nExplain the purpose and benefits of using Postman for API testing?

Postman is a powerful tool for API testing that offers numerous benefits. Its purpose is to simplify and streamline the process of testing APIs. Some benefits of using Postman for API testing include:

* Easy API Exploration: Postman provides a user-friendly interface for creating and sending requests to APIs, allowing you to explore and understand their functionalities.
* Efficient Testing: With Postman, you can quickly test various scenarios by sending different types of requests (GET, POST, PUT, DELETE) and examining the responses.
* Request and Response Validation: Postman allows you to validate API responses, ensuring they meet the expected criteria and identifying any errors or inconsistencies.
* Environment Management: Postman lets you create and manage environments, making it convenient to switch between different setups, such as development, staging, and production.
* Collaboration and Sharing: Postman facilitates teamwork by enabling the sharing of collections, requests, and test suites, fostering collaboration among developers, testers, and other stakeholders.

2. How do you create and send a request in Postman?

To create and send a request in Postman, follow these steps:

* Open Postman and create a new request by selecting the appropriate HTTP method (e.g., GET, POST) from the dropdown menu.
* Enter the request URL in the address bar.
* Add any required headers, query parameters, or request body data, depending on the API endpoint's requirements.
* Click the "Send" button to send the request to the API server.

Postman will display the response, including the status code, headers, and body, allowing you to analyze the results of your request.

3. What are some key features of Postman that you find most useful in your testing workflow?

 Some key features of Postman that are often found useful in testing workflows include:

* Collections: It allows you to organize and group related requests, making managing and executing test suites easier.
* Variables and Environments: Postman allows you to define variables and environments, which help parameterize requests and manage different testing environments.
* Test Scripts: Postman's test scripts, written in JavaScript, enable you to automate assertions, extract data from responses, and perform complex validations.
* Pre-request Scripts: These scripts allow you to perform actions before sending a request, such as setting up dynamic data or configuring authentication.
* Request History: Postman keeps a history of previously sent requests, allowing you to revisit and retest them without re-entering the details.
* Collection Runner: The Collection Runner feature allows you to execute multiple requests in a collection as a test suite, enabling automation and bulk testing.

4. How do you handle authentication and authorization in Postman?

Postman provides multiple ways to handle authentication and authorization in API requests. Some commonly used methods include:

* Basic Authentication: You can include the username and password in the request headers using the "Authorization" header.
* Token-based Authentication: Postman allows you to include tokens (such as JWT or OAuth) in the request headers or as query parameters.
* API Key: If an API requires an API key, you can pass it as a request header or query parameter.
* OAuth 2.0: Postman has built-in OAuth 2.0 support, allowing you to configure and authenticate using various OAuth flows, such as Authorization Code or Client Credentials.
* Custom Authentication: Postman's scripting capabilities enable you to implement custom authentication mechanisms by modifying request headers or using specialized libraries.

5. Can you describe the process of creating and running automated tests in Postman?

To create and run automated tests in Postman, follow these steps:

* Create a new request in Postman or open an existing one.
* Write test scripts using JavaScript to validate the API response, assert specific conditions, or extract data.
* Save the request in a collection.
* Go to the "Collections" tab and click on the three-dot menu next to the collection name.
* Select "Edit" to open the collection, then click on the "Tests" tab.
* Write and organize test scripts for the entire collection or individual requests within the collection.
* Save the collection.
* To run the automated tests, go back to the "Collections" tab and click on the arrow icon next to the collection name.
* Choose the desired test environment, if applicable.
* Click on "Run" to execute the tests in the collection.
* Postman will run the requests and display the test results, highlighting any failures or errors encountered during the execution.

Postman Interview Questions for Freshers

6. What is Postman?

Postman is a collaboration platform for API development. It is a recognized API client that enables you to organize the creation, division, testing, and documentation of APIs. We can send HTTP/s requests to a repair and receive their responses using the Postman tool. It will enable us to approve the service's uptime and functionality.

7. What is a collection in Postman?

In Postman, a collection allows you to group related requests. You can also use it to organize the requests into folders systematically.

8. Why do we use Postman?

Following are some of the most important reasons for using Postman:

* It is software that aids with API testing and is available for free use
* It aids in managing the complete API lifecycle
* It provides Runtime Service to assist with managing API collections, workspaces, environments, and many examples
* Additionally, Postman may get integrated with [CI/CD](https://www.simplilearn.com/best-ci-cd-tools-article) programs like Circle CI, [Jenkins](https://www.simplilearn.com/tutorials/jenkins-tutorial/what-is-jenkins), etc
* It offers a sizable community forum where you can quickly find solutions to any technical problems you run across while using the tool

9. What is an API?

Application Programming Interface is referred to as API. Technically speaking, an API is a collection of steps, functions, and other points of access that a program, an operating system, a library, etc. makes available to[programmers](https://www.simplilearn.com/how-to-become-programmer-article) so they can use it to communicate with other software programs.

