[](http://career.guru99.com/wp-content/uploads/2014/04/Git.jpg)

[**Top 40 GIT Interview Questions & Answers**](http://career.guru99.com/top-40-interview-questions-on-git/)

**1)      What is GIT?**

GIT is a distributed version control system and source code management (SCM) system with an emphasis to handle small and large projects with speed and efficiency.

**2)      What is a repository in GIT?**

A repository contains a directory named .git, where git keeps all of its metadata for the repository. The content of the .git directory are private to git.

**3)      What is the command you can use to write a commit message?**

The command that is used to write a commit message is “git commit –a”.  The –a on the command line instructs git to commit the new content of all tracked files that have been modified. You can use “git add<file>” before git commit –a if new files need to be committed for the first time.

**4)      What is the difference between GIT and SVN?**

The difference between GIT and SVN is

a)      Git is less preferred for handling extremely large files or frequently changing binary files while SVN can handle multiple projects stored in the same repository.

b)      GIT does not support ‘commits’ across multiple branches or tags.  Subversion allows the creation of folders at any location in the repository layout.

c)        Gits are unchangeable, while Subversion allows committers to treat a tag as a branch and to create multiple revisions under a tag root.

**5)      What are the advantages of using GIT?**

a)      Data redundancy and replication

b)      High availability

c)       Only one.git directory per repository

d)      Superior disk utilization and network performance

e)      Collaboration friendly

f)       Any sort of projects can use GIT

**6)      What language is used in GIT?**

GIT is fast, and ‘C’ language makes this possible by reducing the overhead of runtimes associated with higher languages.

**7)      What is the function of ‘GIT PUSH’ in GIT?**

‘GIT PUSH’ updates remote refs along with associated objects.

**8)      Why GIT better than Subversion?**

GIT is an open source version control system; it will allow you to run ‘versions’ of a project, which show the changes that were made to the code overtime also it allows you keep the backtrack if necessary and undo those changes.  Multiple developers can checkout, and upload changes and each change can then be attributed to a specific developer.

**9)      What is “Staging Area” or “Index” in GIT?**

Before completing the commits, it can be formatted and reviewed in an intermediate area known as ‘Staging Area’ or ‘Index’.

**10)   What is GIT stash?**

GIT stash takes the current state of the working directory and index and puts in on the stack for later and gives you back a clean working directory.  So in case if you are in the middle of something and need to jump over to the other job, and at the same time you don’t want to lose your current edits then you can use GIT stash.

**11)   What is GIT stash drop?**

When you are done with the stashed item or want to remove it from the list, run the git ‘stash drop’ command.  It will remove the last added stash item by default, and it can also remove a specific item if you include as an argument.

**12)   How will you know in GIT if a branch has been already merged into master?**

Git branch—merged lists the branches that have been merged into the current branch

Git branch—-no merged lists the branches that have not been merged

**13)   What is the function of git clone?**

The git clone command creates a copy of an existing Git repository.  To get the copy of a central repository, ‘cloning’  is the most common way used by programmers.

**14)   What is the function of ‘git config’?**

The ‘git config’ command is a convenient way to set configuration options for your Git installation.  Behaviour of a repository, user info, preferences etc. can be defined through this command.

**15)   What does commit object contain?**

a)      A set of files, representing the state of a project at a given point of time

b)      Reference to parent commit objects

c)       An SHAI name, a 40 character string that uniquely identifies the commit object.

**16)   How can you create a repository in Git?**

In Git, to create a repository, create a directory for the project if it does not exist, and then run command “git init”. By running this command .git directory will be created in the project directory, the directory does not need to be empty.

**17)   What is ‘head’ in git and how many heads can be created in a repository?**

A ‘head’ is simply a reference to a commit object. In every repository, there is a default head referred as “Master”.  A repository can contain any number of heads.

**18)   What is the purpose of branching in GIT?**

The purpose of branching in GIT is that you can create your own branch and jump between those branches. It will allow you to go to your previous work keeping your recent work intact.

**19)   What is the common branching pattern in GIT?**

The common way of creating branch in GIT is to maintain one as “Main“

branch and create another branch to implement new features. This pattern is particularly useful when there are multiple developers working on a single project.

**20)   How can you bring a new feature in the main branch?**

To bring a new feature in the main branch, you can use a command “git merge” or “git pull command”.

**21)   What is a ‘conflict’ in git?**

A ‘conflict’ arises when the commit that has to be merged has some change in one place, and the current commit also has a change at the same place. Git will not be able to predict which change should take precedence.

