What is Maven?

Apache Maven is a popular build automation and project management tool used primarily for Java projects, although it can be used with other languages and project types as well. It is an open-source software developed by the Apache Software Foundation.

Maven provides a way to define a project's structure, dependencies, and build process in a declarative XML file called "pom.xml" (Project Object Model). This file acts as the project's configuration file and contains information such as project metadata, dependencies, build plugins, and other configuration settings.

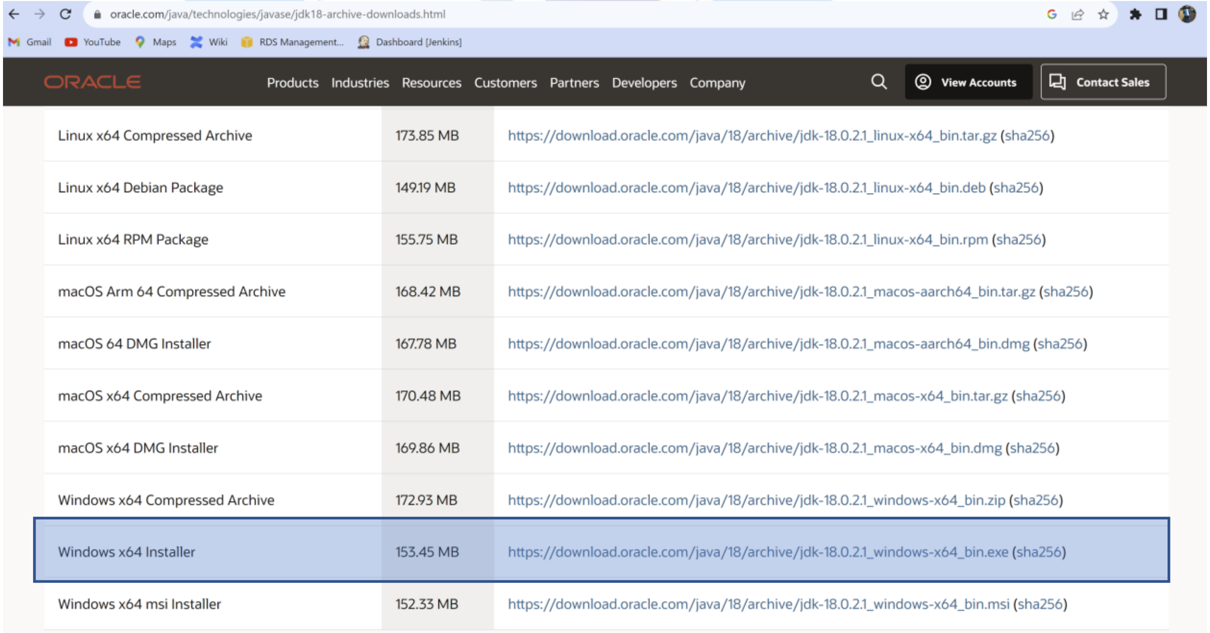
Install Java on widnows10:

To install Java on Windows 10, you can follow these step-by-step instructions:

Step 1:

- Download Java JDK (Java Development Kit):
- Go to the Oracle website: https://www.oracle.com/java/technologies/javase/jdk18-archive-downloads.html

  (or the website of your preferred JDK provider).



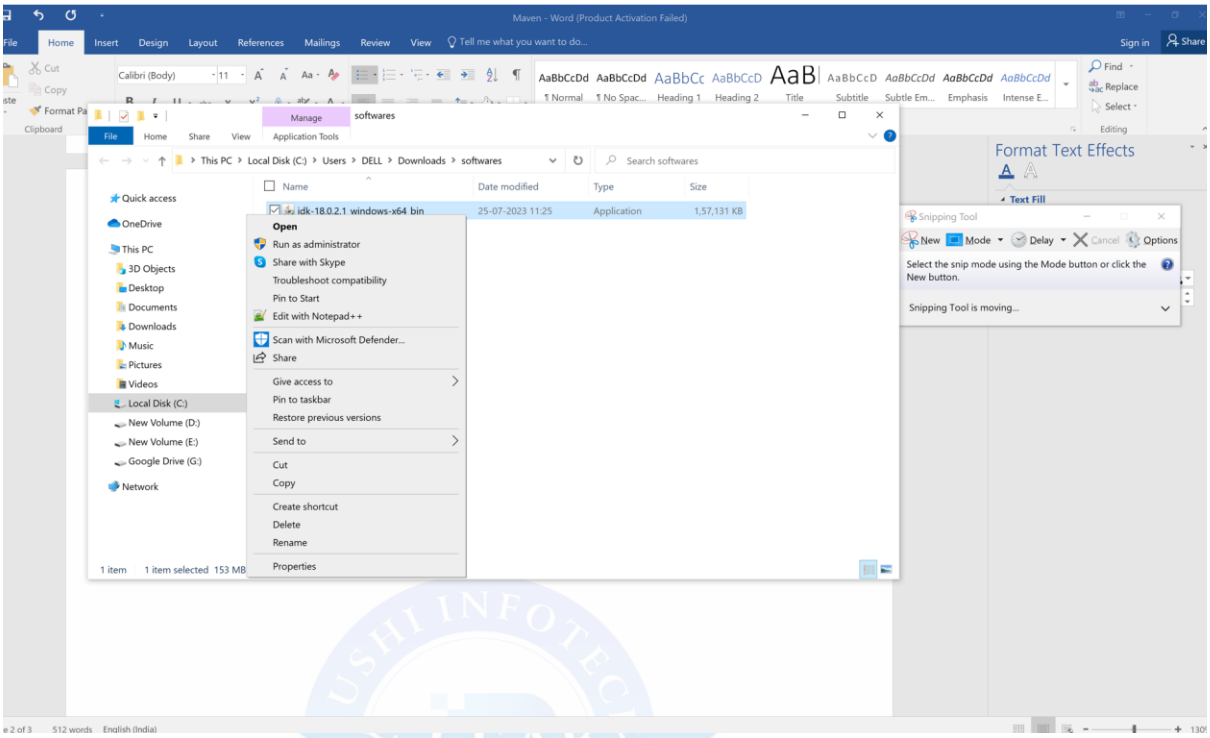
oracle.com/java/technologies/javase/jdk18-archive-downloads.html

Linux x64 Compressed Archive	173.85 MB	https://download.oracle.com/java/18/archive/jdk-18.0.2.1_linux-x64_bin.tar.gz (sha256)
Linux x64 Debian Package	149.19 MB	https://download.oracle.com/java/18/archive/jdk-18.0.2.1_linux-x64_bin.deb (sha256)
Linux x64 RPM Package	155.75 MB	https://download.oracle.com/java/18/archive/jdk-18.0.2.1_linux-x64_bin.rpm (sha256)
macOS Arm 64 Compressed Archive	168.42 MB	https://download.oracle.com/java/18/archive/jdk-18.0.2.1_macos-aarch64_bin.tar.gz (sha256)
macOS 64 DMG Installer	167.78 MB	https://download.oracle.com/java/18/archive/jdk-18.0.2.1_macos-aarch64_bin.dmg (sha256)
macOS x64 Compressed Archive	170.48 MB	https://download.oracle.com/java/18/archive/jdk-18.0.2.1_macos-x64_bin.tar.gz (sha256)
macOS x64 DMG Installer	169.86 MB	https://download.oracle.com/java/18/archive/jdk-18.0.2.1_macos-x64_bin.dmg (sha256)
Windows x64 Compressed Archive	172.93 MB	https://download.oracle.com/java/18/archive/jdk-18.0.2.1_windows-x64_bin.zip (sha256)
Windows x64 Installer	153.45 MB	https://download.oracle.com/java/18/archive/jdk-18.0.2.1_windows-x64_bin.exe (sha256)
Windows x64 msi Installer	152.33 MB	https://download.oracle.com/java/18/archive/jdk-18.0.2.1_windows-x64_bin.msi (sha256)

- Accept the license agreement and choose the appropriate version of JDK for your system (e.g., Windows x64 or x86, depending on your computer's architecture).
- Click on the download link to save the installer file on your computer.

