

MAVEN INTERVIEW QUESTIONS

http://www.tutorialspoint.com/maven/maven_interview_questions.htm

Copyright © tutorialspoint.com

Dear readers, these **Maven Interview Questions** have been designed specially to get you acquainted with the nature of questions you may encounter during your interview for the subject of **Maven**. As per my experience good interviewers hardly plan to ask any particular question during your interview, normally questions start with some basic concept of the subject and later they continue based on further discussion and what you answer:

What are the steps involved in project deployment?

Normally a deployment process consists of following steps –

- Check-in the code from all projects in progress into the SVN or source code repository and tag it.
- Download the complete source code from SVN.
- Build the application.
- Store the build output either WAR or EAR file to a common network location.
- Get the file from network and deploy the file to the production site.
- Updated the documentation with date and updated version number of the application.

What is Maven?

Maven is a project management and comprehension tool. Maven provides developers a complete build lifecycle framework. Development team can automate the project's build infrastructure in almost no time as Maven uses a standard directory layout and a default build lifecycle.

What does it mean when you say Maven uses Convention over Configuration?

Maven uses Convention over Configuration which means developers are not required to create build process themselves. Developers do not have to mention each and every configuration details.

What are the aspects Maven manages?

Maven provides developers ways to manage following –

- Builds
- Documentation
- Reporting
- Dependencies
- SCMs
- Releases
- Distribution
- mailing list

How do you know the version of mvn you are using?

Type the following command –

```
mvn --version
```

What is POM?

POM stands for Project Object Model. It is fundamental Unit of Work in Maven. It is an XML file. It always resides in the base directory of the project as pom.xml. It contains information about the project and various configuration details used by Maven to build the projects.

What information does POM contain?

POM contains the some of the following configuration information –

- project dependencies
- plugins
- goals
- build profiles
- project version
- developers
- mailing list

What is Maven artifact?

An artifact is a file, usually a JAR that gets deployed to a Maven repository. A Maven build produces one or more artifacts, such as a compiled JAR and a "sources" JAR.

Each artifact has a group ID *usually are reversed domain name, like com.example.foo*, an artifact ID *just a name*, and a version string. The three together uniquely identify the artifact. A project's dependencies are specified as artifacts.

What is Maven Build Lifecycle?

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.

Name the 3 build lifecycle of Maven.

The three build lifecycles are –

- **clean:** cleans up artifacts created by prior builds.
- **default or build:** This is used to build the application.
- **site:** generates site documentation for the project.

What is the command to quickly build your Maven site?

Type the command –

```
mvn site
```

What would the command *mvn clean* do ?

This command removes the target directory with all the build data before starting the build process.

What are the phases of a Maven Build Lifecycle?

Following are the phases –

- **validate** – validate the project is correct and all necessary information is available.
- **compile** – compile the source code of the project.
- **test** – test the compiled source code using a suitable unit testing framework. These tests should not require the code be packaged or deployed

- **package** – take the compiled code and package it in its distributable format, such as a JAR.
- **integration-test** – process and deploy the package if necessary into an environment where integration tests can be run.
- **verify** – run any checks to verify the package is valid and meets quality criteria.
- **install** – install the package into the local repository, for use as a dependency in other projects locally.
- **deploy** – done in an integration or release environment, copies the final package to the remote repository for sharing with other developers and projects.

What is a goal in Maven terminology?

A goal represents a specific task which contributes to the building and managing of a project. It may be bound to zero or more build phases. A goal not bound to any build phase could be executed outside of the build lifecycle by direct invocation.

What would this command do *mvn clean dependency:copy-dependencies package*?

This command will clean the project, copy the dependencies and package the project (executing all phases up to *package*).

What phases does a Clean Lifecycle consist?

The clean lifecycle consists of the following phases –

- pre-clean
- clean
- post-clean

What phases does a Site Lifecycle consist?

The phases in Site Lifecycle are –

- pre-site
- site
- post-site
- site-deploy

What is Build Profile?

A Build profile is a set of configuration values which can be used to set or override default values of Maven build. Using a build profile, you can customize build for different environments such as Production v/s Development environments.

What are different types of Build Profiles?

Build profiles are of three types –

- **Per Project** – Defined in the project POM file, pom.xml.
- **Per User** – Defined in Maven settings xml file .
- **Global** – Defined in Maven global settings xml file

How can you activate profiles?

