**1. What is API testing ?**

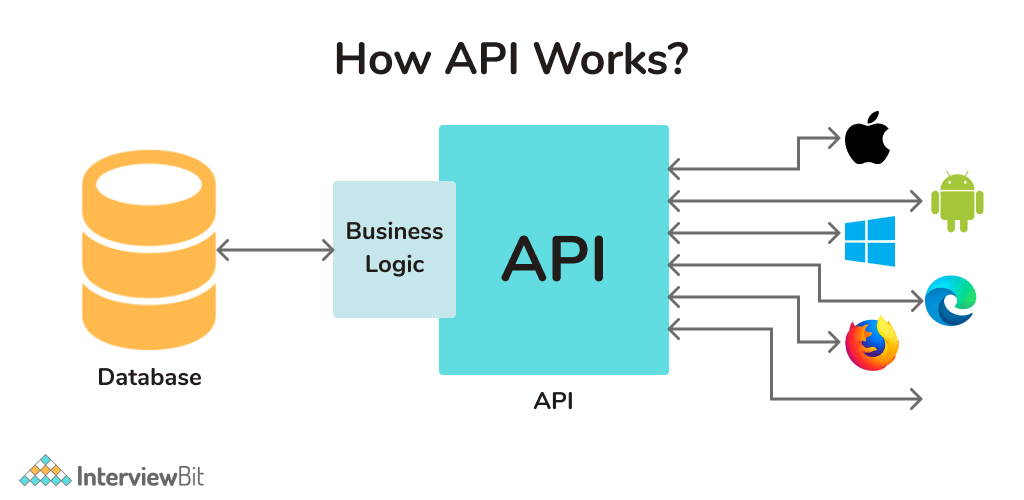
API testing is a category of software integration testing that deals with the testing of Application Programming Interfaces (APIs) directly. It deals with checking if the APIs developed work as expected in terms of reliability, functionality, security, and performance of the business logic covered by the applications.

**2. What is API?**

API stands for Application Programming Interface that is useful for communication between different software systems. It facilitates data exchange between systems located in different remote places. They are nothing but a collection of functions that are executable by other functions of the software application.

**3. How do APIs work?**

The general workflow of API is that it takes a request, processes it which might involve data validation, database interaction, data processing, and then the resultant of this is sent back to the source. APIs provide an abstraction to the internal business logic as they are not exposed to the world.



Examples of APIs: Amazon API, Google Map API, Twitter API, etc.

**You can download a PDF version of Api Testing Interview Questions.**

[**Download PDF**](javascript:void(0))

**4. What are the different types of API testing?**

There are various types of API testing, they are:

* Functional Testing
* Unit Testing
* Load Testing
* Security Testing
* UI Testing
* Interoperability and WS compliance Testing
* Penetration Testing (Pen Test)
* Fuzz Testing.

**5. What protocols can be tested using API Testing?**

API testing can be used for testing the following protocols:

* HTTP
* REST
* SOAP
* JMS
* UDDI

**6. What are the most commonly used tools for API testing?**

The most popularly used tool in the market is PostMan. This tool helps to create manual and automated test cases for testing the APIs in a well-designed manner. Apart from this, there are different tools like JMeter, Parasoft SOAtest, SoapUI, Apigee, API fortress, JUnit, etc.

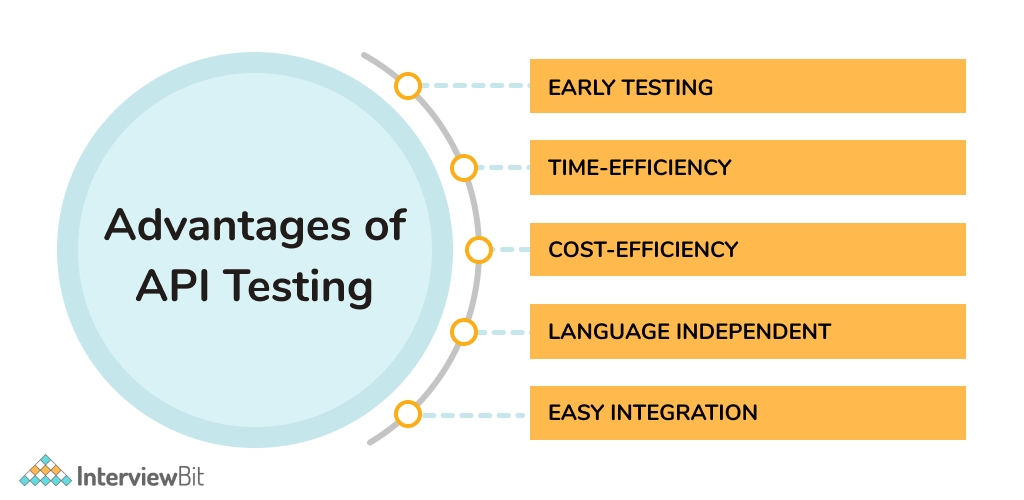
**7. What are the differences between API Testing and Unit Testing?**

| **API testing** | **UNIT testing** |
| --- | --- |
| This testing is owned by the Quality Analyst team. | This testing is owned by the developers working on the corresponding modules. |
| This belongs to the category of black box testing. | This belongs to white box testing. |
| Full system functionality is considered in API testing as the API would be used by external developers. | As the name indicates, this testing verifies whether the unit of code works as expected or not in isolation. |
| In this testing, the testers do not have access to the internal source code and it focuses only on the functionality of the API. | The developers who work on developing unit test cases have access to the source code as they need to ensure the modules developed are passed before delivery. |

**8. What are the advantages of API Testing?**

There are various advantages of API testing. Some of the most significant advantages are as follows:

* **Core Functionality Testing:** This kind of testing provides access to the entire system without the need for a user interface. The functionalities of the application would be evaluated end-to-end without the need for GUI (Graphical User Interface) which helps in detecting issues that can prove to be bigger at the time of GUI testing.
* **Time and Cost Effectiveness:** This is usually less time-consuming when compared to GUI testing. It also requires less code for testing the functionalities thereby making it easier to set up and get faster access to test coverage. It also results in effective cost savings for the project.
* **Language-Independent:** The data transfer between the test platform to the applications is done utilizing XML or JSON and is completely independent of the languages used in developing the systems. The test automation suite can be developed in any language.
* **Ease of Integration with GUI:** API testing provides highly flexible test suites that help in easier integration with the GUI tests. For instance, before the GUI test cases are initiated, employing API test cases, we can create sample users that can act as an initial base for the GUI tests.



**9. What is the approach followed in API Testing?**

We follow the below approaches in API Testing:

* Firstly, write required test cases for testing the APIs by making use of different testing techniques like equivalence class, boundary value analysis etc that helps to verify the functionalities.
* Clearly define the scope and functionality of the APIs.
* Define the different input parameters that you want to test the API with.
* Verify the test cases by passing the input parameters.
* Compare the results of the different test cases based on their expected behavior.
* Also subject the API to different conditions depending on the description of the functionality.

**10. What needs to be verified in API testing?**

In API Testing, we send a request to the API and then we analyze the responses based on the following parameters:

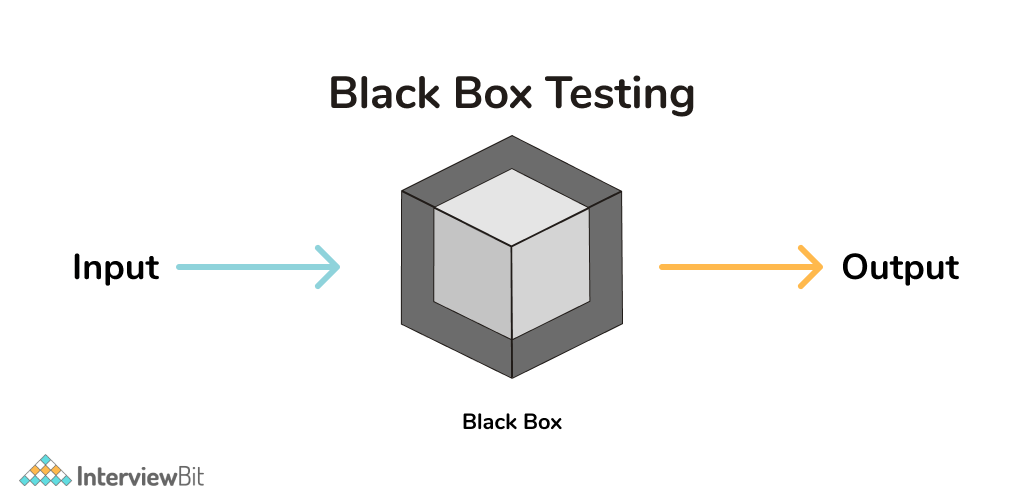
1. Data accuracy of the actual response with the expected response.
2. HTTP status codes of the resultant API.
3. Response time of the API.
4. Error codes if the APIs are expected to throw an error.
5. Authorization Details
6. Test non-functional specifications of the applications like security, performance, availability etc.

**11. What are the best practices that need to be followed for writing test cases?**

* We need to write test cases that correspond with the perspective of end-users.
* Steps defined in the test cases needs to be simple so that anyone can replicate the steps.
* Ensure that the test cases are reusable.
* Define and set the priority of test cases.
* Provide a valid description, test input parameters, test data, expected outcome after running the test cases so that we can compare the actual outcomes of the test cases with the expected ones.
* Make sure to develop test cases that cover negative test scenarios too.
* Naming conventions need to be properly followed while developing test cases.
* Review them regularly and update them as and when the functionality gets updated.

**12. What do you understand by Black Box Testing?**

Black Box Testing is one of the methods in software testing where the testers evaluate the software functionality without the knowledge of the internal source code. This ideology can be applied in every branch of testing such as unit testing, integration testing, system testing and acceptance testing.



**13. Define Test Data.**

Test data is the input data used by the testers to execute their test cases. This data can be prepared either manually or by making use of tools. For instance, to test the login functionality of an application, testers would need input data such as username and password which constitutes the test data.

**14. Define test coverage.**

Test coverage is a measure that signifies the amount of testing performed by making use of our test cases. It can be either functional testing or non-functional testing of the application. This provides a fair idea about what the testers need to cover in their test cases.

**15. Does the API tester need to have coding knowledge to perform API testing?**

API testing covers manual and automated testing. Manual testing does not require a tester to know to code. We just need API request details, headers, payload, credentials and know how to use the tools required to test the system. But in the case of automated testing, a tester needs to know how to code to automate test cases and develop a sophisticated optimised test suite.

**16. What is the process of API Specification Review?**

API Specification Review is the first and foremost step of documentation of the API testing needs and requirements. It should clearly state the purpose of the API, the application workflow and its features along with all other required details which could possibly help to plan the API testing process smoothly.

**17. What is Latency in API testing?**

Latency refers to the response time or the delay taken by the request to reach the server. We need to ensure that the latency involved in reaching the server is minimum as higher the latency, greater is the impact in the application’s speed and performance.

**18. What do you understand by Throughput in Performance testing?**

Throughput refers to the number of transactions per second that an application can handle under the influence of lot of users (load). The API needs to ensure that required throughput is met before it is deployed on production. We can identify this by performing the Load Testing of the APIs. We can do this by identifying multiple transactions of different priorities and check how many requests are successfully passed in acceptable time governed by the SLAs (Service Level Agreements) defined by us.

**API Testing Interview Questions for Experienced**

**19. How do you document an API functionality? What are the tools available for achieving the same?**

API documentation represents any description of the functionality of the API. Since this documentation will be used by external developers, we need to follow some best practices. They are:

* Plan what needs to be shown in your documentation.
* Do not exclude any fundamental sections of the API functionality.
* Use simple words by avoiding technical jargon.
* Include various interactive examples and resources to understand the API functionality clearly.
* Consistently maintain the documentation as and when the functionality gets updated.

Some of the popular tools used for achieving API documentation are:

* JavaDoc
* Doxygen
* OpenAPI
* Redoc
* Swagger UI

**20. What is the most important difference between API testing and UI testing?**

UI testing represents testing by using Graphical User Interface. The main area of focus in this is to test the look and feel of the software application by focusing on how the application is feasible for the end-users, do the functionalities of all the items shown on the UI screen - images, fonts, buttons, layouts etc are appearing properly as expected.  
On the other hand, API testing ensures the testing of communication of data between various software systems. It mostly falls under the validation of back-end functionality.

**21. What are the major blockers or challenges faced while performing API testing?**

Some of the challenges faced while doing API testing are:

* Proper Parameter Selection
* Proper Parameter Combination
* Knowing which API needs to be called in what sequence
* Proper knowledge of output verification
* Knowing what are the proper input values that needs to be provided to the API inputs.

**22. What are the principles that need to be followed while performing API Testing?**

There are 7 principles in API testing design. They are:

* **Optimal Testing:** Since it is not possible to test anything exhaustively, there needs to be at least some optimal amount of testing depending on the risk assessment performed on the application.
* **Defect Clustering:** This represents the clustering of modules containing most defects. As a general rule, almost 80% of the defects are found in 20% of the application modules. These risky modules can be identified by following this approach.
  + However this principle has a problem. We might have to perform API testing repeatedly on the application which might be obstructive to finding new bugs.
* **Pesticide Paradox:** This paradox states that if the testers use the same test cases repeatedly, then they would not be capable of finding new bugs over a period of time. Hence, the testers need to be fully equipped by reviewing and revising the test cases regularly. New test cases are more effective in finding new bugs. However, no amount of testing can ensure that the product is 100% bug-free.
* **Presence of defects:** Another principle of testing is that the aim of testing should be the presence of bugs and not the absence of defects. The goal of testing is to reduce the probability of finding undiscovered bugs. In case 0 defects are found, yet we cannot fully claim that our software is 100% bug-free.
* **Error absence fallacy:** There can be an increased possibility of software being unusable even if it is 99% bug-free which can be wrong. The main focus of software testing is to ensure that the software addresses the business requirements correctly. This principle states that even if we identify and fix the defects if the software does not fulfil the business requirements, then the system is unusable.
* **Early Testing:** This principle states that it is easier to fix bugs early in the software development lifecycle. Hence, it is recommended to start testing the application as early as possible.
* **Context Dependent Testing:** The way we test an application depending on the type of the system. We use different techniques, methodologies and test cases to test the applications depending on the type. For example, the way we test e-commerce applications is different from the way we test online streaming applications.

**23. What are the different bugs that can be found in API testing?**

We can find the below bugs at the time of API testing:

* Duplicate or missing API functionality
* Failure to handle negative test cases
* Failure to handle a sudden spike in load or stress
* Reliability of the application behavior
* Failure to handle requests securely
* Unused flags
* Unimplemented errors
* Poor Performance
* Issues in Multi-threading
* Improper error responses
* Improper status codes

**24. Define Test API.**

Test API refers to a set of APIs or library utilities that helps developers to create automated test cases for testing .NET or WIN 32 systems. It has a set of basic data building blocks, data types, data structures etc.

**25. What is Payload?**

Payload is the most common term used in the case of REST APIs. It refers to the actual data sent to the server in the API request in different formats like JSON, XML etc.

**26. What is Run Scope?**

Run Scope is an API testing tool that is typically a web application supporting an easier user interface platform to test back-end services. For more information, visit [here](https://www.runscope.com/).

**27. What is the importance of caching mechanism?**

Caching mechanism is the practice of storing data temporarily to retrieve data for repeated requests. This increases the performance of the system by obtaining the data from the cached copy instead of hitting the database and getting the original data.