Step 2:

- Run the JDK Installer:
- Once the download is complete, locate the downloaded JDK installer (e.g., jdk-11.x.x_windows-x64_bin.exe).
  Double-click on the installer to start the installation process.



Step 3:

- Start JDK Installation:
- The JDK installer will open. Click "Next" to proceed.

Step4:

- Choose Installation Directory:
- By default, the JDK will be installed in the "C:\Program Files\Java" directory. You can choose a different location if needed, but it's recommended to keep the default directory. Click "Next" to continue.

Step 5:

- Choose JDK Components:
- The installer will ask you to select which components you want to install. Usually, all components are selected by default, and you can leave them as is. Click "Next" to proceed.

Step 6: Install JDK:

- The installer will now start installing the JDK and its components. Wait for the installation to complete.

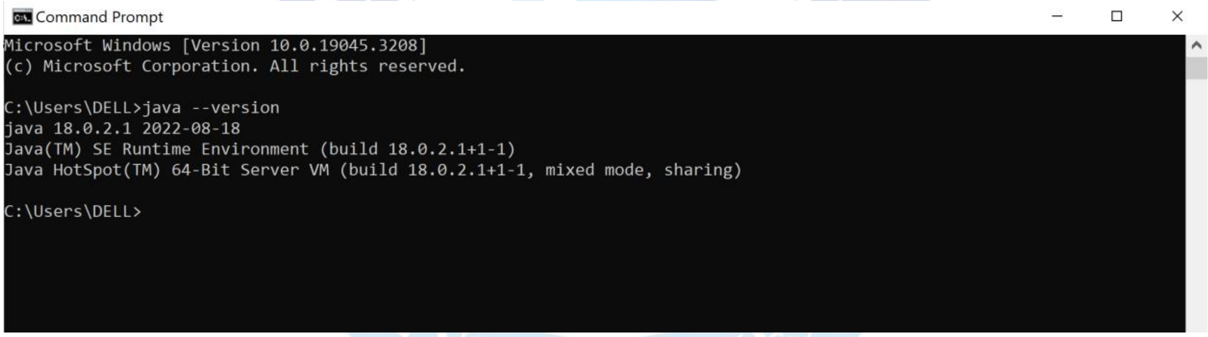
Step 7: Verify JDK Installation:

- Once the installation is finished, you should verify that Java is installed correctly. Open a Command Prompt or PowerShell window.

Type the following command to check the JDK version:

Command: java -version

You should see the installed Java version information printed on the screen.

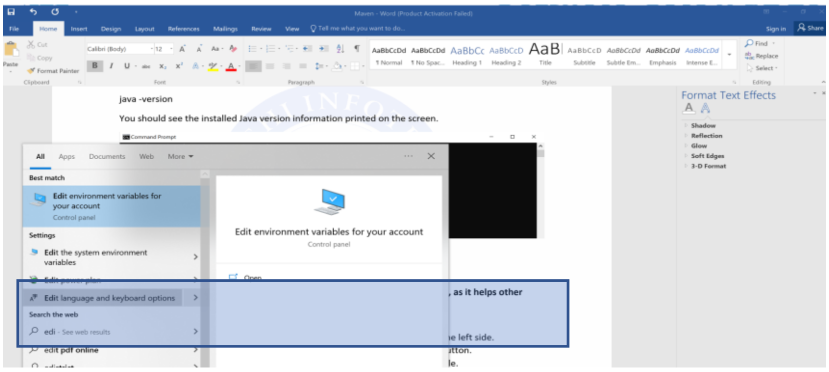


```
Microsoft Windows [Version 10.0.19045.3208]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL>java --version
java 18.0.2.1 2022-08-18
Java(TM) SE Runtime Environment (build 18.0.2.1+1-1)
Java HotSpot(TM) 64-Bit Server VM (build 18.0.2.1+1-1, mixed mode, sharing)

C:\Users\DELL>
```

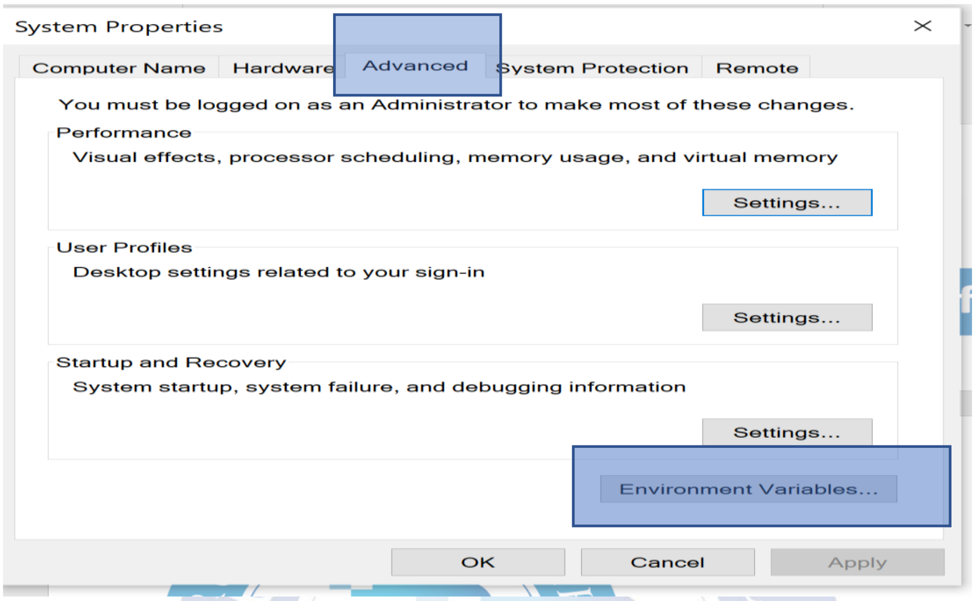
Step 8: Set JAVA_HOME Environment Variable

Setting the JAVA_HOME environment variable is optional but recommended, as it helps other programs find the JDK installation. To set the JAVA_HOME variable:

- Select "edit system environment variables" in the search bar



- In the System Properties window, click the "Environment Variables" button.



