**Jenkins**

**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.

[Add to Bookmark](https://www.edureka.co/blog/interview-questions/jenkins-interview-questions/)  Email this Post  21.6K [**0**](https://www.edureka.co/blog/interview-questions/jenkins-interview-questions/#disqus_thread)

**Jenkins Interview Questions**

Just commit changes to the SCR (Source Code Repository) and Jenkins can automate the rest of the process for you with the help of plugins. So that makes it a very important DevOps tool. There is a high possibility that you encounter many Jenkins questions if you go for a DevOps job interview. Below are the most frequently asked Jenkins interview questions. I have collected these questions after doing a lot of research and after discussing with some DevOps experts who are directly involved in the hiring process. Curious to know more about Jenkins [***check out this Jenkins blog series***](https://www.edureka.co/blog/what-is-jenkins/).

This Jenkins Interview Questions blog is a part of parent blog [***DevOps Interview Questions***](https://www.edureka.co/blog/interview-questions/top-devops-interview-questions-2016/). It includes all the DevOps Stages.

First question in this Jenkins Interview Questions blog has to be:

**DevOps Interview Questions and Answers | DevOps Training | Edureka**

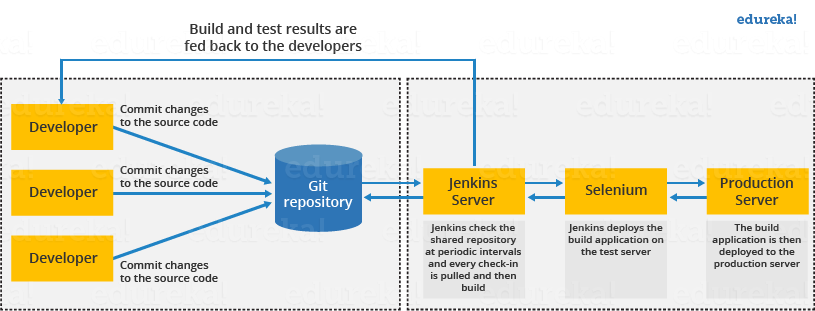
**Q1. What is Jenkins?**

My suggestion is to start this answer by giving a definition of Jenkins.

Jenkins is an open source automation tool written in Java with plugins built for Continuous Integration purpose. Jenkins is used to build and test your software projects continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build. It also allows you to continuously deliver your software by integrating with a large number of testing and deployment technologies.

Once you have defined Jenkins give an example, you can refer the below mentioned use case:

* First, a developer commits the code to the source code repository. Meanwhile, the Jenkins server checks the repository at regular intervals for changes.
* Soon after a commit occurs, the Jenkins server detects the changes that have occurred in the source code repository. Jenkins will pull those changes and will start preparing a new build.
* If the build fails, then the concerned team will be notified.
* If built is successful, then Jenkins deploys the built in the test server.
* After testing, Jenkins generates a feedback and then notifies the developers about the build and test results.
* It will continue to check the  source code repository for changes made in the source code and the whole process keeps on repeating.



*Interviewer now knows what is Jenkins but why we use it, there are many other CI tools as well, so why Jenkins?, the next question in this Jenkins interview questions will deal with that answer.*

**Q2. What are the benefits of using Jenkins?**

I will suggest you to include the following benefits of Jenkins, if you can recall any other benefit apart from the below mentioned points you can include that as well.

* At integration stage, build failures are cached.
* For each change in the source code an automatic build report notification is generated.
* 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.

*Interviewer: Okay Jenkins looks like a really cool tool, but what are the requirements for using Jenkins?*

**Q3.** **What are the pre-requisites for using Jenkins?**

Answer to this is pretty straightforward To use Jenkins you require:

