How to Create a Maven Project in Eclipse

What is Maven?

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. The tool provides allows developers to build and document the lifecycle framework.

Maven is written in Java and is used to build projects written in C#, Scala, Ruby, etc. Based on the Project Object Model (POM), this tool has made the lives of Java developers easier while developing reports, checks build and testing automation setups.

Maven focuses on the simplification and standardization of the building process, taking care of the following:

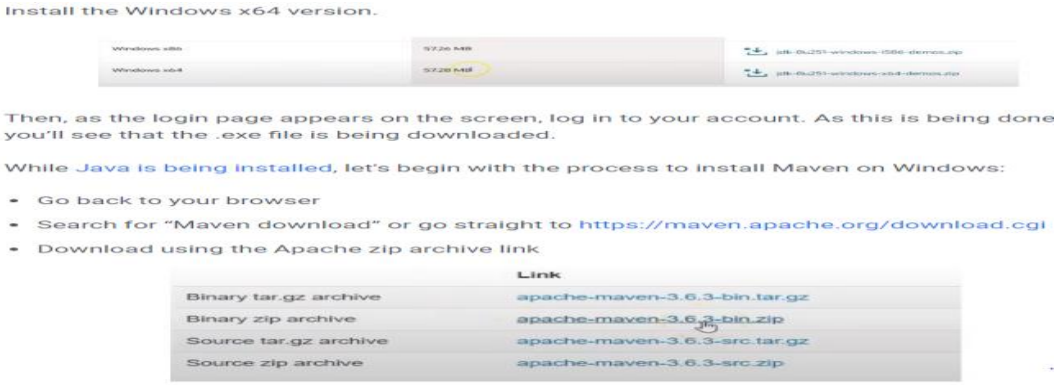
Builds
Documentation
Dependencies
Reports
SCMs
Distribution
Releases
Mailing list

Install Maven on Windows

First, we need JDK (Java development kit) installed in our system, which is then followed by installing Maven on Windows.

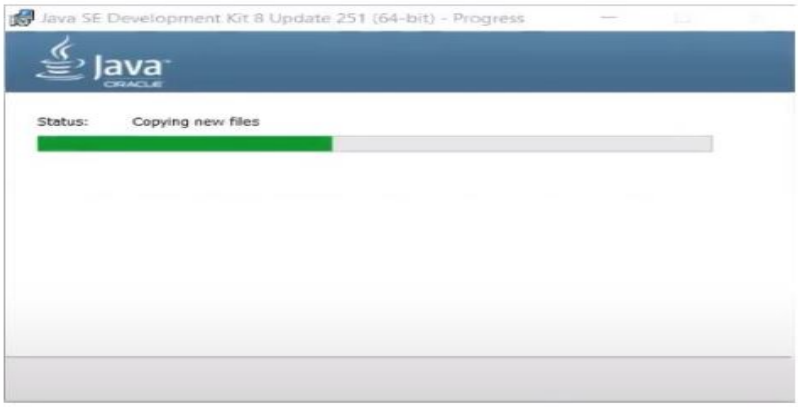
Let's begin with opening the chrome browser and searching for "JDK 8 download." There will be a link to to Oracle. Navigating to that page, you'll find the JDK for different platforms and operating systems.

Install the Windows x64 version.



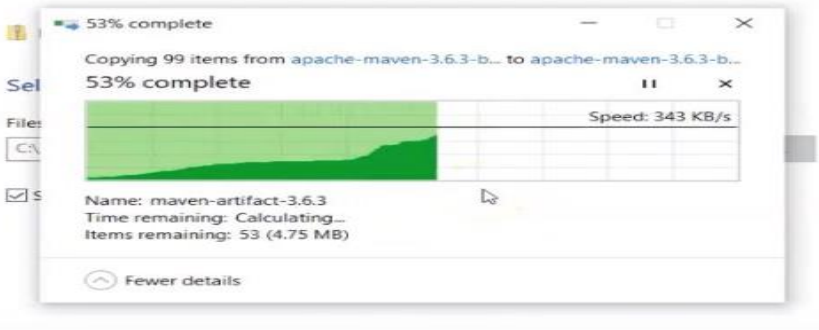
Once both JDK and Apache Maven are downloaded, we'll open the directory where the Java executables are.