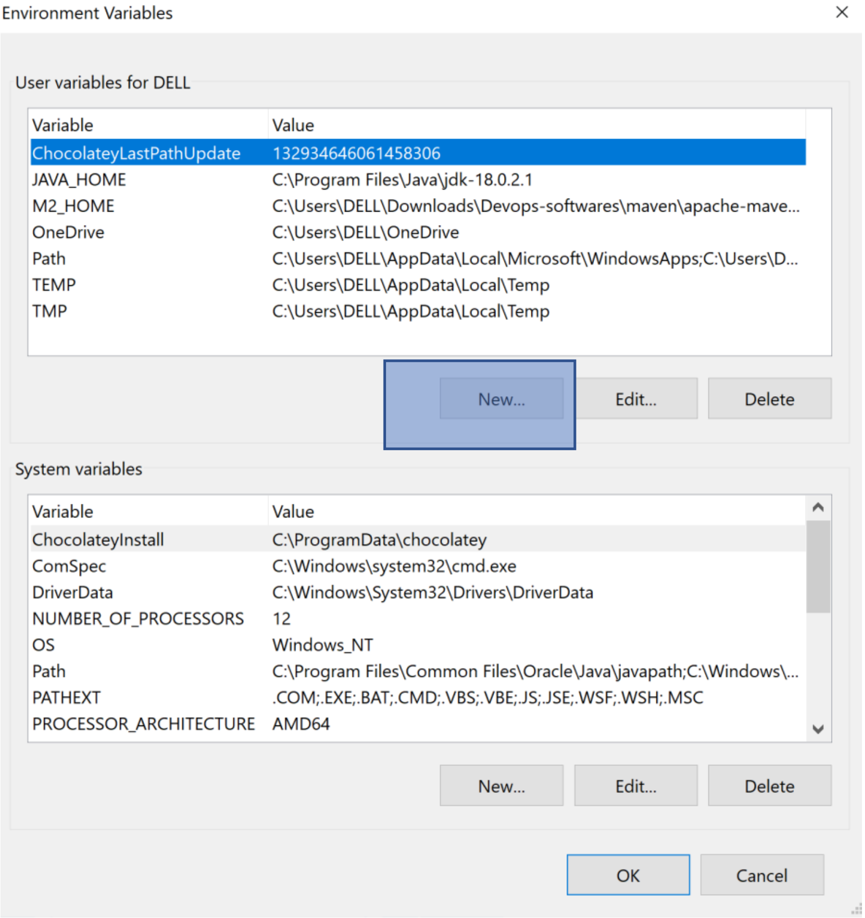
You must be logged on as an Administrator to make most of these changes.

Performance
Visual effects, processor scheduling, memory usage, and virtual memory

User Profiles
Desktop settings related to your sign-in

Startup and Recovery
System startup, system failure, and debugging information

- Under the "System variables" section, click "New" to add a new variable.



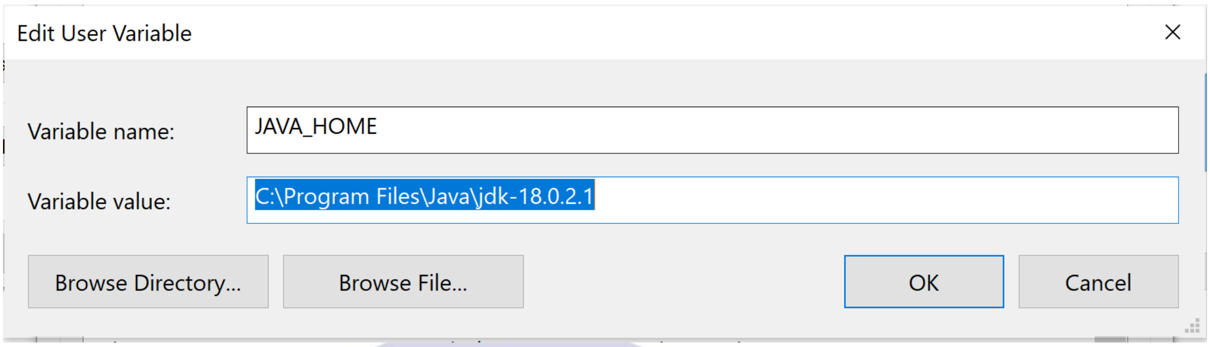
User variables for DELL

Variable	Value
ChocolateyLastPathUpdate	132934646061458306
JAVA_HOME	C:\Program Files\Java\jdk-18.0.2.1
M2_HOME	C:\Users\DELL\Downloads\Devops-softwares\maven\apache-mave...
OneDrive	C:\Users\DELL\OneDrive
Path	C:\Users\DELL\AppData\Local\Microsoft\WindowsApps;C:\Users\D...
TEMP	C:\Users\DELL\AppData\Local\Temp
TMP	C:\Users\DELL\AppData\Local\Temp

System variables

Variable	Value
ChocolateyInstall	C:\ProgramData\chocolatey
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	12
OS	Windows_NT
Path	C:\Program Files\Common Files\Oracle\Java\javapath;C:\Windows\...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64

Variable Name: JAVA_HOME
Variable Value: C:\Program Files\Java\jdk-18.0.2.1



Edit User Variable

Variable name: JAVA_HOME

Variable value: C:\Program Files\Java\jdk-18.0.2.1

- Set the variable name as "JAVA_HOME" and the variable value to the installation path of your JDK (e.g., "C:\Program Files\Java\jdk-11.x.x").
- Click "OK" to save the changes.

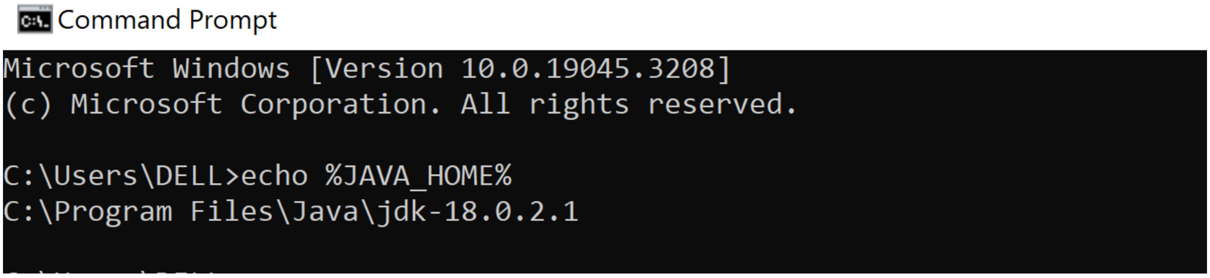
Step 9: Verify JAVA_HOME Environment Variable (Optional):

To verify that the JAVA_HOME variable is set correctly, open a Command Prompt or PowerShell window.

Type the following command:

Command: echo %JAVA_HOME%

It should display the path to your JDK installation.



```
Microsoft Windows [Version 10.0.19045.3208]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL>echo %JAVA_HOME%
C:\Program Files\Java\jdk-18.0.2.1
```

Congratulations! You have now successfully installed Java JDK on your Windows 10 system. You can start developing and running Java applications on your computer.

