Mavern

Install and Maven Setup

For Windows:

1. **Download Maven:**
   - Go to the *Apache Maven download page*.
   - Download the binary zip archive (`apache-maven-x.y.z-bin.zip`, where `x.y.z` is the version number).
2. **Extract the Archive:**
   - Extract the downloaded zip file to a directory of your choice (e.g., `C:\Program Files\Apache\maven`).
3. **Set Environment Variables:**
   - Open the Start Menu, search for "Environment Variables," and select "Edit the system environment variables."
   - Click on the "Environment Variables" button.
   - In the "System Variables" section, click "New" to create a new environment variable:
     - **Variable name:** `MAVEN_HOME`
     - **Variable value:** Path to your Maven directory (e.g., `C:\Program Files\Apache\maven\apache-maven-x.y.z`)
   - Find the "Path" variable in the "System Variables" section, select it, and click "Edit."
   - Add a new entry with the path to the Maven `bin` directory (e.g., `C:\Program Files\Apache\maven\apache-maven-x.y.z\bin`).
4. **Verify Installation:**
   - Open Command Prompt.
   - Type `mvn -version` and press Enter.
   - You should see Maven version information, Java version, and OS details.

For macOS/Linux:

1. **Install Homebrew (macOS Only):**

- If you don't have Homebrew installed, open Terminal and paste the following command:

  ```
  /bin/bash -c "$(curl -fsSL
  https://raw.githubusercontent.com/Homebr
  ew/install/HEAD/install.sh)
  ```

- Follow the on-screen instructions to complete the installation.

2. **Install Maven Using Homebrew (macOS Only):**
   - Open Terminal and run:

     ```
     brew install maven
     ```

3. **Download and Extract Maven (Linux and Alternative macOS Method):**
   - Go to the *Apache Maven download page*.
   - Download the binary tar.gz archive (`apache-maven-x.y.z-bin.tar.gz`).
   - Open Terminal and navigate to the download location. Extract the archive using:

     ```
     tar -xvf apache-maven-x.y.z-bin.tar.gz
     ```

   - Move the extracted folder to a directory of your choice (e.g., `/usr/local/apache-maven`
)

     ```
     sudo mv apache-maven-x.y.z
     /usr/local/apache-maven
     ```

4. **Set Environment Variables:**
   - Open your profile file in a text editor (`~/.bash_profile`, `~/.bashrc`, `~/.zshrc`, or `~/.profile`
     ), depending on your shell:

     ```
     nano ~/.bash_profile
     ```

   - Add the following lines:

     ```
     export MAVEN_HOME=/usr/local/apache-
     maven/apache-maven-x.y.zexport
     ```