10. Name some tools used for API Testing?

The following list includes some of the tools used for API testing:

* Postman 2. Tricentis Tosca 3. Tricentis Tosca, Katalon Studio , Apigee , Jmeter

11. What API information is exposed in Web Developer tools?

API information for request headers, response body, and response cookies is exposed in web developer tools.

12. What can we use to get API information from web developer tools into Postman?

Copy as cURL allows Postman to receive API data from web developer tools.

13. Where are query parameters stored in a GET request?

The query parameters for the GET request are saved in Postman's URL.

14. What is meant by the term environment in Postman?

An environment in Postman is a group of key-value pairs. Postman allows you to create various environments that you can switch instantly with a button. There are two different sorts of environments: global and local. They define the variable's scope so you can use it in requests.

15. What are the core components of an HTTP request?

An HTTP request is made up of five major components:

* HTTP methods: A collection of request techniques used to carry out particular actions on resources (GET, PUT, POST, DELETE)
* URI (Uniform Resource Identifier): locates a resource.
* Version of HTTP (example- HTTP v1.1)
* Content-Type: application/JSON, Content-Length: 511) Request Headers
* Payload: The message content is contained in the request body.



16. State the Core Components of an HTTP Response?

Four essential components are present in each HTTP response:

* According to the HTTP request's resource, the server's status is indicated by the Status/Response Code. Examples include 404 (resource not found) and 200 (response ok).
* HTTP Version - Indicates the version of HTTP being used, such as HTTP v1.1.
* The response header is where key-value pairs of metadata for the HTTP response message are stored—for instance, content length, content type, response date, and server type.
* The response body indicates response message content or resource representation.

**17. What is GUID?**

Global Unique Identifier is referred to as GUID. It consists of hexadecimal digits separated by hyphens. This Postman identifier GUID fulfills the goal of uniqueness.

**18. What is the HTTP response code for a POST request with incorrect parameters**?

The correct response code for a request with incorrect parameters is 400 Bad Request.

**19. Can local variables be imported in Postman Monitors?**

Yes, you can import local variables into Postman Monitors. However,[global variables](https://www.simplilearn.com/tutorials/python-tutorial/global-variable-in-python) cannot get imported into Postman Monitors.

**20. How can you iterate a request 100 times in Postman?**

With the help of Collection Runner, Postman allows us to iterate a request 100 times.

**21. Which programming language is used for Postman tests?**

Postman tests are run using [JavaScript](https://www.simplilearn.com/best-javascript-books-to-read-article).

**22. What is a Postman Collection?**

We can group several needs using a Postman Collection. It merely enables us to organize the requests into folders.

23**. What do you understand by the term Postman Collection runners**?

Data-driven testing is carried out by using a Postman collection runner. API requests are collected for numerous iterations with various data sets.

24**. What is Postman cloud’s purpose if we work in a company? Why?**

A Postman cloud is a shared repository where businesses can access Postman collections. Work can be immediately saved in the Postman cloud after logging in. Data and collections are accessible to team members from any location.

**25. Why is it not preferred to save work in Postman cloud?**

Since company work cannot be disclosed and must be kept confidential, it is not advised or desired to save your work in the Postman cloud. Since Postman requires sign-in, saving your work there could lead to a security risk.



**26. How will you log variable values in Postman?**

You may use the following command in Postman to log the variable values on the console:

console.log(pm.variables.get("variable\_name"));

**27. How do you access postman variables?**

You may access a Postman variable using the variable name {{var}}.

**28. What are the various authorization methods provided by Postman?**

Postman provides the following API request authorization options:

* API Key
* Basic auth
* Digest auth
* Hawk Authentication
* Oauth 1.0
* Oauth 2.0
* Bearer Token
* NTLM Authentication
* AWS Signature

**29. What are the different types of API requests supported in Postman?**

The following is a list of the various API requests that Postman supports:

* GET
* POST
* PUT
* PATCH
* COPY
* DELETE
* HEAD
* OPTIONS
* LINK
* UNLINK
* PURGE
* LOCK
* UNLOCK
* PROPFIND
* **VIEW**

**30. How are Query Params different from Path Variables?**

Query parameters are used to sort or filter the resources, and Path Variables gets used to identify specific resources.

**31. What is Basic Auth in Postman?**

Basic Auth is a method of authorization offered by Postman for HTTP user agents, such as web browsers, to enter login and password. It becomes connected with the request after the username and password are entered.