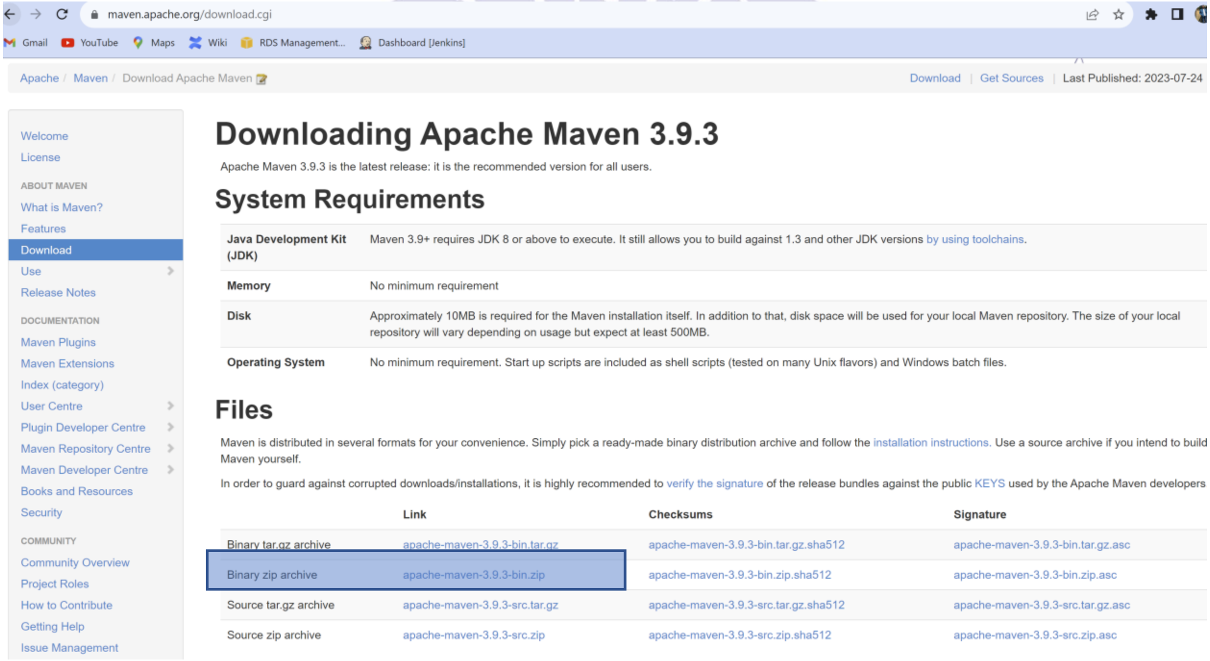
To install Apache Maven on Windows 10, you can follow these step-by-step instructions:

Step 1: Download Apache Maven:

Go to the Apache Maven website: https://maven.apache.org/download.cgi

Under "Files," click on the link to download the latest version of Maven (e.g., "Binary zip archive").

Save the downloaded ZIP file to a location on your computer.



Downloading Apache Maven 3.9.3

Apache Maven 3.9.3 is the latest release: it is the recommended version for all users.

System Requirements

Java Development Kit (JDK)	Maven 3.9+ requires JDK 8 or above to execute. It still allows you to build against 1.3 and other JDK versions by using toolchains.
Memory	No minimum requirement
Disk	Approximately 10MB is required for the Maven installation itself. In addition to that, disk space will be used for your local Maven repository. The size of your local repository will vary depending on usage but expect at least 500MB.
Operating System	No minimum requirement. Start up scripts are included as shell scripts (tested on many Unix flavors) and Windows batch files.

Files

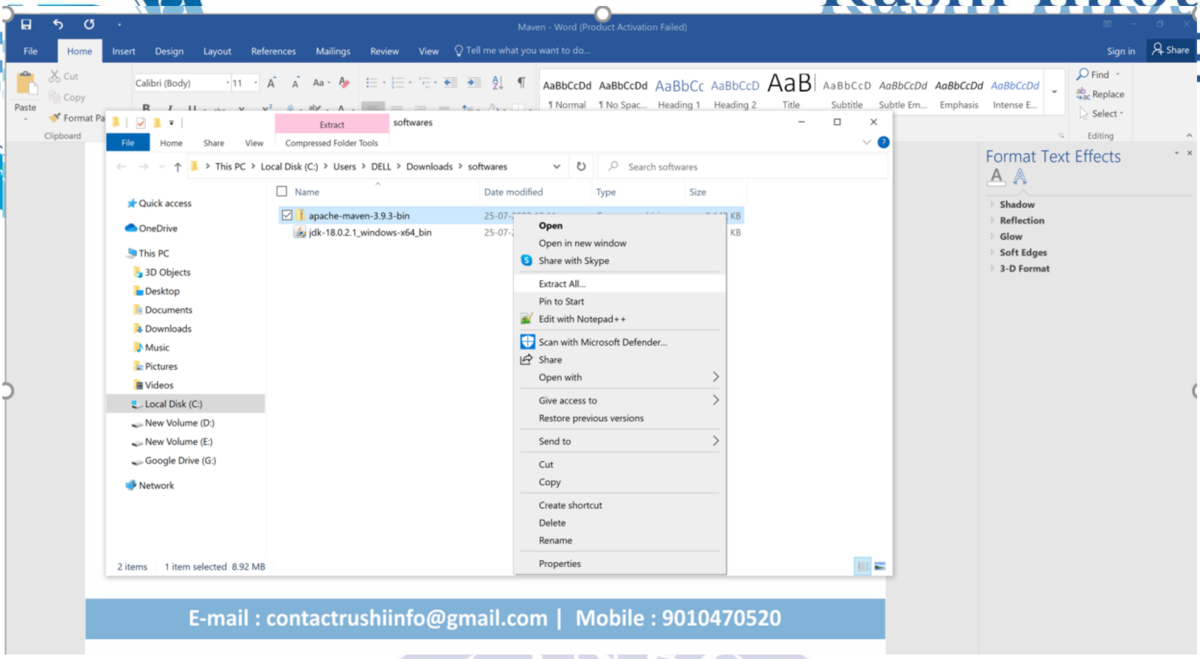
Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the installation instructions. Use a source archive if you intend to build Maven yourself.