**28. Why is automated API testing useful?**

Automated testing is useful in the long run as it helps to maximize the test coverage of the applications in a shorter period of time meaning it helps to test large test sets very easily and quickly. It enables parallel execution and helps to reduce human-generated errors in testing. It saves the time required to test applications thereby saving the overall cost.

**29. What do you understand by Input injection?**

Input injection is the act of simulating inputs for testing APIs. It can be simulated in different ways:

* Direct Method Invocation
* Accessibility interface invocation
* Low-level input simulation
* Device driver simulation
* Robot Simulation.

**30. What do you understand by the test environment for API?**

The test environment is an environment that helps test the APIs developed by providing a feature to send requests and get the responses from the server. It is similar to the production environment where the Quality Analyst has enough data to perform their functionality testing. In a test environment, we can have a test database, a localized gateway, a server and a load balancer.

**31. Is it possible to hack API while testing?**

Yes, it is possible. This is because we are sending requests over the internet which mostly follows HTTP protocol. This protocol is text-based and is easier to read. Hence, it is required to perform security testing of the APIs to ensure safer systems.

**32. How should we test the API security?**

To test the security of the API during API testing, we need to validate 2 things:

* **Authentication**: Whether the identity of the end-user is correct.
* **Authorization**: Whether the user is allowed to access the resource.

We can also validate whether the TLS or the SSL certificate used over the HTTPS protocol is valid or not.

**33. What do you understand by Big Bang Approach in testing?**

The big bang approach is the approach of combining all modules at once and then performing verification of the functionality after the individual modules are tested. It belongs to the category of integration testing of the applications.

**34. How do you perform API Load Testing?**

* Load Testing is a category of Performance Testing that is used for checking an application’s capability to perform under various user loads.
* This is done for identifying bottlenecks in performance before the application becomes live.
* It is done by simulating many users hitting the API at the same time or in other words, artificial traffic is simulated to identify if the application is capable of handling the load by maintaining consistency in the response times and not impacting the functionality.
* One such tool to perform Load Testing is JMeter. It provides flexibility to create a test plan, define the thread groups and record test scripts to simulate artificial load to the API. It finally provides a feature to visualize the result of load testing done. For more information about using JMeter, you can refer [here](https://www.loadview-testing.com/blog/load-test-using-jmeter/).

**Useful Interview Resources:**

* [Automation Testing](https://www.interviewbit.com/automation-testing-interview-questions/)
* [JMeter](https://www.interviewbit.com/jmeter-interview-questions/)
* [Web Services](https://www.interviewbit.com/web-services-interview-questions/)
* [Web API](https://www.interviewbit.com/web-api-interview-questions/)
* [REST API](https://www.interviewbit.com/rest-api-interview-questions/)
* [Robot Framework](https://www.interviewbit.com/robot-framework-interview-questions/)
* [Database Testing](https://www.interviewbit.com/database-testing-interview-questions/)
* [UFT/QTP](https://www.interviewbit.com/uft-interview-questions/)

**What is an API?**

An API (Application Programming Interface) is a software intermediary that enables two applications to communicate with each other. It comprises a number of subroutine definitions, logs, and tools for creating application software.

In an API testing interview, you could be asked to give some API examples, here are the well-known ones: Google Maps API, Amazon Advertising API, Twitter API, YouTube API, etc.

**2. What are the main differences between API and Web Service?**

* All Web services are APIs but not all APIs are Web services.
* [Web services](https://katalon.com/web-testing)might not contain all the specifications and cannot perform all the tasks that APIs would perform.
* A Web service uses only three styles of use: SOAP, REST, and XML-RPC for communication whereas API may be exposed in multiple ways.
* A Web service always needs a network to operate while APIs don’t need a network for operation.

**3. What are the Limits of API Usage?**

Many APIs have a certain limit set up by the provider. Thus, try to estimate your usage and understand how that will impact the overall cost of the offering. Whether this will be a problem depends in large part on how data is leveraged. Getting caught by a quota and effectively cut off because of budget limitations will render the service (and any system or process depending on it) virtually useless.

**Creating an API (Common Web API Testing interview questions)**

**4. What are some architectural styles for creating a Web API?**

This is one of the fundamental Web API interview questions. Bellows are four common Web API architectural styles:

* HTTP for client-server communication
* XML/JSON as formatting language
* Simple URI as the address for the services
* Stateless communication

**5. Who can use a Web API?**

Web API can be consumed by any clients which support HTTP verbs such as GET, PUT, DELETE, and POST. Since Web API services do not require configuration, they can be easily used by any client. In fact, even portable devices such as mobile devices can easily use Web API, which is undoubtedly the biggest advantage of this technology.

**Testing an API – Top Web API Testing interview questions & answers**

**6. What is API Testing?**

[API testing](https://katalon.com/api-testing/) is a kind of software testing that determines if the developed APIs meet expectations regarding the functionality, reliability, performance, and security of the application.

**7. What are the advantages of API Testing?**

In an API interview, they are likely to ask about the advantages of API testing. So be prepared with the significant ones such as:

* ***Test for Core Functionality:***API testing provides access to the application without a user interface. The core and code-level of functionalities of the application will be tested and evaluated early before the GUI tests. This will help detect minor issues which can become bigger during the GUI testing.
* ***Time Effective:***API testing usually is less time-consuming than functional GUI testing. The web elements in GUI testing must be polled, which makes the testing process slower. Particularly, API test automation requires less code so it can provide better and faster test coverage compared to GUI test automation. These will result in cost saving for the testing project.
* ***Language-Independent:*** In API testing, data is exchanged using XML or JSON. These transfer modes are completely language-independent, allowing users to select any coding language when adopting automation testing services for the project.
* ***Easy Integration with GUI:*** API tests enable highly integrable tests, which is particularly useful if you want to perform functional GUI tests after API testing. For instance, simple integration would allow new user accounts to be created within the application before a GUI test started.

**8. Some common protocols used in API testing?**

Many protocols are now available to be used in API testing, such as JMS, REST, HTTP, UDDI and SOAP.

**9. What is the test environment of API?**

Setting up the API’s test environment is not an easy task, so you should have a ready answer if your API testing interview is coming. The test environment of API is a bit complete and requires the configuration of the database and server, depending on the software requirements. No GUI (Graphical User Interface) is available in this test form.

When the installation process is complete, API is verified for proper operation. Throughout the process, the API called from the original environment is set up with different parameters to study the test results.

**10. What are the principles of an API test design?**

The five most important principles of an API test design are:

* Setup: Create objects, start services, initialize data, etc
* Execution: Steps to apply API or the scenario, including logging
* Verification: Oracles to evaluate the result of the execution
* Reporting: Pass, failed, or blocked
* Clean up: Pre-test state

**11. What are the common API testing types?**

While there are certainly specialty tests, and no list can be asked to be comprehensive in this realm, most tests fit broadly into the following nine categories that you should remember before attending an API testing interview.

1. Validation Testing
2. Functional Testing
3. UI testing
4. Load testing
5. Runtime/ Error Detection
6. Security testing
7. Penetration testing
8. Fuzz testing
9. Interoperability and WS Compliance testing

[](https://katalon.com/sign-up)

**12. What is the procedure to perform API testing?**

1. Choose the suite to add the API test case
2. Choose the test development mode
3. Demand the development of test cases for the required API methods
4. Configure the control parameters of the application and then test conditions
5. Configure method validation
6. Execute the API test
7. Check test reports and filter API test cases
8. Arrange all API test cases

**13. What must be checked when performing API testing?**

During the API testing process, a request is raised to the API with the known data. This way you can analyze the validation response. While testing an API, you should consider:

* Accuracy of data
* Schema validation
* HTTP status codes
* Data type, validations, order, and completeness
* Authorization checks
* Implementation of response timeout
* Error codes in case API returns, and
* Non-functional testing like performance and security testing

**14. What is the best approach method to perform API testing?**

The following factors should be considered when performing API testing:

* Defining the correct input parameters
* Verifying the calls of the mixture of two or more added value parameters
* Defining the basic functionality and scope of the API program
* Writing appropriate API test cases and making use of testing techniques such as equivalence class, boundary value, etc. to check the operability
* Testing case execution
* Comparing the test result with the expected result
* Verifying the API behavior under conditions such as connection to files and so on.

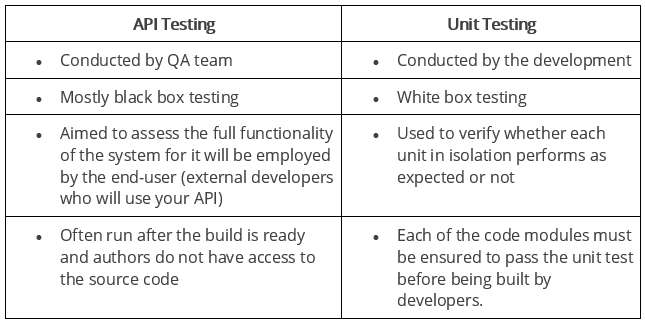
**15. What tools could be used for API testing?**

There is myriad different [API testing tools](https://www.katalon.com/resources-center/blog/top-5-free-api-testing-tools/) available. A few common tools are Katalon Studio, Postman, SoapUi Pro, Apigee, etc.  While doing Unit and API testing, both target source code. If an API method uses code based in .NET then another supporting tool must have .NET.

Learn more: [SoapUI vs Postman, Katalon Studio: A Review of Top 3 API Tools](https://www.katalon.com/resources-center/blog/soapui-vs-postman-katalon-api-tools/)

[](https://katalon.com/sign-up)

**16. What are the differences between API Testing and Unit Testing?**



**17. What are the differences between API Testing and UI Testing?**

* API enables the communication between two separate software systems. A software system implementing an API contains functions or subroutines that can be executed by another software system.
* On the other hand, UI ( User Interface) testing refers to testing graphical interfaces such as how users interact with the applications, and testing application elements like fonts, images, layouts, etc. UI testing basically focuses on the look and feel of an application.

**18. What are the major challenges faced in API testing?**

If you can overcome the challenges in API Testing, you can be confident in the API testing interview too. They are:

* Parameter Selection
* Parameter Combination
* Call sequencing
* Output verification and validation
* Another important challenge is providing input values, which is very difficult as GUI is not available in this case.

**19. What are the testing methods that come under API testing?**

One of the most common Web API testing interview questions is about the testing methods. They are:

* Unit testing and Functional testing
* Load testing to test the performance under load
* Discovery testing to list, create and delete the number of calls documented in the API
* Usability and Reliability testing to get consistent results
* Security and Penetration testing to validate all types of authentication
* Automation testing to create and run scripts that require regular API calls
* End to end Integration and Web UI testing
* API documentation testing to determine its efficiency and effectiveness

**20. Why is API testing considered as the most suitable form for Automation testing?**

API testing is now preferred over GUI testing and is considered as most suitable because:

* It verifies all the functional paths of the system under test very effectively.
* It provides the most stable interface.
* It is easier to maintain and provides fast feedback.

**21. What are common API errors that are often founded?**

Not only API fundamental questions, the interviewer also determine your knowledge and experience by asking about the API errors in a Web API testing interview. So the most common ones are:

* Missing module errors
* Documentation errors
* Parameter validation errors
* And some standard error expectations as if the result is not so predicted then the occurrence of errors can be seen and the same warnings are specified in the form of a message. There can be one or more warnings within an individual module.

**22. What kinds of bugs that API testing would often find?**

* Missing or duplicate functionality
* Fails to handle error conditions gracefully
* Stress
* Reliability
* Security
* Unused flags
* Not implemented errors
* Inconsistent error handling
* Performance
* Multi-threading issues
* Improper errors

**Documenting the API (Common Web API Testing interview questions)**

**23. What is API documentation?**

The API documentation is a complete, accurate technical writing giving instructions on how to effectively use and integrate with an API. It is a compact reference manual that has all the information needed to work with the API and helps you answer all the API testing questions with details on functions, classes, return types, arguments, and also examples and tutorials.

**24. What are API documentation templates that are commonly used?**

There are several available API documentation templates that help to make the entire process simple and straightforward, which could be  answered in your API testing interview, such as:

* Swagger
* Miredot
* Slate
* FlatDoc
* API blueprint
* RestDoc
* Web service API specification

**25. When writing API document, what must be considered?**

* Source of the content
* Document plan or sketch
* Delivery layout
* Information needed for every function in the document
* Automatic document creation programs

**26. How often are the APIs changed and, more importantly, deprecated?**

APIs, especially modern RESTful APIs, is a nice creation that can certainly simplify and accelerate integration efforts, which makes it more likely you will benefit from them. But APIs can and do change for various reasons, sometimes abruptly, and hence REST APIs do not differ from traditional integration methods in this respect. If an API call is obsolete and disappears, your procedure will interrupt and it is important to understand how often the APIs you depend on change or are deprecated.

**REST (Common Web API Testing interview questions)**

**27. What is REST?**

REST (Representational State Transfer) is an architectural style for developing web services that exploit the ubiquity of HTTP protocol and uses the HTTP method to define actions. It revolves around resources where every component is a resource that can be accessed through a shared interface using standard HTTP methods.   
  
In REST architecture, a REST Server provides access to resources and REST client accesses and makes these resources available. Here, each resource is identified by URIs or global IDs, and REST uses multiple ways to represent a resource, such as text, JSON, and XML. XML and JSON are nowadays the most popular representations of resources.

**28. What is a RESTFul Web Service?**

Mostly, there are two kinds of Web Services that should be remembered in your next API testing interview:

1. SOAP (Simple Object Access Protocol) – an XML-based method to expose web services.
2. Web services developed in the REST style are referred to as RESTful web services. These web services use HTTP methods to implement the concept of REST architecture. A RESTful web service usually defines a URI, Uniform Resource Identifier a service, and provides resource representation like JSON and a set of HTTP methods.

**29. What is a “Resource” in REST?**

REST architecture treats any content as a resource, which can be either text files, HTML pages, images, videos, or dynamic business information.   
REST Server gives access to resources and modifies them, where each resource is identified by URIs/ global IDs.

**30. What is the most popular way to represent a resource in REST?**

REST uses different representations to define a resource like text, JSON, and XML.   
XML and JSON are the most popular representations of resources.

**31. Which protocol is used by RESTful Web services?**

RESTful web services use the HTTP protocol as a medium of communication between the client and the server.

**32. What are some key characteristics of REST?**

Key characteristics of REST are likely asked in a Web API Testing interview. So please get the answer ready in your mind with these 2 ones:

* REST is stateless, therefore the SERVER has no status (or session data)   
  With a well-applied REST API, the server could be restarted between two calls, since all data is transferred to the server
* Web service uses POST method primarily to perform operations, while REST uses GET for accessing resources.

**33. What is messaging in RESTful Web services?**

RESTful web services use the HTTP protocol as a communication tool between the client and the server. The technique that when the client sends a message in the form of an HTTP Request, the server sends back the HTTP reply is called Messaging. These messages comprise message data and metadata, that is, information on the message itself.

**34. What are the core components of an HTTP request?**

An HTTP request contains five key elements:

1. An action showing HTTP methods like GET, PUT, POST, and DELETE.
2. Uniform Resource Identifier (URI), which is the identifier for the resource on the server.
3. HTTP Version, which indicates HTTP version, for example-HTTP v1.1.
4. Request Header, which carries metadata (as key-value pairs) for the HTTP Request message. Metadata could be a client (or browser) type, format supported by the client, format of a message body format, cache settings, and so on.
5. Request Body, which indicates the message content or resource representation.

