

## 1. What is Maven?

**Answer:** Maven is a build automation tool used primarily for Java projects. It simplifies the build process by managing project dependencies, compiling code, running tests, and packaging applications. Maven uses an XML file (pom.xml) to define project structure, dependencies, and build processes.

## 2. What does POM stand for in Maven?

**Answer:** POM stands for Project Object Model. It is an XML file (pom.xml) that contains configuration details for a Maven project, including project dependencies, build settings, plugins, and other project-specific information.

## 3. What is a Maven lifecycle?

**Answer:** Maven defines a build lifecycle as a sequence of phases that represent the stages of the build process. The three main lifecycles are:

- **Default Lifecycle:** Handles project deployment (e.g., clean, compile, test, package, install, deploy).
- **Clean Lifecycle:** Handles project cleaning (e.g., pre-clean, clean, post-clean).
- **Site Lifecycle:** Handles generating project documentation (e.g., pre-site, site, post-site, site-deploy).

## 4. What is the role of the pom.xml file?

**Answer:** The pom.xml file is the core configuration file in a Maven project. It defines the project's dependencies, build plugins, goals, and other configuration details necessary for building and managing the project.

## 5. What is a Maven dependency?

**Answer:** A Maven dependency is an external library or component that a project requires to compile or run. Dependencies are specified in the pom.xml file and are automatically downloaded from Maven repositories.

## 6. How do you add a dependency in Maven?

**Answer:** To add a dependency in Maven, you need to include it in the <dependencies> section of the pom.xml file. For example:

xml

Copy code

```
<dependency>

  <groupId>com.example</groupId>

  <artifactId>example-library</artifactId>

  <version>1.0.0</version>

</dependency>
```

## 7. What is a Maven plugin?

**Answer:** A Maven plugin is a piece of software that provides additional capabilities to the Maven build process, such as compiling code, running tests, or creating JAR files. Plugins are defined and configured in the pom.xml file.

#### **8. What is the mvn clean command used for?**

**Answer:** The mvn clean command is used to remove all files generated by the previous build. It typically deletes the target directory where compiled classes and build artifacts are stored.

#### **9. What is the purpose of the mvn install command?**

**Answer:** The mvn install command compiles the code, runs tests, and packages the project into a JAR or WAR file, then installs the resulting artifact into the local Maven repository. This allows other projects on the same machine to use this artifact as a dependency.

#### **10. What is the mvn package command used for?**

**Answer:** The mvn package command compiles the code and packages it into a distributable format, such as a JAR or WAR file. It prepares the project for deployment or distribution.

#### **11. What is the mvn validate command used for?**

**Answer:** The mvn validate command validates the project's configuration and ensures that all necessary information is available before the build process begins. It does not perform any actual build tasks.

#### **12. What is the difference between mvn clean and mvn clean install?**

**Answer:**

- **mvn clean** removes all files generated by the previous build, such as the target directory.
- **mvn clean install** performs the clean operation and then compiles the code, runs tests, packages the project, and installs the artifact into the local Maven repository.

#### **13. What is a Maven repository?**

**Answer:** A Maven repository is a location where Maven stores and retrieves project artifacts (such as JAR files). There are local repositories (on your machine), remote repositories (e.g., Maven Central), and central repositories.

#### **14. What is the difference between compile and provided scope in Maven dependencies?**

**Answer:**

- **compile** scope: The dependency is available for compiling, testing, and running the project. It is included in the final artifact.
- **provided** scope: The dependency is required for compiling and testing but is expected to be provided by the runtime environment (e.g., servlet API). It is not included in the final artifact.

#### **15. What is a Maven profile?**

**Answer:** A Maven profile is a set of configuration settings that can be activated or deactivated to customize the build process. Profiles allow you to define different build configurations for different environments (e.g., development, testing, production).

**16. What is the purpose of the mvn deploy command?**

**Answer:** The mvn deploy command builds the project, runs tests, packages the project, and uploads the resulting artifact to a remote repository. This is typically used for sharing the artifact with other developers or projects.

**17. What is the role of the <build> section in the pom.xml file?**

**Answer:** The <build> section in the pom.xml file specifies build-related configurations, such as the source directory, output directory, plugins, and goals to be used during the build process.

**18. What is the purpose of the mvn test command?**

**Answer:** The mvn test command compiles the test code and runs the unit tests defined in the project. It helps ensure that the code behaves as expected before deployment.

**19. How does Maven handle transitive dependencies?**

**Answer:** Maven automatically manages transitive dependencies. When you add a dependency to your project, Maven also includes its dependencies (transitive dependencies) in the build process. This is done by resolving and downloading all necessary dependencies from the Maven repository.

**20. What is the mvn site command used for?**

**Answer:** The mvn site command generates a site for the project, including reports and documentation. It creates a set of HTML pages that provide information about the project, such as code coverage, dependencies, and project metrics.