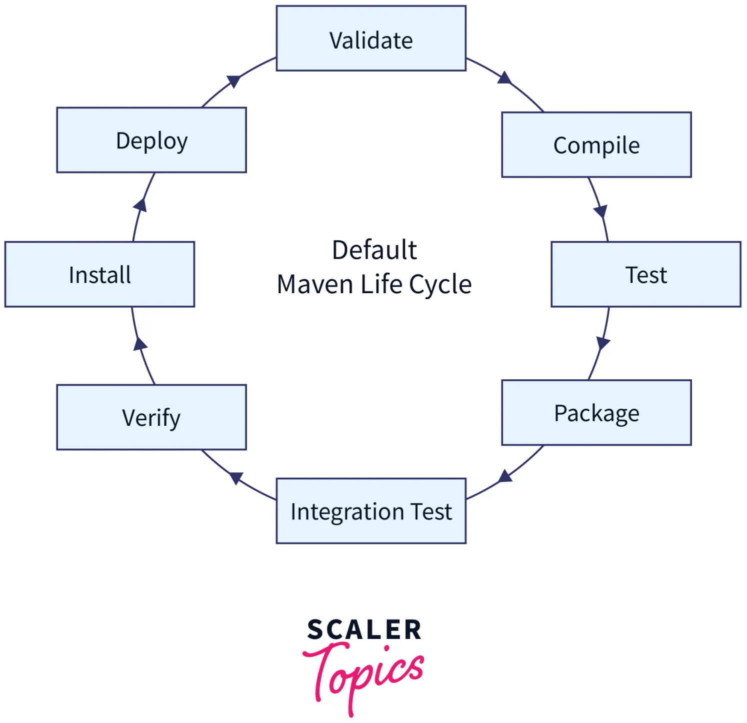
```
PATH=$MAVEN_HOME/bin:$PATH
```

- Save the file and apply the changes:

```
source ~/.bash_profile
```

5. **Verify Installation:**
    - Open Terminal.
    - Type `mvn -version` and press Enter.
    - You should see Maven version information, Java version, and OS details.

Maven LifeCycle



***clear maven cache** rm -rf ~/.m2/repository*

validate: Checks project configuration and dependencies.
compile: Transforms source code into bytecode.
test: Executes unit tests.
package: Creates a distributable archive(.jar , .war , zip file)
verify: Performs additional checks on packaged artifacts (optional).
*Installs the Package**:** It installs the packaged artifacts into the local Maven repository, located usually in the `~/.m2/repository` directory. This makes the artifacts available for other projects on the*

same machine to reference as dependencies.
install: Deploys the package to your local Maven repository.
deploy: Deploys the package to a remote Maven repository (optional).
mvn site: generate a site or project documentation.

Maven™

20+ MUST KNOW COMMANDS

1 mvn clean

Cleans the maven project by deleting the target directory.

2 mvn compiler:compile

Compiles the Java source classes. Use "**mvn compiler:testCompile**" to compile the test classes.

3 mvn package

Build the maven project and create JAR, WAR files.

4 mvn install

Build the maven project and install the package files (JAR, WAR, pom.xml, etc) to the local repository.

5 mvn deploy

Deploy the build artifact to the remote repository

6 mvn validate

validate the project is correct and all necessary information is available

7 mvn dependency:tree

Generates the dependency tree of the maven project.

8 mvn dependency:analyze

Analyze the maven project to identify the unused declared and used undeclared dependencies

9 mvn archetype:generate

Used to create a maven project from the archetype template project

10 mvn -help

Prints the usage and all the different options we can use with the mvn command

11 mvn site:site

Generate a site for the maven project.

12 mvn test

test the compiled source code using a suitable unit testing framework.

13 mvn compile

compile the source code of the project

14 mvn verify

run any checks on results of integration tests to ensure quality criteria are met

15 mvn -f dir/pom.xml package

Force the use of an alternate POM file (or directory with pom.xml)

16 mvn -o package

runs the maven command in the offline mode.

17 mvn -q package

runs the maven command in the quiet mode, only show errors and the test cases results.

17 mvn -X package

runs the build and produces output in the debug mode.

19 mvn -v

Display maven version information.

20 mvn -V package

Display maven version information and continue with the build

21 mvn -DskipTests package

skips running the test cases of the project, you can also use **-Dmaven.test.skip=true** option.

22 mvn -T 4 clean install

parallel build with 4 threads, useful to increase the build performance in the multiple module project.

Create project using maven command

mvn archetype:generate -DgroupId=com.example -DartifactId=my-app -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
There are two modes , interactive and non inrrteractive , in interactive mode you need to tell every information at the terminal,where as in non interactive , you need to write it correspondingly.

Maven Not Found Error

Step 1: Clear Maven Cache
Sometimes, Maven's local repository cache might cause issues. You can clear the cache by deleting the relevant plugin directory:

1. Navigate to your local Maven repository, typically located at ***`~/.m2/repository/org/apache/maven/plugins/`***.
2. Delete the ***`maven-surefire-plugin`*** directory.
3. Run your Maven build again (`mvn test`).

Step 2: Update the Surefire Plugin Version

Ensure you're using a valid and stable version of the Surefire
Step 3 : Clear invalidate caches from project top left side in intelij