Install the complete JDK.



After installing the JDK, we need to go to the Maven directory and extract the files.

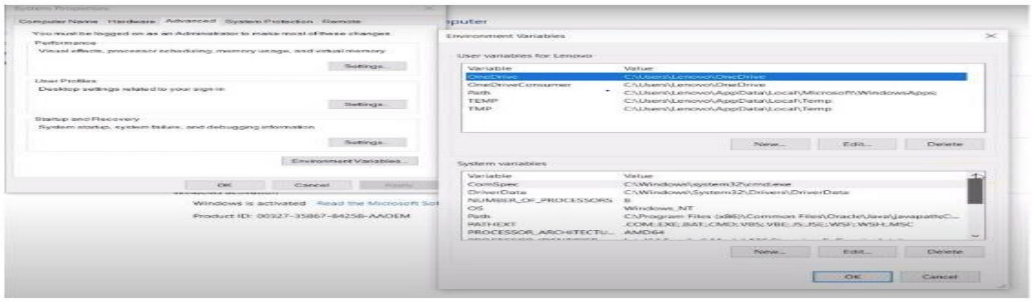
After installing the JDK, we need to go to the Maven directory and extract the files.



The file we get can be renamed and placed in the C drive, and that will be the Maven home path we'll use.

Then do the following:

- Go to the system properties
- In the system properties, go to the advanced system settings option
- In the dialog box that appears, go to the environment variables to set the path



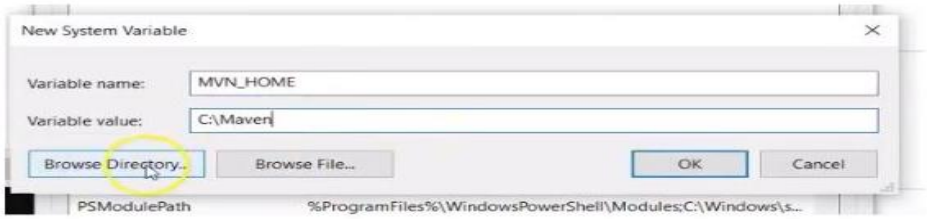
The next step is to open the CMD prompt.

In the command prompt, type the command to check the Java version:

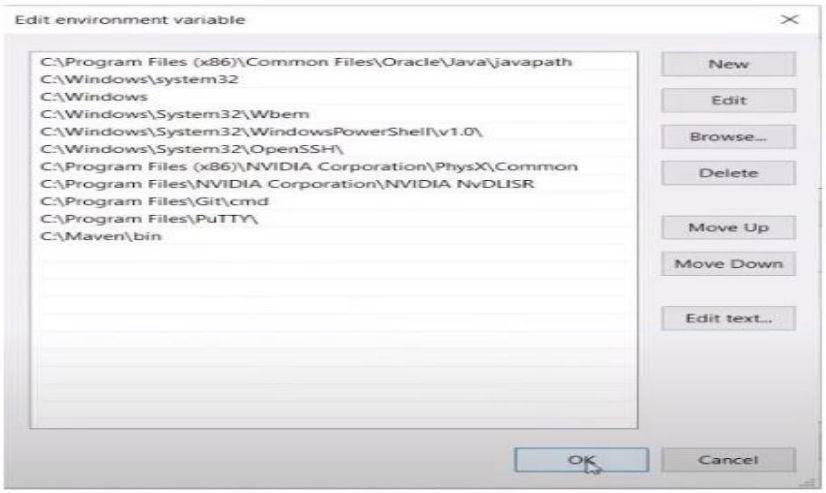
The version of Java appears on the screen.