**32. What is digest auth in Postman?**

One of Postman's authorization techniques is digest auth or digest authorization. Through this method, clients can send requests to the API first and then receive responses from the server, such as 401 illegal responses and numbers that can only be used once as absolute values.

**33. What encoding is accepted by Postman in authorization credentials?**

Postman only takes authorization credentials encoded in Base64. Postman already has this feature, or you can use a third-party service to convert the credentials to base64.

**34. Can we have the same names for global variables in Postman?**

Global variables cannot have identical names since doing so would confuse the software because they are devoid of any environment. The terms of local variables might be the same across various environments.

**35. What do you know about the postman monitor?**

For running collections, the postman monitor is employed. Collections are kept running for the duration of the user-defined time. The user must get logged in to utilize Postman Monitor. Users regularly send emails to one another to share the monitor reports.

**36. What is a binary form in POST methods?**

Postman's binary form is made to send data in a format that you cannot manually type. You use these options, which cannot be manually typed, such as an image, a file, etc., because everything in a computer is transformed into binary.

**37. What are the limitations of Postman?**

The following is a list of Postman's main drawbacks:

* Postman cannot process 1000+ API requests.
* It might be challenging to manage collections and requests for large projects.
* For managing the workspace as code, Postman is not appropriate. It is because dynamic API requests would result in a lot of code duplication.

**38. How can you save the responses of API to a file in Postman?**

There are two ways to save an API response to a file in Postman:

* First, in the response section, click the download button.
* Then, press the arrow next to the send button. An option to send and download is available here. After clicking it, Postman will prompt you for the location to save the response after the request gets successfully executed.

**39. What is the significance of 301 status code?**

When a page has been permanently redirected from one website page to another, Postman's 301 status code gets used to indicate this. The search engine is informed that the old page is out-of-date and has to index the URL of the new page.

**40. Define status code 201.**

When you successfully create a resource using a POST or PUT request, the status code 201 denotes that the resource has been created. It uses the location header to return a link to a newly built resource.

**41. When do we use global variables, collection variables, and local variables?**

Global variables are all-purpose variables perfect for prototyping and quick results. They get utilized when transferring data to other requests.

You can most frequently use collection variables to store constants that don't change while the Collection is being executed. They are used for constants that remain the same throughout execution and for URLs and authentication credentials when just one environment is present.

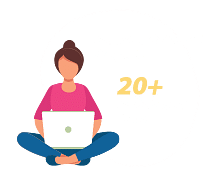
Local variables are only accessible for the duration of the request that set them or while using the Newman/Collection runner. They get used whenever you want to override all other variable scopes.

**42. How do you remove local variables?**

Once you run and finish the tests, the local variables are automatically eliminated.

**43. What is the difference between form data and x-www-form-urlencoded?**

The primary distinction between form data and x-www-form-urlencoded is that the URL is pre-arranged when sent via x-www-form-



**44. How do you access the history of requests in Postman?**

The History tab in the Postman application allows us to access the history of requests. The record will be synced amongst all your logged-in devices when we login into the Postman account. The view will load the request we saved while working on it earlier after clicking on any of the requests shown in the History tab.

Postman Interview Questions for Experienced

**45. Why is Base64 encoding primarily used in Postman?**

As a result of the data being transmitted in textual form and being sent in a more accessible format, like HTML form data, the Base64 authorization credentials are typically employed. Base64 is so popular because we can rely on the same 64 characters regardless of the encoding language we use.

**46. What is the purpose of the 304 status code?**

The meaning of the status code 304 is "Not Modified." It is utilized in conditional GET requests to reduce network bandwidth usage. The response's body must be empty. Dates, locations, and other information should be in the headers.

**47. Is it preferable to save our work on Postman Cloud?**

Saving your work in the Postman cloud is not recommended because business work cannot be disclosed and must be kept private. Since Postman cloud requires sign-in, security breaches could happen if it gets used. Therefore, using Postman Cloud to save work is discouraged, and team workspace is strongly recommended.

**48. What are the various variable scopes provided by Postman?**

Postman provides the following variable scopes:

* Global Variables
* Local Variables
* Environment Variables
* Collection Variables
* Data Variables

**49. Is it possible to reuse the authentication token for multiple requests?**

You can indeed use the authentication token more than once. To do this, create a collection, add all requests with the same authentication token, and then assign the Collection with the auth token to the Collection. By choosing "Inherit auth from parent" under the Authorization tab, we may apply it to each request separately.