**22)   How can conflict in git resolved?**

To resolve the conflict in git, edit the files to fix the conflicting changes and then add the resolved files by running “git add” after that to commit the repaired merge,  run “git commit”.  Git remembers that you are in the middle of a merger, so it sets the parents of the commit correctly.

**23)   To delete a branch what is the command that is used?**

Once your development branch is merged into the main branch, you don’t need

development branch.  To delete a branch use, the command “git branch –d [head]”.

**24)   What is another option for merging in git?**

“Rebasing” is an alternative to merging in git.

**25)   What is the syntax for “Rebasing” in Git?**

The syntax used for rebase is “git rebase [new-commit] “

**26)   What is the difference between ‘git remote’ and ‘git clone’?**

‘git remote add’  just creates an entry in your git config that specifies a name for a particular URL.  While, ‘git clone’ creates a new git repository by copying and existing one located at the URI.

**27)   What is GIT version control?**

With the help of GIT version control, you can track the history of a collection of files and includes the functionality to revert the collection of files to another version.  Each version captures a snapshot of the file system at a certain point of time. A collection of files and their complete history are stored in a repository.

**28)   Mention some of the best graphical GIT client for LINUX?**

Some of the best GIT client for LINUX is

a)      Git Cola

b)      Git-g

c)       Smart git

d)      Giggle

e)      Git GUI

f)       qGit

**29)   What is Subgit? Why to use Subgit?**

‘Subgit’ is a tool for a smooth, stress-free SVN to Git migration.  Subgit is a solution for a company -wide migration from SVN to Git that is:

a)      It is much better than git-svn

b)      No requirement to change the infrastructure that is already placed

c)       Allows to use all git and all sub-version features

d)      Provides genuine stress –free migration experience.

**30)   What is the function of ‘git diff ’ in git?**

‘git diff ’ shows the changes between commits, commit and working tree etc.

**31)   What is ‘git status’ is used for?**

As ‘Git Status’ shows you the difference between the working directory and the index, it is helpful in understanding a git more comprehensively.

**32)   What is the difference between the ‘git diff ’and ‘git status’?**

‘git diff’ is similar to ‘git status’, but it shows the differences between various commits and also between the working directory and index.

**33)   What is the function of ‘git checkout’ in git?**

A ‘git checkout’ command is used to update directories or specific files in your working tree with those from another branch without merging it in the whole branch.

**34)   What is the function of ‘git rm’?**

To remove the file from the staging area and also off your disk ‘git rm’ is used.

**35)   What is the function of ‘git stash apply’?**

When you want to continue working where you have left your work, ‘git stash apply’ command is used to bring back the saved changes onto the working directory.

**36)   What is the use of ‘git log’?**

To find specific commits in your project history- by author, date, content or history ‘git log’ is used.

**37)   What is ‘git add’ is used for?**

‘git add’ adds file changes in your existing directory to your index.

**38)   What is the function of ‘git reset’?**

The function of ‘Git Reset’ is to reset your index as well as the working directory to the state of your last commit.

**39)   What is git Is-tree?**

‘git Is-tree’ represents a tree object including the mode and the name of each item and the SHA-1 value of the blob or the tree.

**40)   How git instaweb is used?**

‘Git Instaweb’ automatically directs a web browser and runs webserver with an interface into your local repository.

**41)   What does ‘hooks’ consist of in git?**

This directory consists of Shell scripts which are activated after running the corresponding Git commands.  For example, git will try to execute the post-commit script after you run a commit.

**42)   Explain what is commit message?**

Commit message is a feature of git which appears when you commit a change. Git provides you a text editor where you can enter the modifications made in commits.

**43)   How can you fix a broken commit?**

To fix any broken commit, you will use the command “git commit—amend”. By running this command, you can fix the broken commit message in the editor.

**44)   Why is it advisable to create an additional commit rather than amending an existing commit?**

There are couple of reason

a)      The amend operation will destroy the state that was previously saved in a commit.  If it’s just the commit message being changed then that’s not an issue.  But if the contents are being amended then chances of eliminating something important remains more.

b)      Abusing “git commit- amend” can cause a small commit to grow and acquire unrelated changes.

**45)   What is ‘bare repository’ in GIT?**

To co-ordinate with the distributed development and developers team, especially when you are working on a project from multiple computers ‘Bare Repository’ is used. A bare repository comprises of a version history of your code.