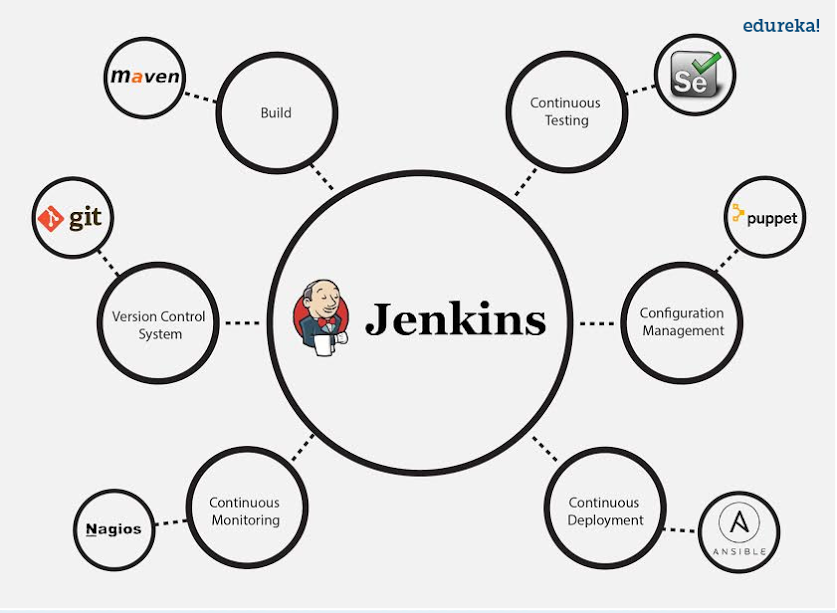
* A source code repository which is accessible, for instance, a Git repository.
* A working build script, e.g., a Maven script, checked into the repository.

*Remember, you have mentioned Plugins in your previous answer, so next question in this Jenkins interview questions blog will be regarding Plugins.*

**Q4.** **Mention some of the useful plugins in Jenkins?**

Below I have mentioned some important Plugins:

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



These Plugins I feel are the most useful plugins, if you want to include any other Plugin that is not mentioned above, you can add that as well, but make sure you first mention the above stated plugins and then add your own.

**Q15.** **Which SCM tools Jenkins supports?**

Below are Source code management tools supported by Jenkins:

* AccuRev
* CVS,
* Subversion,
* Git,
* Mercurial,
* Perforce,
* Clearcase
* RTC

*Now, the next set of Jenkins interview questions will test your experience with Jenkins.*

**Q4.** **Mention what are the commands you can use to start Jenkins manually?**

For this answer I will suggest you to go with the below mentioned flow:

To start Jenkins manually open Console/Command line, then go to your Jenkins installation directory. Over there you can use the below commands:

To start Jenkins: **jenkins.exe start**  
To stop Jenkins: **jenkins.exe stop**  
To restart Jenkins: **jenkins.exe restart**

**Q6.** **Explain how you can set up Jenkins job?**

My approach to this answer will be to first mention how to create Jenkins job.

Go to Jenkins top page, select “New Job”, then choose “Build a free-style software project”.

Now you can tell the elements of this freestyle job:

* Optional SCM, such as CVS or Subversion where your source code resides.
* Optional triggers to control when Jenkins will perform builds.
* Some sort of build script that performs the build (ant, maven, shell script, batch file, etc.) where the real work happens.
* Optional steps to collect information out of the build, such as archiving the artifacts and/or recording javadoc and test results.
* Optional steps to notify other people/systems with the build result, such as sending e-mails, IMs, updating issue tracker, etc..

**Q7. Explain how to create a backup and copy files in Jenkins?**

Answer to this question is really direct.

To create a backup all you need to do is to periodically back up your JENKINS\_HOME directory. This contains all of your build jobs configurations, your slave node configurations, and your build history. 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.

[**Learn Jenkins With DevOps Now**](https://www.edureka.co/devops?utm_source=blog&utm_medium=button&utm_campaign=jenkins-interview-questions-learn-jenkins-with-devops-now)

**Q8. How will you secure Jenkins?**

The way I secure Jenkins is mentioned below, if you have any other way to do it than mention that:

* Ensure global security is on.
* Ensure that Jenkins is integrated with my company’s user directory with appropriate plugin.
* Ensure that matrix/Project matrix is enabled to fine tune access.
* Automate the process of setting rights/privileges in Jenkins with custom version controlled script.
* Limit physical access to Jenkins data/folders.
* Periodically run security audits on same.