**35. What are the most commonly used HTTP methods supported by REST?**

* GET is only used to request data from a specified resource. Get requests can be cached and bookmarked. It remains in the browser history and haS length restrictions. GET requests should never be used when dealing with sensitive data.
* POST is used to send data to a server to create/update a resource. POST requests are never cached and bookmarked and do not remain in the browser history.
* PUT replaces all current representations of the target resource with the request payload.
* DELETE removes the specified resource.
* OPTIONS is used to describe the communication options for the target resource.
* HEAD asks for a response identical to that of a GET request, but without the response body.

**36. Can GET request be used instead of PUT to create a resource?**

The PUT or POST method should be used to create a resource. GET is only used to request data from a specified resource.

**37. Is there any difference between PUT and POST operations?**

PUT and POST operations are quite similar, except for the terms of the result generated by them.

PUT operation is idempotent, so you can cache the response while the responses to POST operation are not cacheable, and if you retry the request N times, you will end up having N resources with N different URIs created on the server.

In a Web API Testing interview, you should give a specific example for PUT and POST operations to make it crystal clear to the interviewer. Below is an example:

*Scenario:* *Let’s say we are designing a network application. Let’s list down a few URIs and their purpose to get to know when to use POST and when to use PUT operations.*

) What is API?

**API** (*Application Programming Interface*) helps in communication and data exchange between two software systems. API act as an interface between two applications and allows the two software systems communicate with one another. API is a collection of functions which can be executed by another software program.

API works as; it takes a request from the source, takes that request to the database, fetches the request data from the database and returns a response to the source. API takes the requests from the user and gives the response without exposing the internal details. API acts as Abstraction.

**Example:** Amazon API, Google Map API

56.4M

1K

Java Try Catch

2) What is API testing?

**API testing** is a type of software testing that involves testing APIs directly. API is a part of integration testing to check whether the API meets expectations in terms of functionality, reliability, performance, and security of applications. Multiple API system can performed API testing. In API testing, our primary focus is on Business Logic Layer of the software architecture.

3) What are the types of API testing?

API testing involves the following types of testing:

* Unit Testing
* Functional Testing
* Load Testing
* Runtime/Error Detection
* Security Testing
* UI Testing
* Interoperability and WS compliance Testing
* Penetration Testing
* Fuzz Testing

4) What are the protocols used in API Testing?

Protocols used in API testing are:

* HTTP
* REST
* SOAP
* JMS
* UDDI

5) What are the tools used for API Testing?

Tools used for API testing are:

* Parasoft SOAtest
* PostMan
* AlertSite API monitoring

6) What is API test environment?

For API the test environment is a quite complex method where the configuration of server and database is done as per the requirement of the software application. API testing does not involve graphical user interface (GUI).

API is checked for its proper functioning after installation.

7) What is API framework?

API framework is described by the config. File which consist of the list of all APIs that are required to be activated and are activated for any particular program run. This is essential as every test run does not require all APIs.

8) What are the limits of API usage?

Many APIs have certain limit set up by the provider. Hence, try to estimate our usage and understand how that will impact the overall cost of the offering.

9) What are the advantages of API testing?

Advantages of API testing are:

* **Test for core functionality:** API testing provides access to the application without the user interface. The core functionality of the application will be tested before the GUI tests. This will help to detect the minor issue which can become bigger during the GUI testing.
* **Time effective:** API testing is less time consuming than GUI testing. Particularly, API test requires less code so it can provide better and faster test coverage compare to GUI test automation. This will reduce the cost for the testing project.
* **Language Independent:** In API testing data is exchange using XML or JSON. These transfer mode are completely language-independent, which allows users to select any code language when adopting automation test service for the project.
* **Easy Integration with GUI:** API tests provide highly integrable tests which is useful to perform functional GUI tests after GUI tests. Simple integration would allow new user accounts to be created within the application before GUI started.

10) What are the principles of an API test design?

Here, are the seven principles of API test design.

1. **Exhaustive Testing:** Exhaustive testing is not possible. Instead we need optimal amount of testing which is based on the risk assessment of the application.
2. **Defect Clustering:** Defect Clustering states that a small number of modules contain the most of the defect detected. Approximately 80% of the defect found in 20% of the modules. By experience we can identify such risky modules. But this approach has its own problems. If the same tests are repeated over and over again, eventually the same test case will no longer find new bugs.
3. **Pesticide Paradox:** Testers cannot depend on existing technique. They must have to look continually to improve the existing method to make testing more effective. But even all these hard work in testing we can never claim our product is bug free. To overcome this, test cases need to be regularly reviewed and revised add new and different test cases to help find more defects.
4. **Testing shows presence of defects:** Testing principle states that- testing talks about the presence of defects not about the absence of defect. Software testing reduces the probability of undiscovered defects remaining in the software but even if no defects found, it is not a proof of correctness.

But if we work hard, taking all precautions and make our software products 99% bug free. The software does not meet the needs and requirements of the client.

1. **Absence of error -fallacy:** This can be possible the software which is 99% bug free is still unusable. The case can be if the system is tested for the wrong requirement. Software testing is not finding the defects but also to check that software addresses the business needs. The absence of error is fallacy i.e. finding and fixing defects does not help if the system build is unusable and doesn't fulfill the user's needs and requirements.
2. **Early Testing:** Testing should start as soon as possible in the software development lifecycle. So that defects in the requirement or design phase captured in the early stages. It is cheaper to fix defect in the early stages of testing. We should start finding the bug at the moment the requirements are defined.
3. **Testing is context dependent:** Testing is context dependent that we test an e-commerce site will be different from the way we test the commercial. All the developed software's are not identical. We will use different methodology; techniques and type of testing depend on the application type.

11) What is API framework?

A framework or software framework is a platform for developing software applications. API framework is a foundation on which software developer can build applications for a specific platform.

**Example:** A framework can include predefined classes and functions that can be used to process input, manage hardware devices and interact with system software.

Framework is similar to an Application Programming Interface, technically framework includes API. Framework serves foundation for programming while API provides access to the elements supported by the framework. Framework also includes code libraries, compiler and other programs used in the software development process.

API framework is defined by configuration file which consists the list of all APIs that is required to be activated and activated for a particular program run.

12) What are the common tests that performed on API?

Here, are the common tests that performed on API are as:

1. Response of the API should be verified based on the request. We will verify that the return value is based on request.
2. When API is updating any data structure we should verify the system is authenticating the outcome.
3. We will verify whether the API is trigger other event or request another API.
4. We will verify the behavior of the API when no value is return.

13) What exactly needs to verify in API testing?

In API testing, we send a request to API with the known data and then analysis the response.

1. We will verify the accuracy of the data.
2. Will see the HTTP status code.
3. We will see the response time.
4. Error codes in case API returns any errors.
5. Authorization would be check.
6. Non-Functional testing such as performance testing, security testing.

14) What are the differences between API and Web Services?

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **API** | **Web Services** |
| **1.** | API may or may not need network for its operations. | Web Services always need network for its operation. |
| **2.** | API can be communicated through SOAP, REST, XML-RPC and CURL calls as well. API can also be exposed in number of ways like JAR, DLL, XML over HTTP, JSON over HTTP etc. | Web service can be communicated through SOAP, REST, AND RPC. |
| **3.** | API can perform all the operations which web service can't perform. | Web service can't perform all the operations like API. |
| **4.** | All APIs are not web service. | All web services are API |

15) What is API documentation?

A good documentation is must for any foundation. API documentation serves as quick reference for accessing library or working within a program.

When we use any such documents, it must consists of proper plan, content source, proper layout, information related to each function etc.

There are various documentation tools like Doxygen and JavaDoc. Here, are the functions which are documented which revolve around the parameters like:

* Function description
* Type and syntax of error message that may occure
* Syntax, elements and sequence needed for each parameter
* Links regarding functions

16) What is the most used template for API documentation?

Here, are the various documentation template that make the whole process simple and easy. They are:

* Swagger
* Miredot
* Slate
* FlatDoc
* API blueprint
* RestDoc
* Web service API specification

17) What are the types of bug that can be found during API testing?

API testing helps us to find many types of bugs which are:

* Stress
* Security
* Duplicate or missing functionality
* Reliability
* Unused flags
* Performance
* Incompatible error handling
* Multi-threaded issue
* Improper errors

18) What are the difference between API testing and UI testing?

UI (User Interface) testing means the testing of the graphical user interface. The focus of UI testing is on the look and feel of the application. In user interface testing the main focus is on how users can interact with app elements such as images, fonts, layout etc. are checked.

API testing allows the communication between two software systems. API testing works on backend also known as backend testing.

19) What is SOAP?

SOAP (Simple Object Access Control) . It is an XML based protocol that helps in exchanging information among computers.

20) What is REST API?

**REST API** is a set of function helps the developers performing requests when the response is receiving. Through HTTP protocol interaction is made in REST API.

REST is defined as Representational state transfer. It is an effective standard for API creation.

21) What are the differences between SOAP and REST API?

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **SOAP API** | **REST API** |
| **1.** | **SOAP** stands as Simple Object Access Protocol. | **REST** stands as Representational State Transfer. |
| **2.** | **SOAP** is a protocol. | **REST** is an architectural pattern. |
| **3.** | **SOAP** can work with XML format. In SOAP all the data passed in XML format. | **REST** permit different data format such as Plain text, HTML, XML, JSON etc. But the most preferred format for transferring data is in JSON. |

22) What are the major challenges faced during API testing?

The major challenges faced during the API testing are:

* Parameter Selection
* Parameter Combination
* Call sequencing
* Output verification and validation
* A major challenge is providing input values which are very difficult because GUI is not available.

23) What are the difference between API Testing and Unit Testing?

Difference between API testing and Unit testing are:

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **API Testing** | **UNIT Testing** |
| **1.** | API testing is a form of black box testing. | Unit testing is a form of white box testing. |
| **2.** | API testing is performed after the project completion during the test. | Unit testing is performed when the project is created. |
| **3.** | In API testing there is a wide scope of testing. | In Unit testing there is a limited scope of testing we can test only the basic functionality. |
| **4.** | API testing is done by the testers. The whole purpose of API testing is end to end testing of the functionality. | Unit testing is done by the developer. In unit testing every functionality is separately tested. |

24) What is a RESTFUL web services?

There are two kinds of web services

1. SOAP Web Services
2. RESTFUL Web Services

**1. SOAP (Simple Object Access Protocol) -** SOAP is a XML based method which is used in Web Services.

**2. RESTFUL Web Services -** To implement the concept of REST architecture HTTP method is used. RESTFUL Web Services defines URI (Uniform Resource Identifier), and also provides resource representation like JSON and a set of HTTP method.

25) What is Resource in REST?

REST architecture treats any content as resource, which can be text files, HTML pages, images, videos or dynamic business information. REST server gives the functionality to access the resources and modifies them. We can identify the each resources by URIs/ global IDs.

26) What is the way to represent the resource in REST?

REST uses different representation to define the resources like text, JSON and XML. The most popular representation of resources is JSON and XML.

27) What protocol is used by the RESTFUL Web Services?

RESTFUL Web Services uses the HTTP protocol. They use the HTTP protocol as a medium of communication between the client and the server.

28) What are the characteristics of REST?

Here, are the two characteristics of REST.

1. REST is stateless. With the use of the REST API the server has no status, we can restart the server between two calls, inspite of all the data is transferred to the server.
2. Web Services uses POST method to perform operations, while REST uses GET method to access the resources.

29) What is messaging in RESTFUL Web Services?

RESTFUL Web Services use the HTTP protocol as a communication tool between the client and the server. This is the technique when the client sends a message in the form of HTTP request the server send back the HTTP reply which is called Messaging. This message consists message data and Meta data i.e. information on the message itself.

30) What are the components of an HTTP request?

An HTTP request have five components. These are:

1. **Action showing HTTP method** like GET, PUT, POST, DELETE.
2. **Uniform Resource Identifier (URI):** URI is the identifier for the resource on the server.
3. **HTTP version:** Indicate the HTTP version like- HTTP V1.1.
4. **Request Header:** Request Header carries metadata for the HTTP request message. Metadata could be a client type, format supported by the client, format of a message body, cache setting etc.
5. **Request Body:** Resource body indicates message content or resource representation.

31) What is the HTTP protocol supported by REST?

**GET:** GET is used to request data from the specified resource.

GET request can be cached and bookmark. It remains in the browser history and has length restriction. When dealing with sensitive data GET requests should not be used.

**POST:** POST is used to send data to server for creation or updating the resources.

POST requests are never cached or bookmark.

**PUT:** PUT replaces the current representation of the target resource with the request payload.

**DELETE:** DELETE removes the specified resource.

**OPTIONS:** OPTION is used to describe the communication option for the target resources.

**HEAD:** HEAD asks for response which is identical to GET requests, but without the response body.

32) Can we use GET request instead of PUT to create a resource?

PUT or POST method is used create a resource. GET is only used to request the resources.

33) What is URI? What is the purpose of web-based service and what is it's format?

URI stands for Uniform Resource Identifier. It is a string of characters designed for unambiguous identification of resources and extensibility by the URI scheme. The purpose of URI is to locate the resource on the server hosting of the web service.

A URIs format is ***<protocol>://<service-name>/<Resource Type>/<ResourceID>***

34) What are SOAP Web Services?

**SOAP** (*Simple Object Access Protocol*) is defined as the XML based protocol. SOAP is also known for developing and designing web services and also enable the communication between the applications developed on different platform by using different programming languages on the internet. SOAP is platform and language independent.

35) When we can use SOAP API?

We can use SOAP API to perform the operation on records like create, retrieve, update or delete. We can use API to manage password, perform searches etc.

**What are the tools used for API testing?**

The tools used for various API testing are

* SoapUI Pro
* PostMan
* Alertsite API monitoring

[👉 Free PDF Download: API Testing Interview Questions](https://drive.google.com/uc?export=download&id=1utHtPR7Ulab33EGBfKphCHSctwOK13q6)

**2) What is API testing?**

API (Application Programming Interface) specifies how some software components should interact with other, in other words it’s a set of functions and procedures that allows the creation of applications which access the features or data of an application or [operating system](https://career.guru99.com/top-50-operating-system-interview-questions/).   Testing of these functions is known as API testing.

**3) What are the common tests performed on API’s?**

The common tests performed on API’s

* Verification of the API whether it is updating any [data structure](https://career.guru99.com/top-50-data-structure-interview-questions/)
* Verify if the API does not return anything
* Based on input conditions, returned values from the API’s are checked
* Verification of the API whether it triggers some other event or calls another API

**Most Frequently Asked API Testing Interview Questions:**

EXPLORE MORETop 10 Behavioral Interview Questions and Answers04:00Top 10 Interview Do's and Don'ts in a Job Interview02:11Top 5 Powerpoint Interview Questions and Answers02:11What to bring to a JOB Interview: Interview Tips01:50What to wear in interview for Women | Working women01:44How To Dress for an Job Interview for Men01:3610 Most Common Interview Questions and Answers 🔝03:545 Secrets Tips to Win an Interview01:42

**4) Mention the key difference between UI level testing and API testing?**

UI ( User Interface) refers to testing graphical interface such as how user interacts with the applications, testing application elements like fonts, images, layouts etc. UI testing basically focuses on look and feel of an application. While, API enables communication between two separate software systems. A software system implementing an API contains functions or sub-routines that can be executed by another software system

**5) Explain what is SOAP?**

API Testing Interview Questions

SOAP-stands for Simple Object Access Protocol, and it is an [XML](https://career.guru99.com/xml-interview-questions/) based protocol for exchanging information between computers.

**6) Explain what is REST API?**

It is a set of functions to which the developers performs requests and receive responses. In REST API interaction is made via HTTP protocol REST – stands for Representational State Transfer, it is quickly becoming defacto standard for API creation.

