MAVEN

- Maven is a automation and project management tool developed by apache software foundation. It is based on POM (project object model).
- Maven can build any number of projects into desired output such as .jar, .war, metadata.
- Mostly used for java-based project.
- It was initially released on 13th July 2004.
- Maven is written in java language.
- Meaning of maven is 'accumulator of knowledge'.
- Maven helps to getting the right .jar file for each project as there may be different version of separate packages.
- To download dependencies, it is no more needed to visit the official website of each software. It could now be easily done by visiting 'mvnrepository.com'
- Dependencies: it refers to the java libraries that are needed for the project.
- Repositories: it refers to the directories of packaged .jar files.

Build tools:

C, C++: make file

.Net: visual studio

Java: Ant, Maven, Gradle

Problems without Maven:

1. Adding set of jar in each project: in case if struts, spring we need to add jar files in each project. It must include all the dependencies of jars also.
2. Creating the right project structure: we must create the right project structure in servlet, struts etc, otherwise it will not be executed.
3. Building and deploying the project: we must have to build and deploy the project so that it may work.

What Maven does?

1. It makes a project easy to build.
2. It provides project information (for e.g: log documents, cross reference sources, mailing list, dependencies list, unit testing)
3. Easy to add new dependencies, therefore apache maven helps to manage
   a. Build
   b. Dependencies
   c. Reports
   d. Release
   e. Distribution

What is Build Tool:

A build tool take care of everything for building a process. It does following:

- Generate source code.
- Generate documentation from source code.

- Compiles source code.
- Install the package code in local repo., server repo. Or central repo.

POM (project object model):

- POM refers to the .xml file that have all the information regarding project and configuration details.
- Main configuration file in pom.xml.
- It has the description of the project details regarding the version and configuration management of the project.
- The .xml file is in the project home directory.
- Pom.xml contains:
    - Metadata
    - Dependencies
    - Kind of project
    - Kind of output (.jar, .war)
    - Description

One project – one workspace – one pom.xml file

Requirement for Build:

- Source code (present in workspace)
- Compiler (remote repo – local repo – work space)
- Dependencies (remote repo – local repo – work space)

Local Repository: local repo. refers to the machine of the developer where all the project material is saved.

Remote Repository: it refers to the repo. present on a webserver which is used when maven needs to download dependencies. This repo. works sane as the central repo. whenever anything is needed form remote repo. it is first downloaded to the local repo. and then it is used.

Central Repository: central repo. refers to the maven community that comes into action when there is a need od dependencies and those dependencies cannot be found in the local repo.

Maven Build Life-Cycle

Goals:

1. Generate resources (dependencies)
2. Compile code
3. Unit test
4. Package (build)

5. Install (into local repo. and artifactory)
6. Deploy (to server)
7. Clean (delete all run time files)
   e.g- mvn install
   mvn clean package
   1 to 6 – default and sequence order
   7 – not default and it won't allow sequence

- Build life cycle consists of a sequence of build phases and each build phase consists of a sequence of goals.
- Each goal is responsible for a particular task.
- When a phase is run all the goals related to that phase and its plugins are also compiled.
  e.g- mvn install
  mvn clean package

ANT:

- Ant doesn't have formal conventions, so we need to provide information of the project structure in build.xml file.
- Ant is procedural, you need to provide info about what to do and when to do through code.
- There is no lifecycle in ant.
- It is a tool box.
- It is mainly a build tool.
- It is less preferred than maven.

MAVEN:

- Maven have a convention to place source code, compiled code etc. so we don't need to provide information about the project structure in pom.xml file.
- Maven is declarative, everything you define in the pom.xml file.
- There is a life cycle in maven.
- It is a framework.
- It is mainly a project management tool.
- It is more preferred.

By:
Akshat Gupta
Linkedin: https://www.linkedin.com/in/akshat-gupta-57342938/
Email: akshu20791@gmail.com