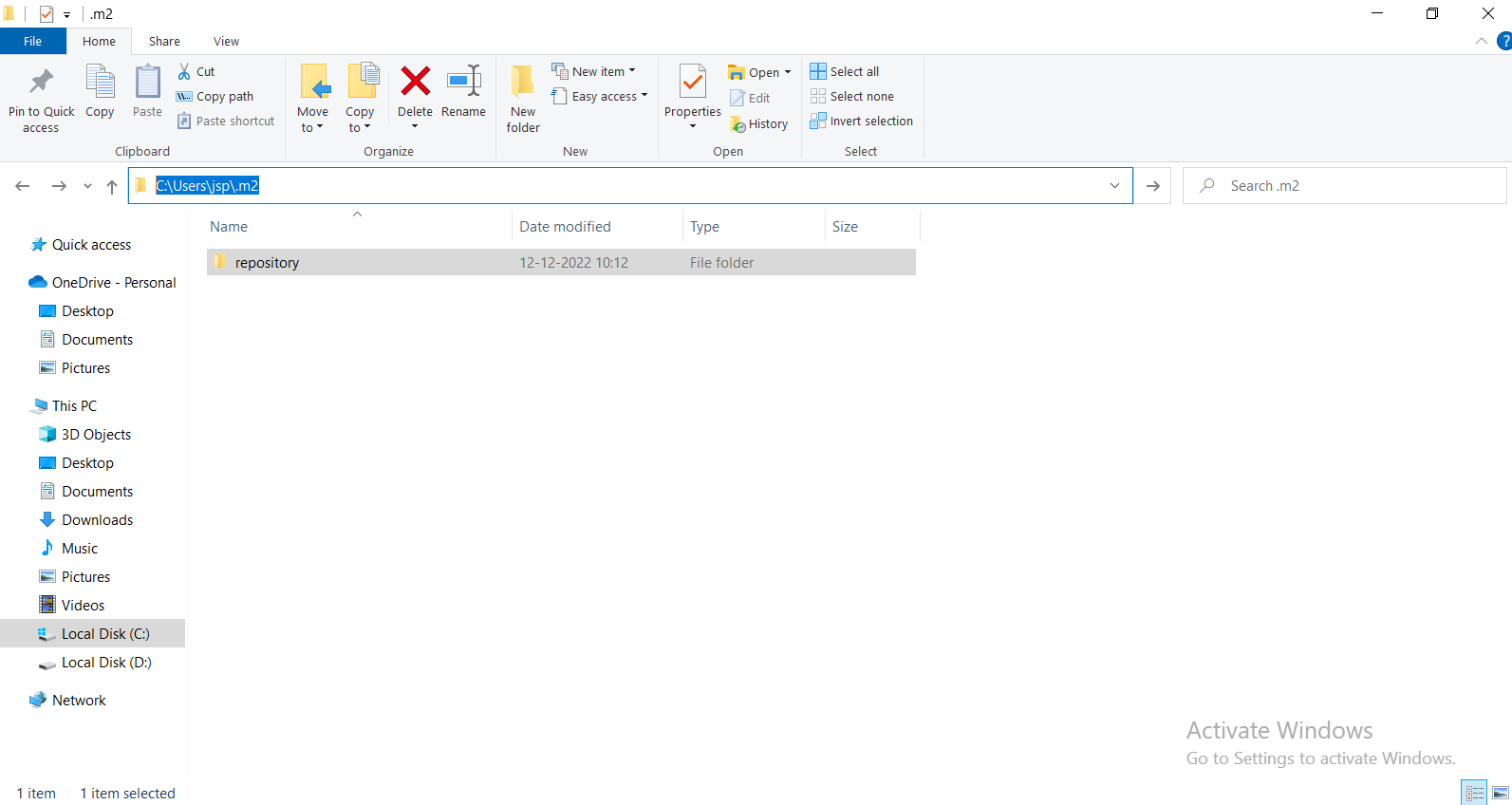
Local Repository🡪Central Repository🡪Remote Repository.

If dependency is not found in these repositories, maven stops processing and throws an error.

### Local Repository:

Maven **local repository** is located in your local system. It is created by the maven when you run any maven command.

By default, maven local repository is %USER\_HOME%/.m2 directory. For example:



## Central Repository:

Maven **central repository** is located on the web. It has been created by the Apache maven community itself.

The path of central repository is: [http://repo1.maven.org/maven2/](https://repo1.maven.org/maven2/).

## Remote Repository:

Maven **remote repository** is located on the web. We can define remote repository in POM.xml

Dependency:

In the context of Maven, **a dependency is simply a JAR file used by a Java application.** Based on the POM file, Maven will download and add the JAR file to our Java path. Java will then be able to find and use the classes in the JAR file.