**50. How do you write test cases for basic authentication in Postman?**

One of the authentication methods offered by Postman, Basic Authentication, ensures we can specify the username and password along with the API calls. We can achieve this by first configuring the API's credentials by:

* Going to the Authorization tab
* From the drop-down option, choose Basic Auth
* Enter the API's username and password in the corresponding fields

You can write the test cases like this:

pm.test("Is the Request Authenticated?", function () {

       var jsonData = pm.response.json();

       //if authenticated then assert to true

pm.expect(jsonData.authenticated).to.eql(true);

  });

pm.test("Is the Content-Type present?", function () {

pm.response.to.have.header("Content-Type");

  });

pm.test("Is it a successful POST Request?", function () {

pm.response.to.have.status(200);

  });

**51. How do you set the same headers for all requests in a Postman Collection?**

Pre-request scripts are supported at the collection level and for individual requests in Postman. Any script that applies to every request in the Collection may get included in the pre-request scripts. The steps are as follows:

To access the pre-request tab, right-click the Collection.

Add the script's lines of code below to add a request header for each request included in the Collection.

pm.request.headers.add({

    key: 'TestHeader',

    value: 'testValue'

});

To save the script, click Update.

Run the request in the Collection and check the Postman console to ensure that the headers have been added.

**52. What are workspaces in Postman? What are their uses?**

Postman workplaces are just areas or spaces where one or more people can collaborate on the same Collection or set of collections. It allows the collections or requests to get logically separated from one another. Postman supports two different types of workspaces: Team and Personal.

**53. Does Postman provide a feature to log requests and responses?**

The Postman software does allow users to view requests and response parameters. But after applying the pre-request scripts, it is essential to see how you sent the request. To inspect every request and response detail in such circumstances, Postman includes an additional tool called "Postman Console." By having console.log statements in the scripts, you can additionally log the information in the console.

**54. How will you stop the execution of upcoming requests or the execution of the collections?**

You can use the code below to stop processing the next request: pm.setNextRequest (null);

**55. What do you understand by the pre-request script?**

Pre-request scripts are those used to run Javascript code before a request is made. Pre-processing tasks, including setting variables, parameters, headers, body data, etc., are performed using it.

56**. How can we use Custom Javascript libraries in our scripts with an example?**

We may utilize the many built-in tools and libraries that Postman offers to include our pre- or post-request scripts or test cases. Consider the use of the moment.js library. It offers a variety of helpful methods for formatting data around time. Consider a POST request that must provide the generated date for the user, who anticipates the format "DD/MM/YYYY." You can use the moment library with just one line of code. To obtain the data with the proper formatting and then store it in an environment variable, we must add the following lines of code to our pre-request script:

var moment = require('moment');

pm.environment.**set('createdDate',moment().format('DD/MM/YYYY'));**

**57. If we have a global and a local variable of the same name, which one will be given the most preference in Postman?**

In certain circumstances, the global variable's value is overwritten to give the local variable higher precedence.

**58. Does Postman allow flexibility to make use of the command-line?**

Any Postman collection can get executed using the Newman command-line tool that Postman offers. This [NodeJS](https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-nodejs" \o "NodeJS" \t "_blank)-based package uses the Newman Collection Runner to execute collections and therefore needs a node environment. It supports running assertions, pre-request scripts, or other request scripts linked with the requests that are a part of the Collection, just like Postman's Collection Runner does.

**59. How will you generate random numbers of a given range in Postman?**

Assuming you want to create numbers between the ranges of 1 to N, the pre-request script can be written as follows:

pm.globals.set('randomNumber', Math.floor(Math.random() \* N));

This variable can then be used in the URL as follows:{{randomNumber}}

**60. What do you understand about ScratchPad?**

Postman allows us to work without a connection to its servers by giving us access to Scratch Pad. It offers the freedom of using some of Postman's functionalities offline. The features include sending requests, creating requests, and adding requests to collections.

**61. How do you get the cURL command based on the details of the REST API obtained from Postman?**

You can use the steps listed below to get the cURL command equivalent:

As indicated below, click the Code icoIn response to the REST API request, you will receive the cURL comman

By choosing the necessary choice from the drop-down, as shown in the image below, we can also obtain the command for the request in several languages, such as C#, Javascript, NodeJs, PHP, etc.

**1) What is Postman?**

The Postman is a highly popular API testing tool that helps the development team create, share, and test an API. The Postman tool provides a GUI interface to the API and a command-line utility for experienced testers.

**2) What is an HTTP request?**

An HTTP request is a program that the client makes to a name host located on a server. It works as a communication interface or a request-response protocol between a client and server. The primary use of the HTTP request is to access a resource on the server. To initiate the HTTP request, the client uses components of a URL (Uniform Resource Locator) that also includes the information needed to access the resource.

Postman Interview Questions

**3) State The Core Components of an HTTP Response?**

In Postman, every HTTP response contains four key elements.