Then come back to the dialog box and click on new, to set up a new environment variable.

In the name, enter MVN_HOME. In the variable value, paste the Maven directory location in the C drive.



Then, in the path variable, we have to add the bin directory.



Press OK on all the dialog boxes.

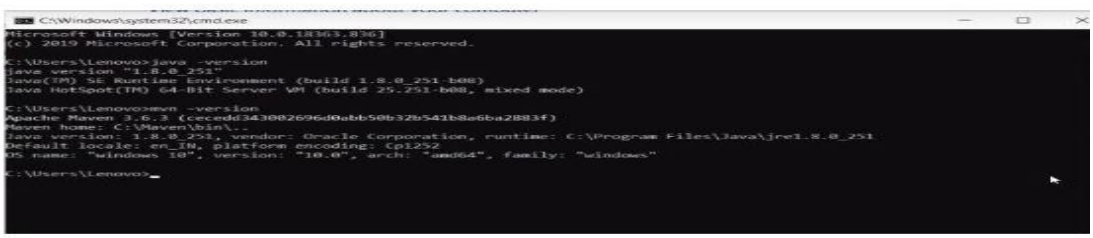
The Maven installation is done. To confirm that Maven is installed, go back to the command prompt window.

Check the Java version:

```
java -version
```

Followed by the Maven version:

```
mvn -version
```



Here, we can see both JDK and Maven installed on our system.

How to Create a Maven Project in Eclipse

Basics of Eclipse Demo –

Maven project in EclipseConclusionWant to up Your Skills in Software? Maven is primarily used for Java-based projects—one of the world's most widely used programming languages. When we talk about Java, Eclipse is the integrated development environment (IDE) that often comes to mind.

Eclipse is one of the most popular IDEs for Java and Android application development. It provides an excellent plugin, M2Eclipse, which integrates Maven and Eclipse.

Demo - Maven project in Eclipse

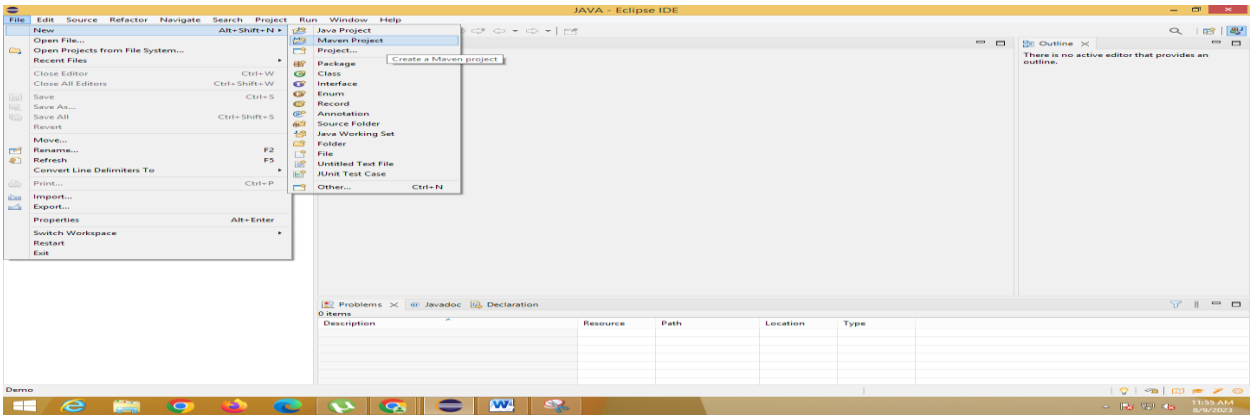
The first step is to open Eclipse, which comes with the integrated Maven environment. For this demo, we are using the Oxygen version of eclipse.

After opening Eclipse, choose the workspace you want to use.

