```
Working with Maven(Only for Java Development)
a. CLI
b. IDE's
CLI : Command Line Interaction
To work with CLI in Maven
 Set up Java in our system(already done)
  Set up Maven in our system
Download maven software from
      a. https://maven.apache.org/download.cgi apache-maven-3.9.7-bin.zip
      b. Extract the file and keep the apache-maven-3.X version in C drive of your local system
      c. Set up environment variable to inform O.S about maven
                  MAVEN_HOME = C:\apache-maven-3.9.1
                                                                              Set using GUI mode of windows os
                                 = C:\apache-maven-3.9.1\bin
      d. run the command from commandline
                   \Users\nitin>mvn -version
                Apache Maven 3.9.1 (2e178502fcdbffc201671fb2537d0cb4b4cc58f8)
                Maven home: C:\apache-maven-3.9.1
                Java version: 18.0.1.1, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-18.0.1.1
               Default locale: en_IN, platform encoding: UTF-8
OS name: "windows 11", version: "10.0", arch: "amd64", family: "windows"
Creating Standalone project using Maven in CLI Mode
  Maven Terminologies(GAV)
   a.groupId —
                                                                                                         Every dependency used in project is identified as "GAV"
    b. artifactId ————
                                                                            --- ProjectName
                                                     → version of development
   d. packaging — → jar/war
Note: Developer should inform maven about what type of project we need to build.
         To say this to maven we need to for "archetype" [2500+]
             1. standalone :: maven-archetype-quickstart
             2. webapp :: maven-archetype-webapp
 Creating a maven project in CLI mode
                                                                                                                                                      (standalone)
 D:\maventools>mvn archetype:generate -DgroupId=pwskills -DartifactId=01-Maven-App -DarchetypeArtifactId=maven-archetype-quickstart -Dpackage=in.pwskills.nit
in.main -Dversion=1.4 -DinteractiveMode=false
                                                                                                Life Cycle of Maven
  01-Maven-App
                                                                                                Maven has 3 Life Cycle.Each Life Cycle contains number of Phases
            -> src/main/java
                     -> in.pwskills.nitin.main | Development code
                                                                                                                                           (ready made/standard plugins)

    Clean

                                       |-> App.java
                                                                                                                        clean :: old .class files it will clean (if generated previously)

    default

            -> src/test/java
                                                                                                                       compile:: Generates the .class file for the source code.
                                                                                                   site
                     -> in.pwskills.nitin.main
                                       |-> AppTest.java | Testing code
                                                                                                                      package :: Zip the .class files and creates a jar/war file
           -> pom.xml
                                                                                                                        test :: run the test cases and generates the report(testing)
                                                                                                                     install :: our project will be used as a jar/war in other projects
             (build file)
                                                                                                                                              (installed locally as a dependency)
                                                                                                                      To run java application using maven standard plugin support not available
      02-Maven-App
                                                                                                                        we use extra plugin in pom.xml file to get the support.
             -> src/main/java
                                                                                                                                       exec-maven-plugin
                       |-> in.pwskills.nitin.main | Development code
                                           |-> App.java
              -> src/test/java
                        -> in.pwskills.nitin.main
                                          -> AppTest.java | Testing code
             -> pom.xml
                (build file)
  >> mvn test(generates the test cases output in console)
  >> mvn site(generates the test cases output in target/site/index.html)
  >> mvn surefire-report:report(generates the test cases output in target/site/surefire-report.html)
      03-JDBC-Maven-App
                                                                                                                           02-Maven-App
                -> src/main/java
                                                                                                                                   -> src/main/java
                          -> in.pwskills.nitin.main Development code
                                                                                                                                              -> in.pwskills.nitin.main
                                                                                                                                                               |-> Arithmetic.java | Development code
                                         -> MainApp.java
                                                                                                use as a service
                 -> src/test/java
                                                                                                                                    -> src/test/java
                          -> in.pwskills.nitin.main
                                                                                                                                              -> in.pwskills.nitin.main
                                                                                                                                                                -> AppTest.java | Testing code
                                                                        Testing code
                                             |-> AppTest.java
               -> pom.xml
                                                                                                                                   -> pom.xml
                                                                                                                                      (build file)
                                                                                                                                                                          mvn install
                                             <groupId>com.oracle.database.jdbc</groupId>
                downloaded
                                            <artifactId>ojdbc6</artifactId>
               and kept
                                             <version>11.2.0.4</version>
               in .m2 folder
                                        </dependency>
                                        <!--Dependency from another project-->
                                              <groupId>pwskills</groupId>
                                             <artifactId>02-Maven-App</artifactId>