**7) Difference API and Unit Testing?**

|  |  |
| --- | --- |
| API testing | UNIT testing |
| * API is owned by QA team | * Unit testing is owned by development team |
| * API is mostly black box testing | * Unit testing is white box testing |
| * Full functionality of the system is considered in API testing as it will be used by the end-user (external developers who will use your API ) | * Unit testing is done to verify whether each unit in isolation performs as expected or not |
| * API test are often run after the build is ready and authors do not have access to the source code | * For each of their module the developers are expected to build unit tests for each of their code modules and have to ensure that each module pass unit test before the code is included in a build |

**8) How to test API’s ?**

To test the API’s you should follow the following steps

* Select the suite in which you want to add the API test case
* Choose test development mode
* Develop test cases for the desired API methods
* Configure application control parameters
* Configure test conditions
* Configure method validation
* Execute API test
* View test reports
* Filter API test cases
* Sequence API test cases

**9) Mention what the main areas to be taken in consideration while writing API document ?**

The key area to be considered when writing API documents are

* Source of the content
* Document plan or sketch
* Delivery layout
* Information required for each function in the document
* Automatic document creation programs

**10) In API document explain how to document each function ?What are the tools used for documentation?**

* **Description:** Small description about what a function does
* **Syntax:** Syntax about the parameter of the code, the sequence in which they occur, required and optional elements etc.
* **Parameters:** Functions parameters
* **Error Messages:** Syntax of error messages
* **Example Code:** Small snippet of code
* **Related Links:** Related functions

Popular tools used for API documentations are JavaDoc (for Java code ) Doxygen (for .Net code)

**11) Explain API framework?**

API framework is self-explanatory. Values for test run and for holding the configurable parts, config file is used.  Automated test cases must represent in “ parse-table” format within config file.  When testing API, it is not necessary to test each API so the config file have some section whose all API are activated for that specific run.

**12) How does the API Builder work?**

API Builder is a PLSQL program consists of four [SQL](https://www.guru99.com/sql-server-questions.html) files

* For setting API parameters and starting the process one file is responsible
* Two files are created for temporary tables and Master package to create the outputted code
* Fourth file creates “spooled” output of the code into a file called “output\_script\_.sql”

**13) Explain what is TestApi ?**

TestApi is a library of utility and test APIs that enables testers and developers to create testing tools and automated tests for .NET and Win32 application.  It provides a set of common test building blocks, types, data-structure and algorithms.

**14) What is Input injection and what are different ways of doing it ?**

Input Injection:  It is the act of simulating user input, in several ways you can simulate user input.

* Direct Method Invocation
* Invocation using an accessibility interface
* Simulation using low-level input
* Simulation using a device driver
* Simulation using a robot

**15) What are the main challenges of API testing?**

The main challenges in API testing is

* Parameter Selection
* Parameter Combination
* Call sequencing

**16) What is API testing with runscope ?**

Runscope is a web application that provides backend services and easy to use interface for testing APIs.

**17) Explain what are the principles of API test design?**

The principle for API test design are

* **Setup :** Create objects, start services, initialize data etc
* **Execution:** Steps to exercise API or scenario, also logging
* **Verification:** Oracles to evaluate execution outcome
* **Reporting:** Pass, failed or blocked
* **Clean up:** Pre-test state

**18) What are the types of Bugs will API testing finds?**

The types of Bugs, API will find

* Missing or duplicate functionality
* Fails to handle error conditions gracefully
* Stress
* Reliability
* Security
* Unused flags
* Not implemented errors
* Inconsistent error handling
* Performance
* Multi-threading issues
* Improper errors

**19) What are the tools used for API test automation?**

While testing Unit and API testing,  both target source code, if an API method is using code  based on .NET then the tool which is supporting should have .NET Automation tools for API testing can be used are

* NUnit for .NET
* JUnit for Java
* HP UFT
* Soap UI

**20) Mention the steps for testing API ?**

API testing steps

* Select the test case that has to be fulfilled
* For API call develop a test case
* To meet the test case configure the API parameters
* Determine how will you validate a successful test
* Using programming language like PHP or .NET execute the API call
* Allow the API call to return the data to validate

**What does API testing mean?**

API stands for Application Programming interface details regarding how some software components must act together. In general terms, API testing is a set of procedures and functions allowing the creation of apps accessing data or features of an operating system or application. All in all, testing of such procedures is acknowledged as API testing.

|  |
| --- |
| **Related Article:**[**API Testing**](https://mindmajix.com/what-is-api-testing) |

**2. What are the names of tests executed on APIs?**

There can be numerous reasons behind executing API testing and there are a number of tests that can be performed on APIs. Some common API test examples are as follows:

* Any data structure which demands proper validation can be updated by API
* In case some interruption arises during the process or any other event is prompted, you can call another API
* Tests for inputting condition-based return values can be performed
* With the help of these tests, some resources can be easily modified like process killing, an update of the database, etc.
* The return values can also be tested which can be even null or are with wrong results

**3. What is the procedure to perform API testing and what exactly needs to be checked?**

During the API testing process, a request is raised to the API with the known data. This way you can analyze the validation response. Basically, things that must be checked during performing API testing are:

1. Accuracy of data
2. Schema validation
3. HTTP status codes
4. Data type, validations, order, and completeness
5. Authorization checks
6. Implementation of response timeout
7. Error codes in case API returns, and
8. Non-functional testing like performance and security testing.

**4. What is the test environment of API?**

To set up the test environment of API is not a cakewalk. It is a bit complete and demands the configuration of the database as well as a server according to the need of the software. There is no availability of [GUI](https://www.computerhope.com/jargon/g/gui.htm) (Graphical User Interface) in this form of testing.

Once the installation process is over, API is verified for proper functioning. During the whole process API that is invoked by the initial environment is being set up with distinct parameters to examine the test results.

[](https://bit.ly/3if9dmk)

**5. What approach should be followed for API testing?**

There are certain factors that determine the API testing approach. Let’s check them below:

1. Defining the accurate input parameters
2. Verifying the calls of the blend of two or more than two value-added parameters
3. Defining the basic functionality and scope of the API program
4. Writing suitable API test cases and making use of testing techniques like equivalence class, boundary-value, etc. to verify the functionality
5. Testing case execution
6. Testing result comparisons with the results expected
7. Verifying behavior of API under conditions like the connection with files etc.

**6. Define the basic difference between API testing and UI level testing?**

UI testing means the testing of the graphical interface. Its focus is basically on the feel and looks of an application. Within user interface testing, things like how the user interacts with app elements such as images, fonts, layouts, etc are checked.

On the other hand API, testing allows communicating between two different software systems. During this testing, a software system that implements an API includes sub-routines or functions that can be performed by other software systems.

**7. Name the common protocols used in API testing.**

Many protocols are there that can be used in API testing. These are as follows:

JMS, REST, HTTP, UDDI, and SOAP

**8. Name different tools used for API testing.**

There are many [testing tools](https://mindmajix.com/testing-tools) available that can be used for API testing. A few of them are:

Postman, SoapUi Pro, Curl, and Alert site API monitoring

**9. What is SOAP?**

The term SOAP refers to Simple Object Access Control. In simple terms, it is an [XML](https://whatis.techtarget.com/definition/XML-Extensible-Markup-Language)-based protocol that helps in exchanging information among computers.

**10. What’s the procedure to test API’s?**

For testing API’s one must follow the below-mentioned steps:

1. Make a selection of the suite you like to add the API test case to
2. Now choose the test development mode
3. Next demands the development of test cases for the required API methods
4. After this, you need to configure the control parameters of the application and then test the conditions
5. Once done with all the previous steps, configure method validation
6. Now is the time for the execution of the API test
7. After this, you can check test reports and filter API test cases
8. Last but not least, sequence all API test cases. That’s it!

|  |
| --- |
| **Related Article:**[**Manual Testing Interview Questions**](https://mindmajix.com/manual-testing-interview-questions) |

**11. What is REST API?**

REST API is a set of functions helping developers in performing requests along with receiving responses. Through HTTP protocol interaction is made in REST API.

The term REST refers to Representational State Transfer. In a very short span of time, it has become an effective standard for API creation.

**12. What are the areas that need to be taken care of while writing API documents?**

The main areas that need your concentration while writing API documents are as follows:

1. You need to check the source of the content
2. Plan or sketch of your document
3. The delivery layout of the same
4. Information needed for each of the functions available in the document
5. Lastly, automatic document creation programs

**13. What is an API framework?**

The API framework is easy to understand. During the process, the config file is used to hold the configurable parts as well as to value the test run. Besides, within the config file, automated test cases should be represented in the format of a parse table. During the process of API testing, it is not mandatory to test each API as a result the config file contains some sections whose API is activated for all that specific run.

**14. What do you mean by input injection? Explain different ways of doing it.**

The term Input injection is the act to stimulate user input. User input can be simulated in many different ways such as:

1. Direct Method Invocation
2. Invocation with the help of accessibility interface
3. Doing simulation with the help of low-level input
4. Doing simulation with the help of a device driver
5. Doing simulation with the help of a robot

**15. Define API testing with Runscope.**

To test APIs, Runscope is used. It is basically a web application providing backend services as well as an easy-to-use interface.

**16. Explain the major challenges that come while API testing.**

The list of major challenges that come while API testing is:

1. Parameter Combination
2. Parameter Selection and
3. Call Sequencing

**17. What are the main principles of API test design?**

There are various principles of API test design. Those are as follows:

1. Setup: this includes the creation of objects, start services and initialize data, etc.
2. Execution: during this principle, there are steps to follow API or scenario as well as logging
3. Verification: for evaluating the execution outcome there are oracles
4. Reporting: keep a tab on the pass, blocked or failed
5. Clean up: this shows the pre-test state

**18. Explain the types of bugs that can be found using API testing?**

API is capable of finding many types of bugs that includes:

1. Stress
2. Security
3. Duplicate or missing functionality
4. Reliability
5. Unused flags
6. Incompatible error handling
7. Multi-threaded issues, and
8. Improper errors

**19. Name various tools used for API test automation.**

While doing API testing and Unit testing, both targeting source code. In case an API method is making use of code based on .NET then other tools that are providing support must have .NET

There are various automation tools for API testing:

1. HP UFT
2. Soap UI
3. JUnit for Java
4. NUnit for .Net

**20. What is the API documentation?**

For any foundation, there is always a need for good documentation. Similarly, API documentation provides a quick reference to access working or library within a program.

While walking through any such documents, a proper plan is a must along with a proper sketch or layout for delivery, there is a need for the content source, information regarding each and every function, etc.

There are various API documentation tools like Doxygen and JavaDoc. Below you can check the various categories in which each and every function is being documented that are revolving around the parameters like:

1. Function description
2. Type and syntax of the error message that may occur
3. Syntax, elements, and sequence needed for each parameter
4. Links regarding functions

**21. List some templates for API documentation that are most used.**

There are various API documentation templates that are making the whole process really simple and easy. Check them below:

1. Swagger
2. Miredot
3. Slate
4. FlatDoc
5. API blueprint
6. RestDoc
7. Web service API specification

**22. Explain the difference between API testing and Unit Testing.**

* Where Unit testing is a form of white-box testing, API testing is a form of black-box testing.
* Unit testing is performed prior to the process of including the code in the build. On the other hand, API testing is done after the build is prepared for testing.
* In Unit testing, the source code is drawn in the form of testing while in API testing the source code is not drawn in.
* In Unit testing, there is a limited scope of testing as a result only basic functionalities are measured for the purpose of testing. Subsequently, in API testing there is a wide scope of testing, thus all the issues that are functional are measured for the purpose of testing.
* Unit testing is done by the testers and wherein every functionality is separately tested. While The API testing is done by the testers for the purpose of end-to-end testing of the whole functionality.

**23. Define TestApi?**

TestApi can be explained as the test building blocks library which is indispensable for testers and developers to create testing tools and automated test suites.

**24. Explain everything about warnings and API errors.**

When something is not going as per expectations like when the outcome is not as predicted then the occurrence of errors can be seen and the same warnings are explained in the form of a message in a proper format. Within a single module, there can be one or many warnings.

A wide range of warnings that can form are:

Missing module warning and parameter validation warning

A wide range of errors that can form are:

**25. Explain the working of API Builder.**

API Builder is a PLSQL utility that includes 4 square files. To place API parameters and to begin the technique only one report is liable. API builder allows you to create and make use of API endpoints that can be guzzled by any client application.

There are several components that lead to the making of API Builder. During the working, files and formed brief tables as well as master bundles for creating the output code. Lastly, the fourth record generates a spooled output of the code into a record relating to output\_script\_.sq.

**API Testing Interview Questions For Experienced**

**26. What are the benefits of API testing?**

* Provides application access without the user interface
* Provision for easy test maintenance
* Less time for resolution
* Speed and coverage of testing
* Protects from malicious code and breakage
* Cost-effective/ reduces testing cost
* Technology independent

**27. What are the challenges faced in API testing?**

Just like other software testing techniques, API testing also faces some challenges like:

* The main challenge is sequencing API calls.
* Strong coding knowledge needed for testers.
* No GUI is available to test the application, which makes it difficult while giving inputs.
* Testers must be aware of parameter combinations and validations.
* Exception handling functions also be tested.
* Validating and verifying different systems is difficult for testers.

**28. Explain how to document each function in the API document?**

Description: Small description of what a function does

Syntax: Syntax about the parameter of the code, the sequence in which they occur, required and optional elements, etc.

Parameters: Functions parameters

Error Messages: Syntax of error messages

Example Code: Small snippet of code

Related Links: Related functions

**29. What are the tools used for API documentation?**

The free tools used for API documentation are ReDoc, Swagger [UI](https://mindmajix.com/ui-developer-training), and DapperDox.

**30. What’s the difference between API and Web Service?**

|  |  |
| --- | --- |
| **API** | **Web Services** |
| API is a set of protocols and definitions which allow one application to interact with another application. | A web service is a way for two machines to interact with each other over a network. |
| API can interact through REST, SOAP. CURL, and XML-RPC calls as well.  Also, through DLL, JAR, XML over HTTP, JSON over HTTP, etc. | A web service uses three styles for communication, such as SOAP, REST, and XML-RPC. |
| All APIs are not web services. | All web services are APIs |
| APIs don’t need a network for operation | Web services always need a network for operations |
| API can perform all the operations which web service can't achieve. | Web services cannot perform all the tasks that API would perform. |

**31. What are the architectural styles used for creating a Web API?**

* HTTP for client-server communication
* Stateless communication
* Simple URI as the address for the services
* XML/JSON as a formatting language

**32. How to perform API testing?**

API testing is a type of integration testing used to test API to validate the functionality, performance, and security of the application.

API testing should perform the following testing methods:

* Discovery testing - This testing manually executes the set of calls documented in the API.
* Usability testing - It verifies whether the API is functional, user-friendly, and does API integrates with another platform or not.
* Automated testing - It comes while creating a set of scripts or a tool to execute API regularly.
* Security testing - It recommends what authentication type is needed and also checks sensitive data encryption.
* Documentation - It's the final phase for a deliverable. The team makes sure the documentation provides enough data to interact with the API.

**33. Why is API testing the most preferred for Automation testing?**

API testing is considered most suitable for automation testing because:

* It effectively verifies all the functional paths of the system under test.
* Provides faster feedback.
* It presents the most stable interface.