The Eclipse window opens on the screen. Since there aren't any projects yet, complete the following steps:

- Go to the File option
- In the drop-down menu, select New
- Select the Project option

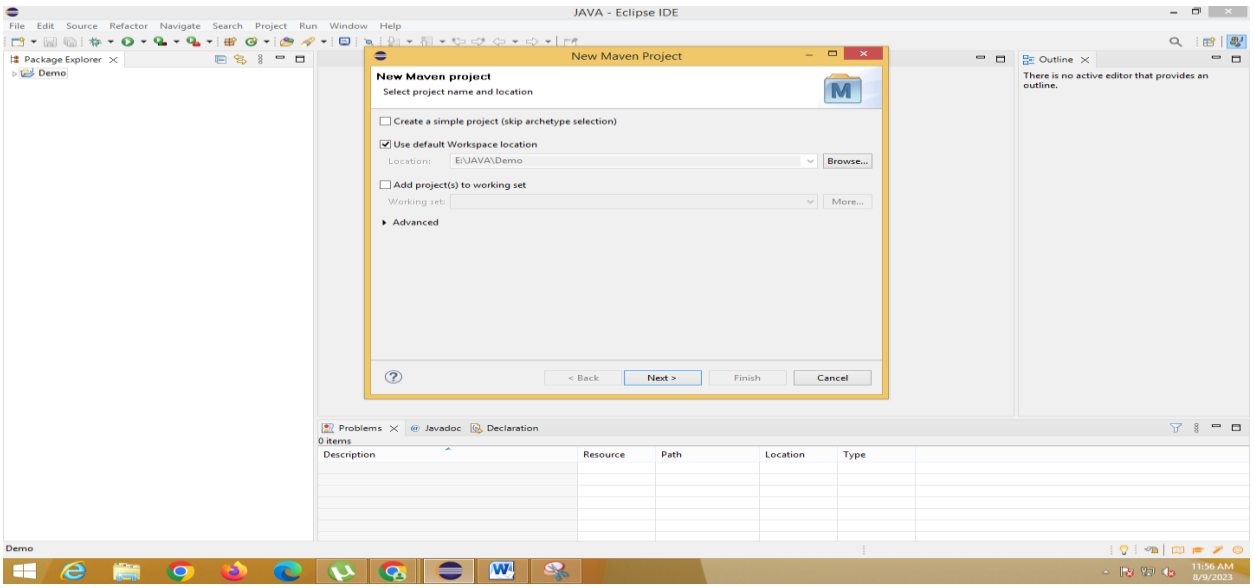
If you want to create a Java project, you can select the "Java Project" option. Since we are not creating a Java project specifically, we have chosen the "Project" option.



The dialog box that appears on the screen will display different types of projects.