A Maven Build Profile can be activated in various ways –

- Explicitly using command console input.
- Through maven settings.

- Based on environment variables *User/System variables*.
- OS Settings *forexample, Windowsfamily*.
- Present/missing files.

What is a Maven Repository?

A repository is a place i.e. directory where all the project jars, library jar, plugins or any other project specific artifacts are stored and can be used by Maven easily.

What types of Maven repository?

Maven repository are of three types: local, central, remote

What is local repository?

Maven local repository is a folder location on your machine. It gets created when you run any maven command for the first time. Maven local repository keeps your project's all dependencies *libraryjars, pluginjarsetc*.

What is the default location for your local repository?

~/m2/repository.

What is the command to install JAR file in local repository?

mvn install

What is Central Repository?

It is repository provided by Maven community. It contains a large number of commonly used libraries. When Maven does not find any dependency in local repository, it starts searching in central repository using following URL: <http://repo1.maven.org/maven2/>.

What is Remote Repository?

Sometimes, Maven does not find a mentioned dependency in central repository as well then it stops the build process and output error message to console. To prevent such situation, Maven provides concept of Remote Repository which is developer's own custom repository containing required libraries or other project jars.

What is the sequence in which Maven searches for dependency libraries?

Following is the search pattern –

- Step 1 – Search dependency in local repository, if not found, move to step 2 else if found then do the further processing.
- Step 2 – Search dependency in central repository, if not found and remote repository/repositories is/are mentioned then move to step 4 else if found, then it is downloaded to local repository for future reference.
- Step 3 – If a remote repository has not been mentioned, Maven simply stops the processing and throws error *Unabletofinddependency*.
- Step 4 – Search dependency in remote repository or repositories, if found then it is downloaded to local repository for future reference otherwise Maven as expected stop processing and throws error *Unabletofinddependency*.

Why are Maven Plugins used?

Maven Plugins are used to –

- create jar file.
- create war file.

- compile code files.
- unit testing of code
- create project documentation.
- create project reports.

What are the types of Maven Plugins?

Maven provides following two types of Plugins –

- **Build plugins** – They execute during the build and should be configured in the `<build/>` element of `pom.xml`
- **Reporting plugins** – They execute during the site generation and they should be configured in the `<reporting/>` element of the `pom.xml`

When does Maven use External Dependency concept?

Maven dependency management using concept of Maven Repositories *Local, Central, Remote*. Suppose dependency is not available in any of remote repositories and central repository; in such scenarios Maven uses concept of External Dependency.

What are the things you need to define for each external dependency?

External dependencies *libraryjarlocation* can be configured in `pom.xml` in same way as other dependencies.

- Specify groupId same as name of the library.
- Specify artifactId same as name of the library.
- Specify scope as system.
- Specify system path relative to project location.

What is Archetype?

Archetype is a Maven plugin whose task is to create a project structure as per its template.

What is the command to create a new project based on an archetype?

Type the following command –

```
mvn archetype:generate
```

What is SNAPSHOT in Maven?

SNAPSHOT is a special version that indicates a current development copy. Unlike regular versions, Maven checks for a new SNAPSHOT version in a remote repository for every build.

What is difference between Snapshot and Version?

In case of Version, if Maven once downloaded the mentioned version say `data-service:1.0`, it will never try to download a newer 1.0 available in repository. To download the updated code, `data-service` version is be upgraded to 1.1.

In case of SNAPSHOT, Maven will automatically fetch the latest SNAPSHOT `data – service: 1.0 – SNAPSHOT` everytime app-ui team build their project.

What is transitive dependency in Maven?

Transitive dependency means to avoid needing to discover and specify the libraries that your own dependencies require, and including them automatically.

What does dependency management mean in the context of transitive dependency?

It means to directly specify the versions of artifacts to be used when they are encountered in transitive dependencies. For an example project C can include B as a dependency in its dependencyManagement section and directly control which version of B is to be used when it is ever referenced.

How Maven handles and determines what version of dependency will be used when multiple version of an artifact are encountered?

Maven determines what version of a dependency is to be used when multiple versions of an artifact are encountered. If two dependency versions are at the same depth in the dependency tree, the first declared dependency will be used. This is called dependency mediation.