**34. Name a few API documentation templates?**

There are several API documentation templates that make the entire process simple, leverage best practices, and will make API users satisfied. They are:

* RestDoc
* RAML
* Miredot
* Calamum
* Swagger
* API Blueprint
* Web Service API Specification Doc Template
* FlatDoc
* apiDoc
* Slate

**35. What to be considered to create great API documentation?**

* Plan your docs
* Include fundamental sections
* Be consistent and avoid jargon
* Include interactive examples and other resources
* Maintain your docs
* Delivery layout
* Information needed for every function in the document
* Automatic document creation programs

**36. What are the differences between SOAP and REST API?**

|  |  |
| --- | --- |
| **SOAP** | **REST API** |
| SOAP stands as Simple Object Access Protocol. | REST stands as Representational State Transfer. |
| It’s largely based and uses only HTTP and XML | It supports different data formats such as HTML, plain text, JSON, XML, and more. But the most preferred format to transfer data is JSON. |
| It’s a protocol | It’s an architectural pattern |
| SOAP uses WS-security and SSL( Secure Socket Layer) for security | On the other hand, REST has SSL and HTTPS for security. |

**37. What is messaging in RESTFUL Web Services?**

RESTFUL Web Services uses HTTP protocol as a source of communication between client and server. The technique when a client sends a message in the form of an HTTP request, and the server responds in the way of an HTTP response is called Messaging. These messages comprise metadata and message data, i.e., information related to the message itself.

**38. What are the main components of an HTTP request?**

* Action showing HTTP methods like PUT, GET, DELETE, POST.
* Uniform Resource Identifier (URI), which is the identifier for the resource on the server.
* HTTP version which represents the HTTP version like- HTTP V1.1.
* Request Header used for carrying metadata to the HTTP request message.
* Request Body describes resource representation or message content.

**39. Which HTTP protocols are supported by REST?**

* GET - Requests data from the defined resource.
* PUT - Replaces the current representation of the target resource with the request payload.
* POST - Sends data for a server to create or update the resources. POST requests are never cached or bookmark.
* OPTIONS - Specifies the communication option for the target resources.
* DELETE - Removes the specified resource.
* HEAD - HEAD requests for a response that is similar to GET requests, but without the response body.

**40. What is URI? What is the purpose of a web-based service, and what is its format?**

Uniform Resource Identifier (URI) is a string of characters used for unambiguous identification of resources and extensibility through the URI scheme.

The purpose of this web-based service is to locate a resource on server hosting.

A URI’s format is <protocol>://<service-name>/<ResourceType>/<ResourceID>.

**41. Define the caching mechanism.**

A caching mechanism is a practice to store data temporarily and retrieve data from a high-performance data store either implicitly or explicitly.

Caching mechanism improves performance by copying the asset requested and obtaining the cached copy instead of the original later.

**42. What’s the difference between PUT and POST operations in Rest API?**

|  |  |
| --- | --- |
| **PUT** | **POST** |
| The PUT method is a call when you have to modify a single resource, which is part of resource collection. | POST method is a call when you have to add a child resource under resource collection. |
| The PUT method is idempotent | POST method is not idempotent |
| PUT for UPDATE operations. | POST for CREATE operations. |
| If the PUT request is used more than one time, the results will remain the same. | If a POST request is used multiple times, the results will be different. |
| PUT works as specific. | POST work as abstract. |

**43. Can we use GET requests instead of PUT to create a resource?**

PUT or POST are used for creating resources. GET is used only for requesting data from a specified resource.

**44. What are the commonly used HTTP methods for RESTful services?**

* **GET -** Retrieves data from a server at the specified resource.
* **HEAD -**Works the same as the GET method, but the server replies without the body.
* **POST-** Creates a new resource
* **PATCH -** Allows partial modifications to a resource
* **PUT -** Replaces all current representations of the target resource
* **DELETE -** Removes the defined resource
* **OPTIONS -** Returns the HTTP methods supported by the server for the specified URL

**45. What is Payload in REST API?**

The Payload in REST API is the actual data pack that is sent with the GET method in HTTP. It’s the crucial information that you submit to the server when making an API request.

The payload is denoted using “{}” in a query string, and it can be sent or received in multiple formats.

**What is API testing?**

API testing is a type of software testing that involves evaluating APIs (Application Programming Interface) to see if they meet the requirements for functionality, dependability, performance, and security. Since APIs don’t have a GUI, API testing is performed at the message layer of the system.

**2. What are the advantages of performing API testing?**

API testing has several advantages. Among the most important, you can mention:

* *Test without GUI*: Testers can conduct API tests without having to use the software directly. This is a huge benefit because it gives QA engineers early insight into flaws and faults, allowing developers to fix them before they have an impact on the GUI.
* *Test for core functionality:* Before performing GUI tests, testing an application's code-level functionality allows for an evaluation of its overall build quality. This helps reveal little errors that may grow into more significant issues at the GUI level. Core access makes it possible to do testing concurrently with development, enabling communication and better teamwork.
* *Time effective:*API tests typically take less time than functional GUI testing. GUI testing takes longer because the web components need to be polled. API test automation particularly involves less code and can therefore give better and faster test coverage, when compared to GUI automated testing.
* *Language dependent:* An API test uses XML or JSON to exchange data. These transfer modes are not language-dependent, therefore you can use any programming language when writing automated tests for your API.

**3. How is API testing different from UI testing?**

API testing focuses a lot more on testing the business logic, data replies, and security, as well as performance bottlenecks. In contrast, UI testing focuses on verifying the look and feel of a web interface or that certain buttons, forms, dropdowns, etc work.

**4. What are the components of an HTTP request?**

An HTTP request has five elements:

* An HTTP method (discussed below) that defines the action.
* A URI (Uniform Resource Identifier), which is the resource’s identifier on the server.
* An HTTP Version, e.g. HTTP v1.1.
* The Request Header, which carries metadata (as key-value pairs) for the HTTP Request message. Client (or browser) type, client-supported formats, message body formats, cache settings, and other information are examples of metadata.
* The Request Body, which represents the data sent by the client to the API.

**5. What are the most used HTTP methods in REST APIs?**

The most important HTTP methods used when performing REST API testing are the methods that perform [CRUD](https://en.wikipedia.org/wiki/Create,_read,_update_and_delete)operations:

* GET is the HTTP method that reads the information from the resource.
* POST method is used to create or update resources.
* PUT modifies an existing resource.
* DELETE will remove a specified resource.

**6. What is the difference between the PUT and the POST methods?**

This is an interview question I got at a lot of jobs, and the answer is partly answered above.

When you need to change a single resource that is a component of a resource collection, you call the PUT method. When you need to add a child resource to a resource collection, you must use the POST method. If the PUT HTTP call is sent more than once, the results will stay the same. If a POST request is sent multiple times, then the results will differ, i.e. multiple resources might be created or an error is returned.

For example, if you have a resource for creating and updating users, sending the same PUT method for a user will update the user each time. Sending the same POST method for a user will result either in multiple users created, or in an error that the username or email address is already in use.

Discover what's new in the QA world!

[Subscribe now](https://theqalead.com/go/newsletter-placeholder-display-ad-takeover/)

**7. What are the HTTP response status code classes?**

Another one of the common interview questions, and also something important to know when performing API testing. The HTTP response code classes are

* 1xx: the response calls in this category are informational responses. They mean that the client should continue the request or ignore the response if the request is finished.
* 2xx: a 200 code means Success.
* 3xx: these responses are redirect responses. This means that there are multiple possible responses to the request. One of them should be selected by the user agent or user.
* 4xx: the codes in this group denote a client error. This means that the server cannot process the request and it perceives it as an error from the client side, such as an unrecognized URL, an incorrect request syntax, and so on.
* 5xx: the 500 HTTP response code is returned when there is an error on the server side and the server is unable of performing the request.

If you want to get into the details of the response statuses, you can find the complete list [online](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes).

**8. What are some common API automation testing tools?**

For this question, I would answer with some tools I have already worked with or am at least a little familiar with. So, if you have experience with any API testing tools, go ahead and mention them. If not, you can answer with some of the popular ones, such as Katalon, Postman, or SoapUI. You can take a look at this [article](https://theqalead.com/tools/best-api-testing-tools/)for some inspiration, too.

**9. What are some commonly used authentication methods in API testing?**

An appropriate answer for this question would be

* Session/Cookies based authentication
* Basic authentication
* Digest authentication
* OAuth.

**10. What is the difference between authentication and authorization?**

In short, authentication is the process of verifying a user's identity, whereas authorization is the process of confirming their level of access.

**11. Why is API testing preferred to UI testing when it comes to automated tests?**

Coming back to the classical test automation pyramid, it is well known in our industry that UI end-to-end tests should be at the top of the pyramid, meaning that they should account for the least number of tests. This is because UI automated tests tend to take more time, and are more prone to flakiness because they have many dependencies. API automated tests represent the integration testing part of the pyramid, and they are a lot faster and usually more reliable.

**12. What is the difference between API and unit testing?**

Unit testing falls under the white box testing category, while API testing is usually black box testing. Since an end-user will engage with the user interface, API testing must represent the system as a whole. In unit testing, a key consideration is whether each component or module functions flawlessly on its own. That is, to achieve a solid module architecture, dependencies should be minimized.

**13. What types of testing can be applied to APIs?**

Most of the testing types applied to UI testing work on APIs at well. A few of the most notable testing types you can mention for this API interview question are

* Functional testing: most of the time, you will want to test that the APIs do what they are designed to do. This means that you’ll be running functional test cases on APIs.
* Manual testing: just because you are not an automation tester doesn’t mean you can’t test APIs. You can use tools such as [Postman](https://www.postman.com/)to send requests and test the responses manually.
* Automated testing: It’s a good idea to automate the API test cases if you can. A lot of the tools mentioned above can help you with that, or you can create your own API framework.
* Load testing: By simulating traffic to APIs, testers can identify bottlenecks before they get into production. In the absence of a production load, it might be challenging to identify these bottlenecks in development environments. There are tools, such as Apache JMeter,  that enable you to send HTTP calls to a given endpoint and measure the response time, errors and error rates, and other useful data from the responses. They can also help simulate large amounts of data to evaluate how an application behaves.
* Security testing: with security testing, the API implementation is protected from outside threats. Phases in security testing include verifying encryption techniques and the architecture of the API access control. User access management and authorization verification are also included.
* Penetration testing: With this type of testing, users unfamiliar with the API will attempt to evaluate the threat vector from a distance, focusing on functionalities, resources, workflows, or the complete API and its components.

**Q #1) What is API Testing?**

**Answer:** API is a collection of routines, tools, protocols that together are required for building the software application. Any system software or application software which consists of multiple APIs can perform Application Programming Interface (API) testing.

This form of testing includes interaction between various or says multiple APIs as well as the interaction between API and application program. The procedure mainly includes making API calls using software and observing system response after receiving the output.

**Q #2) Enlist some common tests that are performed on APIs.**

**Answer:**There can be multiple reasons for performing API testing.

**Let us see some common test examples, where this form of testing is used to verify:**

* Any data structure updated by API which requires proper validation.
* Input condition-based return values.
* Call to another API or if any other event is triggered or some interruption is raised.
* The return values can also be null or wrong results.
* Modification of some resources like an update of the database, process killing, etc.

**Q #3) What is the API test environment?**

**Answer:**Setting up a test environment of API is a complex method where the configuration of the server and database is done as per the requirement of the software application. Graphical User Interface (GUI) is not available in this form of testing.

After installation, API is verified for its proper functioning. In this process initial environment that invokes API is being set up with a defined set of parameters so that test results can be examined.

**Q #4) Explain the API testing approach.**

**Answer: Mentioned below are the factors which determine the approach:**

* Write appropriate test cases for the APIs and use testing techniques like boundary value analysis, equivalence class, etc. for verifying the functionality.
* Verify the calls of the combination of two or more value-added parameters.
* Define the scope and basic functionality of the API program.
* Define the accurate input parameters.
* Test case execution and comparison of the results with expected results.
* Determining API behavior under conditions like the connection with files, etc.

**Q #5) Explain in brief the different types of output observed of an API.**

**Answer:**API is considered as the essential connecting part of this digital world. It basically resides in the business logic layer where it performs functions like processing commands, application coordination, initiates logical decisions, etc.

The main consideration is returning correct results under any type of conditions. Mainly, the output or results observed of an API are divided into three sections as follows:

* Returning the result status values as ‘Pass’ or ‘Fail’.
* Result as data or any specific information.
* An event where the call to any API function will initiate the call to another API function.

**Q #6) Enlist some best practices that are followed to make API testing successful.**

**Answer:**Performing tests repeatedly define some best practices for making testing successful.

**Enlisted below are some best practices for API testing:**

* Test cases should be grouped under category with expected results that happen consistently and other typical results.
* Test cases should include selected parameters as well as API call declarations.
* API load tests are performed to determine system application stress.
* Maintain the limits of the variables used in the tests as well as avoid ‘Test Chaining’.
* To make ease for the testers, API call is being prioritized and call sequencing is planned.
* Every input combination and dependencies are considered for complete test coverage.
* Automation of the test cases, documentation is done as and when required.

**Q #7) What are the tools used for API testing?**

**Answer: Best API Testing tools:**

* [SOAPUI](https://www.softwaretestinghelp.com/web-services-api-testing-tool-soapui-tutorial-1/)
* Runscope
* LOADUI
* Automated API testing
* Curl

There are a few more others than the above-listed tools that are used for API testing.

**Q #8) What are the tools used for API test automation?**

**Answer:**Automation testing is a must when we talk about agile development in API testing. However, the language in which the code is written is also an important factor as it decides the tool language.

**Some important API test automation tools are:**

* **SOAPUI:** It is an open-source API testing tool which is considered as the best testing tool because of its feature like creating complex validation scripts and test cases, efficient test coverage, etc.
* **HP QTP/UFT:** This is now known as HP UFT i.e. Unified Functional Testing. This tool is basically used for systems without user interface like web services, etc.
* **PARASOFT:** This testing tool runs on various platforms and is used to test API which does not have a Graphical User interface (GUI).
* HTTP master
* NUnit and JUnit testing tools are used where the code is written in .Net and Java respectively.

**Q #9) What is the API framework?**

**Answer:**API framework is described by the config file which consists of the list of all APIs that are required to be activated and are activated for any particular program run. This is essential as every test run does not require all APIs.

The purpose of the ‘Config’ file is to describe and enlist every configurable component within a test run.

**Q #10) Explain API documentation.**

**Answer:**As it is a well aware fact that, for any foundation, there has to be good documentation. API documentation likewise, serves as a quick reference for accessing the library or working within a program.

When we go through any such documents, it must consist of a proper plan, content source, proper layout or sketch for delivery, information related to each function, etc.

**API documentation tools are:**

* JavaDoc
* Doxygen

**Enlisted below are the categories in which every function is being documented which mainly revolve around the parameters:**

* Function description
* Sequence, syntax, and elements required for each parameter.
* Syntax and type of error message that can occur.
* Links related to functions.

**Q #11) Name some most used templates for API documentation.**

**Answer: Some free templates which makes API documentation much easier and simple are:**

* Slate
* FlatDoc
* Swagger
* API blueprint
* RestDoc
* Miredot
* Web service API Specification.

**Q #12) Enlist some of the API examples which are very well known and popular.**

**Answer:**There are several such examples.**Enlisted below are some most popular ones:**

* **Google Maps API:** These are designed mainly for mobile and desktop use with the help of a flash interface and JavaScript.
* **Amazon Advertising API:** Amazon is known for their products and thus their advertising API accesses their product to discover their functionality and thus advertise accordingly.
* **Twitter:** The API for twitter is usually in two categories, one for accessing data and the other for interacting with the twitter search.
* **YouTube:** This API used for YouTube includes various functionalities including videos, live streaming, player, etc.