* **Response/Status Code-**There are response code issues by a sever for client’s request, as 404 means Page Not Found.
* **HTTP Version-** HTTP version name. For example, HTTP v2.2
* **Response Header-**It included information for the HTTP response message. For example. The content length, date, status, server type, etc.
* **Response Body –**It contains the data which a client requested from the server.

**4) Why do we use Postman?**

Here are some most prominent reasons for using Postman:

* It is free to use software which is helpful for API testing
* It helps you to manage the end-to-end lifecycle of API
* It offers Runtime Service that helps manage API collections, workspaces, environments, and different examples.
* You can also integrate Postman with [CI/CD tools](https://www.guru99.com/top-20-continuous-integration-tools.html) like Circle CI, Jenkins, etc.
* It has a vast community forum that can easily address any technical issues you face while using the tool.

**5) What is API?**

Application Programming Interface (API) is a software interface that enables two applications to interact with each other without any user intervention. It is a collection of software functions and procedures. API is defined as a code that helps two different software communicate and exchange data with each other.

9M

What to bring to a JOB Inte**at are the important authorization methods supported by Postman?**

[Postman](https://www.guru99.com/postman-tutorial.html) offers the following API request authorization options 1) API Key, 2) Oauth 1.0 and 2.0, 3) Basic auth, 4) Digest auth, 5) Bearer Token, 6) AWS Signature, 7) Hawk Authentication, 8) NTLM Authentication.

**7) How do you log variable values in Postman?**

Postman allows to log the variable values in Postman in the console by using the following command:

console.log(pm.variables.get("guru99"));

**8) Postman is available as a native desktop app for?**

[Postman API testing tool](https://www.guru99.com/top-6-api-testing-tool.html) is currently available for Mac, Windows (32-bit / 64-bit), and Linux (32-bit / 64-bit)

**9) What is the History tab in Postman?**

All the request you send in Postman appears under the History tab of the sidebar. It is very much similar to browser history, which you can clear whenever you want.

**10) What is Basic Auth in Postman?**

In Postman, Basic Auth is an authorization method provided for HTTP user agents like web browsers to enter username and password. After entering the username and password that you can associate with the request.

**11) What is a binary form in POST methods?**

Post binary firm is designed to send the information in a format that is impossible to enter manually. These options are used while sending larger files like CSV files, etc.

**12) What is the main difference between Authorization and authentication?**

Here are a few differences between authorization and Authentication:

* Authorization is the act of allowing or permitting someone, whereas authentication is proving that something is genuine.
* Authentication always comes first, while authorization comes after authentication.
* Authorization is open to anyone with permission, whereas authentication requires you to have a password.

**13) What is the Payload in Postman?**

The Payload of an API Module is the body of your request and response message. When making an API request, it contains the data you send to the server. You can send and receive Payload in various formats, for example, JSON or XML.

**14) What is a collection in Postman?**

A collection in Postman enables you to group similar requests. It also allows you to systematically arrange the requests into folders.

**15) What is a Pre-Request script?**

Pre-request scripts help you to execute JavaScript before a request runs. It allows you to accomplish pre-processing tasks like setting variable values, parameters, headers, and body data.

**16) How the Postman variables are accessed?**

The Postman variables are always accessed by using the variable name:

{{variable name}}

**17) What is the meaning of the term environment in Postman?**

The environment in Postman is a set of key-value pairs. Postman allows you to build multiple environments and switch among them with a click of a button.

**18) Why does Postman never accepts any other encoding apart from Base64?**

You can use base64 as it helps us transmit the data into the textual form and send it as HTML form data. Moreover, we must rely upon the same 64 characters in any encoding language.

**19) Which kind of encoding does Postman accepts for authorization credentials?**

Postman only accepts Base64 encoding, which is provided inbuilt in Postman. Otherwise, it would benefit when you use 3rd party websites that help you to convert the credentials into base64.

**20) What are the different scopes of an environment variable in Postman?**

Scope of a variable in Postman is defined as the boundaries it can access.

Here are important scopes of an environment variable in Postman:

* **Local Scope:** This type of variable can be accessed only in the environment in which it was created
* **Global Scope:** This type of variable can be accessed globally in any environment or no environment.

**21) Is it possible to import local variables in Postman Monitors?**

Postman monitors allow you to import local variables but not global variables.

**22) What is the Postman execution order for a collection?**

For all the requests in a collection, the scripts will execute in the following given order:

**Step 1)**A pre-request script associated with a collection will run before every request.  
**Step 2)**A pre-request script associated with a folder will run before every request in a specific folder.  
**Step 3)**A test script associated with a collection will run after every request.  
**Step 4)**A test script associated with a folder will run after the request in the specific folder.