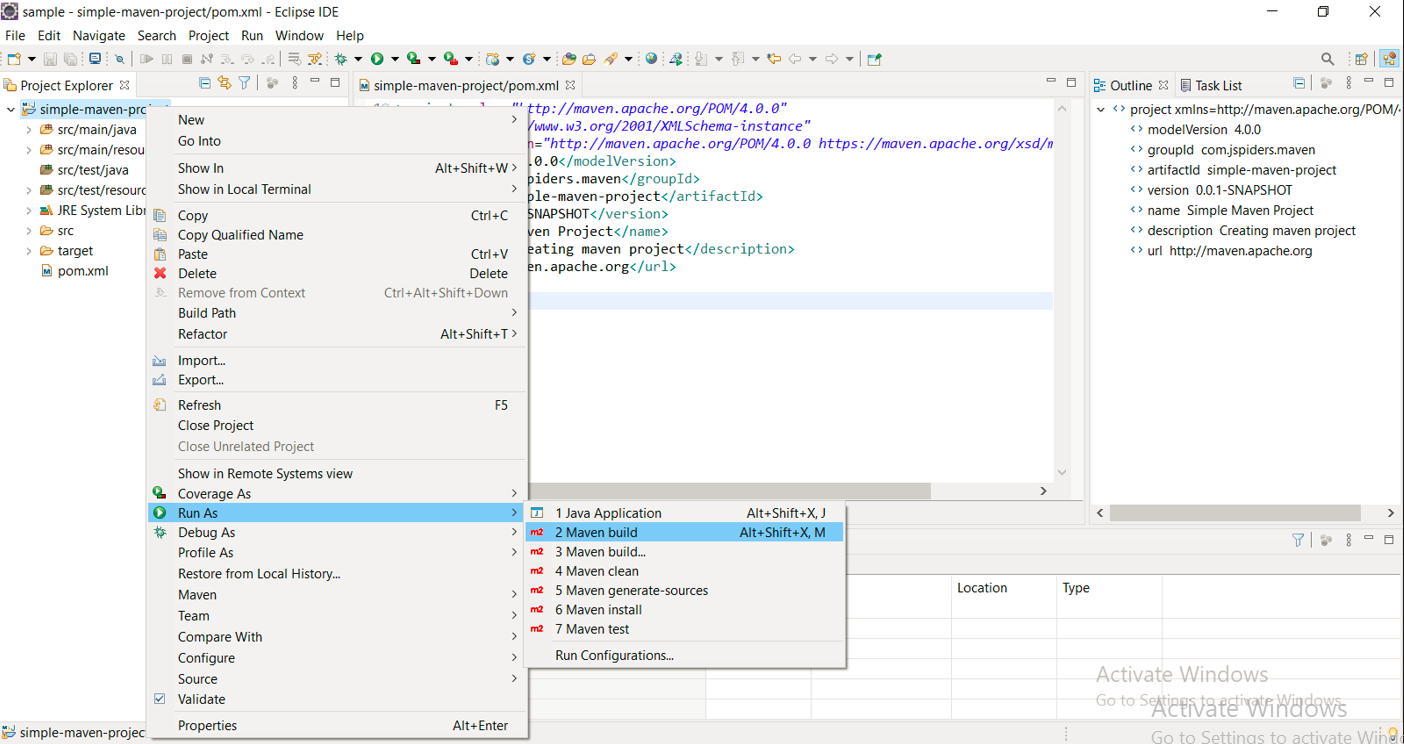
## POM (Project Object Model):

A Project Object Model or POM is the fundamental unit of work in Maven. It is an XML file that contains information about the project and configuration details used by Maven to build the project.

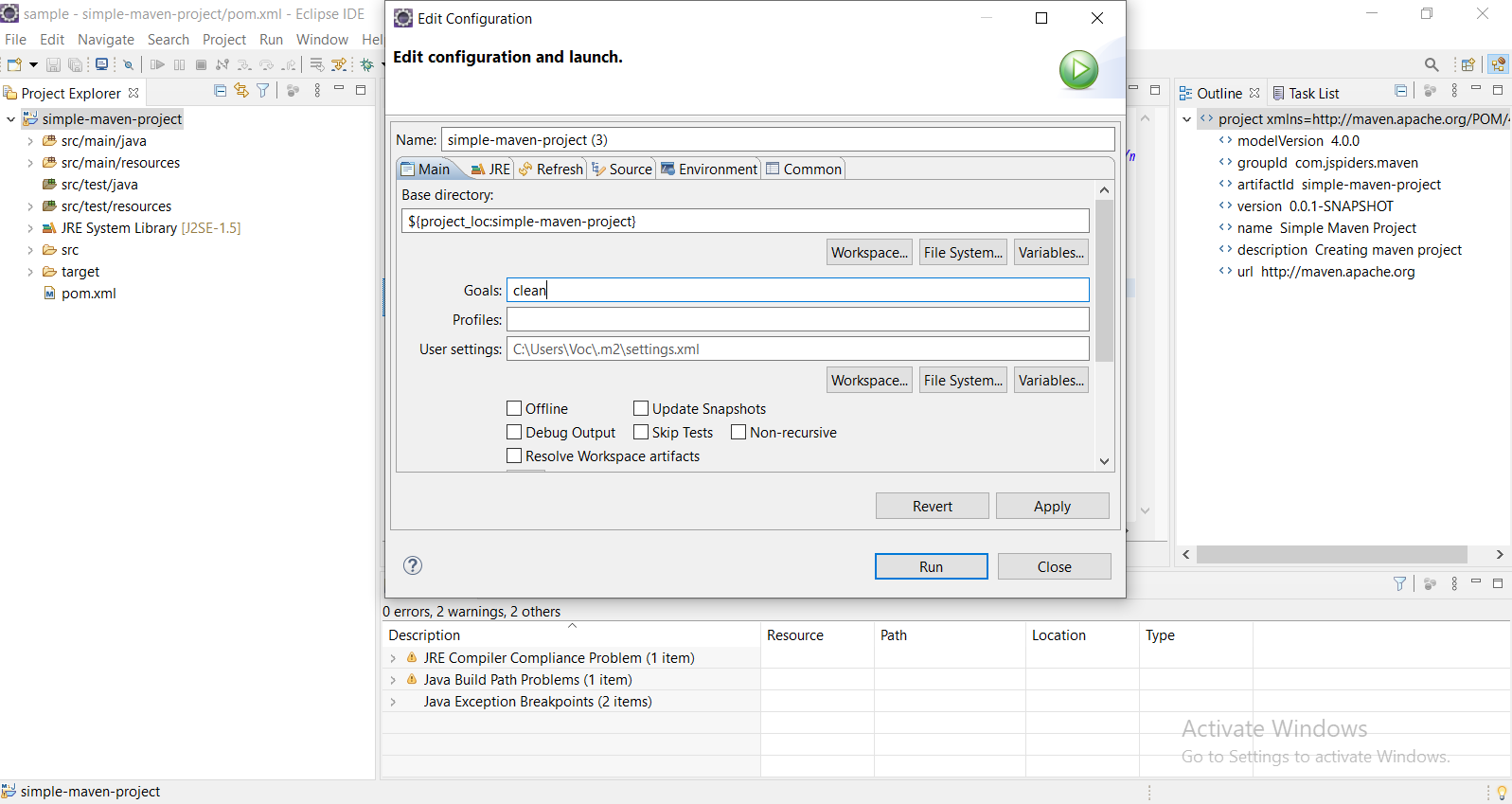
When executing a task or goal, Maven looks for the POM in the current directory. It reads the POM, gets the needed configuration information, then executes the goal.