What is dependency scope? Name all the dependency scope.

Dependency scope includes dependencies as per the current stage of the build. Various Dependency Scopes are –

- **compile** – This scope indicates that dependency is available in classpath of project. It is default scope.
- **provided** – This scope indicates that dependency is to be provided by JDK or web-Server/Container at runtime.
- **runtime** – This scope indicates that dependency is not required for compilation, but is required during execution.
- **test** – This scope indicates that the dependency is only available for the test compilation and execution phases.
- **system** – This scope indicates that you have to provide the system path.
- **import** – This scope is only used when dependency is of type pom. This scope indicates that the specified POM should be replaced with the dependencies in that POM's <dependencyManagement> section.

What is the minimal set of information for matching a dependency references against a dependencyManagement section ?

{groupId,artifactId,type,classifier}.

How do you reference a property defined in your pom.xml file?

To reference a property defined in your pom.xml, the property name uses the names of the XML elements that define the value, with "pom" being allowed as an alias for the project *root* element.

So *pom.name* refers to the name of the project, {pom.version} refers to the version of the project, \${pom.build.finalName} refers to the final name of the file created when the built project is packaged, etc.

What are the default values for packaging element? If there is no packaging element defined? What is the default value for that?

Some of the valid packaging values are jar, war, ear and pom. If no packaging value has been specified, it will default to jar.

What is the value for packaging element in pom for a project that is purely meta-data?

pom

What is the use of execution element in pom file?

The <execution> element contains information's required for the execution of a plugin.

What is a project's fully qualified artifact name?

<groupId>:<artifactId>:<version>

If you do not define any information, where does your pom inherits that information from?

All POMs inherit from a parent *despite explicitly defined or not*. This base POM is known as the Super POM, and contains values inherited by default.

How profiles are specified in Maven?

Profiles are specified using a subset of the elements available in the POM itself.

What are the elements in POM that a profile can modify when specified in the POM?

<repositories>, <pluginRepositories>, <dependencies>, <plugins>, <properties>, <modules>, <reporting>, <dependencyManagement>, <distributionManagement>

Why profile is used in Maven?

To give portability to projects *e. g. windows, linux etc.*

What are the benefit of storing JARS/external dependencies in local repository instead of remote one?

It uses less storage, it makes checking out project quicker, non need for versioning JAR files.

How can you build your project offline?

Use the command –

```
mvn o package.
```

How do you exclude dependency?

Using the *exclusion* element.

What is a system dependency?

Dependency with scope system are always available and are not looked up in repository, they are usually used to tell Maven about dependencies which are provided by the JDK or the VM. Thus, system dependencies are especially useful for resolving dependencies on artifacts which are now provided by the JDK.

What is the use of optional dependency?

Any transitive dependency can be marked as optional using "optional" element. As example, A depends upon B and B depends upon C. Now B marked C as optional. Then A will not use C.

What is dependency exclusion ?

Any transitive dependency can be exclude using "exclusion" element. As example, A depends upon B and B depends upon C then A can mark C as excluded.

How can you run the clean plugin automatically during the build?

You can put the clean plugin inside the execution tag in pom.xml file.

How to stop the propagation of plugins to child POMs?

set <inherited> to false.

What does the "You cannot have two plugin executions with the same *or missing* elements" message mean?

It means that you have executed a plugin multiple times with the same <id>. Provide each <execution> with a unique <id> then it would be ok.

What is a Mojo?

A mojo is a Maven plain Old Java Object. Each mojo is an executable goal in Maven, and a plugin is a distribution of one or more related mojos.

What is difference between Apache Ant and Maven?

Ant is simply a toolbox whereas Maven is about the application of patterns in order to achieve an infrastructure which displays the characteristics of visibility, reusability, maintainability, and comprehensibility. It is wrong to consider Maven as a build tool and just a replacement for Ant.

What is Next ?

Further you can go through your past assignments you have done with the subject and make sure you are able to speak confidently on them. If you are fresher then interviewer does not expect you will answer very complex questions, rather you have to make your basics concepts very strong.

Second it really doesn't matter much if you could not answer few questions but it matters that whatever you answered, you must have answered with confidence. So just feel confident during your interview. We at tutorialspoint wish you best luck to have a good interviewer and all the very best for your future endeavor. Cheers :-)

Processing math: 100%