Maven

**1.What is life cycle of Maven?**

**2.How to check the dependency of project?**

Ans:mvn dependency:tree

**3.what is pom? And give the details of entry in pom’s file?**

Pom stands for Project Object Model.It is an xml reside in base directory of project as pom.xml.

Pom contained the details of Project and various configuration details used by the Maven to build the application.

Pom also contains the goal and plugin. While executing the task or goals ,Maven looks for the POM in the current directory .It reads the pom ,get the needed configuration details and the executed the goals .

Some of the configuration that can be provide as following

* project dependencies
* plugins
* goals
* build profiles
* project version
* developers
* mailing list

Before creating a POM, we should first decide the project group (groupId), its name (artifactId) and its version as these attributes help in uniquely identifying the project in repository.

POM Example

<project xmlns = "http://maven.apache.org/POM/4.0.0"

xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation = "http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.companyname.project-group</groupId>

<artifactId>project</artifactId>

<version>1.0</version>

</project>

**4.what is Maven?**

Ans:-Maven is Proejct management and comprehensive tools which provides developer complete build life cycle framework.

**5. Default (or Build) Lifecycle**

This is the primary life cycle of Maven and is used to build the application. It has the following 23 phases.

|  |  |
| --- | --- |
| Sr.No. | Lifecycle Phase & Description |
| 1 | validate  Validates whether project is correct and all necessary information is available to complete the build process. |
| 2 | initialize  Initializes build state, for example set properties. |
| 3 | generate-sources  Generate any source code to be included in compilation phase. |
| 4 | process-sources  Process the source code, for example, filter any value. |
| 5 | generate-resources  Generate resources to be included in the package. |
| 6 | process-resources  Copy and process the resources into the destination directory, ready for packaging phase. |
| 7 | compile  Compile the source code of the project. |
| 8 | process-classes  Post-process the generated files from compilation, for example to do bytecode enhancement/optimization on Java classes. |
| 9 | generate-test-sources  Generate any test source code to be included in compilation phase. |
| 10 | process-test-sources  Process the test source code, for example, filter any values. |
| 11 | test-compile  Compile the test source code into the test destination directory. |
| 12 | process-test-classes  Process the generated files from test code file compilation. |
| 13 | test  Run tests using a suitable unit testing framework (Junit is one). |
| 14 | prepare-package  Perform any operations necessary to prepare a package before the actual packaging. |
| 15 | package  Take the compiled code and package it in its distributable format, such as a JAR, WAR, or EAR file. |
| 16 | pre-integration-test  Perform actions required before integration tests are executed. For example, setting up the required environment. |
| 17 | integration-test  Process and deploy the package if necessary into an environment where integration tests can be run. |
| 18 | post-integration-test  Perform actions required after integration tests have been executed. For example, cleaning up the environment. |
| 19 | verify  Run any check-ups to verify the package is valid and meets quality criteria. |
| 20 | install  Install the package into the local repository, which can be used as a dependency in other projects locally. |
| 21 | deploy  Copies the final package to the remote repository for sharing with other developers and projects. |

**6.)How many types of build profile is present in Maven ?**

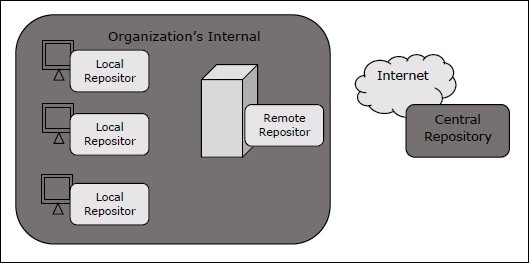
|  |  |
| --- | --- |
| Per Project | Defined in the project POM file, pom.xml |
| Per User | Defined in Maven settings xml file (%USER\_HOME%/.m2/settings.xml) |
| Global | Defined in Maven global settings xml file (%M2\_HOME%/conf/settings.xml) |

7.What is repository in maven?How many types of repository are available to use?

In Maven terminology, a repository is a directory where all the project jars, library jar, plugins or any other project specific artifacts are stored and can be used by Maven easily.

Maven repository are of three types. The following illustration will give an idea regarding these three types.

* local
* central
* remote



Local Repository

Maven local repository is a folder location on your machine. It gets created when you run any maven command for the first time.

Maven local repository keeps your project's all dependencies (library jars, plugin jars etc.). When you run a Maven build, then Maven automatically downloads all the dependency jars into the local repository. It helps to avoid references to dependencies stored on remote machine every time a project is build.

Maven local repository by default get created by Maven in %USER\_HOME% directory. To override the default location, mention another path in Maven settings.xml file available at %M2\_HOME%\conf directory.

<settings xmlns = "http://maven.apache.org/SETTINGS/1.0.0"

xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation = "http://maven.apache.org/SETTINGS/1.0.0

http://maven.apache.org/xsd/settings-1.0.0.xsd">

<localRepository>C:/MyLocalRepository</localRepository>

</settings>

When you run Maven command, Maven will download dependencies to your custom path.

Central Repository

Maven central repository is repository provided by Maven community. It contains a large number of commonly used libraries.

When Maven does not find any dependency in local repository, it starts searching in central repository using following URL − <https://repo1.maven.org/maven2/>

Key concepts of Central repository are as follows −

* This repository is managed by Maven community.
* It is not required to be configured.
* It requires internet access to be searched.

To browse the content of central maven repository, maven community has provided a URL − <https://search.maven.org/#browse>. Using this library, a developer can search all the available libraries in central repository.

Remote Repository

Sometimes, Maven does not find a mentioned dependency in central repository as well. It then stops the build process and output error message to console. To prevent such situation, Maven provides concept of Remote Repository, which is developer's own custom repository containing required libraries or other project jars.

For example, using below mentioned POM.xml, Maven will download dependency (not available in central repository) from Remote Repositories mentioned in the same pom.xml.

<project xmlns = "http://maven.apache.org/POM/4.0.0"

xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation = "http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.companyname.projectgroup</groupId>

<artifactId>project</artifactId>

<version>1.0</version>

<dependencies>

<dependency>

<groupId>com.companyname.common-lib</groupId>

<artifactId>common-lib</artifactId>

<version>1.0.0</version>

</dependency>

<dependencies>

<repositories>

<repository>

<id>companyname.lib1</id>

<url>http://download.companyname.org/maven2/lib1</url>

</repository>

<repository>

<id>companyname.lib2</id>

<url>http://download.companyname.org/maven2/lib2</url>

</repository>

</repositories>

</project>

Sequence

Local🡪central🡪remote repository.