

## Maven Build Process (Building Java Source Code in Linux machine )

### Download code from github

\$ git clone <https://github.com/thej950/maven>

From above steps code will be downloaded into system

Build the java code using maven

# mvn archetype:generate

- Apache Maven, archetype:generate is a command used to create a new Maven project from an existing project template, known as an "archetype."
- An archetype is essentially a project template that provides a predefined directory structure, build configuration, and other resources to kickstart the development of a particular type of project.

```
ubuntu@Docker-Host:~$ mvn archetype:generate
[INFO] Scanning for projects...
[INFO]
[INFO] -----< org.apache.maven:standalone-pom >-----
[INFO] Building Maven Stub Project (No POM) 1
[INFO] -----[ pom ]-----
[INFO]
[INFO] >>> maven-archetype-plugin:3.2.1:generate (default-cli) > generate-sources @ standalone-pom >>>
[INFO]
[INFO] <<< maven-archetype-plugin:3.2.1:generate (default-cli) < generate-sources @ standalone-pom <<<
[INFO]
```

Below image shows a default number go with that number (that number indicates build folder structure of project ) Go with default

```
3306: remote -> za.co.absa.hyperdrive:component-archetype_2.11 (-)
3307: remote -> za.co.absa.hyperdrive:component-archetype_2.12 (-)
Choose a number or apply filter (format: [groupId:]artifactId, case sensitive contains): 2067:
```

Asking to choose number go with default

Go with default Number 8: Press Enter

```
8: 1.4
Choose a number: 8:
```

Asking group id here i am providing com.web

```
Choose a number or apply filter (format: [groupId:]artifactId, case sensitive contains): 2067:
Define value for property 'groupId': com.web
```

Providing artifactId : webapp

```
Define value for property 'artifactId': webapp
```

Press Enter

```
Define value for property 'version' 1.0-SNAPSHOT: :
```

Press Enter

```
Define value for property 'package' com.web: :
```

Go with Default Press Enter

```
Confirm properties configuration:
groupId: com.web
artifactId: webapp
version: 1.0-SNAPSHOT
package: com.web
Y: :
```

From here its show build success

```
[INFO] Parameter: package, Value: com.web
[INFO] Parameter: groupId, Value: com.web
[INFO] Parameter: artifactId, Value: webapp
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Project created from Archetype in dir: /home/ubuntu/webapp
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 59.816 s
[INFO] Finished at: 2023-07-21T13:50:43Z
[INFO] -----
ubuntu@Docker-Host:~$
```

Now from above steps a New webapp Directory will be created  
Enetr into that directory Perform mvn compile

```
ubuntu@Docker-Host:~$ ls
maven webapp
ubuntu@Docker-Host:~$
```

```
ubuntu@Docker-Host:~$ cd webapp/
ubuntu@Docker-Host:~/webapp$ ls
pom.xml src
ubuntu@Docker-Host:~/webapp$
```

Performing # mvn compile

```
ubuntu@Docker-Host:~/webapp$ mvn compile
[INFO] Scanning for projects...
[INFO] -----< com.web:webapp >-----
[INFO] Building webapp 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO] --- maven-resources-plugin:3.0.2:resources (default-resources) @ webapp ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/ubuntu/webapp/src/main/resources
[INFO] --- maven-compiler-plugin:3.8.0:compile (default-compile) @ webapp ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to /home/ubuntu/webapp/target/classes
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.426 s
[INFO] Finished at: 2023-07-21T14:03:43Z
[INFO] -----
ubuntu@Docker-Host:~/webapp$
```

From above image mvn compile make all .java file to .class

Note: here mvn compile also will perform downloads the third party dependencies those dependencies normally we added inside "pom.xml" file based on this file dependencies will be downloaded

After build we will get a target directory inside target directory inside directory we have necessary files

```
ubuntu@Docker-Host:~/webapp$ tree target/
target/
├── classes
│   ├── com
│   │   └── web
│   │       └── App.class
├── generated-sources
│   └── annotations
├── maven-status
│   └── maven-compiler-plugin
│       └── compile
│           └── default-compile
│               ├── createdFiles.lst
│               └── inputFiles.lst
9 directories, 3 files
ubuntu@Docker-Host:~/webapp$
```

## Maven Build Process (or) Maven Lifecycle

### 1. prepare-resources

- it is a default stage by default it gets executed

### 2. validate

- In this validate stage maven will check project folder structure created or not according to the ID

### 3. Compile

- When java developer create program that contains default file extension should be .java only (it is called source code )
- We need to execute this java program directly which means it has to be compile
- When compile that java program it creates .class files it can also called as bite code ( which system understandable language )

- When that .class files archive or bundled together is called an artifact it could be a jar file or war file or ear file depending on the kind of java programming

#### 5. package

- this stage make all .class file archive or bundled into artifact )

#### 6. install

- In this install stage developers want to use artifact for others projects they install into maven repository command called is mvn install
- Maven also work like application server

#### 7. deploy

=====

Q: how can you add dependencies in maven

Q: what is maven lifecycle (or) how many steps in maven explain them