In order to guard against corrupted downloads/installations, it is highly recommended to verify the signature of the release bundles against the public KEYS used by the Apache Maven developers

	Link	Checksums	Signature
Binary tar.gz archive	apache-maven-3.9.3-bin.tar.gz	apache-maven-3.9.3-bin.tar.gz.sha512	apache-maven-3.9.3-bin.tar.gz.asc
Binary zip archive	apache-maven-3.9.3-bin.zip	apache-maven-3.9.3-bin.zip.sha512	apache-maven-3.9.3-bin.zip.asc
Source tar.gz archive	apache-maven-3.9.3-src.tar.gz	apache-maven-3.9.3-src.tar.gz.sha512	apache-maven-3.9.3-src.tar.gz.asc
Source zip archive	apache-maven-3.9.3-src.zip	apache-maven-3.9.3-src.zip.sha512	apache-maven-3.9.3-src.zip.asc

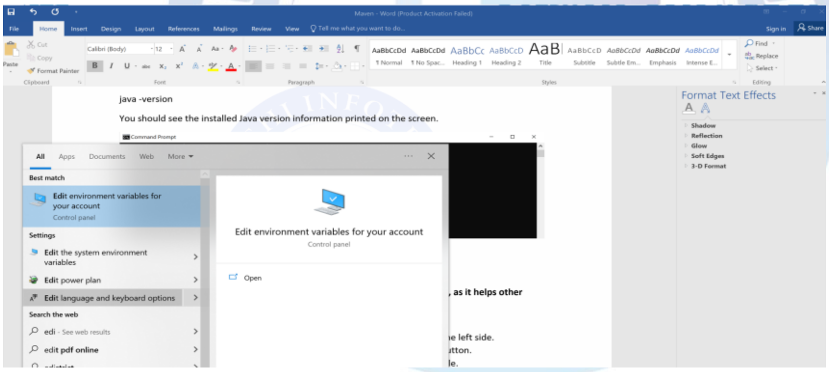
Step 2: Extract Maven Archive:

Locate the downloaded ZIP file and extract its contents to a directory of your choice. For example, you can extract it to "**C:\Program Files\Apache\maven**" **or any other location you prefer**.



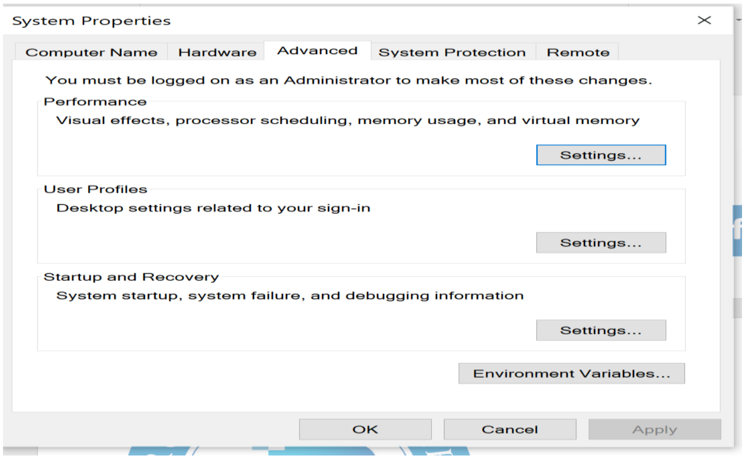
Step 3: Set MAVEN_HOME Environment Variable:

- Select "edit system environment variables” in the search bar


java -version

You should see the installed Java version information printed on the screen.

- In the System Properties window, click the "Environment Variables" button.


System Properties

You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

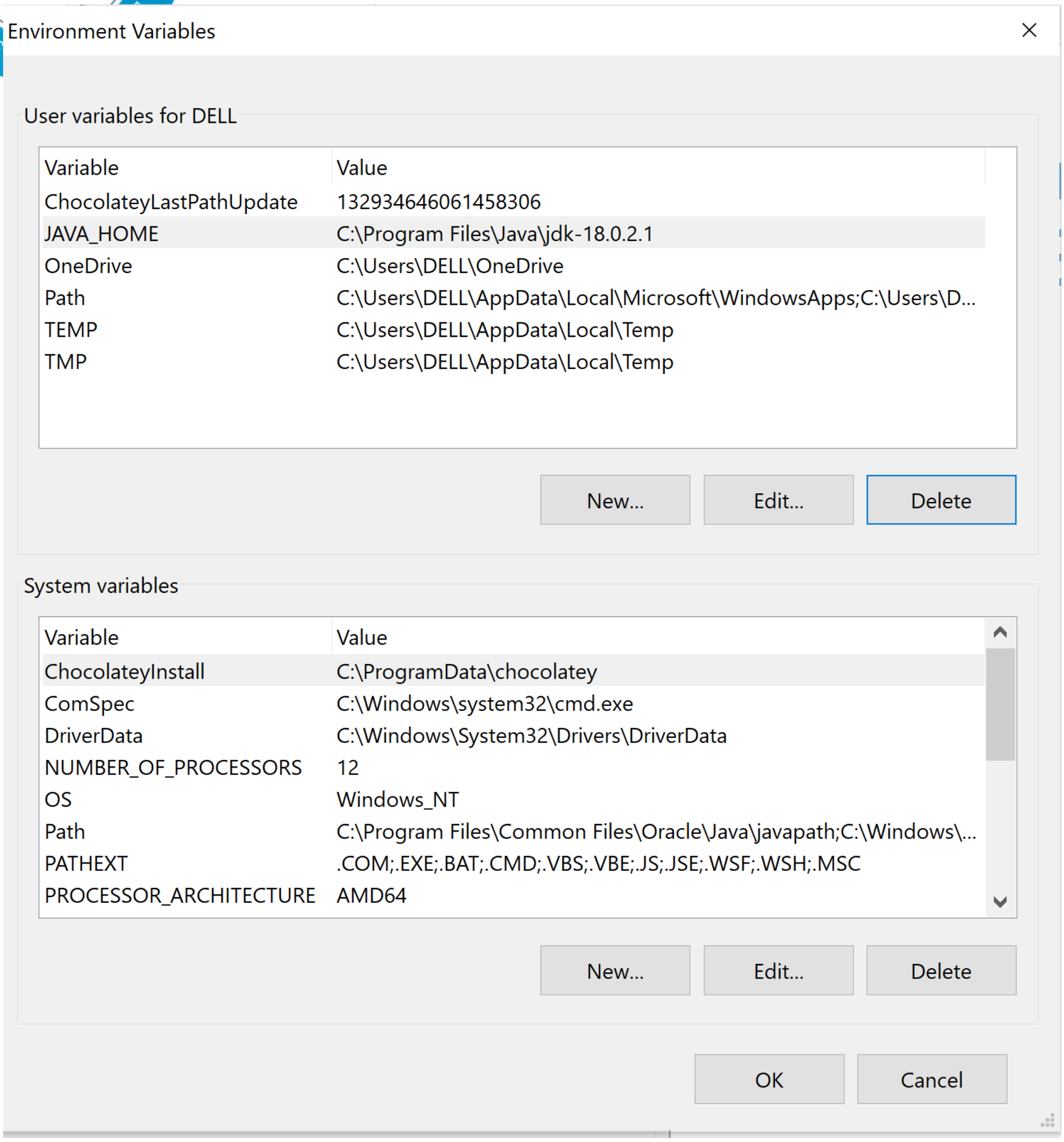
User Profiles

Desktop settings related to your sign-in

Startup and Recovery

System startup, system failure, and debugging information

- Under the "System variables" section, click "New" to add a new variable.