**Q #13) What are the testing methods that come under API testing?**

**Answer: API testing generally involves the following testing methods:**

* Unit testing and Functional testing
* Load testing for testing the performance under load.
* Discovery testing for listing, creating and deleting the number of calls that have been documented in API.
* Usability testing and Reliability testing for obtaining consistent results.
* Security testing and Penetration testing for validating all types of authentication.
* Automation testing for creating and executing scripts that require API calls execution regularly.
* End to end Integration testing and Web UI testing.
* API documentation testing for determining its efficiency and effectiveness.

**Q #14) Differentiate API testing and Unit Testing.**

**Answer:**The difference between API testing and Unit testing can be understood from the below table:

| **UNIT testing** | **API Testing** |
| --- | --- |
| Unit testing is usually performed by developers where every functionality is tested separately. | API testing is performed by the testers for end to end testing of the functionality. |
| As they have the limited scope of testing, thus basic functionalities are only considered for testing. | As they have the broader scope of testing, all issues that are functional are considered for testing. |
| It is a form of white box testing. | It is a form of black box testing. |
| Usually, unit testing is done before the code is included in the build. | API testing is performed after the build is ready for testing. |
| The Source code is involved in this form of testing. | Source code is not involved in this form of testing. |

**Q #15) What challenges are included under API testing?**

**Answer:**Challenges are the part of every form of testing and the same goes with API testing too.

**Mentioned below are some common challenges that are faced in API testing:**

* The first and foremost challenge is selecting an appropriate parameter and then its combination.
* Parameter categorization
* Proper sequencing of call is required as this may lead to inadequate coverage in testing.
* Output verification and validation
* Another important challenge is providing input values, which is very difficult as GUI is not available in this case.

**Q #16) What are the types of issues observed while performing API testing?**

**Answer:**When testing is performed, then there have to be issues associated with them. Issues observed while performing this form of testing are not new or much different but they are common in this category.

**Find below the list of such issues/defects:**

* Inconsistent or absence of error handling mechanism
* Repetition or redundancy of the functionalities
* Missing required functionality in some cases
* Passing incorrect argument to the input values
* Improper messaging
* Stress and performance issues
* Reliability issues with respect to connection with other APIs
* Multithreading and improper handling issues.

**Q #17) Why API testing is determined as the most suitable form for Automation testing?**

**Answer:**Yes, it’s true that API testing is now preferred over GUI testing and is considered as most suitable.

**Below are the few reasons behind this statement.**

* Verify all the functional paths of the system under test very effectively.
* Provides the most stable interface.
* Easier to maintain and provides fast feedback.

**Q #18) How is UI level testing different from API testing?**

**Answer:**The main consideration of the UI (User Interface) level testing is to test the graphical interface part of the application include features like font, layout, etc.

Whereas, the main consideration of the API testing is establishing communication between different software systems and it mainly resides in business logic layer. It never concentrates on the look of the application.

**Q #19) What is TestApi?**

**Answer:**TestApi is known as the library of test building blocks which are essential for developers and testers for creating testing tools as well as automated test suites.

**Q #20) What do you know about API errors and warnings?**

**Answer:**When something goes wrong i.e. the outcome is not as expected then the error occurs and warnings are described as a message in the proper format. There can be one or multiple warnings within the same module.

**Different types of warnings that can occur are:**

* Parameter validation warning
* Missing module warning

**Different types of errors that can occur are:**

* Documentation errors
* Missing module errors
* Parameter validation errors
* Some standard error messages.

#### **What is an API?**

API (Application Programming Interface) is a software interface that connects two or more applications to exchange data. It improves an organization's productivity and keeps data in sync.

#### **2. What is API testing?**

API testing validates the functionality, behavior, security, and performance of the application. It ensures that the developed APIs meet expectations.

#### **3. Is API considered a software?**

API is not a software but rather an interface to provide data exchange and functionality among different software applications or processes. While an API provides access to data as well as functionality, it can be thought of as software that fulfills our needs, too.

#### **4. Name the five important principles of an API design.**

The five important principles of API design are:

**Setup**: Create objects, start services, initialize data, etc.  
**Execution**: Steps to apply API or the scenario, including logging  
**Verification**: Oracles to evaluate the result of the execution  
**Reporting**: Pass, failed, or blocked  
**Clean up**: Pre-test state

#### **5. Name some of the types of API testing.**

Some of the most common API types are:

* Unit Testing
* Functional Testing
* Load Testing
* Runtime/Error Detection
* Security Testing
* UI Testing
* Interoperability and WS compliance Testing
* Penetration Testing
* Fuzz Testing

#### **6. What are some commonly used authentication techniques in API testing?**

Some of the most common authentication techniques are: Session/Cookies based authentication, basic authentication, digest authentication, and OAuth.

#### **7. What are some common tools you can use for API testing?**

There are many tools available for API testing. Here is a list of the most common:

* Postman
* SoapUi Pro
* Apigee
* JMeter
* API fortress

#### **8. Share some of the advantages of API testing.**

The advantages of API testing are:

* API testing is less time-consuming than functional testing.
* It is cost-effective.
* It is language-independent and time-effective.

#### **9. What are the challenges faced in API testing?**

The challenges faced in API testing are:

* API chaining or sequencing the API calls
* Testing parameter combinations
* Frequent Schema changes
* Access to the database

#### **10. What is the procedure for performing API testing?**

The API testing steps/procedures are:

1. Select the test case that has to be fulfilled
2. For API calls, develop a test case
3. To meet the test case, configure the API parameters
4. Determine how will you validate a successful test
5. Using programming languages like PHP or .NET, execute the API call
6. Allow the API call to return the data to validate

#### **11. What must be checked when performing API testing?**

In order to perform an API test you must check the following:

* Accuracy of data
* Schema validation
* HTTP status codes
* Data type, validations, order, and completeness
* Authorization checks
* Implementation of response timeout
* Error codes in case API returns
* Non-functional testing like performance and security testing

#### **12. Differentiate API testing from UI testing.**

API testing allows communication between two software systems. It determines if the developed APImeets the expectation regarding functionality, reliability, performance, and security. It works on the backend and also knows the backend testing.

[UI (User Interface) testing](https://blog.hubspot.com/website/ui-design?_ga=2.153240312.112504332.1656682773-1810929624.1656682773&hubs_content=blog.hubspot.com/website/api-testing-interview-questions&hubs_content-cta=UI%20(User%20Interface)%20testing%20)means testing the graphical user interface. The focus of UI testing is on the look and feel of the application like how the user interacts with the application elements, such as images, font, layout, etc.

#### **13. What is API documentation?**

An API documentation serves as a quick reference for accessing the library or working within a program.

#### **14. What is API automation?**

We often need to automate the test cases which are repeatedly executed in every sprint like regression cases. Similarly, in the case of API testing, there are some cases that we need to execute before every release and those cases should be automated.

There are many tools for API automation like:

* SOAPUI
* Katalon studio
* Postman
* Jmeter
* RestAssured
* CloudQA TruAPI

#### **15. Name some of the common API documentation templates.**

The most common API documentation templates are: Swagger, RestDoc, FlatDoc, Slate, Web Services API Specification, API Blueprint, and Miredot.

#### **16. What is TestApi?**

TestApi is known as the library of test building blocks which are essential for developers and testers when creating testing tools as well as automated test suites.

#### **17. What kinds of bugs does API testing find most commonly?**

The types of bugs that APIs will find are:

* Missing or duplicate functionality
* Fails to handle error conditions gracefully
* Stress
* Reliability
* Security
* Unused flags
* Not implemented errors
* Inconsistent error handling
* Performance
* Multi-threading issues
* Improper errors

#### **18. Describe some of the types of status codes.**

Some status codes are:

* 1xx informational response – the request was received, continuing process
* 2xx successful – the request was successfully received, understood, and accepted
* 3xx redirection – further action needs to be taken to complete the request
* 4xx client error – the request contains bad syntax or cannot be fulfilled
* 5xx server error – the server failed to fulfill a valid request

#### **19. What are the most common API/HTTP methods?**

The most used HTTP methods are GET, POST, PUT, PATCH, and DELETE.

#### **20. What is the difference between PUT, POST, and PATCH?**

**PUT**: Put request is used for both creating and updating a new object in the database. If the resource already exists, then Put will update the resource. If not, it will create one.

**POST**: Post request is used for creating a new object in the database. It allows clients to create resources without knowing the URI of the new resources.

**PATCH**: Patch is used to apply the partial modification to a resource.

#### **21. What should the Delete request return?**

Delete request returns the HTTP status code 200(OK) if the response contains an entity describing the status. If the response does not include an entity then it will return 204(No content) and we will get 202(Accepted) if the action has been queued.

#### **22. What is payload?**

Payload/body is a secured input data that is sent to API to process the request. The payload is generally constructed in JSON format in REST API.

#### **23. How can we pass dynamic data for a request?**

We can write a pre-request script to pass dynamic data or we can create a collection and pass dynamic data using a CSV file.

#### **24. What is the difference between API testing and Unit Testing?**

Unit testing is white-box testing while API testing is black-box testing. Unit testing is used to verify that each unit in isolation works as expected while API testing is used to assure full functionality of the system.

#### **25. What do you understand about browser APIs?**

Browser APIs are built-in with the browsers. They enable developers to implement complex operations without administering the sophisticated lower-level code.

#### **26. What is REST?**

[REST](https://blog.hubspot.com/website/what-is-rest-api?_ga=2.159025914.112504332.1656682773-1810929624.1656682773&hubs_content=blog.hubspot.com/website/api-testing-interview-questions&hubs_content-cta=REST) stands for Representational State Transfer. REST is an architectural style for web development. REST architecture lays out guidelines for transferring resource representations between clients and servers on the web.

#### **27. How is restful API implemented?**

The implementation consisted of running the code from the JUnit tests into the APIs and then refreshing the tests to summon those APIs. The modifyCertificate method, which gives the implementation for the certificates resource PUT method, is the most difficult REST API to implement.

#### **28. What are some key characteristics of REST?**

Key characteristics of REST include:

* REST is stateless, therefore the SERVER has no state (or session data).
* With a well-applied REST API, the server could be restarted between two calls as every data is passed to the server.
* Web service mostly uses POST method to make operations, whereas REST uses GET to access resources.

#### **29. What are the HTTP methods supported by REST?**

* **GET**: This requests a resource at the request URL. It should not contain a request body as it will be discarded. It can be cached locally or on the server.
* **POST**: This submits information to the service for processing; it should typically return the modified or new resource.
* **PUT**: At the request URL, this updates the resource.
* **DELETE**: At the request URL, this removes the resource.
* **OPTIONS**: This indicates which techniques are supported.
* **HEAD**: This returns meta-information about the request URL.

#### **30. What is the most popular way to represent a resource in REST?**

JSON is the most popular and important way to represent resources.

#### **31. Explain caching in Rest API.**

Caching is used for network optimization by reducing the load on servers. It is the ability to store copies of frequently accessed data. Get Requests are by default cacheable, however, Post can be made cacheable.

#### **32. What is a Pre-Request script in Postman?**

In short, a pre-request script is a script that runs before the execution of a request.

#### **33. What is the importance of setNextRequest in Postman?**

setNextRequest is used to define the workflow of API testing. setNextRequest is needed to control the order of the execution of requests.

#### **34. What are the two types of scripts in Postman?**

Two types of scripts in Postman are test script and pre-request script.

#### **35. Which type of communication API works in IoT?**

REST is the most popular IoT Communication APIs.

#### **36. Define SOAP.**

[SOAP](https://blog.hubspot.com/website/rest-vs-soap?_ga=2.159025914.112504332.1656682773-1810929624.1656682773&hubs_content=blog.hubspot.com/website/api-testing-interview-questions&hubs_content-cta=SOAP) stands for “Simple Object Access Control,” and it is an XML based protocol for exchanging information between computers.

#### **37. Why is SOAP more reliable than REST?**

SOAP is more reliable than REST as it practices WS-Security for transmission with Secure Socket Layer. Also, SOAP is state-full as it takes the request as a whole, unlike REST which gives independent processing of various methods. No independent processing is present in SOAP.

#### **38 . What is the major drawback of using SOAP?**

When using SOAP, users often experience firewall security mechanisms as the biggest obstacle. This blocks all the ports, leaving a few like HTTP port 80 and the HTTP port used by SOAP that bypasses the firewall. The technical complaint against SOAP is that it mixes the specification for message transport with the specification for message structure.

#### **39. Can we automate API testing?**

Yes, we can automate API testing by using automation testing tools for API testing.

#### **40. Is JMeter good for API testing?**

Yes, JMeter is good for API testing, especially REST API. JMeter is an open source tool used for performance or load testing.

#### **41. Can GET request be used instead of PUT to create a resource?**

The PUT or POST method should be used to create a resource. GET is only used to request data from a specified resource.

#### **42. What is URI? What is the main purpose of REST-based web services and what is its format?**

[URI](https://blog.hubspot.com/website/uri-vs-url?_ga=2.159099770.112504332.1656682773-1810929624.1656682773&hubs_content=blog.hubspot.com/website/api-testing-interview-questions&hubs_content-cta=URI) stands for Uniform Resource Identifier. It is a string of characters designed for unambiguous identification of resources and extensibility via the URI scheme.

The purpose of a URI is to locate a resource(s) on the server hosting of the web service.

A URI’s format is <protocol>://<service-name>/<ResourceType>/<ResourceID>.

#### **43. What are the syntax rules for a SOAP message?**

The syntax rules for a SOAP message include the following:

* Must use encoded XML
* Envelope namespace must be used
* Encoding namespace must be used
* Must not consist of a DTD reference
* Must not have XML processing instruction

#### **44. What is Run Scope?**

Run Scope is an API testing tool that is typically a web application supporting an easier user interface platform to test back-end services.

#### **45. Is it possible to hack API while testing?**

Yes, it is possible. This is because we are sending requests over the internet which mostly follows HTTP protocol. This protocol is text-based and is easier to read. Hence, it is required to perform security testing of the APIs to ensure safer systems.

#### **46. How should we test the API security?**

To test the security of the API during API testing, we need to validate two components:

1. **Authentication:** Whether the identity of the end-user is correct
2. **Authorization:** Whether the user is allowed to access the resource

We can also validate whether the TLS or the [SSL certificate](https://blog.hubspot.com/website/get-ssl-certificate-wordpress?_ga=2.159099770.112504332.1656682773-1810929624.1656682773&hubs_content=blog.hubspot.com/website/api-testing-interview-questions&hubs_content-cta=SSL%20certificate) used over the HTTPS protocol is valid or not.

#### **47. What helps hide the distinction between different micro-services?**

The API Gateway hides the distinction between different micro-services.

#### **48. What is the most commonly used command-line tool to explore API?**

The most commonly used command-line tool to explore API is cURL.

#### **49. When a user attempts to access records that do not belong to them, what should the API return in the response status code?**

The API should return the 401 status code.

#### **50. What is the kind of API traffic that is internal to the organization?**

The Internal Traffic is internal to the organization.

#### **51 Which among the following options is used for requesting JSON instead of XML from API?**

The Use Accept Header would be used for requesting JSON instead of XML with API.

#### **52. Which options help to identify the type of API requests?**

The HTTP Methods help identify the type of API requests.

#### **53. What category does API testing belong to?**

API testing is generally black-box testing. We don't look at what happens behind the API server. We only validate the responses.