**23) Can you have two global scope variables with the same name in Postman?**

No, the global scope never has duplicate/same names, while variables having local scope can have the same name in various environments.

**24) How do you access postman variables?**

You can log the variable values in the console by using the command:

console. Log (pm. Variables.

**25) What is the use of the collection in Postman?**

In Postman, a collection is used to group similar requests. It helps you to arrange requests systematically into folders.

**26) How can you use POSTMAN to generate random numbers in a specific range?**

In your Pre-request script define your variable with:

pm.globals.set('randomNum', Math.floor(Math.random() \* 5));

Then in your URL call your variable in the URL like so:  
**Output:**

{{randomNum}}

**27) How do you remove local variables?**

Local variables are

removed once the tests have been executed.

**28) What is ‘Postman Collection runners?**

Postman contains a collection runner that is useful for automating API testing. It helps visualize details of each iteration and test results. A postman collection runner is also used for Data-driven testing.

**29) What do you mean by postman monitors?**

The postman monitor feature is used for running collections until the user’s specified time. However, the user must be logged in to their account to use this feature.

**30) Why saving your work in the Postman cloud is not advisable?**

You should not save your work in Postman as your business details do not remain confidential. Moreover, saving your on-Postman cloud may cause a security breach

**1. What is version control?**

Version control is a set of practices and tools for managing codebases. Developers use version control to keep track of every line of code, and share, review, and synchronize changes among a team.

**2. What is**[**Git**](https://www.atatus.com/blog/git-tutorial-for-beginners/)**?**

Created by Linus Torvalds to support the open-source development of Linux, Git is the most popular version control tool. It uses a distributed repository model that can efficiently handle projects of any size.

**3. What is a Git repository?**

A Git repository keeps track of every file in a software project. The repository serves as an index for all files and changes in the project, allowing developers to navigate to any point in the project’s history.

**4. Which other version control tools do you know of?**

* Mercurial
* Subversion (SVN)
* Concurrent Version Systems (CVS)
* Perforce
* Bazaar
* Bitkeeper
* Fossil

**5. What is a Git branch?**

A Git branch is an independent line of development, usually created for working on a feature. Branches let developers code without affecting the work of other team members.

**6. What is merging?**

Merging consists of joining branches. For example, when developers incorporate their peer-reviewed changes from a feature branch into the main branch.

**7. What is trunk-based development?**

[Trunk-based development](https://trunkbaseddevelopment.com/) is a branching model where most of the work takes place in a single trunk, usually called trunk, master, or main. The trunk receives daily merges from all developers in the team.

Trunk-based development is a popular development model because it simplifies version control. Since the trunk is a single source of truth, this model minimizes the chances of merge conflict.

**8. What is Gitflow, and how does it compare to trunk-based development?**

Gitflow is a workflow for Git that makes heavy use of branches. In Gitflow, all the code is merged into the develop branch instead of the main branch, which serves as an abridged version of the project’s history.

Features are worked on specific “feature branches” (typically prefixed with feature/). In the same fashion, releases also create a dedicated release/ branch.

Compared with trunk-based development, Gitflow is more complex and has a higher chance of inducing merge conflicts, which is why it has fallen out of favor among the development community.

**9. How long should a branch live?**

In the context of continuous integration, branches should follow trunk-based development practices and thus be short-lived. Ideally, a branch should last for a few hours or, at most, a day.

**CI/CD**

Your CI/CD interview questions will, at the minimum, cover some basic concepts such as what is CI and how it works.

**10. What is continuous integration?**

[Continuous Integration](https://semaphoreci.com/continuous-integration) (CI) is a software development methodology where developers — following the trunk-based model — merge their changes to the main branch many times per day.

CI is supported by automated tests and a build server that runs them on every change. As a result, [failures are made visible as soon as they are introduced](https://qameta.io/blog/test-automation-as-a-service/?utm_source=semaphore&utm_medium=blog&utm_campaign=semaphore-cicd-questions&utm_content=2022-06-07) and can be fixed within minutes.

**11. How do CI and version control relate to one another?**

Every change in the code must trigger a continuous integration process. This means that a CI system must be connected with a Git repository to detect when changes are pushed, so tests can be run on the latest revision.

**12. What’s the difference between continuous integration, continuous delivery, and continuous deployment?**

**Continuous integration** (CI) executes the sequence of steps required to build and test the project. CI runs automatically on every change committed to a shared repository, offering developers quick feedback about the project’s state.

[Continuous delivery](https://semaphoreci.com/blog/2017/07/27/what-is-the-difference-between-continuous-integration-continuous-deployment-and-continuous-delivery.html) is an extension of CI. Its goal is to automate every step required to package and release a piece of software. The output of a continuous delivery pipeline takes the form of a deployable binary, package, or container.