User variables for DELL

Variable	Value
ChocolateyLastPathUpdate	132934646061458306
JAVA_HOME	C:\Program Files\Java\jdk-18.0.2.1
OneDrive	C:\Users\DELL\OneDrive
Path	C:\Users\DELL\AppData\Local\Microsoft\WindowsApps;C:\Users\D...
TEMP	C:\Users\DELL\AppData\Local\Temp
TMP	C:\Users\DELL\AppData\Local\Temp

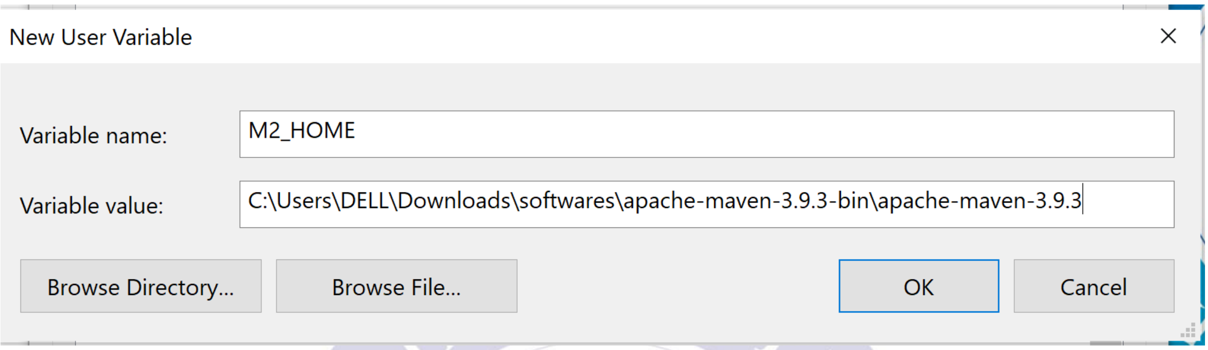
System variables

Variable	Value
ChocolateyInstall	C:\ProgramData\chocolatey
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	12
OS	Windows_NT
Path	C:\Program Files\Common Files\Oracle\Java\javapath;C:\Windows\...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64

Under New user variable:

Variable Name: M2_HOME

Variable value: C:\Users\DELL\Downloads\softwares\apache-maven-3.9.3-bin\apache-maven-3.9.3



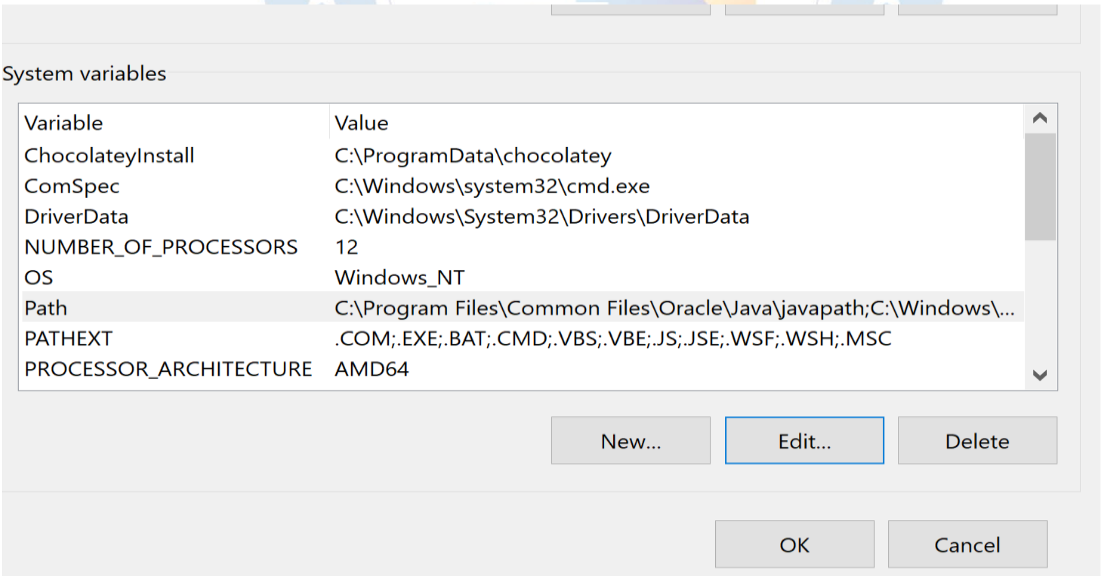
New User Variable

Variable name: M2_HOME

Variable value: C:\Users\DELL\Downloads\softwares\apache-maven-3.9.3-bin\apache-maven-3.9.3

And also add under the system variable:

Under system variable click on path → new → paste below path → and click on OK

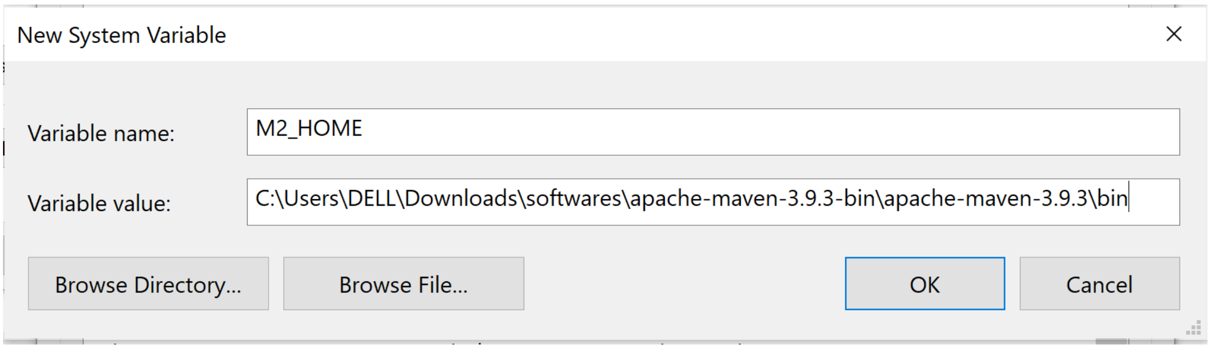


System variables

Variable	Value
ChocolateyInstall	C:\ProgramData\chocolatey
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	12
OS	Windows_NT
Path	C:\Program Files\Common Files\Oracle\Java\javapath;C:\Windows\...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64

Variable Name: M2_HOME

Variable value: **C:\Users\DELL\Downloads\softwares\apache-maven-3.9.3-bin\apache-maven-3.9.3\bin**



New System Variable

Variable name: M2_HOME

Variable value: C:\Users\DELL\Downloads\softwares\apache-maven-3.9.3-bin\apache-maven-3.9.3\bin

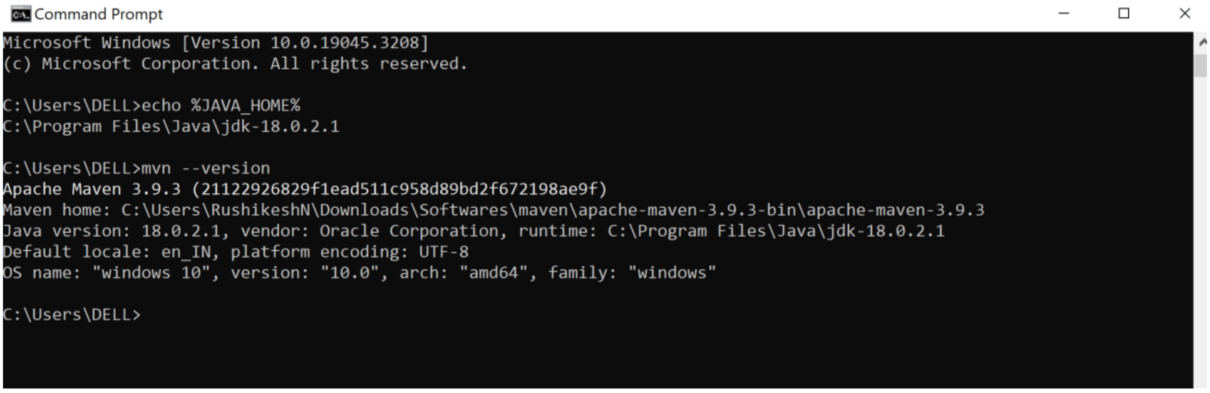
- Set the variable name as "M2_HOME" and the variable value to the path where you
- extracted Maven (e.g., "C:\Program Files\Apache\maven\apache-maven-3.x.x").
- Click "OK" to save the changes.
-

Step 5: Verify Maven Installation:

Open a Command Prompt or PowerShell window.

