APACHE MAVEN

WHAT IS APACHE MAVEN ?

Apache Maven is **a software/Build project management and comprehension tool**. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.

WHY DO I NEED MAVEN ?

Maven is chiefly used for Java-based projects, **helping to download dependencies**, which refers to the libraries or JAR files. The tool helps get the right JAR files for each project as there may be different versions of separate packages.

WHAT IS BUILD CYCLE ?

A Build Lifecycle is a well-defined sequence of phases, which define the order in which the goals are to be executed. Here phase represents a stage in life cycle. As an example, a typical Maven Build Lifecycle consists of the following sequence of phases.

APACHE MAVEN — POM

POM stands for Project Object Model. It is fundamental unit of work in Maven. It is an XML file that resides in the base directory of the project as pom.xml.
The POM contains information about the project and various configuration detail used by Maven to build the project(s).
POM also contains the goals and plugins. While executing a task or goal, Maven looks for the POM in the current directory. It reads the POM, gets the needed configuration information, and then executes the goal. Some of the configuration that can be specified in the POM are 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.

MAVEN BUILD CYCLE

Phase	Handles	Description
prepare-resources	resource copying	Resource copying can be customized in this phase.
validate	Validating the information	Validates if the project is correct and if all necessary information is available.
compile	compilation	Source code compilation is done in this phase.
Test	Testing	Tests the compiled source code suitable for testing framework.
package	packaging	This phase creates the JAR/WAR package as mentioned in the packaging in POM.xml.
install	installation	This phase installs the package in local/remote maven repository.
Deploy	Deploying	Copies the final package to the remote repository.

Clean Lifecycle

When we execute mvn post-clean command, Maven invokes the clean lifecycle consisting of the following phases.

- pre-clean
- clean
- post-clean

Maven clean goal (clean:clean) is bound to the clean phase in the clean lifecycle. Its clean:cleangoal deletes the output of a build by deleting the build directory.

Thus, when mvn clean command executes, Maven deletes the build directory. We can customize this behavior by mentioning goals in any of the above phases of clean life cycle.
In the following example, We'll attach maven-antrun-plugin:run goal to the pre-clean, clean, and post-clean phases.

FULL MAVEN LINK