# [images](http://career.guru99.com/wp-content/uploads/2014/12/images1.jpg)

# [Top 12 JENKIN Interview Questions](http://career.guru99.com/top-12-jenkin-interview-questions/)

**1) Mention what is Jenkins?**

Jenkins is an open source tool with plugin built for continuous integration purpose.  The principle functionality of Jenkins is to keep a track of version control system and to initiate and monitor a build system if changes occur. It monitors the whole process and provides reports and notifications to alert.

**2) Explain what is continuous integration?**

In software development, when multiple developers or teams are working on different segments of same web application, we need to perform integration test by integrating all modules.  In order to do that an automated process for each piece of code is performed on daily bases so that all your code get tested.

**3) What is the requirement for using Jenkins?**

To use Jenkins you require

* A source code repository which is accessible, for instance, a Git repository
* A working build script, e.g., a Maven script, checked into the repository

**4) Mention what are the advantages of Jenkins?**

Advantage of Jenkins include

* At integration stage, build failures are cached
* For each code commit changes an automatic build report notification generates
* To notify developers about build report success or failure, it is integrated with LDAP mail server
* Achieves continuous integration agile development and test driven development
* With simple steps, maven release project is automated
* Easy tracking of bugs at early stage in development environment than production

**5) Explain how you can move or copy Jenkins from one server to another?**

* Slide a job from one installation of Jenkins to another by copying the related job directory
* Make a copy of an already existing job by making clone of a job directory by a different name
* Renaming an existing job by renaming a directory.

**6) Mention what are the commands you can use to start Jenkins manually?**

To start Jenkins manually, you can use either of the following

* (Jenkins\_url)/restart: Forces a restart without waiting for builds to complete
* (Jenkin\_url)/safeRestart: Allows all running builds to complete

**7) Mention some of the useful plugins in Jenkin?**

Some of the important plugins in Jenkin includes

* Maven 2 project
* Amazon EC2
* HTML publisher
* Copy artifact
* Join
* Green Balls

**8) Explain how you can deploy a custom build of a core plugin?**

To deploy a custom field of a core plugin, you have to do following things

* Stop Jenkins
* Copy the custom HPI to $Jenkins\_Home/plugins
* Delete the previously expanded plugin directory
* Make an empty file called <plugin>.hpi.pinned
* Start Jenkins

**9) Explain how can create a backup and copy files in Jenkins?**

Jenkins saves all the setting, build artifacts and logs in its home directory, to create a back-up of your Jenkins setup, just copy this directory. You can also copy a job directory to clone or replicate a job or rename the directory.

**10) Explain how you can clone a Git repository via Jenkins?**

To clone a Git repository via Jenkins, you have to enter the e-mail and user name for your Jenkins system.  For that, you have to switch into your job directory and execute the “git config” command.

**11) Explain how you can set up Jenkins job?**

To create a project that is handled via jobs in Jenkins.  Select New item from the menu, once this done enter a name for the job and select free-style job. Then click OK to create new job in Jenkins.  The next page enables you to configure your job.

**12) Mention what are the two components Jenkins is mainly integrated with?**

Jenkin is mainly integrated with two components

* Version Control system like GIT, SVN
* And build tools like Apache Maven.

**13)  What are the various ways in which build can be scheduled in Jenkins ?**

 Builds can be triggered by source code management  commits. Can be triggered after completion of other builds. Can be scheduled to run at specified time ( crons ) Manual Build Requests

**14) What you do to make sure that your project build doesn't break in Jenkins ?**

I make sure that I perform successful clean install on my local machine with all unit tests.

Then I make sure that I check in all code changes.

Then I do a Synchronize with repository to make sure that all required config and POM changes and any difference is checked into the repository.

**15)  What you do when you see a broken build for your project in Jenkins ?**

I will open the console output for the build and will try to see if any file changes were missed.

If not able to find the issue that way, Will clean and update my local workspace to replicate the problem on my local and will try to solve it.



**1) Explain what is Maven? How does it work?**

Maven is a project management tool. It provides the developer a complete build lifecycle framework. On executing Maven commands, it will look for POM file in Maven; it will run the command on the resources described in the POM.

**2) List out what are the aspects does Maven Manages?**

Maven handles following activities of a developer

• Build

• Documentation

• Reporting

• Dependencies

• SCMs

• Releases

• Distribution

• Mailing list

**3) Mention the three build lifecycle of Maven?**

• Clean: Cleans up artifacts that are created by prior builds

• Default (build): Used to create the application

• Site: For the project generates site documentation

**4) Explain what is POM?**

In Maven, POM (Project Object Model) is the fundamental unit of work. It is an XML file which holds the information about the project and configuration details used to build a project by Maven.