Type the following command to check the Maven version:

Command: mvn --version



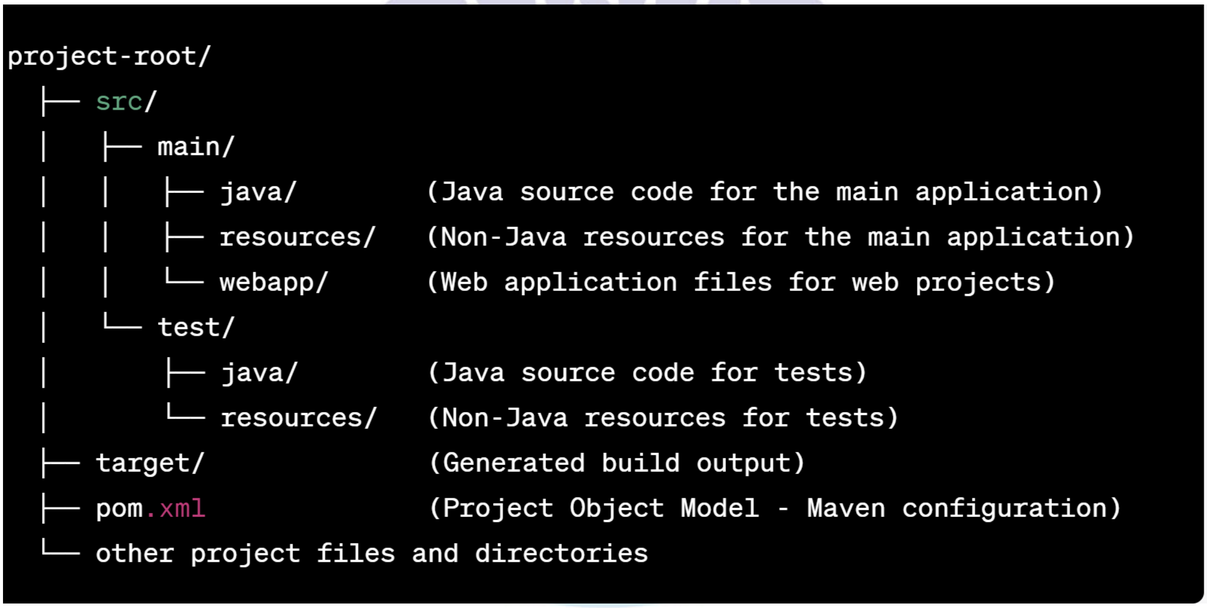
Command Prompt

```
Microsoft Windows [Version 10.0.19045.3208]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL>echo %JAVA_HOME%
C:\Program Files\Java\jdk-18.0.2.1

C:\Users\DELL>mvn --version
Apache Maven 3.9.3 (21122926829f1ead511c958d89bd2f672198ae9f)
Maven home: C:\Users\RushikeshN\Downloads\Softwares\maven\apache-maven-3.9.3-bin\apache-maven-3.9.3
Java version: 18.0.2.1, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-18.0.2.1
Default locale: en_IN, platform encoding: UTF-8
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

C:\Users\DELL>
```

What is Maven Folder structure?

In Apache Maven, the folder structure of a standard project follows a convention known as the "Standard Directory Layout." This structure is designed to promote consistency, ease of collaboration, and best practices across Maven projects. The Standard Directory Layout defines a set of predefined directories and their purposes.

Here's the typical folder structure of a Maven project:



```
project-root/
  ├── src/
  │   ├── main/
  │   │   ├── java/        (Java source code for the main application)
  │   │   ├── resources/   (Non-Java resources for the main application)
  │   │   └── webapp/      (Web application files for web projects)
  │   └── test/
  │       ├── java/        (Java source code for tests)
  │       └── resources/   (Non-Java resources for tests)
  ├── target/              (Generated build output)
  ├── pom.xml              (Project Object Model - Maven configuration)
  └── other project files and directories
```

Here's a brief explanation of each directory:

src/: This directory is the source root of your project, and it contains the main application code and test code.

main/: This directory contains the source code and resources for the main application. It includes:

java/: Java source code for the main application.

resources/: Non-Java resources (e.g., configuration files, property files) for the main application.

webapp/: For web projects, this directory contains web application files (HTML, CSS, JavaScript, etc.).

test/: This directory contains the source code and resources for testing the main application. It includes:

java/: Java source code for test cases.

resources/: Non-Java resources for testing (e.g., test data, configuration files).

target/: This directory is created by Maven and is used to store generated build output, including compiled classes, JARs, and other artifacts.

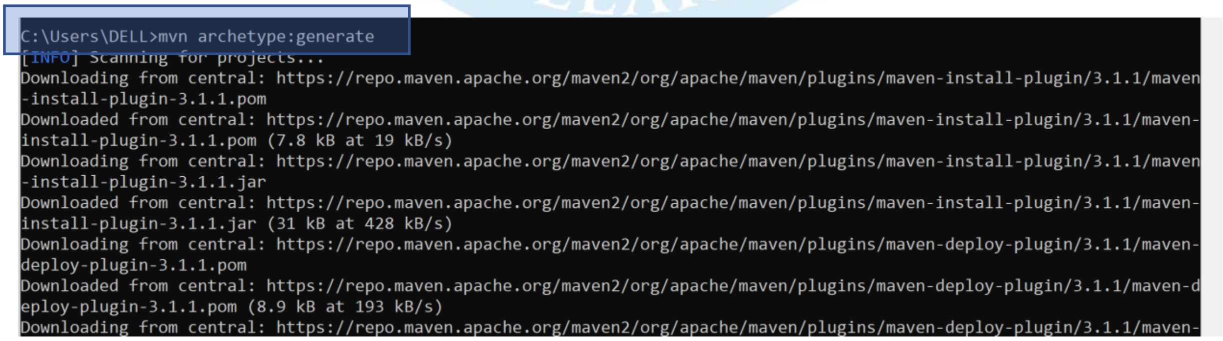
pom.xml: The Project Object Model (POM) file is the heart of Maven. It is an XML file that defines the project configuration, including dependencies, build settings, plugins, and other project-related information.