*I hope you have enjoyed the above set of Jenkins interview questions, the next set of questions will be more challenging, so be prepared.*

**Q9** **Explain how you can deploy a custom build of a core plugin?**

Below are the steps to deploy a custom build of a core plugin:

* 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.

**Q10. What is the relation between Hudson and Jenkins?**

You can just say Hudson was the earlier name and version of current Jenkins. After some issue, the project name was changed from Hudson to Jenkins.

**Q11. What you do when you see a broken build for your project in Jenkins?**

There can be multiple answers to this question I will approach this task in the following way:

I will open the console output for the broken build and try to see if any file changes were missed. If I am unable to find the issue that way, then I will clean and update my local workspace to replicate the problem on my local and try to solve it.

If you do it in a different way then just mention that in your answer.

**Q12.** **Explain how you can move or copy Jenkins from one server to another?**

I will approach this task by copying the jobs directory from the old server to the new one. There are multiple ways to do that, I have mentioned it below:

You can:

* Move a job from one installation of Jenkins to another by simply copying the corresponding job directory.
* Make a copy of an existing job by making a clone of a job directory by a different name.
* Rename an existing job by renaming a directory. Note that if you change a job name you will need to change any other job that tries to call the renamed job.

**Q13.** **What are the various ways in which build can be scheduled in Jenkins ?**

You can schedule a build in Jenkins in the following ways:

* By source code management commits
* After completion of other builds
* Can be scheduled to run at specified time ( crons )
* Manual Build Requests

**Q14.** **What is the difference between Maven, Ant and Jenkins?**

Maven and Ant are Build Technologies whereas Jenkins is a continuous integration tool.

Q16. What are the two components Jenkins is mainly integrated with?

According to me Jenkins is mainly integrated with the following:

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

If you have anything else in your mind then mention that as well but make sure you include the above two components in your answer.

Github

**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.

**46)   Name a few Git repository hosting services**

* Pikacode
* Visual Studio Online
* GitHub
* GitEnterprise
* SourceForge.net

CA TDM(Test Data Manager)

CA Test Data Manager offers an automated solution to one of the most time-consuming and resource-intensive problems in Continuous Delivery—the creating, maintaining and provisioning of the test data needed to rigorously test evolving applications. CA Test Data Manager uniquely combines elements of data subsetting, masking and synthetic, on-demand data generation to enable testing teams to meet the agile needs of the organization.

**Selenium**

I have used selenium for automated browser testing. I have created required browser environment and tested all the elements in the browser using selenium web drivers.

**Q #1) What is Automation Testing?**

Automation testing or Test Automation is a process of automating the manual process to test the application/system under test. Automation testing involves use to a separate testing tool which lets you create test scripts which can be executed repeatedly and doesn’t require any manual intervention.

**Q #2) What are the benefits of Automation Testing?**

Benefits of Automation testing are:

1. Supports execution of repeated test cases
2. Aids in testing a large test matrix
3. Enables parallel execution
4. Encourages unattended execution
5. Improves accuracy thereby reducing human generated errors
6. Saves time and money

**Q #3) Why should Selenium be selected as a test tool?**

Selenium

1. is free and open source
2. have a large user base and helping communities
3. have cross Browser compatibility (Firefox, chrome, Internet Explorer, Safari etc.)
4. have great platform compatibility (Windows, Mac OS, Linux etc.)
5. supports multiple programming languages (Java, C#, Ruby, Python, Pearl etc.)
6. has fresh and regular repository developments
7. supports distributed testing

**Q #4) What is Selenium? What are the different Selenium components?**

Selenium is one of the most popular automated testing suites. Selenium is designed in a way to support and encourage automation testing of functional aspects of web based applications and a wide range of browsers and platforms. Due to its existence in the open source community, it has become one of the most accepted tools amongst the testing professionals.

Selenium is not just a single tool or a utility, rather a package of several testing tools and for the same reason it is referred to as a Suite. Each of these tools is designed to cater different testing and test environment requirements.