                                              <version>1.4</version>
mvn package
                                         </dependency>
CONNECTION object created...
STATEMENT object created...
RESULTSET object created...
         SNAME SAGE SADDRESS
           schin 49
                                 ΜI
           dhoni 41
                                 CSK
          kohli 35
                                 RCB
           rohith 38
                                 ΜI
                                 LSG
          rahul 31
Getting extra service
                                                   Service used from
The sum is :: 30 👉—
Closing the resources.
                                                  another project
Creating a webapplication using maven in CLI Mode
 D:\maventools>mvn archetype:generate -DgroupId=pwskills -DartifactId=04-Maven-webApp -DarchetypeAr
 tifactId=maven-archetype-webapp -Dpackage=in.pwskills.nitin.main -Dversion=1.0 -DinteractiveMode=f
 alse=false -Dpackaging=war
    04-Maven-webapp
                                            ── src/main
                                                                    <del>-</del>resources
                                                                                                                       -in.pwskills.nitin.controller
                                                                                                                                   WishServlet.java
                                                                                                                                     (/wish)
                                                                           pom.xml
                                                                          index.jsp
                  (build file)
  mvn package :: creates a war, deploy it manually in tomcat9 folder
  mvn tomcat7:run :: creates a tomcat environment and runs our war file(logical binding)
Working with Maven in Eclipse
  Note: no need to install any plugin for maven, by default eclipse provides maven supoprt
maven will follow transitive dependency management to download the dependencies to the project.

    Maven Dependencies

                                                                                                                                              \( \mathbb{\overline} \) HibernateProject
                                                                                   > 📠 junit-4.11.jar - C:\Users\nitin\.m2\repository\junit
    <dependency>
                                                                                   > 📠 hamcrest-core-1.3.jar - C:\Users\nitin\.m2\reposi
               <groupId>org.hibernate.orm
                                                                                                                                                 in.pwskills.nitin.bean
                                                                                   > 뤒 hibernate-core-6.2.12.Final.jar - C:\Users\nitin\.m
              <artifactId>hibernate-core</artifactId>
                                                                                                                                                   > I Student.java
                                                                                    📠 jakarta.persistence-api-3.1.0.jar - C:\Users\nitin\.r
               <version>6.2.12.Final

¬ 

in.pwskills.nitin.cfgs

in.pwskills.nitin.cfgs
                                                                                  > 둶 jakarta.transaction-api-2.0.1.jar - C:\Users\nitin\.r
                                                                    hibernate >  iboss-logging-3.5.0.Final.jar - C:\Users\nitin\.m2\
                                                                                                                                                  🗷 hibernate.cfg.xml
                                                                                                                                                  in.pwskills.nitin.main
                                                                                   > 📠 hibernate-commons-annotations-6.0.6.Final.jar -
                                                                                                                                                     > 🗾 LoadRecordApp.java
                                                                                 > 👼 jandex-3.0.5.jar - C:\Users\nitin\.m2\repository\ic
                                                                                                                                                   > 🖶 in.pwskills.nitin.util
                                                                 dependencies classmate-1.5.1.jar - C:\Users\nitin\.m2\repositor
                                                                                                                                                 > 乃 src/test/java
                                                                                   > 👼 jakarta.xml.bind-api-4.0.0.jar - C:\Users\nitin\.m2
                                                                                                                                                 > A JRE System Library [JavaSE-1.8]
                                                                                                                                                 > 🚵 Maven Dependencies
                                                                                   > 👼 jakarta.activation-api-2.1.0.jar - C:\Users\nitin\.m
                                                                                                                                                 > 🗁 src
                                                                                    > 📠 jaxb-runtime-4.0.2.jar - C:\Users\nitin\.m2\reposi
                                                                                                                                                 > 🗁 target
  developer
                                                                                   > 👼 jaxb-core-4.0.2.jar - C:\Users\nitin\.m2\repository
                                                                                                                                                  pom.xml
                                                                                   > 📠 angus-activation-2.0.0.jar - C:\Users\nitin\.m2\rep
                     → version mismatch problem (which version will match if
                                                                                   > 📠 txw2-4.0.2.jar - C:\Users\nitin\.m2\repository\org
                                                                                   > 📠 istack-commons-runtime-4.1.1.jar - C:\Users\nitir
                            we work with multiple
                                                                                   > 📠 jakarta.inject-api-2.0.1.jar - C:\Users\nitin\.m2\re
                                                                                   > 📠 antlr4-runtime-4.10.1.jar - C:\Users\nitin\.m2\rep
                            dependencies)
                                              solution

    Maven inheritance

  2. Maven MultiModule
```

Maven Build tool(Developer/Expert)

Central Repository

(maven team)

c. Adds the required jars to classpath of our project.

search dependency

a. Creates a Standard folder structure for our project based on developer requirement.

download

add dependencies

(GAV)

d. Compilation + Execution + Packaging(jar/war) would be done based on developer needs(goals)

b. Downloads the dependency from central/remote repository to Local Repository(.m2 folder of our local system).

Note: Developer while using project management build tool, he would focus only on "Business Logic/Development".

Local Repository

ひ ひ

pom.xml

search

dependency

Wipro

Infosys

Oracle

(Individual Company)

Remote Repository

c:/users/{username}/.m2/repository

download

Developer input file(build file)