Maven

Maven is a software management and comprehension tool based on the concept of Project Object Model(POM) which can manage project build,reporting and documentation from a central piece of information.

All build systems are essentially perform same tasks.

- Compile Source Code
- Copy Resources
- Compile and Run Test
- Package Project
- Deploy Project
- CleanUp

What is POM(Project Object Model)?

POM is an XML file that contains information about project and configuration details used by Maven to build the project.

- Describes a Project
- Name and Version,Artifact Type,Source code Locations,Dependencies
- Plugins
- Profiles(Alternate build configurations)
- Use XML by default
- Not the way Ant uses XML.

Maven Objectives:

- Making the build process easy
- providing a uniform build system
- providing quality project information
- providing guidelines for best practices development
- allowing transparent migration to new features

Installing Maven:

Pre-Requisite:

Install java version 1.8 and set Environmental variable path.(..\jdk1.8\bin)

Steps to Install Maven:

Go to Download Apache Maven 3.3.9

Download the Binary Zip archive file.

Extract the zip-file at your desired location

System Environment Variable --> set path -->(../apache-maven-3.3.9/bin)

To chcek version: Command Prompt --> mvn -version.

Creating a maven Project using CLI:

1.Go to the folder you wanna save your project

>mvn archetype:generate

Type of project:1368

Archetype id:

Group id:

Version:

Successfully created….

2.Go to Eclipse and import the existing project.

We can add dependencies to POM.xml file using Maven Repository.

Generate a Jar file for Maven project.

1. mvn clean
2. mvn compile
3. mvn test-compile
4. mvn test
5. mvn install (create a jar file in Target folder)

Maven build lifecycle

- Validate
- Compile
- Test
- Package
- Integration-test
- Verify
- Install
- Deploy

Transitive Dependencies in Maven:

Excluding Maven Dependencies .

```
 <excludes>
        <exclude>dependency name and its tag</exclude>
</excludes>
```

Scope Dependencies in Maven:

Specifically for test only then you can add <scope>test</scope>

How to Setup Jenkins for Maven Project:

1. Generate maven-demo project and go to directory where pom.xml resides.
2. On Github → Create new Repository → maven-demo
3. On Gitbash :
    a. git init
    b. git add .
    c. git commit -m “Created Maven Project”
    d. git remote add origiN CLONED IINK
    e. git push -u origin master
    f. check on Github account
4. Go to→ localhost:8080 (Jenkins)
    a. New Item →choose Freestyle Project (Name: Maven-Demo)

b. General >> Select github url = https://github.com/angelD25/maven-demo
c. Source code management >> Select git → Repository url = https://github.com/angelD25/maven-demo.git
d. Build Triggers>> Select → GitHub hook trigger for GITScm polling
e. Build environment >> Select → Delete workspace before build starts
f. Build >> Select → Invoke top-level maven target
   Goals : clean test
g. Save and Build the project and check console output.