**5) Explain what is Maven artifact?**

Usually an artifact is a JAR file which gets arrayed to a Maven repository. One or more artifacts a maven build produces such as compiled JAR and a sources JAR.

Each artifact includes a group ID, an artifact ID and a version string.

**6) Explain what is Maven Repository? What are their types?**

A Maven repository is a location where all the project jars, library jars, plugins or any other particular project related artifacts are stored and can be easily used by Maven.

Their types are local, central and remote

**7) Why Maven Plugins are used?**

Maven plugins are used to

• Create a jar file

• Create war file

• Compile code files

• Unit testing of code

• Documenting projects

• Reporting

**8) List out the dependency scope in Maven?**

The various dependency scope used in Maven are:

• Compile: It is the default scope, and it indicates what dependency is available in the classpath of the project

• Provided: It indicates that the dependency is provided by JDK or web server or container at runtime

• Runtime: This tells that the dependency is not needed for compilation but is required during execution

• Test: It says dependency is available only for the test compilation and execution phases

• System: It indicates you have to provide the system path

• Import: This indicates that the identified or specified POM should be replaced with the

dependencies in that POM’s section

**9) Mention how profiles are specified in Maven?**

Profiles are specified in Maven by using a subset of the elements existing in the POM itself.

**10) Explain how you can exclude dependency?**

By using the exclusion element, dependency can be excluded

**11) Mention the difference between Apache Ant and Maven?**

Apache Ant Maven

• Ant is a toolbox - Maven is a framework

• Ant does not have formal conventions like project directory structure - Maven has conventions

• Ant is procedural; you have to tell to compile, copy and compress - Maven is declarative ( information on what to make & how to build)

• Ant does not have lifecycle; you have to add sequence of tasks manually - Maven has a lifecycle

• Ant scripts are not reusable - Maven plugins are reusable

**12) In Maven what are the two setting files called and what are their location?**

In Maven, the setting files are called settings.xml, and the two setting files are located at

• Maven installation directory: $M2\_Home/conf/settings.xml

• User’s home directory: ${ user.home }/ .m2 / settings.xml

**13) List out what are the build phases in Maven?**

Build phases in Maven are

• Validate

• Compile

• Test

• Package

• Install

• Deploy

**14) List out the build, source and test source directory for POM in Maven?**

• Build = Target

• Source = src/main/java

• Test = src/main/test

**15) Where do you find the class files when you compile a Maven project?**

You will find the class files ${basedir}/target/classes/.

**16) Explain what would the “jar: jar” goal do?**

jar: jar will not recompile sources; it will imply just create a JAR from the target/classes directory considering that everything else has been done

**17) List out what are the Maven’s order of inheritance?**

The maven’s order of inheritance is

• Parent Pom

• Project Pom

• Settings

• CLI parameters

**18) For POM what are the minimum required elements?**

The minimum required elements for POM are project root, modelVersion, groupID, artifactID and version

**19) Explain how you can produce execution debug output or error messages?**

To produce execution debug output you could call Maven with X parameter or e parameter

**20) Explain how to run test classes in Maven?**

To run test classes in Maven, you need surefire plugin, check and configure your settings in setting.xml and pom.xml for a property named “test.”

**Q1.  What is the best practice configuration usage for files - pom.xml or settings.xml ?**  
Ans. The best practice guideline between settings.xml and pom.xml is that configurations in settings.xml must be specific to the current user and that pom.xml configurations are specific to the project.  
 **Q2.  How can I change the default location of the generated jar when I command "mvn package"?**  
Ans. By default, the location of the generated jar is in ${project.build.directory} or in your target directory. We can change this by configuring the outputDirectory of maven-jar-plugin.  
 **Q3.  What is Maven's order of inheritance?**  
Ans.   
  
1. parent pom  
2. project pom  
3. settings  
4. CLI parameters  
 **Q4.  What is a Mojo?**  
Ans. 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.  
 **Q5.  How do I determine which POM contains missing transitive dependency?**  
Ans. run "mvn -X"  
  
**Q6. How would you see the version of Maven ?**  
  
Ans. mvn --version  
  
**Q7. Where do we configure repositories in Maven ?**  
  
Ans. Within settings.xml in either MAVEN\_HOME or .M2 directory.  
  
**Q8. What are different type of repositories in Maven ?**  
  
Ans. Local and Remote.   
  
