

Ex1:

Build Tools - Maven and Gradle.

3 types repository - local repository, server or central repository

POM → Project Object Model.

Gradle → Kotlin, Groovy, Scala

Lifecycle → clean

validate

compile

test

package

verify

install

site

deploy

Installation of Maven :

→ Goto chrome → Maven download in suspension.

Download Apache maven



Binary zip archive → windows.

Once completed go to maven file.

window to → WinRAR → Right click → WinRAR extract

Extract all

→ desktop. → two folder.

Maven ^{folder} → copy to c drive or d drive
double click, copy the path

cmd java --version

Setup environment

Right click on this PC | → properties
↳ Advanced system setting

↓
Click on environment variable.

In system variable not user variable.

Step 1 ↓ variable name

New → MAVEN_HOME (All should be in capital letters).

variable value → Open maven folder copy + paste
root path
Paste.

Step 2

OK

Again in system variable → New

variable name → M2_HOME

Step 3 variable value -

Again in system variable

Goto path → Press edit → copy the path of
bin folder which is in apache-maven-3.8.6

New → Path - Paste → OK.

Installation of Gradle :-

Go to cmd → gradle -v

Go to chrome - Gradle download
↓

Installation - Gradle or Release
↓

Go down → Download Binary-only
↓

Once download completed
↓

Go to folder
↓

copy → Go to C drive
↓

Create a new folder gradle
↓

Paste.
↓

Extract file.

↓

Open the file
↓

Copy → Cut, → Paste.

Copy the bin path

Right click of This pc → properties
path → environmental settings.
↓

In user variables

↳ Go to path → (New) or Edit → New → Paste
Variable name → → OK → OK → OK.

cmd → gradle -v

Gradle 6.0.1

Java(TM) SE Runtime Environment

Java HotSpot(TM) 64-Bit Server VM

Build #gradle-6.0.1-20190711-1258-username

Using 'gradle' script from: /home/username/.gradle/wrapper/bin

Using Java JRE: /usr/lib/jvm/java-11-openjdk-amd64/jre

Using Java VM Args: -Xmx3072m -XX:+HeapDumpOnOutOfMemoryError

Using Gradle VM Args: -Dfile.encoding=UTF-8 -Duser.country=US -Duser.language=en -Duser.variant=

Using system classpath: [empty]

Using system properties: [empty]

Using system environment: [empty]

Using system classpath: [empty]

Using system properties: [empty]

Using system environment: [empty]

Using system classpath: [empty]

Using system properties: [empty]

Using system environment: [empty]

Exp 2:-

* In IntelliJ →

New → Project

Name → shwija-maven

Project Type - Maven

Location → Home → editise → This window.

POM XML → Full remove

→ GitHub → POM.XML → Copy & Paste POM.XML

src → main - java → right click → New →

In file copy logo, style, index

index.html

resource → new HTML file → GitHub → docs →

index.html → Save

Resource → File → style.css → GitHub → docs.

copy style.css → save
Paste

Resource → File → logo.png - GitHub

Github → copy image → click on resource → Add

↳ Rename what code has name.

To push git

* First create new repository in git

↳ Give → test → name.

* Public

* Create repository

Go to IntelliJ → Terminal

↳ 1. git init

2. git add.

3. git commit -m "Initial commit".

4. git remote add origin

copy the link in GitHub

3. git push -u origin master

6. give username

In index.html click ~~change~~
Title driver - get → Website
inPath copy
Paste it in

Token creation

- Profile → settings → Developer setting → Access token
- Token class → note → abcde → select all
- Generate token → copy token → paste in IntelliJ

* Paste the Token.

Configure → Skip → write only.

To build project:

* In terminal mvn clean install

docs → created

3 file it will take.

to add git

git add ~~docs~~ docs

git commit -m "deploy" (do not give -u none)

git push origin master

In git → check in your repository → check
docs

settings → page → none change to master →
root change to /docs

b Refresh the page then you
will get logo website.

* Concentrate on logo.png it is not jpg in ubuntu

WebpageTest

in terminal →

src → test → java - new → java class →
WebPageTest

↳ In git → go to previous repository → go to

WebPageTest → cop.g → save in → WebPageTest

* Right click Run WebPageTest

Jar file :-

* Skip step 1.

→ src → main → java → org.example → Main

→ In terminal → mvn clean package → In target

check SNAPSHOT jar file

→ Then in terminal java -ja target/

copy the SNAPSHOT path

Documentation command

mvn site

→ ~~jar~~

→ in target → sites → index.html → Right select
chrome.

3

Gradle (Groovy)

- File → New → Project → maven → test-gradle →
- Gradle → Groovy → Create ·
- build.gradle →
- create repository → from previous repository
- copy paste the build.gradle → click on → To run
- Gradle icon → Tasks → application → run.
- Terminal - run gradle
- In root directory → test-gradle → new directory dars.
- Then follow same step of maven
- create WebPageTest
- ↳ change the Title
- driver.get
- To run
- Task → verification → test.
- Next in terminal → gradle test
- Next jar file
- Terminal - grade jar
- Then in → in build → libs → test-gradle.jar
- Then in terminal java -jar build/libs/test-

Kotlin

F5 → project → gradle → Kotlin → Gradle → Kotlin

src → main → build.gradle.kts → Paste → run make changes.

4th Exp:-

F5 → project → migration → Groovy maven → create pom.xml → Paste.

Terminal → mvn clean compile
target

mvn package

target → SNAPSHOT jar file →

Then in terminal
java -jar target/(copy paste the path)

Next

in terminal gradle init --type pom

Select → 2 (Groovy)

yes

in build.gradle

in terminal → gradle clean build

in build.gradle

in terminal java -jar build/libs/ path

5th →

Check java installed → java --version

In browser

jenkins.io

download

Generic java package(war)

Once completed paste it in d drive

create folder and
paste it there

In cmd.

d: (folder name)

cd jenkins

cls.

D:\Jenkins\java -jar jenkins.war

Extracted file name

Copy the password:

In browser → search localhost:8080

In Administrator password: copy the password

Install suggested plugins:

Create first Admin user. →

Ashwija
ashwija@004

Once completed click on get started.