**Continuous deployment** is an optional step-up from continuous delivery. It is a process that takes the output from the delivery pipeline and deploys it to the production system in a safe and automated way.

A complete CI/CD workflow

**13. Name some benefits of CI/CD**

* **Less risk**: automated tests reduce the chance of introducing bugs, creating a safety net that increases the developer’s confidence in their code.
* **More frequent releases**: the automation provided by continuous delivery and continuous deployment allows developers to release and deploy software safely many times per day.
* **Improved productivity**: freed from the manual labor of building and testing the code, developers can focus on the creative aspects of coding.
* **Elevated quality**: CI acts as a quality gate, preventing code that is not up to standards from getting released.
* **Better design**: the iterative nature of continuous integration lets developers work in small increments, allowing a higher degree of experimentation, which leads to more innovative ideas.

**14. What are the most important characteristics in a CI/CD platform?**

* **Reliability**: the team depends on the CI server for testing and deployment, so it must be reliable. An unreliable CI/CD platform can block all development work.
* **Speed**: the platform should be fast and scalable to obtain results in a few minutes.
* **Reproducibility**: the same code should always yield the same results.
* **Ease of use**: easy to configure, operate, and troubleshoot.

**15. What is the build stage?**

The [build stage](https://semaphoreci.com/blog/build-stage) is responsible for building the binary, container, or executable program for the project. This stage validates that the application is buildable and provides a testable artifact.

**16. What’s the difference between a hosted and a cloud-based CI/CD platform?**

A hosted CI server must be managed like any other server. It must be first installed, configured, and maintained. Upgrades and patches must be applied to keep the server secure. Finally, failures in the CI server can block development and stop deployments.

On the other hand, a cloud-based CI platform does not need maintenance. There’s nothing to install or configure, so organizations can immediately start using them. The cloud provides all the machine power needed, so scalability is not a problem. Finally, the reliability of the platform is guaranteed by SLA.

**17. How long should a build take?**

Developers should get results from their CI pipeline in [less than 10 minutes](https://semaphoreci.com/blog/2017/03/02/what-is-proper-continuous-integration.html). That’s the longest time that’s practical to wait for results.

**18. Is security important in CI/CD? What mechanisms are there to secure it?**

Yes. CI/CD platforms have access to all kinds of sensitive data such as API keys, private repositories, databases, and server passwords. An improperly secured CI/CD system [is a prime target for attacks](https://research.nccgroup.com/2022/01/13/10-real-world-stories-of-how-weve-compromised-ci-cd-pipelines/) and can be exploited to release compromised software or to get unauthorized access. A CI/CD platform must support mechanisms to securely manage secrets, and control access to logs and private repositories.

**19. Can you name some deployment strategies?**

* **Regular release/deployment**: releases software to everyone at once, making it available to the general public.
* [**Canary releases**](https://semaphoreci.com/blog/what-is-canary-deployment): this is a method that reduces the chance of failure by exposing a small portion of the userbase (around 1%) to the release. With a canary release, developers gradually switch users to the latest release in a controlled way.
* [**Blue-green releases**](https://semaphoreci.com/blog/blue-green-deployment): consists of running two simultaneous instances of an application; one is the stable version currently serving users and the other the latest release. Users are switched from the former to the latter all at once. This method is safer than the regular or big bang releases because users can instantly be routed back to the previous version if there is a problem.
* **Dark launches**: are deployments where new features are released without being announced. Features can be enabled in a very fine-grained way with [feature flags](https://semaphoreci.com/blog/feature-flags).

**Testing**

Testins is inseparable from CI, so your CI/CD interview questions will include this topic.

**20. How does testing fit into CI?**

Testing is integral to and inseparable from CI. The main benefit teams get from CI is continuous feedback. Developers set up tests in the CI to check that their code behaves according to expectations. There would be no feedback loop to determine if the application is in a releasable state without testing.

**21. Should testing always be automated?**

Yes, CI requires that [all tests are automated](https://semaphoreci.com/blog/test-automation). They must work without human intervention.

That is not to say that manual or exploratory testing don’t have their places. They are very useful for discovering potential features and finding further test cases to automate.

**22. Name a few types of tests used in software development**

There are more [types of tests](https://semaphoreci.com/blog/20-types-of-testing-developers-should-know) than we can count with both hands, but the most common ones are:

* **Unit tests**: validate that functions or classes behave as expected.
* **Integration tests**: are used to verify that the different components of an application work well together.
* **End-to-end tests**: check an application by simulating user interaction.
* **Static tests**: finds defects in code without actually executing it.
* **Security tests**: scans the application’s dependencies for known security issues.
* [**Smoke tests**](https://semaphoreci.com/community/tutorials/smoke-testing): fast tests that check if the application can start and that the infrastructure is ready to accept deployments.

