**Maven**

**Maven** is a [build automation](https://en.wikipedia.org/wiki/Build_automation) tool used primarily for [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) projects.

* **Pom:**

Pom stands **Project Object Model**. Here we define all the configuration for the maven project.

We define <groupId> and <artifactId> for every project or module which will further used for identified in next project. If we want to use this project generated jar in another project then we will have to use this groupid and artifactid. <version> Represent version of project and snapshot represent project under development. <packaging> as name clear we define jar and war.

* Each dependency has groupId, artifactId and version (optional).

**Maven Build Life cycle**

* **Validate**
* **Compile**
* **Test**
* **Package**
* **Integration Test**
* **Verify**
* **Install**
* **Deploy**

First Maven **Validates** the project then it **Compiles** the project. Compile does have two steps first is to compile the source code and then second is to compile the test code. Next stage is **Test** in which the maven run all the unit test in project. If all the unit test run successful then maven will go to **Package** stage which means it would actually build a jar or war. Then if any **Integration** **test** then those will run then it will got to verify stage. Then it will install the jar into the local repository means maven will take artifact which is generated then put it into local repository and **Deploy** means maven will take artifact and deploy it to remote repository.

Important pre-configuration of maven:

1. All the main source codes resides under the src/main/java
2. All the properties files and configurations files should be in src/main/resource.
3. All the test files should be in src/test/java

Note: Maven use this strategy on the concept of convention over configuration.

**Maven repository**

1. **Local repository**

This repository store on local system.

1. **Remote repository**

Central repository which store all the versions of dependencies.

**Maven commands**

1. **Mvn Install** – Means maven will run all the life cycle till install.
2. **Mvn clean** – This command will delete all the target folder. (Then compile).
3. **Mvn compile**- This command compiles the code. If we want to compile the test code.
4. **Mvn test-compile –** This will compile the test code after compile the source code.
5. **Mvn test –** It is used to run the unit test. Means it will follow build life cycle i.e validate, compile then test.
6. **Help:**effective-pom – The entire content of pom using super pom will be printed.
7. **Help:**effective-setting – All the setting repository will be printed on cmd.
8. **Dependency:**tree – This will show all the dependency.
9. **Dependency:**source – All the dependency will be downloaded into local machine.
10. **–-** debug – Maven will execute in debug mode.