## Building the Maven Project in Eclipse.

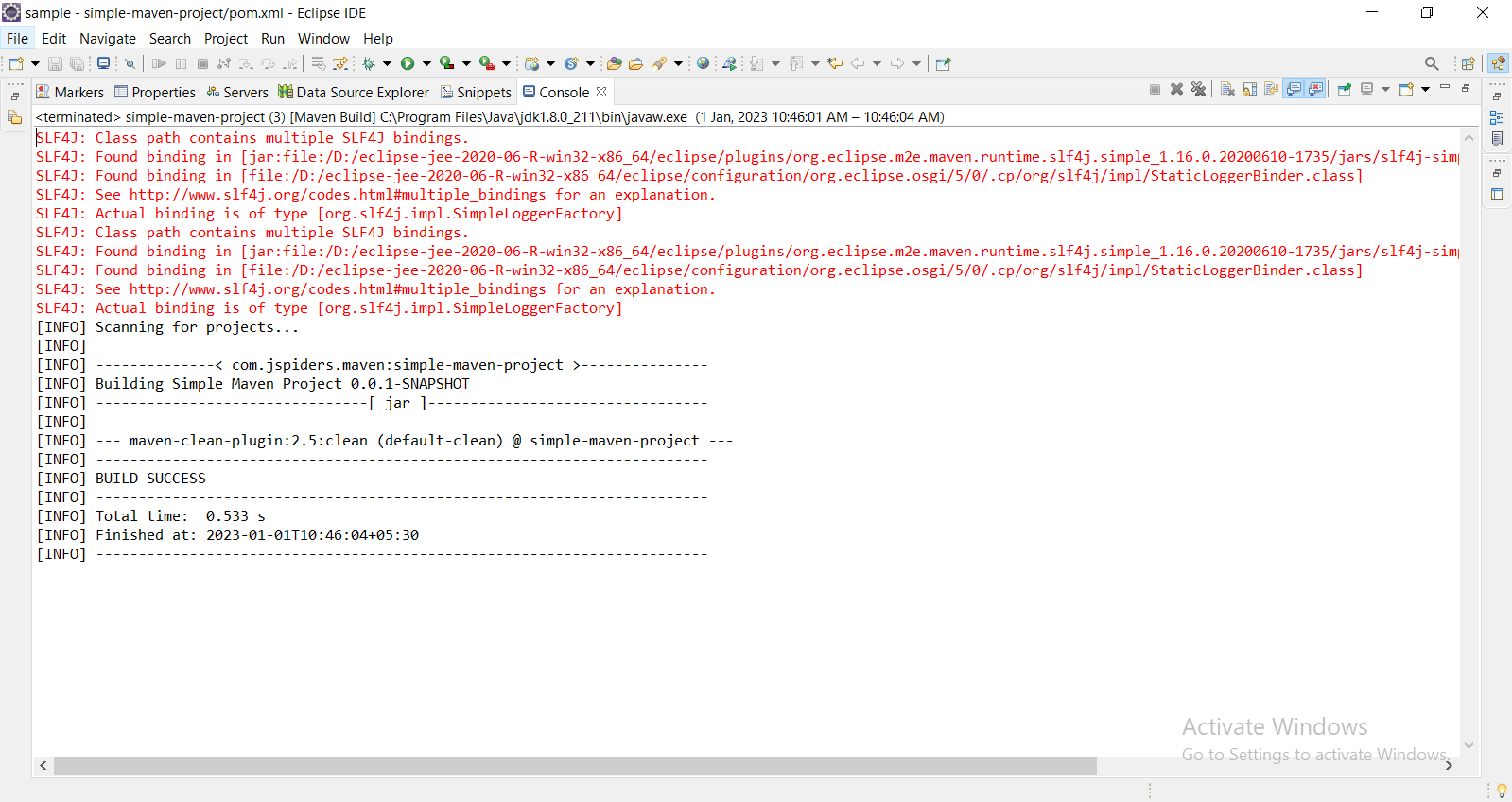
* Step 1: First of all, select the project and go to “Run As -> Maven Build”.



* Step 2: The “Edit Configuration” popup window will open. Enter the “Goals” as “clean” to build the project and click on the Run button.

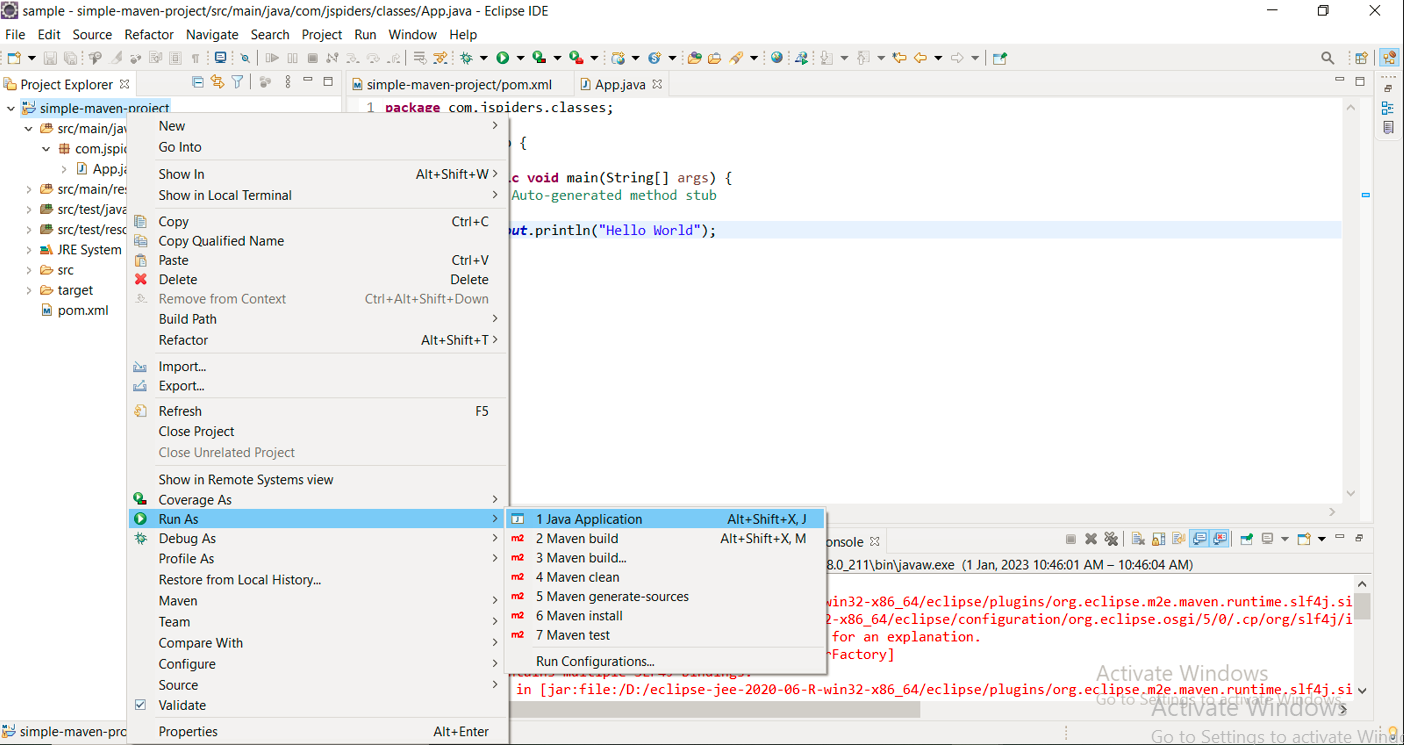


* Step 3: After Clicking on Run button. Maven project will be builded.