**23. How many tests should a project have?**

There is no single answer as it depends on the size and nature of the project. That being said, for various reasons, test suites tend to follow in distribution the [testing pyramid](https://semaphoreci.com/blog/testing-pyramid).

What is Git?

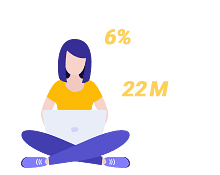
Git is a widely used modern version control system for tracking changes in computer files. The term version control system suggests a system that records all the changes made to a file or set of data, so a specific version can be considered whenever needed. This feature makes the process of collaboration so feasible with all team members, making it considerably more comfortable to work over a big project.

Git makes it possible for several people involved in the project to work together and track each other's progress over time. In software development, the tool helps in Source Code Management. Git favors not only programmers but also non-technical users by keeping track of their project files.

While [working on Git,](https://www.simplilearn.com/tutorials/git-tutorial/git-tutorial-for-beginner) we actively use two repositories.

* Local repository: The local repository is present on our computer and consists of all the files and folders. This Repository is used to make changes locally, review history, and commit when offline.
* Remote repository: The remote repository refers to the server repository that may be present anywhere. This repository is used by all the team members to exchange the changes made.

Both repositories have their own set of commands. There are separate Git Commands that work on different types of repositories.



Git Commands: Working With Local Repositories

* git init
* The command git init is used to create an empty Git repository.
* After the git init command is used, a .git folder is created in the directory with some subdirectories. Once the repository is initialized, the process of creating other files begins.
* git add
* Add command is used after checking the status of the files, to add those files to the staging area.
* Before running the commit command, "git add" is used to add any new or modified files.
* git commit

|  |
| --- |
|  |

* The commit command makes sure that the changes are saved to the local repository.
* The command "git commit –m <message>" allows you to describe everyone and help them understand what has happened.
* git status
* The git status command tells the current state of the repository.

|  |
| --- |
|  |

* The command provides the current working branch. If the files are in the staging area, but not committed, it will be shown by the git status. Also, if there are no changes, it will show the message no changes to commit, working directory clean.
* git config
* The git config command is used initially to configure the user.name and user.email. This specifies what email id and username will be used from a local repository.
* When git config is used with --global flag, it writes the settings to all repositories on the computer.

|  |
| --- |
| git config --global user.nam “any user name”  git config --global user.email <email id> |

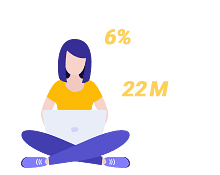
* git branch
* The git branch command is used to determine what branch the local repository is on.
* The command enables adding and deleting a branch.
* Create a new branc  
    git branch <branch\_name>
* git checkout
* The git checkout command is used to switch branches, whenever the work is to be started on a different branch.
* The command works on three separate entities: files, commits, and branches.
* git merge
* The [git merge](https://www.simplilearn.com/tutorials/git-tutorial/merge-conflicts-in-git) command is used to integrate the branches together. The command combines the changes from one branch to another branch.
* It is used to merge the changes in the staging branch to the stable branch.



However, these are popular and basic git commands used by developers.

Git Commands: Working With Remote Repositories

* git remote
* The git remote command is used to create, view, and delete connections to other repositories.
* The connections here are not like direct links into other repositories, but as bookmarks that serve as convenient names to be used as a reference.
* git clone
* The git clone command is used to create a local working copy of an existing remote repository.
* The command downloads the remote repository to the computer. It is equivalent to the Git init command when working with a remote repository.
* git pull
* The [git pull command](https://www.simplilearn.com/tutorials/git-tutorial/git-pull-request) is used to fetch and merge changes from the remote repository to the local repository.
* The command "git pull origin master" copies all the files from the master branch of the remote repository to the local repository.
* git push
* The command [git push](https://www.simplilearn.com/tutorials/git-tutorial/git-push-command) is used to trunsfer the commits or pushing the content from the local repository to the remote repository.
* The command is used after a local repository has been modified, and the modifications are to be shared with the remote team members.

  
zome Advanced Git Commands

* git stash
* The git stash command takes your modified tracked files and saves it on a pile of incomplete changes that you can reapply at any time. To go back to work, you can use the stash pop.
* The git stash command will help a developer switch branches to work on something else without committing to incomplete work.
* git log
* The git log command shows the order of the commit history for a repository.
* The command helps in understanding the state of the current branch by showing the commits that lead to this state.