Select the Maven Project option

Click on Next

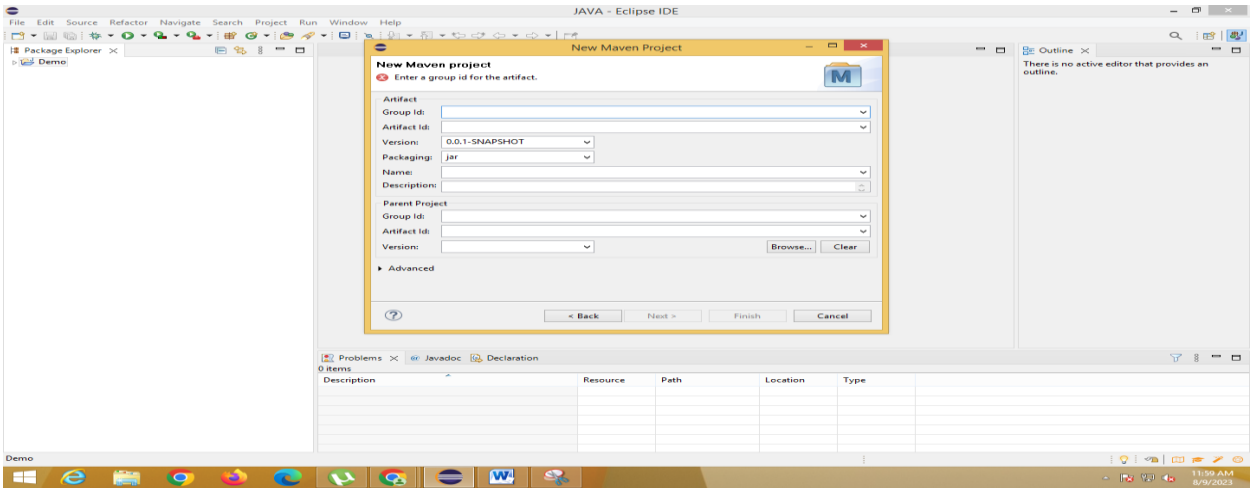


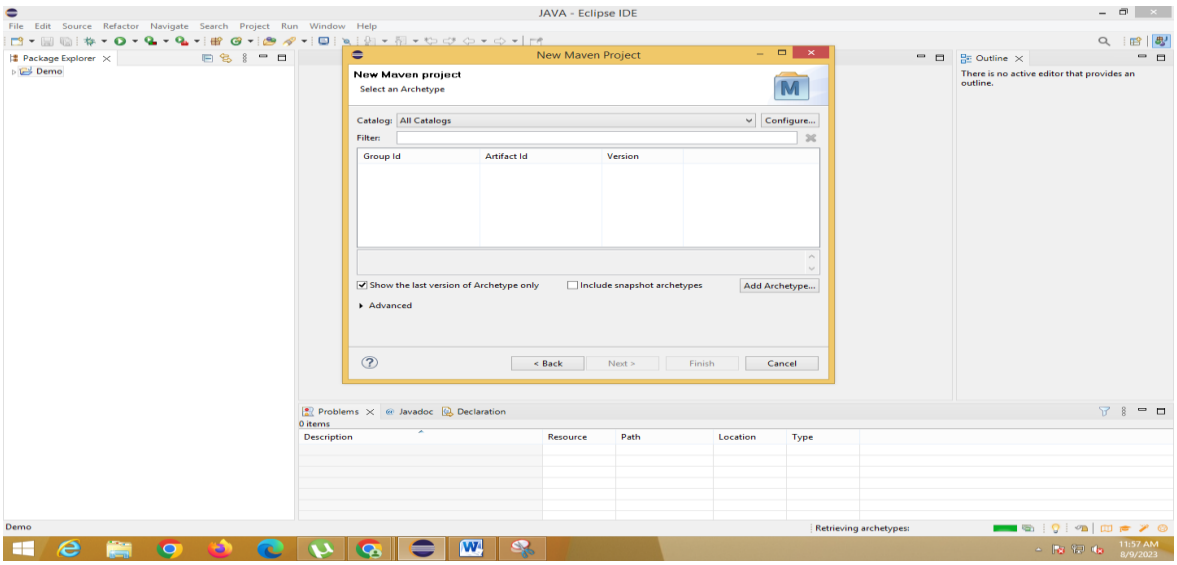
A dialog box will appear. Select the default workspace.

Click on “Next”

Several Group IDs, Artifact IDs, and Versions will then appear.