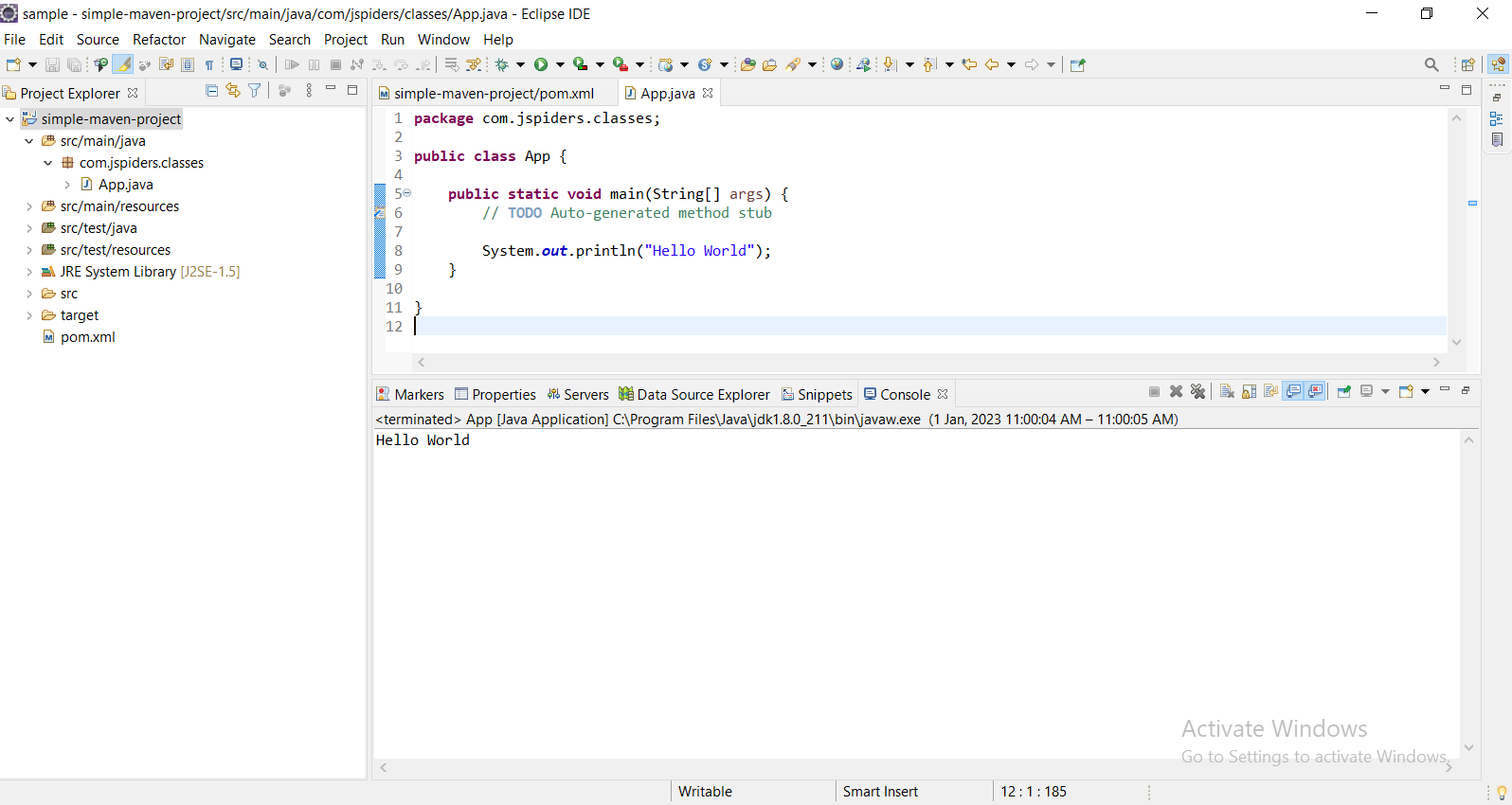


## Running the Maven Project.

* Step 1: select the project and go to “Run As -> Java Application”.