#### **54. Which markup language can be used in Restful Web API?**

[JSON](https://blog.hubspot.com/website/json-files?_ga=2.159099770.112504332.1656682773-1810929624.1656682773&hubs_content=blog.hubspot.com/website/api-testing-interview-questions&hubs_content-cta=JSON) and XML are the two markup languages that can be used in Restful Web API.

#### **55. What is the upper limit for a payload to pass in the POST method?**

GET appends data to the service URL. But, its size shouldn’t exceed the maximum URL length. However, POST doesn’t have any such limit.

So, theoretically, a user can pass unlimited data as the payload to POST method. But, if we consider a real use case, then sending POST with a large payload will consume more bandwidth. It’ll take more time and present performance challenges to your server. Hence, a user should take action accordingly.

**What does API mean?**

During your technical interview, the interviewer may start things off with questions that seem easy. An ice-breaker question can be an opportunity for you to shake off some nerves while also answering the question with details that may catch the interviewer’s attention. This might be a recent milestone you hit or a fun project you’re working on.

For this question, you should answer by first defining the acronym [API](https://www.codecademy.com/resources/blog/what-to-know-about-apis/?utm_source=ccblog&utm_medium=ccblog&utm_campaign=ccblog&utm_content=cw_api_testing_interview_questions_blog), which is Application Programming Interface, and then you can outline what this means. Specifically, you should mention that an API is a set of functions and procedures that allows a program to access features or data from another piece of software or service. As a result, it enables multiple applications to interact with each other.

**2. What is API testing?**

This is another ice-breaker question, and it gives you a chance to show that you know some of the “why” behind API testing. You should discuss how an API test checks to see if the function being called is behaving as expected. During an API test, a function is tested against a wide range of inputs to see if it returns the expected value from each test. Edge cases and corner cases are included in the test to ensure that the function can handle extreme cases. API testing is generally done at the end of development to check and see if the software is performing as expected.

**3. What are some of the most common protocols used during API testing?**

This is a straightforward question, and getting it right shows you understand the API testing environment on a higher level. In your answer, you should mention [REST](https://www.codecademy.com/articles/what-is-rest/?utm_source=ccblog&utm_medium=ccblog&utm_campaign=ccblog&utm_content=cw_api_testing_interview_questions_blog), SOAP, [HTTP](https://www.codecademy.com/resources/docs/general/http/?utm_source=ccblog&utm_medium=ccblog&utm_campaign=ccblog&utm_content=cw_api_testing_interview_questions_blog), UDDI, and JMS.

**4. What are the most common kinds of API testing?**

While there are many different ways of using API testing, the typical methods are:

* Function testing, a test used to test the API’s functionality against functional requirements.
* Load testing, a test to emulate expected load, such as traffic, in a steady stream that can determine issues that may arise from running an API for a prolonged period of time.
* [Unit testing](https://www.codecademy.com/resources/docs/general/unit-testing/?utm_source=ccblog&utm_medium=ccblog&utm_campaign=ccblog&utm_content=cw_api_testing_interview_questions_blog), which verifies whether individual parts of the source code behave and work as expected.
* User interface testing, which tests the aspects of an API that a user will interact with.
* Security testing, a series of tests that test the security of the API. Entry points, flow of data, and shadow APIs are all areas that should be tested during security testing.
* Penetration testing (or pen test or ethical hacking), a type of security testing that simulates a cyber attack against your API to uncover vulnerabilities.
* Fuzz testing, a test that sends random data to all the endpoints of the APIs. During fuzz testing, servers shouldn’t crash or display any odd behavior.
* Runtime and error detection is used during all of these tests and enables APIs to report any issues that occur while it’s on.

When answering this question, you could focus on the tests that are used by the company to show your specific knowledge in that area.

**5. Describe a typical API testing environment**

This is your chance to show you know the difference between API and other kinds of [software testing](https://www.codecademy.com/resources/blog/what-is-software-testing/?utm_source=ccblog&utm_medium=ccblog&utm_campaign=ccblog&utm_content=cw_api_testing_interview_questions_blog). When testing an API, you first have to configure the database and server to ensure the API can be installed. After installing it, you can then start verifying its correct function. As the test continues, you evaluate the environment using API calls and analyze the results to see if it’s working as it should.

Here’s an example of a testing environment that you can give when answering this question: You can use [Postman](https://www.postman.com/api-platform/api-testing/) to test APIs. Postman allows you to make HTTP requests to an API and view the responses you get formatted in [JSON](https://www.codecademy.com/resources/docs/general/json/?utm_source=ccblog&utm_medium=ccblog&utm_campaign=ccblog&utm_content=cw_api_testing_interview_questions_blog).

**6. Which architectural styles are used to create web APIs?**

You may be asked this question because the type of architectural style used to create an API can inform the API testing protocol. The more common architectural styles are:

* REST API Style
* GraphQL API Style
* RPC API Style
* SOAP API Style
* gRPC API Style
* Falcor API Style

**7. What does SOAP mean?**

This is almost guaranteed to come up during your interview. SOAP is an acronym for Simple Object Access Protocol. This is a messaging protocol based on [XML](https://www.codecademy.com/resources/blog/what-is-xml-used-for/?utm_source=ccblog&utm_medium=ccblog&utm_campaign=ccblog&utm_content=cw_api_testing_interview_questions_blog), and it makes it easier for computers to communicate with each other.

**8. What is a REST API?**

This is another question that’s likely to be asked. In fact, it’s hard to imagine an API testing interview without it.

A REST API (also known as RESTful API) is an API that meets the constraints of REST architectural style, which is a set of architectural constraints for providing standards between computer systems on the web. These constraints make it easier for systems to communicate with each other. [REST](https://www.codecademy.com/articles/what-is-rest/?utm_source=ccblog&utm_medium=ccblog&utm_campaign=ccblog&utm_content=cw_api_testing_interview_questions_blog) is an acronym for REpresentational State Transfer.

**9. What are some differences between REST and SOAP?**

Because SOAP and REST can accomplish the same objectives, there’s a high probability that this question will pop up. Here’s a basic breakdown of the differences between REST and SOAP:

* SOAP is a protocol computers use to share XML documents, while REST is a design and service architecture used to set up software environments on a network.
* SOAP strictly supports XML, while REST supports several data formats.
* SOAP doesn’t support caching, while REST does.
* SOAP is slower.
* SOAP is like a normal computer app, while a REST client is more like a browser that allows apps to run inside it.
* SOAP envelops a message while running on HTTP, while REST makes use of HTTP headers to hold its metadata.

**10. What are some of the most common API tests performed?**

This question has several good answers, and your answer will likely depend on the tests you have the most experience with.

For example, you can discuss how an API is tested to see how it handles an HTTP (GET, POST, or other) request. With this test, you’re evaluating if the variables sent with the HTTP request were processed properly. This test also looks to see if there is a request error (40X error), and if so, what is it and why is it happening? Then, if the request is successful, did the API respond with the expected values properly formatted? An API test will also see if it’s communicating with the proper endpoint.

**11. What are some of the benefits of API testing?**

This is another chance to dig into the “why” of API testing. In your answer, you can highlight how API testing can save time because it may require less code than graphical user interface (GUI) testing. You can test APIs without a [user interface](https://www.codecademy.com/resources/blog/what-is-user-interface-design/?utm_source=ccblog&utm_medium=ccblog&utm_campaign=ccblog&utm_content=cw_api_testing_interview_questions_blog), meaning you don’t have to wait for an application to be available. Another benefit is that API testing isn’t limited to a specific coding language. And API testing is a good way to reduce risk cost-effectively, especially if you’re searching for minor bugs.

**What is an API?**

API is an acronym and it stands for **A**pplication **P**rogramming **I**nterface. API is a set of routines, protocols, and tools for building Software Applications. APIs specify how one software program should interact with other software programs.

In simple words, API stands for **A**pplication **P**rogramming **I**nterface. API acts as an interface between two software applications and allows the two software applications to communicate with each other. API is a collection of software functions that can be executed by another software program.

**2. What is API Testing?**

API testing is a type of [software testing](https://www.softwaretestingmaterial.com/software-testing/) that involves testing APIs directly and also as a part of integration testing to check whether the API meets expectations in terms of functionality, reliability, performance, and security of an application. In API Testing our main focus will be on the Business logic layer of the [software architecture](https://www.softwaretestingmaterial.com/software-architecture/). API testing can be performed on any software system which contains multiple APIs.

**3. What are the common API Testing Types?**

API testing typically involves the following practices:

* Unit testing
* Functional testing
* Load testing
* Runtime/ Error Detection
* Security testing
* UI testing
* Interoperability and WS Compliance testing
* Penetration testing
* Fuzz testing

Learn more on [API Testing Types](https://www.softwaretestingmaterial.com/api-testing/#API_Testing_Types)

**4. Name some of the common protocols used in API Testing?**

Some of the protocols using in API Testing are as follows:

* HTTP
* REST
* SOAP
* JMS
* UDDI

**What are some of the architectural styles for creating a Web API?**

Some of the architectural styles for creating web api are as follows.

* Simple URI as the address for the services
* Stateless communication
* HTTP for client-server communication
* XML/JSON as formatting language

**What is API test environment?**

In API testing environment, no GUI (Graphical User Interface) is available.

For API, the test environment is a rather sophisticated approach that configures the server and database to match the requirement of the software application. After the installation process is done, API will be verified for correct functioning.

Throughout the process, various parameters for the original environment are established via API calls to examine the test results.

**5. Difference between API and Web services?**

**Web services:**

1. All web services are APIs  
2. All web services need to be exposed over web(HTTP)  
3. A Web service uses only three styles of use: SOAP, REST and XML-RPC for communication  
4. A Web service always needs a network to operate

**APIs:**

1. All APIs are not web services  
2. All APIs need not be exposed over web(i.e. HTTP)  
3. API uses multiple ways for communication e.g. DLL files in C/C++, Jar files/ RMI in java, Interrupts in Linux kernel API etc.  
4. APIs don’t need a network for operation

**6. What is Soap?**

SOAP stands for Simple Object Access Protocol. It is an XML based messaging protocol. It helps in exchanging information among computers.

**7. What is Rest API?**

REST stands for Representational State Transfer. It is a set of functions helping developers in performing requests and receive responses. Interaction is made through HTTP Protocol in REST API.

**8. Difference between SOAP and REST?**

**SOAP:**  
1. SOAP is a protocol through which two computers communicate by sharing XML document  
2. SOAP supports only XML format  
3. SOAP does not support caching  
4. SOAP is slower than REST  
5. SOAP is like a custom desktop application, closely connected to the server  
6. SOAP runs on HTTP but envelopes the message

**REST:**  
1. REST is a service architecture and design for network-based software architecture  
2. REST supports different data formats  
3. REST supports caching  
4. REST is faster than SOAP  
5. REST client is just like a browser and uses standard methods An application has to fit inside it  
6. REST uses the HTTP headers to hold meta information

**9. What are the common tests that are performed on APIs?**

Some of the common tests we perform on APIs are as follows.

1. Verify whether the return value is based on the input condition. The response of the APIs should be verified based on the request.  
2. Verify whether the system is authenticating the outcome when the API is updating any data structure  
3. Verify whether the API triggers some other event or request another API  
4. Verify the behavior of the API when there is no return value

**10. What are the advantages of API Testing?**

* API Testing is time effective when compared to GUI Testing. API test automation requires less code so it can provide faster and better test coverage.
* API Testing helps us to reduce the testing cost. With API Testing we can find minor bugs before the GUI Testing. These minor bugs will become bigger during GUI Testing. So finding those bugs in the API Testing will be cost-effective to the Company.
* API Testing is language independent.
* API Testing is quite helpful in testing Core Functionality. We can test the APIs without a user interface. In GUI Testing, we need to wait until the application is available to test the core functionalities.
* API Testing helps us to reduce the risks.

**11. What exactly needs to be verified in API Testing?**

Basically, on API Testing, we send a request to the API with the known data and we analyze the response.  
1. Data accuracy  
2. HTTP status codes  
3. Response time  
4. Error codes in case API return any errors  
5. Authorization checks  
6. Non-functional testing such as performance testing, security testing

**12. Name some tools used for API Testing?**

Some of the tools used for API Testing are as follows:

* [Postman](https://www.getpostman.com/)
* [Katalon Studio](https://www.katalon.com/)
* [SoapUI](https://www.soapui.org/)
* [Assertible](https://assertible.com/)
* [Tricentis Tosca](https://www.tricentis.com/software-testing-tools/)
* [Apigee](https://apigee.com/)
* [JMeter](https://jmeter.apache.org/)
* [Rest-Assured](http://rest-assured.io/)
* [Karate DSL](https://github.com/intuit/karate)
* [API Fortress](http://apifortress.com/)
* [Parasoft](https://www.parasoft.com/)
* [HP QTP(UFT)](https://software.microfocus.com/)
* [vREST](https://vrest.io/)
* [Airborne](https://github.com/brooklynDev/airborne)
* [API Science](https://www.apiscience.com/)
* [APIary Inspector](https://help.apiary.io/tools/api-inspector/)
* [Citrus Framework](https://citrusframework.org/)
* [Hippie-Swagger](https://github.com/CacheControl/hippie-swagger)
* [HttpMaster Express](https://www.httpmaster.net/)
* [Mockbin](http://mockbin.org/)
* [Ping API](https://ping-api.com/)
* [Pyresttest](https://github.com/svanoort/pyresttest)
* [Rest Console](https://github.com/ahmadnassri/restconsole)
* [RoboHydra Server](http://robohydra.org/)
* [SOAP Sonar](http://www.crosschecknet.com/products/soapsonar.php)
* [Unirest](https://www.npmjs.com/package/unirest)
* [WebInject](http://www.webinject.org/)

***Learn more on***[***API Testing Tools***](https://www.softwaretestingmaterial.com/best-api-testing-tools/)

**13. List some most used templates for API documentation?**

Some of the API documentation templates are as follows.

* Swagger
* FlatDoc
* RestDoc
* API blueprint
* Slate
* Miredot
* Web service API Specification.

**14. Name some of the API examples which are quite popular.**

Some of the popular API examples are

* Google Maps API
* YouTube
* Twitter
* Amazon Advertising API

**15. Difference between API testing and Unit Testing?**

**UNIT TESTING:**

* Unit testing is conducted by the Development Team
* Unit testing is a form of White box testing
* Unit testing is conducted prior to the process of including the code in the build
* Source code is involved in Unit testing
* In unit testing, the scope of testing is limited, so only basic functionalities are considered for testing

**API TESTING:**

* API testing is conducted by QA Team
* API testing is a form of Black box testing
* API testing is conducted after the build is ready for testing
* Source code is not involved in API testing
* In API testing, the scope of testing is wide, so all the issues that are functional are considered for testing

**16. What are the main challenges faced in API testing?**

Some of the challenges we face while doing API testing are as follows

* Selecting proper parameters and its combinations
* Categorizing the parameters properly
* Proper call sequencing is required as this may lead to inadequate coverage in testing
* Verifying and validating the output
* Due to the absence of GUI, it is quite difficult to provide input values

**17. What are the types of bugs we face when performing API testing?**

Issues observed when performing API testing are

* Stress, performance, and security issues
* Duplicate or missing functionality
* Reliability issues
* Improper messaging
* Incompatible error handling mechanism
* Multi-threaded issues
* Improper errors

**18. How is UI testing is not similar to API testing?**

UI (User Interface) testing is to test the graphical interface part of the application. Its main focus is to test the look and feel of an application. On the other hand, API testing enables the communication between two different software systems. Its main focus is in the business layer of the application.