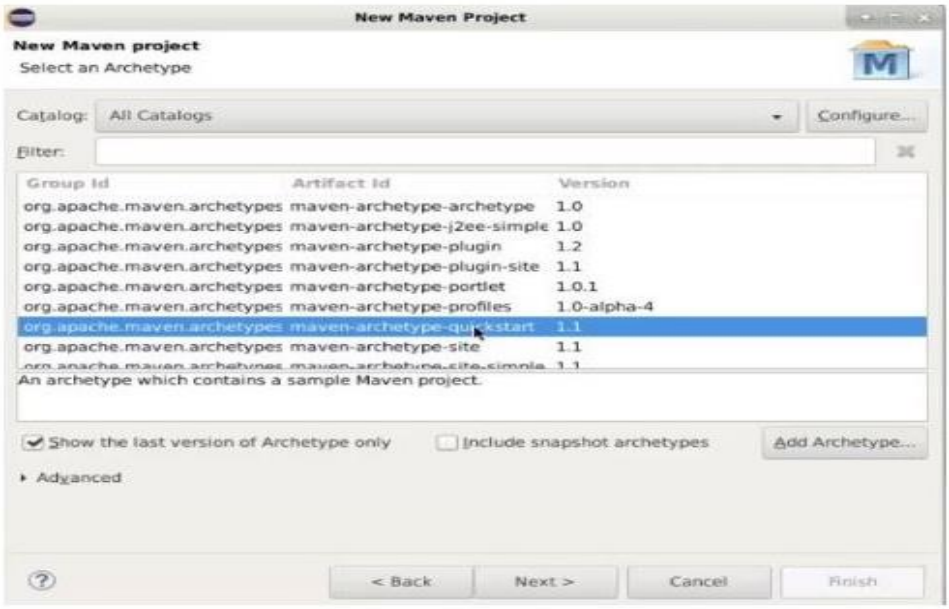
Select a plugin there and click on “Next”





Several Group IDs, Artifact IDs, and Versions will then appear.

- Select a plugin there and click on “Next”



In the next dialog box that appears, you’ll complete the following steps:

- Enter the Group ID

“com.simplilearn”

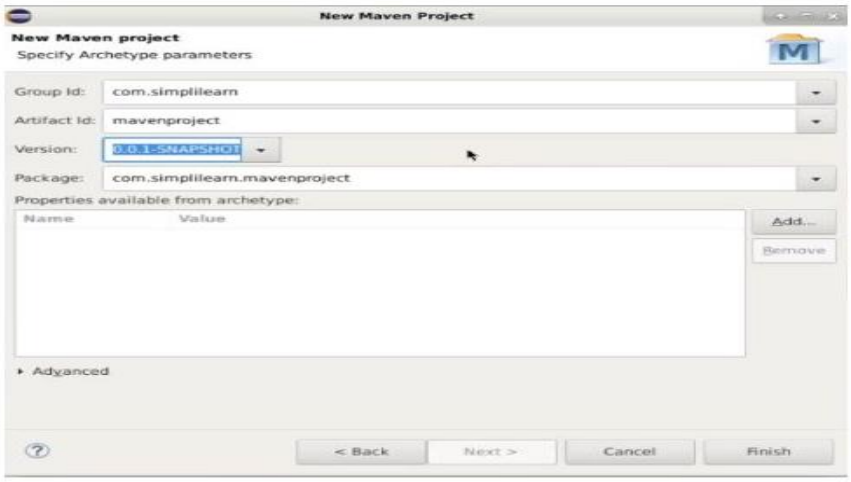
- Enter the Artifact ID

“mavenproject”

- The version will appear on the screen

These items can all be modified at a later time if needed.

- Click on “Finish”



The project is now created.

- Open the pom.xml file

You can see all the basic information that you have entered on the screen, such as the Artifact ID, Group ID, etc.

You can see the junit dependencies have been added.

This process takes place by default in Eclipse. There will also be some by default test cases.

There you can find AppTest.java to be a default test case.

When you click on that, you can see the test cases written in JUnit on your Eclipse screen.

If we try to remove certain dependencies from our file, we will receive error messages. To troubleshoot this, complete the following steps:

- Go to another tab: mavenproject/pom.xml
- Delete any dependencies
- Save the file

Immediately, there will be several error messages in the AppTest.java.

Return to the previous screen and undo the deletion. The errors that occurred will disappear.