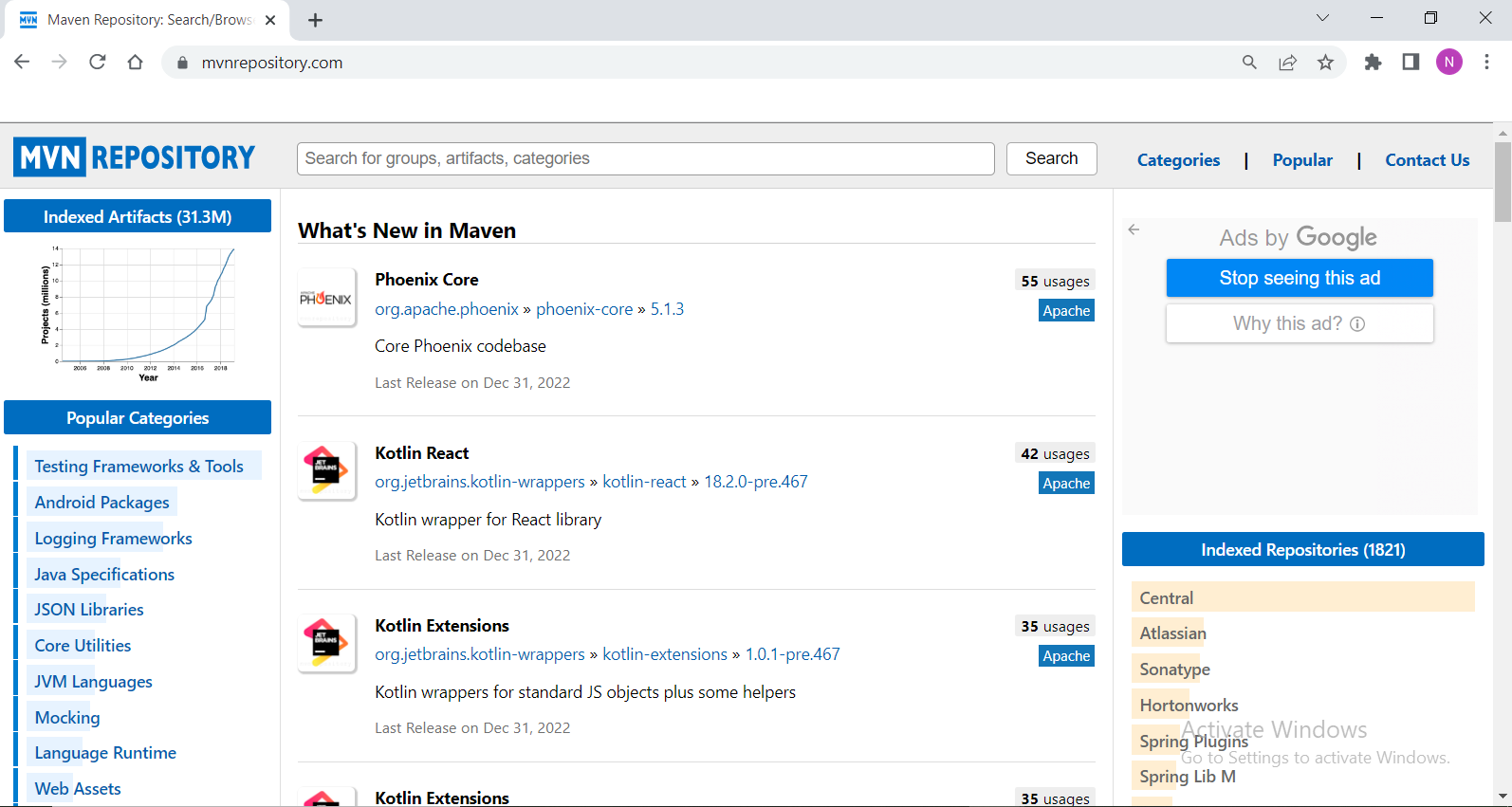


* Step 2: After above Step, the output will be display on the console.



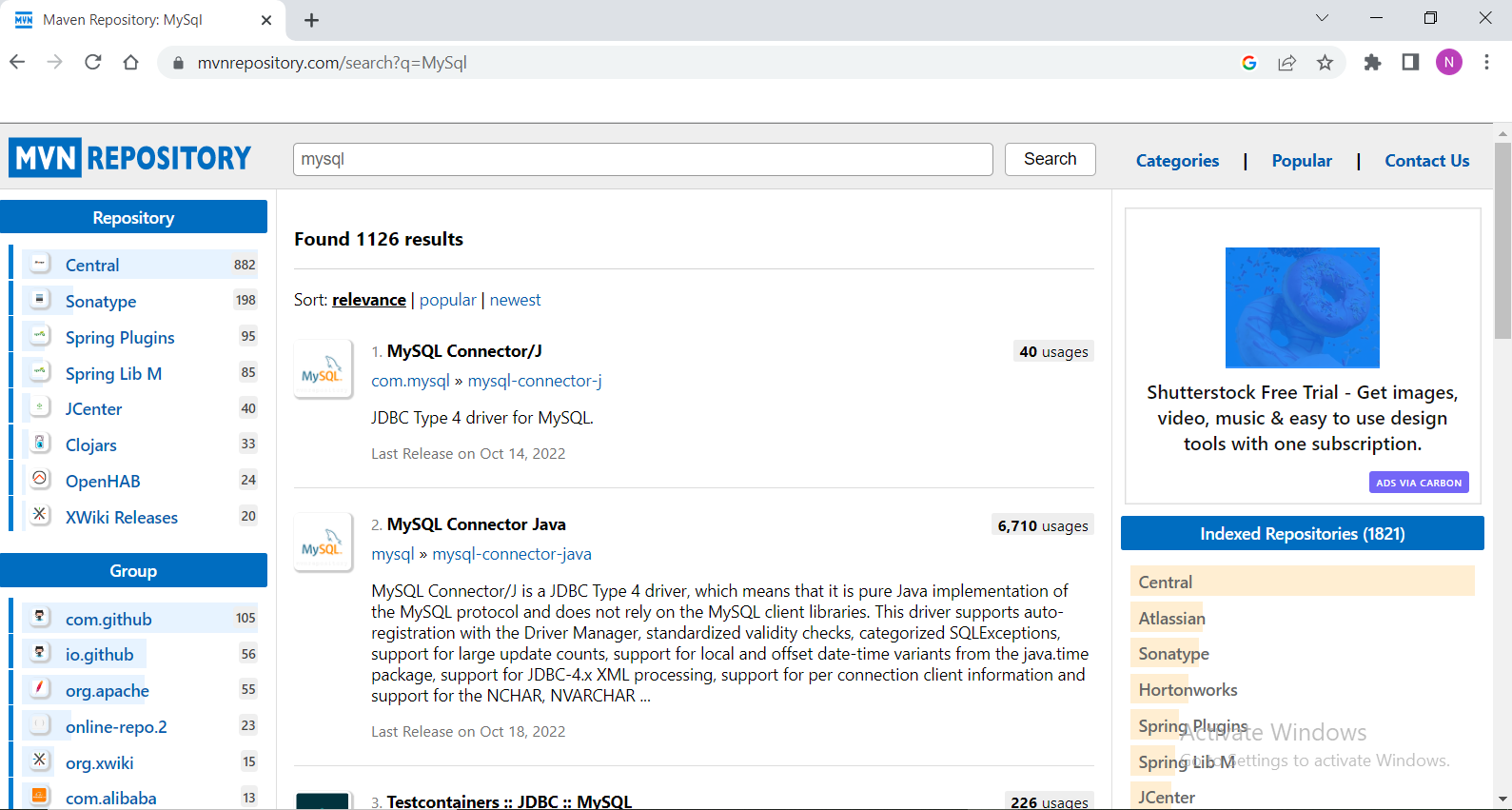
## Adding Dependencies to POM.xml through Maven Repository.

