

## 25 Maven Interview Questions and Answers

In this section, we'll look at some common Maven interview questions and answers that are asked in technical interviews.

### Q1: What are the different aspects that the Maven tool manages?

Maven manages the following aspects of project management:

- Building
- Reporting
- Documentation
- Distribution
- Dependencies
- SCMs
- Releases
- Mailing lists

### Q2: What are the three build life cycles in Maven?

The three build lifecycles in Maven include:

1. Clean: This function cleans up artifacts created by previous builds
2. Default: This function is used to design the application
3. Site: This function generates the site documentation

### Q3: Why should one use Maven for project management?

Maven works wonders in setting up the project quickly and doesn't include complex build files like build.xml. The dependencies of your Java project are automatically downloaded by Maven and stored in a local repository and can be retrieved in quick time, whenever needed.

Maven also helps keep the deployment file light and easy to deploy, as it brings all the jars together in a bundle and stores them in the local repository.

### Q4: What are the steps involved to install Maven for Windows?

Maven can be installed in multiple operating systems, including Windows, Linux, and Mac. Here are the steps to install Maven for Windows:

- Download the Maven installation file and extract the file
- Add Java\_Home and Maven\_Home to the environment variables
- IN the Maven variable, add the environment path
- Check the version and verify the authenticity of the software

## **Q5: How do you know what version of Maven you're currently using?**

You can know the current Maven version by typing the command: `maven-version`.

## **Q6: What is POM in Maven?**

POM in Maven stands for Project Object Model. POM is an XML file in Maven and the fundamental unit that contains crucial information regarding the project and other related configuration details needed to run the project.

## **Q7: What is a Maven artifact?**

An artifact is a Jar file that is usually deployed to a repository. The build function creates artifacts such as a source Jar or a compiled Jar. Every artifact that is deployed comprises an artifact ID, a group ID, and a version string. Artifacts in Maven are identified by these three parameters.

## **Q8: What is the Maven command to build your site?**

The command to build your site in Maven is: `mvn site`

## **Q9: What is the function of the mvnclean command?**

The mvnclean command deletes the target directory with all its data before commencing the build process for the project.

## **Q10. What do you understand about the Maven Build Life Cycle?**

A Build Life Cycle in Maven specifies the order in which tasks need to be carried out and accomplished. Every build phase in Maven consists of a set of tasks or goals. This essentially implies that if one build life cycle is executed, all the tasks in that particular phase are executed. If a build phase is executed, the phases before it are executed.

## **Q11. What are the phases that a clean life cycle in Maven consists of?**

A clean life cycle in Maven consists of the following three phases:

- The pre-clean phase
- The clean phase
- The post-clean phase

## Q12. What is the function of the command `-- mvn clean dependency:copy-dependencies` package?

This command cleans the entire project and copies the dependencies in the project before packaging it.

## Q13. What do you understand about the Maven repository?

A Maven repository is a place where all the artifacts related to the project, such as library jars, project jars, and plugins, are stored in order to be used for different project tasks.

## Q14. What are some of the uses of plugins used in Maven?

Plugins in Maven are primarily used to:

- Create code files
- Create Jar files
- Create War files
- Create documentation for projects
- Create detailed project reports

## Q15. Explain the different types of Maven repositories

There are three main types of Maven repositories. They are:

1. The local repository: A local repository is a folder that is present in your current system/machine. Local repositories are created when you run Maven commands for the first time. The dependencies of your current project can be found in the local Maven repository.
2. The central repository: This repository contains a bunch of libraries and dependencies provided by the Maven community. If Maven doesn't locate a certain dependency in the local repository, it begins searching in the central repository.
3. The Remote Repository: If Maven is unable to locate a dependency in the central repository, it halts the build process and returns an error message. To avoid this, the provision for a remote repository has been made. A remote repository is a collection of custom libraries and other dependencies specific to the project.

## Q16. What are the different Build Profiles in Maven?

There are three types of Build Profiles in Maven.

1. **Per project:** This is defined in the POM.xml file
2. **Global:** This is defined in the Maven Global XML file:  
`<code>~/.m2/conf/settings.xml</code>`
3. **Per user:** This is defined in the Maven XML file: `<code>~/.m2/settings.xml</code>`

## Q17. What are the different phases that a given site life cycle consists of?

The different phases in a site life cycle are:

1. Pre-site
2. Site
3. Post-site
4. Site Deploy

## Q18. What is the default location of the local Maven repository?

The default location of the local Maven repository is `~/.m2/repository`

## Q19. What are the different types of plugins used in Maven?

There are basically two types of plugins that Maven provides for:

1. **The build plugins:** These plugins are used during the Build process and can be configured in the `<build/>` element of the POM.xml file.
2. **Reporting plugins:** These plugins come into effect during the site generation process. They can be configured in the `<reporting/>` element of the POM.xml file.

## Q20. What is an Archetype?

Archetypes are Maven plugins that are used to create the structure of the project in accordance with the specifications of a predefined template.

## Q21. What Maven command would you use to create a new project that's based on an Archetype?

The command used for this function is `-- maven archetype:generate`.

## Q22. When does the “External Dependency” activity come into the picture while executing a Maven project?

The External Dependency feature comes to the fore when dependencies aren't located in the local repository, and certain libraries need to be fetched from the central repository.

## Q23. What do you understand about transitive dependency in Maven?

Transitive dependency is when Maven can automatically locate libraries and other dependencies required for the project without you needing to manually locate these dependencies.

## Q24. What are the different dependency scopes in Maven?

Dependency scopes include project dependencies that are in line with the current, active stage of the build process. The different Dependency scopes in Maven are:

1. **Compile:** This scope specifies that the current dependency is available in the classpath of the current project.
2. **Provided:** This scope specifies that the web server will provide the dependency during runtime.
3. **Runtime:** This particular scope indicates that the dependency isn't needed during compile time and is only needed during execution.
4. **Test:** This scope indicates that the dependency is available limitedly during the test compilation phase.
5. **System:** This scope specifies that the system path for the given build phase needs to be provided.
6. **Import:** This scope indicates that the given POM should be replaced with congruent dependencies in the POM's <DependencyManagement> section.

## Q25. How would you run the “clean” plugin during the build process?

You can run the clean plugin by placing it inside the execution tag of the POM.xml file.