The suite package constitutes of the following sets of tools:

* [**Selenium Integrated Development Environment (IDE)**](http://www.softwaretestinghelp.com/selenium-ide-download-and-installation-selenium-tutorial-2/) – Selenium IDE is a record and playback tool. It is distributed as a Firefox Plugin.
* **Selenium Remote Control (RC)** – Selenium RC is a server that allows user to create test scripts in a desired programming language. It also allows executing test scripts within the large spectrum of browsers.
* [**Selenium WebDriver**](http://www.softwaretestinghelp.com/selenium-webdriver-selenium-tutorial-8/) – WebDriver is a different tool altogether that has various advantages over Selenium RC. WebDriver directly communicates with the web browser and uses its native compatibility to automate.
* [**Selenium Grid**](http://www.softwaretestinghelp.com/selenium-grid-selenium-tutorial-29/) – Selenium Grid is used to distribute your test execution on multiple platforms and environments concurrently.

**Q #5) What are the testing types that can be supported by Selenium?**

Selenium supports the following types of testing:

1. Functional Testing
2. Regression Testing

**Q #6) What are the limitations of Selenium?**

Following are the limitations of Selenium:

* Selenium supports testing of only web based applications
* Mobile applications cannot be tested using Selenium
* Captcha and Bar code readers cannot be tested using Selenium
* Reports can only be generated using third party tools like TestNG or Junit.
* As Selenium is a free tool, thus there is no ready vendor support though the user can find numerous helping communities.
* User is expected to possess prior programming language knowledge.

# Top 20 Automation Testing Interview Questions and Answers

**Here are the most frequently asked Software Automation testing interview questions with answers.**

Test automation plays a very important role in the entire life cycle. Most of the time when want to prepare for an automation testers interview, we focus only on the tool specific question.

We should also consider the fact that learning and knowing the tool is just a mean and it’s not the ultimate goal.

So whenever we are **preparing for an automation tester interview**, we have to consider “Automation” as a whole and focus around the framework and the steps involved.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2015/10/20-Interview-Questions-on-Automation-testing.jpg)

Here I have tried to cite down the questions very much specific to the automation as a whole and not specific to any “tool”.

### Top 20 Automation testing interview questions

**Q #1) What is Automation?**

Automation is any action which can reduce human efforts.

**Q #2) What all things can you automate?**

The right candidates for automation are:

* Regression test suite
* Smoke / Sanity test suite
* Build deployment
* Test data creation
* Automating behind the GUI like testing of APIs and methods

**Q #3) How do you identify the test cases which are suitable for automation?**

[Identify the appropriate test cases for automation](http://www.softwaretestinghelp.com/manual-to-automation-testing-process-challenges/) is the most important step towards automation.

**Q #4) Can you achieve 100% automation?**

100% automation would be difficult to achieve because there would be many edge test cases and some cases which are executed seldom. Automating these cases which are not executed that often will not add value to the automated suite.

**Q #5) Currently I do not have any automation in place in my project, now I want to implement automation, what would be my steps?**

* First identify which type of testing / test cases you want to automate
* Identify the tool
* Design the framework
* Create the utility files and environment files
* Start scripting
* Identify and work on the reporting
* Allocating time for enhancing and maintaining the scripts.

**Q #6) How do you decide which tool you have to use?**

Concluding [which tool is best suitable](http://www.softwaretestinghelp.com/choosing-automation-tool-for-your-organization/) for the project requires a lots of brainstorming and discussions.

**Q #7) Once you identify the tool what would be your next steps?**

Once we finalize the tool, our next step would be to design the framework.