Other project files and directories: Depending on the project type and specific requirements, there might be additional files and directories in the project root. For example, for a web application, you might have a web.xml file, or for a multi-module project, you might have submodules with their own POM files.

The Standard Directory Layout ensures that Maven projects have a consistent and easily understandable structure, making it easier for developers to navigate, collaborate, and follow best practices when developing and building projects.

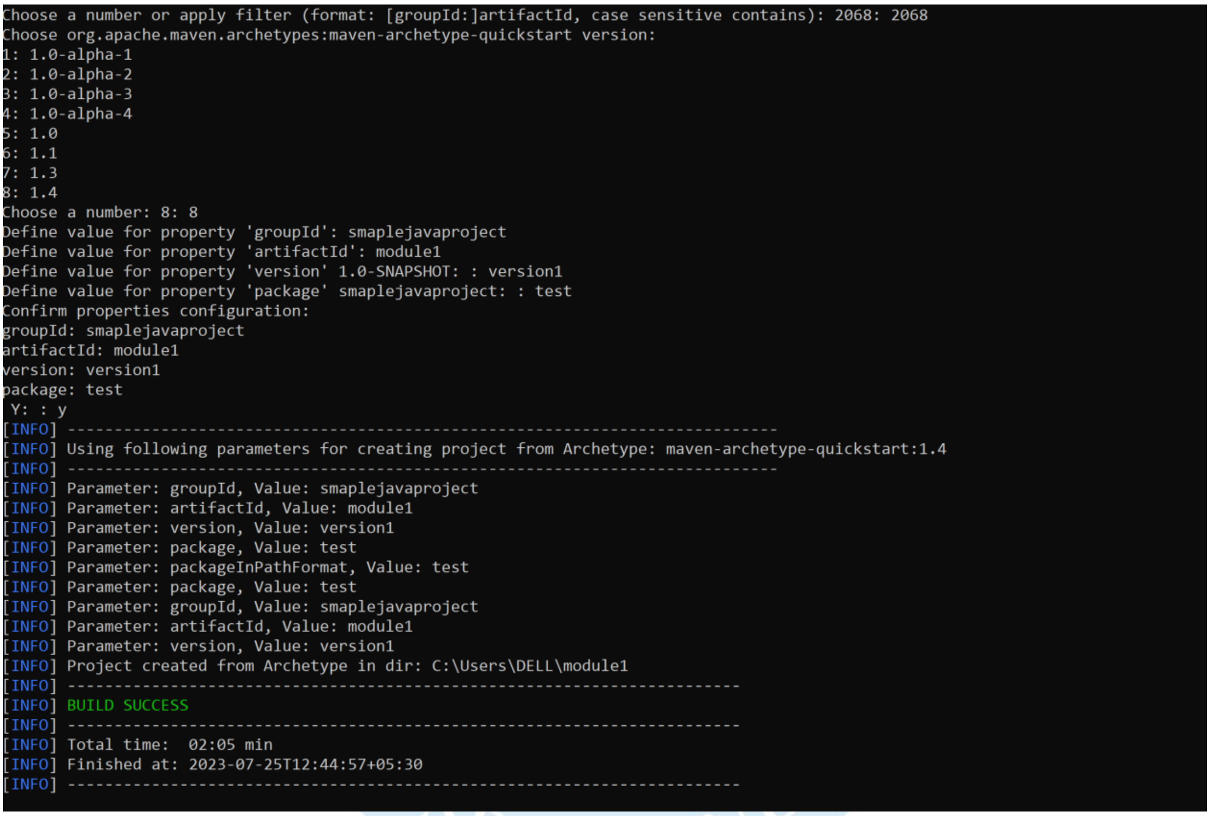
Command for generating maven Folder structure:

Command: mvn archetype:generate



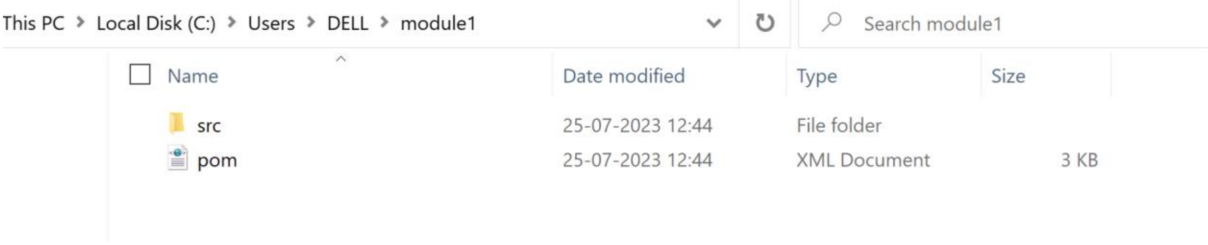
```
C:\Users\DELL>mvn archetype:generate
[INFO] Scanning for projects...
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/3.1.1/maven
-install-plugin-3.1.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/3.1.1/maven-
install-plugin-3.1.1.pom (7.8 kB at 19 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/3.1.1/maven
-install-plugin-3.1.1.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/3.1.1/maven-
install-plugin-3.1.1.jar (31 kB at 428 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-deploy-plugin/3.1.1/maven-
deploy-plugin-3.1.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-deploy-plugin/3.1.1/maven-d
eploy-plugin-3.1.1.pom (8.9 kB at 193 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-deploy-plugin/3.1.1/maven-
```

Sample java project folder structure generated from maven:



```
Choose a number or apply filter (format: [groupId:]artifactId, case sensitive contains): 2068: 2068
Choose org.apache.maven.archetypes:maven-archetype-quickstart version:
1: 1.0-alpha-1
2: 1.0-alpha-2
3: 1.0-alpha-3
4: 1.0-alpha-4
5: 1.0
6: 1.1
7: 1.3
8: 1.4
Choose a number: 8: 8
Define value for property 'groupId': smaplejavaproject
Define value for property 'artifactId': module1
Define value for property 'version' 1.0-SNAPSHOT: : version1
Define value for property 'package' smaplejavaproject: : test
Confirm properties configuration:
groupId: smaplejavaproject
artifactId: module1
version: version1
package: test
 Y: : y
[INFO] ----------------------------------------------------------------------------
[INFO] Using following parameters for creating project from Archetype: maven-archetype-quickstart:1.4
[INFO] ----------------------------------------------------------------------------
[INFO] Parameter: groupId, Value: smaplejavaproject
[INFO] Parameter: artifactId, Value: module1
[INFO] Parameter: version, Value: version1
[INFO] Parameter: package, Value: test
[INFO] Parameter: packageInPathFormat, Value: test
[INFO] Parameter: package, Value: test
[INFO] Parameter: groupId, Value: smaplejavaproject
[INFO] Parameter: artifactId, Value: module1
[INFO] Parameter: version, Value: version1
[INFO] Project created from Archetype in dir: C:\Users\DELL\module1
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  02:05 min
[INFO] Finished at: 2023-07-25T12:44:57+05:30
[INFO] ------------------------------------------------------------------------
```

Output:



Name	Date modified	Type	Size
src	25-07-2023 12:44	File folder	
pom	25-07-2023 12:44	XML Document	3 KB

Task: 1. upload this folder structure into Github Repository