## Step 1: Visit the <https://mvnrepository.com/>

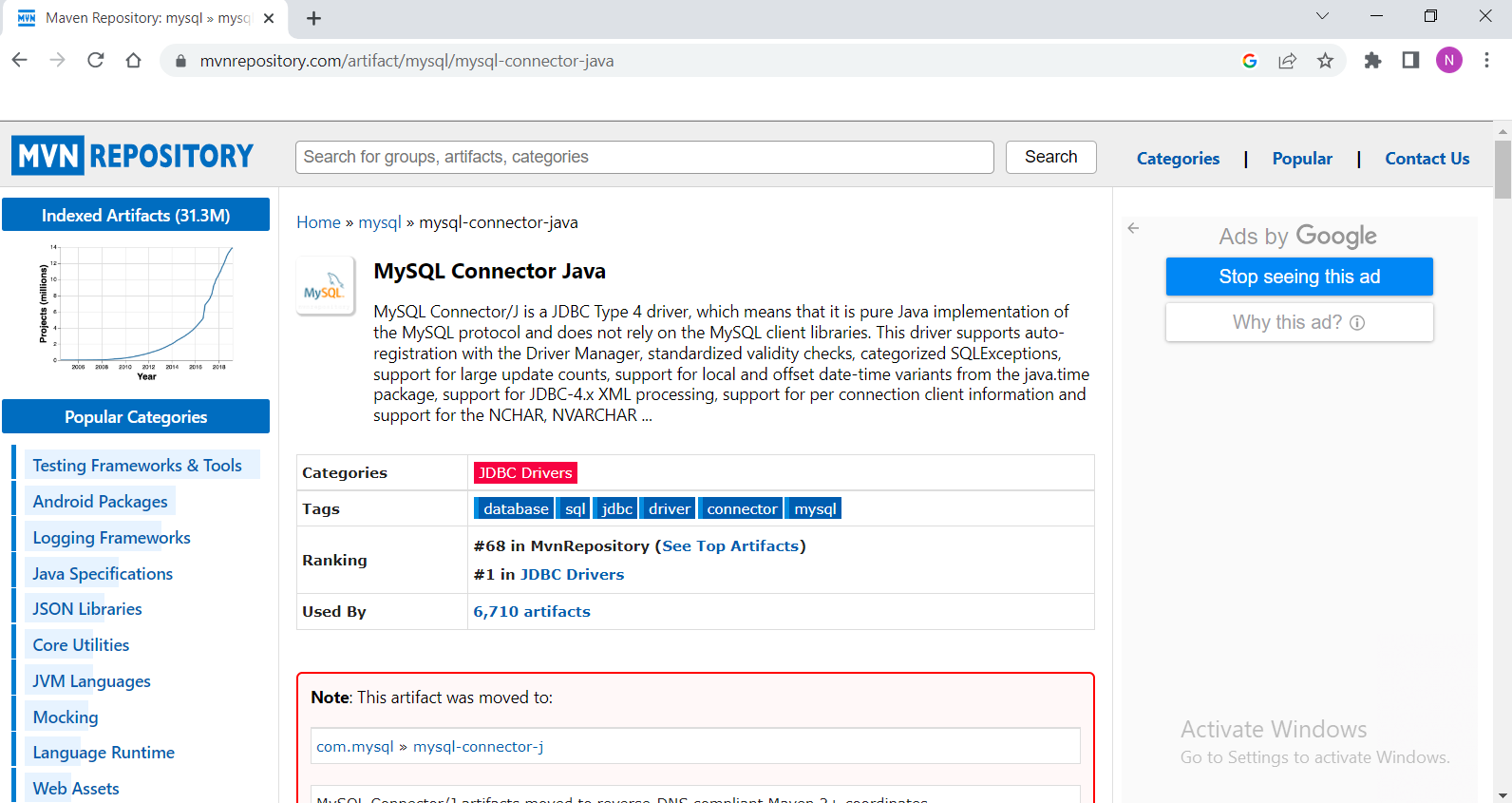


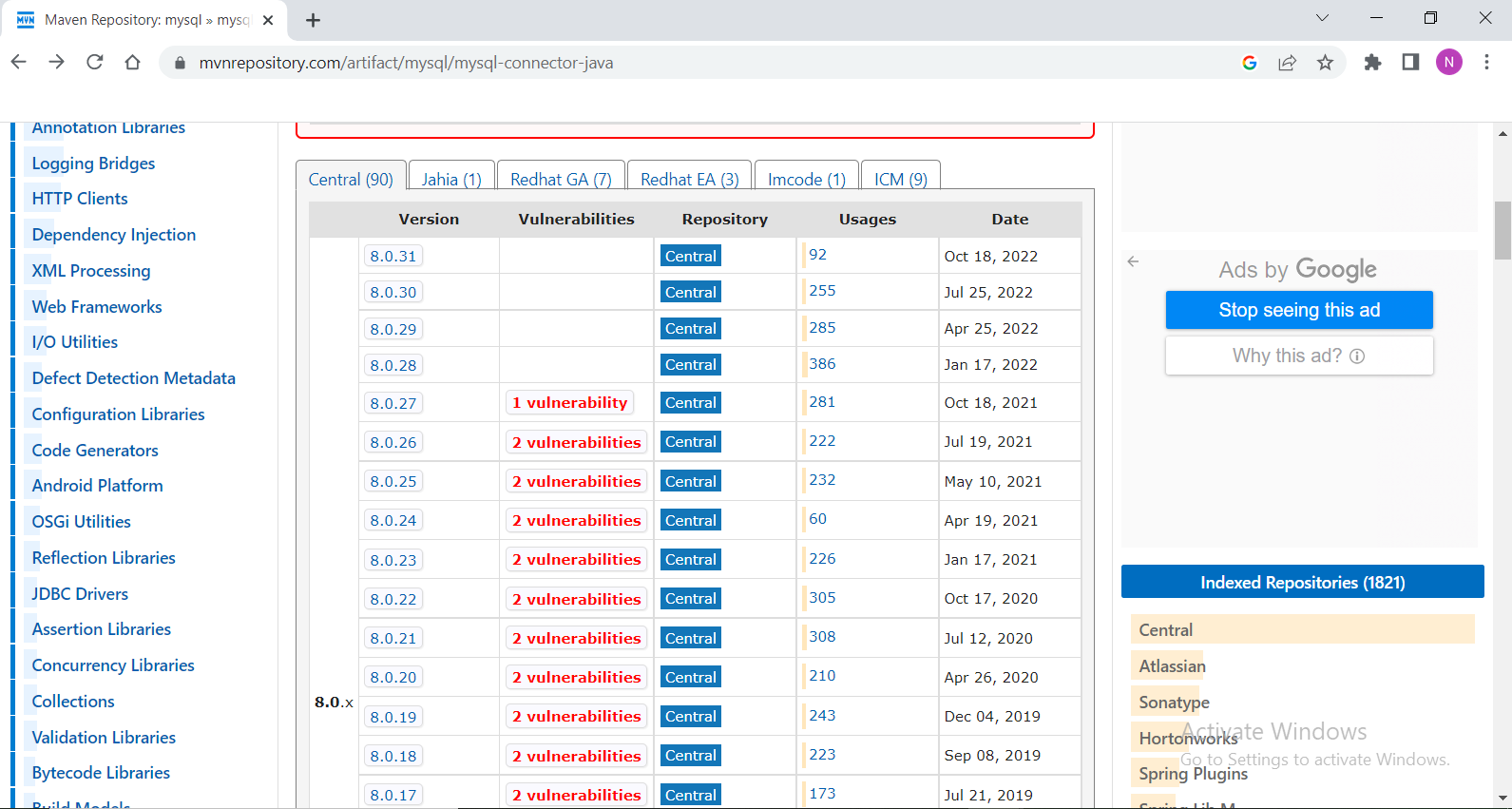
* Step 2: Click on Search bar and enter groups or artifacts or categories.