**Q #8) What is a framework?**

A framework is a set of structure of the entire automation suit. It is also a guideline, if followed can result in a structure which is easy to maintain and enhance. These guidelines include:

* Coding standards
* Handling the test data
* Maintaining and handling the elements (object repository in QTP)
* Handling of environment files and properties file
* Reporting of data
* Handling logs

**Q #9) What are the attributes of a good framework?**

The characteristics are:

* Modular – The framework should be adaptable to change. Testers should be able to modify the scripts as per the environment or login information change
* Reusable – The commonly used methods or utilities should be written in a common file which is accessible to all the scripts.
* Consistent – The suite should be written in a consistent format by following all the accepted coding practices.
* Independent – The scripts should be written in such a way that they are independent of each other. In case one test fails, it should not holdback remaining test cases (unless it is a login page)
* Logger – It is good to have implemented the logging feature in the framework. This would help in case our scripts run for longer hours (say nightly mode), if the script fails at any point of time, having the log file will help us to detect the location and the type of error.
* Reporting – It is good to have reporting feature automatically embedded into the framework. Once the scripting is done, we can have the results and reports sent via an email.
* Integration – Automation framework should be such that it is easy to integrate it with other application like continuous integration or triggering the automated script as soon as the build is deployed.

**Q #10) Can you do without a framework?**

Frameworks are guidelines and not mandatory rules, so we can do without a framework, but if we create it and follow it, enhancing and maintaining would be easy to implement.

**Q #11) What are the different types of automation tool you are aware of?**

Open source tool   like Selenium, JMeter

Paid tools like QTP, Load Runner, Ranorex, RFT, and Rational Robot.

**Q #12) What generally is the structure of a framework?**

Normally the structure should have – (It would differ from project to project)

* A “src” (source) folder having the actual test scripts
* A”lib” (library) folder having all the libraries and common methods
* A “class” folder having all the class file (in case using java)
* A “log” folder having the log file(s)
* A file / folder having all the web element Ids
* A file containing the URL, environment and login information.

**Q #13) Where you maintain information like URL, login, password?**

This information should always be maintained in a separate file.

**Q #14) Why do you want to keep this kind of information in separate file and not directly in code?**

URL, Login and Password are the kind of fields which are used very often and these changes as per the environment and authorization. In case we hardcode it into our code, we have to change it in every file which has its reference. In case there are say more than 100 files, then it becomes very difficult to change in all the 100 files and hence can lead to errors. So this kind of information is maintained in a separate file so that updating becomes easy.

**Q #15) What are the different types of frameworks?**

Different types of [framework](http://www.softwaretestinghelp.com/test-automation-frameworks-selenium-tutorial-20/)available are:

* Keyword driven framework
* Data Driven framework
* Hybrid Framework
* Linear Scripting

**Q #16) Can you tell some good coding practices while automation?**

Some of the good coding practices include:

* Add appropriate comments
* Identify the reusable methods and write it in separate file
* Follow the language specific coding conventions
* Maintain the test data in a separate file
* Run your scripts regularly

**Q #17) Any kind of test which you thing should not be automated?**

* Test which are seldom executed
* Exploratory testing
* Usability testing
* Test which are executed fairly quickly when done manually

**Q #18) Do you think that testing can be done only at the UI level?**

Today as we are moving to Agile mode, testing is not limited to the UI layer. Early feedback is imperial for any agile project. If we concentrate only on the UI layer, we are actually waiting until the UI is developed and available to test. Rather we can test even before the UI is actually developed. We can directly test the APIs or the methods using tools like Cucumber and [Fitnesse](http://www.softwaretestinghelp.com/getting-started-with-fitnesse-a-collaboration-tool-for-testers-and-developers/).

In this way we are giving the feedback much early and even are testing before the UI is developed. Following this approach will help us to test only the GUI aspect like small cosmetic changes or some validations on the UI and will help the developers by giving more time to fix the bugs.

**Q #19) How do you select which automation tool is best suited for you?**

Selecting the automation tool depends upon various factors like:

* Scope of the application which we want to automate
* Management overhead like cost and budget
* Time to learn and implement the tool
* Type of support available for the tool.
* Limitation of the tool

**Q #20) What do you think holds testers back to do automation? Is there a way to overcome it?**

The major hurdle for testers is to learn programming / coding when they want to automate. Since testers do not code, adapting to coding is a bit challenging for testers. We can overcome it by:

* Collaborating with developers when automating
* Considering that automation is the responsibility of the whole team and not only of the testers
* Giving a dedicated time and focus on automation.
* Getting proper management support.

You can save these automation testing interview questions as pdf and print for further reading.

s