

MAVEN..?

Maven is a popular open-source build tool developed by the Apache Group to build, publish, and deploy several projects at once for better project management.

The tool provides allows developers to build and document the lifecycle framework.

USE OF MAVEN ?

When you are ready to start using Maven,keep these 3 things in mind.

Configure Maven in Java, using Project Object Model (POM) found in a pom.xml file.

All Maven-related configuration settings are found in the POM. You can edit and configure plug-ins in the <plugins> tag of a pom.xml file.

Maven provides default settings for configurations, so you don't have to add every configuration into the pom.xml file.

Features of Maven....?

- * A huge, continuously growing repository of user libraries
- * The ability to set up projects easily, using best practices
- * Dependency management, featuring automatic updating
- * Backwards compatible with previous versions
- * Strong error and integrity reporting
- * Automatic parent versioning

Alternate For Maven...?

- * Ant is the alternate of Maven.
- * Apache Ant tutorial provides basic and advanced concepts of Apache Ant technology. Our Apache Ant tutorial is designed for beginners and professionals.
- * Apache Ant is a Java library that is used to handle Microsoft Office Documents. Apache Ant is open source, can be used by JVM based programming languages

Advantages of Maven....?

- * Helps manage all the processes, such as building, documentation, releasing, and distribution in project management
- * Simplifies the process of project building
- * Increases the performance of the project and the building process
- * The task of downloading Jar files and other dependencies is done automatically
- * Provides easy access to all the required information

Maven Repository..?

- * A maven repository is a directory of packaged JAR file with pom.xml file.
- * Maven searches for dependencies in the repositories.

Commands Of Maven...?

- * mvn clean: Cleans the project and removes all files generated by the previous build.
- * mvn compile: Compiles source code of the project.
- * mvn test-compile: Compiles the test source code.
- * mvn test: Runs tests for the project.
- * mvn package: Creates JAR or WAR file for the project to convert it into a distributable format.
- * mvn install: Deploys the packaged JAR/ WAR file to the local repository.
- * mvn deploy: Copies the packaged JAR/ WAR file to the remote repository after compiling, running tests and building the project.

Create Maven...?

- * Create a new Maven project in Eclipse.
- * From the File menu, choose New, and then choose Project.
- * In the New Project window, choose Maven Project.
- * In the New Maven Project window, choose Create a simple project, and leave other default selections.
- * In the New Maven Project, Configure project windows, type the following Artifact information:

Group Id: doc-examples

Artifact Id: lambda-java-example

Version: 0.0.1-SNAPSHOT

Packaging: jar

Name: lambda-java-example

What Is POM...?

- * POM (Project Object)A Project Object Model or POM is the fundamental unit of work in Maven.
- * It is an XML file that contains information about the project and configuration details used by Maven to build the project.
- * It contains default values for most projects.

What is POM file used for?

POM Files: Project Object Model(POM) Files are XML file that contains information related to the project and configuration information such as dependencies,

source directory, plugin, goals etc. used by Maven to build the project.

When you should execute a maven command you give maven a POM file to execute the commands.

GroupId

- * GroupId uniquely identifies your project across all projects.
- * A group ID should follow Java's package name rules.
- * This means it starts with a reversed domain name you control.

artifactId:

- * ArtifactId is the name of the jar without version.
- * If you created it, then you can choose whatever name you want with lowercase letters and no strange symbols.
- * If it's a third party jar, you have to take the name of the jar as it's distributed.