For example: If you want to add MySQL Connector Java dependency, just enter MySQL, then you can see the following page.

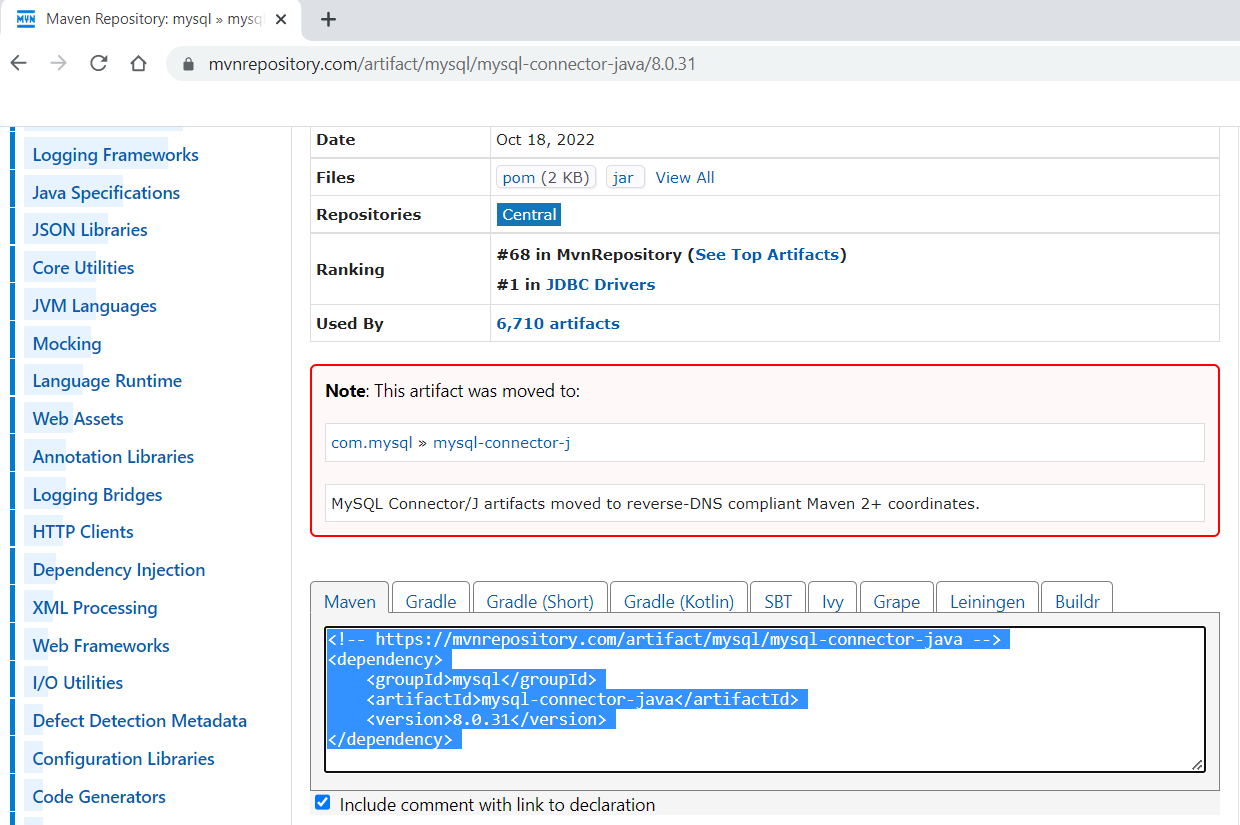


* Step 3: Click on MySQL Connector Java.