**Q9. How does Maven looks for a dependency or resource ?**   
  
Ans. It refers to the settings.xml to look for the repositories to look for the resource. First It looks into the configured local repository, then it looks into the configured Remote repositories. If the resource is still not found , it looks it within maven repository central i.e repo1.maven.org. If its still not found, it throws the exception saying "Unable to find resource in repository central".  
  
**Q10. What is maven repository central ?**  
  
Ans. Its the repository provided by Maven. In case your POM specify the dependencies and its not available in the configured local and the remote repository. It then looks for the resource in Maven Central. Maven provides most of the generic dependency resources at this remote location.   
  
**Q11. What would you do if you have to add a jar to the project using Maven ?**  
  
Ans. If its already there in Maven local repository, We can add that as a dependency in the project pom file with its Group Id, Artifact Id and version.  
  
We can provide additional attribute SystemPath if its unable to locate the jar in the local repository.  
  
If its not there in the local repository, we can install it first in the local repository and then can add it as dependency.

**Q12. Have you ever had problem getting your projects in eclipse refreshed after you made changes in the Pom files ?**

Ans. Yes, It happens many times but I would usually perform mvn eclipse:eclipse and this would resolve the project refresh problems.

**Q13. What is the difference between compile and install ?**

Ans. Compile compiles the source code of the project

whereas  
  
Install installs the package into the local repository, for use as a dependency in other projects locally  
  
**Q14.  How can we see Dependencies for the project and where exactly they are defined ?**  
  
Ans. Using    
  
mvn dependency:tree

**Q15.  What is a transitive dependency ? Can we override Transitive Dependency version and If Yes, how ?**

Ans. Transitive dependency is the dependencies not defined directly in the current POM but the POM of the dependent projects.  Transitive dependencies allows to avoid specifying the libraries that are required by the project which are specified in other dependent projects - Remote or Local.

Yes we can override transitive dependency version by specifying the dependency in the current POM.

**Q16.  What is a cyclic dependency ?**

Ans. A has dependency of B, B has dependency of C and C has dependency of A.

With Maven 2 , came transitive dependency wherein in above scenario, C will acts as a dependency of A as if this dependency has been defined directly in A but the negative side is that if it leads to cyclic dependency , it creates problems.

**Multiple Choice Questions**  
 **Q14.  What is Maven's order of inheritance?**  
  
 a. parent pom -> project pom -> settings -> CLI parameters  
 b. project pom -> parent pom -> settings -> CLI parameters  
 c. settings -> parent pom -> project pom -> CLI parameters  
 d. CLI parameters -> settings -> parent pom -> project pom  
  
Ans. parent pom -> project pom -> settings -> CLI parameters  
  
**Q15.  How do I determine which POM contains missing transitive dependency?**  
  
 a. mvn -A  
 b. mvn -M  
 c. mvn -R  
 d. mvn -X  
  
Ans. mvn -X  
  
**Q16.  What is the sequence in which Maven looks for the resources ?**  
  
 a. Remote-> Maven Central - > Local  
 b. Local -> Remote - Maven Central  
 c. Remote-> Local -> Maven Central  
 d. Maven Central -> Local -> Remote  
  
Ans. Local -> Remote - Maven Central  
  
**Q17.  POM stands for ...**   
  
 a. Project Object Model  
 b. Project Oriented Model  
 c. Prospective Object Model  
 d. Prospective Objective Model  
  
Ans. Project Object Model  
  
**Q18.  Which of the following is not type of Maven Repository ?**  
  
 a. Local  
 b. Remote  
 c. Maven Central  
 d. Maven Local  
  
Ans. Maven Local  
  
**Q19.  What is the default location of local repository ?**  
  
 a. ~/.m2/repo  
 b. ~/m2./home/repository  
 c. ~/m2./repository  
 d. ~/m2./home/repo  
  
Ans. ~/m2./repository  
  
**Q20.  Which of the following is not type of Maven Plugin ?**  
  
 a. Build  
 b. Reporting  
 c. Remote  
 d. All are valid Maven Plugin types  
  
Ans. Remote  
  
**Q21.  Which of the following is not a dependency scope in Maven ?**  
  
 a. Compile  
 b. Test  
 c. System  
 d. Export  
  
Ans. Export  
  
**Q22.  What is a project's fully qualified artifact name?**  
  
 a. <groupId>:<artifactId>:<version>  
 b. <groupId>:<artifactId>  
 c. <artifactId>:<groupId>:<version>  
 d. <artifactId>:<version>  
  
Ans. <groupId>:<artifactId>:<version>