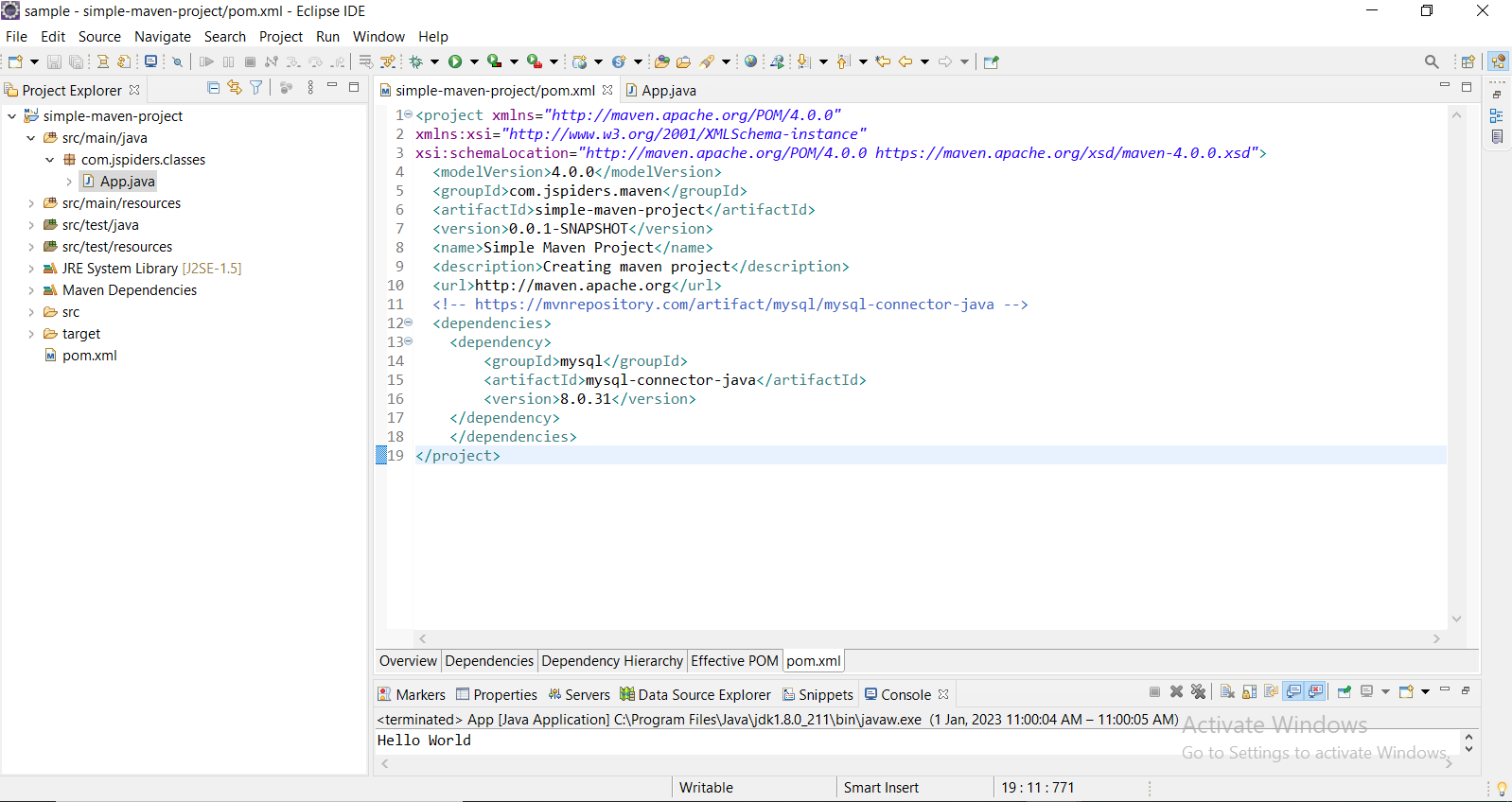




* Step 3: Select required version. Copy the selected xml and paste it into POM.xml

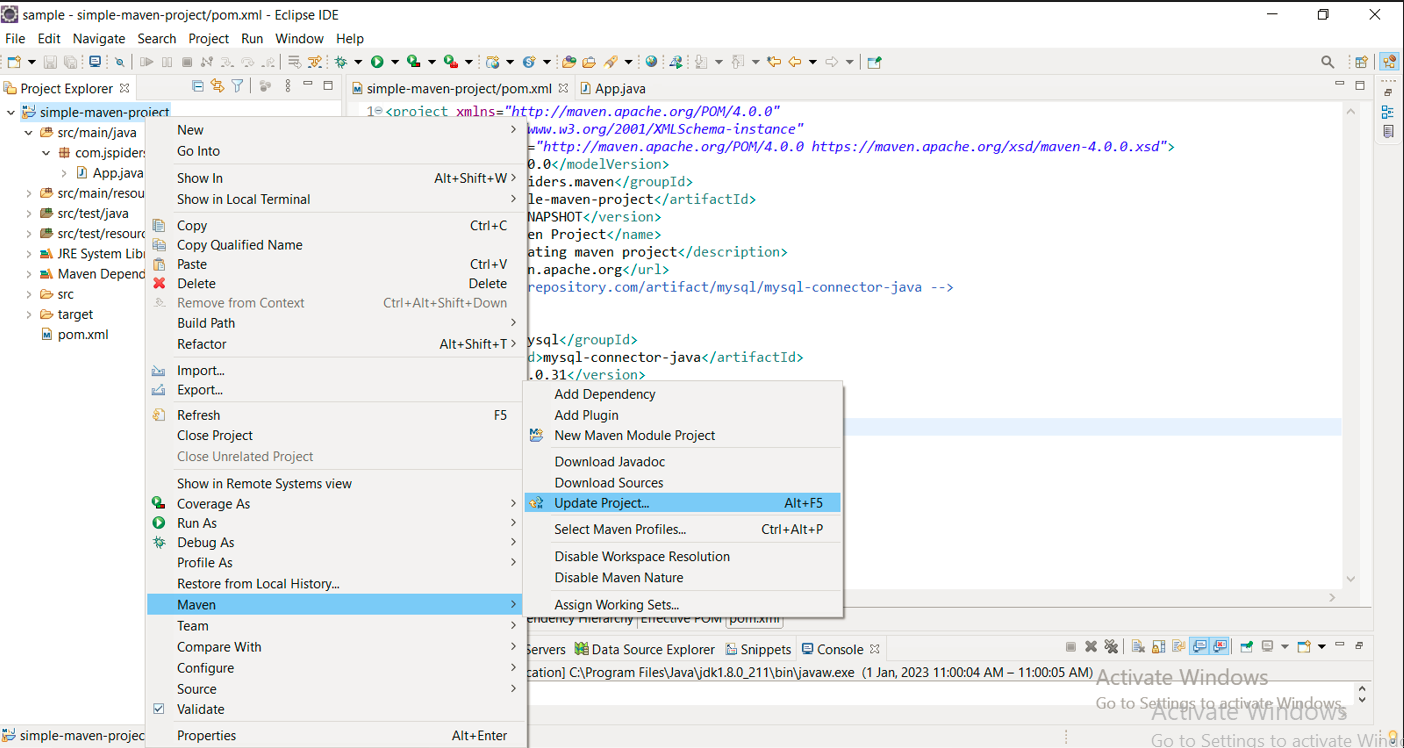


* Step 4: After adding dependency the POM.xml will be look like.



## Updating the Maven Project.

Step 1: Select the project and go to “Maven-> Update Project”.



* Step 2: After Step 1, you will get the following screen, click on particular project name checkbox which you need to update and click on Force Update of Snapshot/Releases. And click on Ok.