**19. Name some most commonly used HTTP methods?**

Some of the HTTP methods are

**GET:** It enables you to retrieve data from a server  
**POST:** It enables you to add data to an existing file or resource in a server  
**PUT:** It lets you replace an existing file or resource in a server  
**DELETE:** It lets you delete data from a server  
**PATCH:** It is used to apply partial modifications to a resource  
**OPTIONS:** It is used to describe the communication options for the target resource  
**HEAD:** It asks for a response identical to that of a GET request, but without the response body

**20. Can you use GET request instead of PUT to create a resource?**

No, GET request only allows read only rights. It enables you to retrieve data from a server but not create a resource. PUT or POST methods should be used to create a resource.

**21. What is the difference between PUT and POST methods?**

PUT and POST methods are sometimes confused in regards to when each should be used. Using POST request, our intent is to create a new object on the server whereas with PUT request, our intent is to replace an object by another object.

POST should be used when the client sends the page to the server and then the server lets the client know where it put it. PUT should be used when the client specifies the location of the page

**What is an API?**

**Ans:-**This is the most basic and crucial question among the API testing interview questions

 API (Application Programming Interface) is a software intermediary that helps communication and data exchange between two software systems.

You can also give more information about the API examples such as Google Maps API, Youtube API, Twitter API, AmazonAdvertising API,  and if you have previously worked on any.

**Q2. What is API Testing?**

This question is mostly asked question among the API interview questions to check if you can efficiently frame your answer.

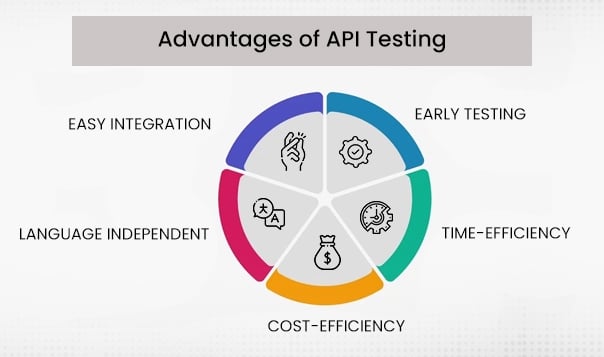
**Ans:** API testing is the testing of the developed so APIs to check their functionality, efficiency, reliability, and security.

**You can mention types of API testing such as**

* Validation testing
* Functional testing
* load testing
* UI testing
* Run time/error detection
* penetration testing
* Fuzz testing
* Interoperability and WS Compliance testing.

**Q3. What are the advantages of API Testing?**

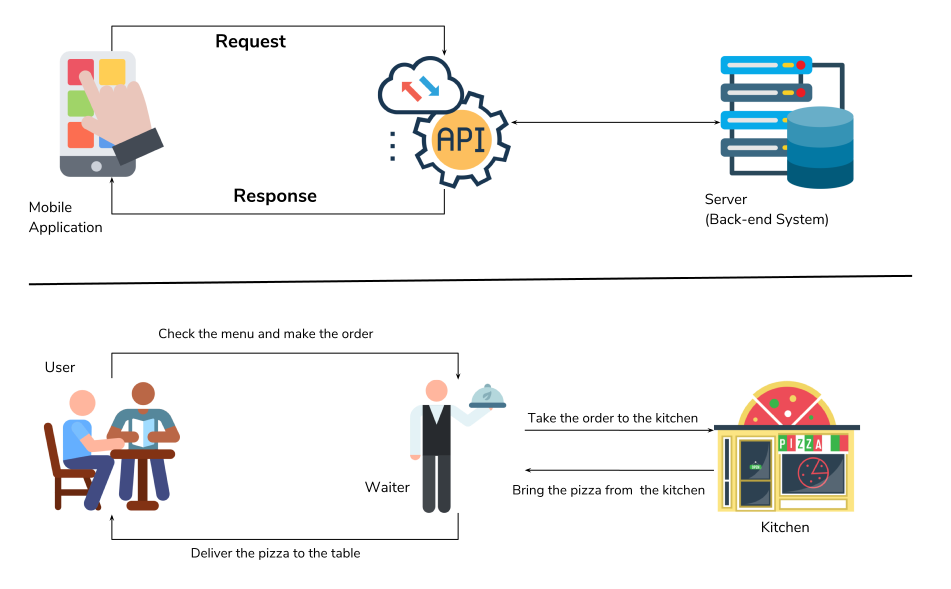
Give this answer in a way to show the interviewer the importance and relevance of API testing and where and how you are going to use it or have already used it.



**Ans:-**

* API testing helps in core functionality by giving direct access to the application, without needing a user interface. It is helpful in detecting minor errors before they turn into major issues during GUI testing.
* API testing uses less code than GUI testing so it gives better coverage. That’s hat it is time-effective too.
* API testing is language-independent as data is exchanged using XML or JSON, which allows users to select any coding language when adopting an automation test service.
* API can be easily integrated with GUI testing.

**Q4. How does API work?**

**Ans:-**Generally, API takes a request, processes it which is data validation, database interaction, data processing, and then the result is reverted to the source. 

**Q5. Do you know about the common tests performed on the APIs?**

**Ans:**-Here is the list of common tests that are performed on APIs –

* You should verify the API first and check whether it is updating any data structure or not.
* You need to check whether the API returns anything.
* As per the given parameters or values, the values returned by APIs need to be checked.
* Verify the API whether it triggers any other event or calls another API.

**Q6. What are the protocols used in API Testing?**

**Ans:-**These are the protocols used in API testing:

* HTTP
* REST
* SOAP
* UDDI
* JMS
* XML-RPC
* JSON-RPC

**Q7. What are the most commonly used tools in API Testing?**

**Ans:-**There are numerous API testing tools available. Some of the tools are Postman, Katalon studio, SoapUi, Apigee, JMeter, Parasoft SOAtest, Apigee, API fortress, JUnit, etc.

**Q8.  What to perform API test?**

* Select the suite where you will add the API test case that you are going to test
* Select the test development mode
* Create test cases for desired API methods
* Configure control parameters, test conditions, and validation method
* Perform API test
* View and analyze test reports
* Filter and sequence API test cases

**Q9. What should you take care of while writing cases?**

**Ans:-**These are the points you can reply with

* The test cases need to e aligned with the end-user interest.
* The steps should be simple so that anyone can use them later.
* Make sure that the test cases are reusable.
* Define the importance of the test cases.
* Provide a valid description, test input parameters, test data, expected outcome after.
* running so that the comparison can be done with the estimated one.
* Test cases should include negative test scenarios too.
* Naming conventions need to be strictly followed while creating the test cases.
* Check for updates daily.

**Q10. What needs to be verified in API testing?**

**Ans:-**In API Testing, we usually send a request to the API, and then we analyze the responses keeping the following parameters in mind:

* Comparing the actual response with the expected response to get data accuracy.
* HTTP status codes of the produced API.
* Response time of the API.
* Error codes if the APIs if there are errors
* Authorization Details
* Test for security, performance, availability, etc

**Q11. Define Test Data.**

**Ans:-**Test data is the input data used to perform test cases. This data can be prepared either manually or by using different tools. For example, to test the login functionality of an application, the input data will be username and password which constitutes the test data.

**Q12. Define test coverage.**

**Ans:-**Test coverage amount of testing performed by making use of test cases. It can be either functional testing or non-functional testing.

**Q13. What is a SOAP web service?**

**Ans:-** SOAP means Simple Object Access Control Protocol and this is an XML-based protocol that is used to exchange information between computer machines.

**Q14. What do you understand about the REST API?**

**Ans:-**Rest ( REpresentational State Transfer) API is defined as the set of functions that helps a developer in sending requests and receiving responses. In this protocol, the interaction is always made through an HTTP protocol. The meaning of REST is Representable State Transfer that has become a defacto standard for API creation these days.

**Q15. How will you differentiate the API testing and the UI testing from each other?**

**Ans:-** UI or User-Interface testing is used to check the graphical interface of an application or software program, how the user interacts and reacts to different elements like fonts, images, layouts, etc. This testing is majorly focused on the look and feel of an application.

At the same time, API enables communication among two different software components. Any software system implementing an API contains functions or subroutines that can be executed by any other software system.

**Q16.  How will you differentiate between API Testing and Unit Testing?**

|  |  |
| --- | --- |
| **Api Testing** | **Unit Testing** |
| Performed by QA team | Performed by the development team |
| This is a form of black-box testing | This is a form of White box testing |
| Full functionality of the software is checked as it will be used by external developers (The end-user) | Here each unit is tested for functionality.If each unit is performing well in isolation. |
| The tester usually does not have the access to the source code, only the functionality test is done. | The developer has the access to the unit test cases as they test these before they proceed for the next step. |

**Q17. What is test documentation?**

**Ans:-**The API documentation is technical writing giving instructions on how to use and integrate with an API efficiently. It gives reference to the information needed to work with the API and helps you get API testing questions.

**Q18. When you are writing an API document, what are the major areas to focus on?**

**Ans:-**Here is the list of major areas that you should focus on while writing an API document –

* Focus on the content source
* Sketch or document the plan well.
* Layout delivery
* Detailed information about each of the functions
* Automatic document creation programs

**Q19.  For any API document, how can we document different functions?**

**Ans:**-Syntax – Here, you have to write the syntax for the parameter of the code in the same sequence as they occur, highlight necessary elements, optional elements, etc.

* **Description –** Give a quick description of each of the functions
* **Error Messages –** Here you need to give the syntax of error messages.
* **Parameters –** Give proper details about function parameters.
* **Related Links**– Connection fields
* **Example Code –**A small snippet of the code

**Q20.  Mention the list of tools that are needed for the documentation.**

**Ans:-**[Read: Automation Testing Tutorial Guide for Beginner](https://www.janbasktraining.com/blog/automation-testing-tutorial/)

For Java Code, you can use JavaDoc and for the .Net code, you can use Doxygen.

**Q21. Difference between API and Web services?**

**Ans:-**

|  |  |
| --- | --- |
| **API** | **Web Services** |
| All APIs are not web services | All web services are APIs |
| All APIs need not be exposed over the web(i.e. HTTP) | All web services need to be exposed over the web(HTTP) |
| API uses multiple ways for communication e.g. DLL files in C/C++, Jar files/ RMI in java, Interrupts in Linux kernel API, etc | A Web service uses only three styles of use: REST, SOAP, and XML-RPC for communication |
| A Web services are network dependent | APIs don’t need a network for operation |

As you are now aware of the API interview questions asked for freshers, Now let’s move towards the advanced API interview questions. Get prepared for all the API interview questions, You never know.

**B. Advanced API Testing Interview Questions and Answers**

Here are the API interview questions that should you prepare while appearing for the senior profiles. These API testing interview questions are asked mostly to test your knowledge and experience, so try to flaunt your experience here to impress the interviewer.

**Q22. What is the basic process to test an API? Explain based on your previous experiences.**

**Ans:- Here are the basic steps to follow to test an API –**

* First of all, decide the suite where you wanted to add the API test case.
* Now select the test development code too.
* Now you have to develop test cases for needed API methods.
* This is the time to configure the app control parameters.
* Configure the test conditions too.
* Validate the methods and configure them
* Execute APIs, check the test reports, filter the test cases, arrange the test cases etc.

By following these steps in the same sequence as given, this is easy to test an API successfully.

**Q23. What are the errors expected during API Testing?**

**Ans:-**These are the errors you can tell to let them know that you have performed such tests before and you know what you claim.

* Security
* Duplicate or missing functionality
* stress
* Reliability
* Unused flags
* Performance
* Incompatible error handling
* Multi-threatening
* Improper errors

**Q24. What are the major challenges faced while performing API Testing?**

**Ans:-** This is one of the most asked API testing interview questions.

You can always talk about the challenges you perform while doing API testing, or the list can go like this.

* Initial setup
* parameter selection
* combination and validation
* sequencing the API calls
* Updating the schema
* tracking system integration.

**Q25. What is the test environment of API?**

**Ans:-**Setting up an API environment is not easy, so try to share your experience and try to communicate with the interviewer on this.

The test environment of API is a bit complete and requires the configuration of both database and server is done without the integration of GUI.

After installation, API is verified for the right operation. Throughout the process, we study the test process by setting up the API with different parameters.

**Q26. What are the principles of an API test design?**

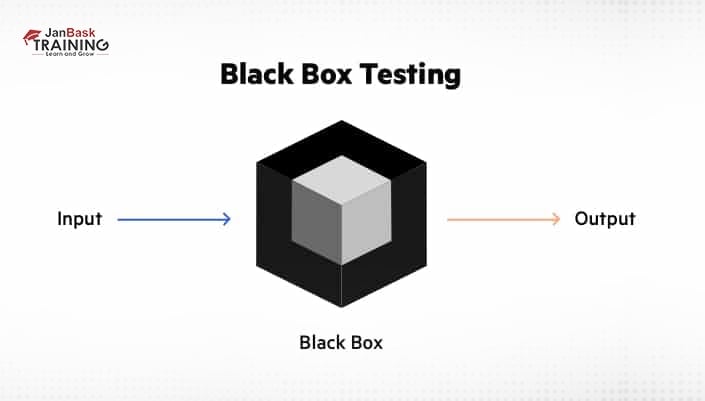
**Ans:-**It is one of the API design interview questions. You can answer it like this:

**The seven principles of API test design are listed below.**

* Exhaustive testing is not possible. In fact, we need to do an optimal amount of testing which relies on the risk assessment of the application.
* Defect Clustering states that a small number of modules carries most of the errors. In 20% of the module, approximately 80% of defects are found. Finding such defects depends on experience.
* You should not conduct the same set of repetitive tests, The same method will not be sufficient to recognize new defects. A metaphor here is the pesticide paradox. Usage of the same pesticide over and over will lead to the evolution of the insects and they will soon become resistant to the pest. The same applies here.
* The absence of defects needs to be taken care of. The testing principle states that the tests inform about the presence of defects not about the absence of a defect. So, even if no defects are found, it can’t be proof of correctness.
* The absence of defect is a fallacy. Sometimes, the software which is 99% bug-free is still not usable. This can happen if the system is tested for the wrong requirements. So, finding errors and correcting them will not help if the requirements of the end-user are not taken care of.
* Early testing is always helpful, if the error is solved in the requirement or design phase, it is easier and cheaper. So, we should start testing in the early stages of the development life-cycle.
* Testing is context-dependent. All the developed software is not identical. You need to test all software based on the context. You might use different approaches, techniques, methodologies, and types of tests for different software and requirement. A POS system will be different for a retail store and an ATM machine.

**Q27. What do you understand by Black Box Testing?**

**Ans:-** Black Box Testing is a method of software testing where the testers evaluate the functionality of the application without peering into the internal source code. This method can be applied in every level of software testing such as integration testing, unit testing, system testing, and acceptance testing.



**Q28.  What do you understand about the API Framework?**

**Ans:-** There is no need to give an introduction to the API Framework as it is self-explanatory. When you are testing APIs then you don’t have to work on each API independently but you can use config files in that case where details for all APIs are given and can be used if needed.

**Q29. Explain the working of API builders.**

**Ans:-**API builder is a PL SQL program that is made up of four SQL files where one file is responsible for starting the process, two files are used to create the temp tables or the master package, and the fourth file will help in generating the final output.

**Q30. What is a Test API?**

**Ans:-** Test API is a set of test APIs or a library of utility that helps developers or testers in creating testing tools and automated test cases for .Net or WIN 32 applications. It also offers a set of basic building blocks, data structures, data types, algorithms, etc.

**Q31. How do you test the API security?**

**Ans:-**For testing the security of API during API testing, we need to validate 2 things:

* **Authentication:**Whether the identity of the end-user is correct.
* **Authorization:**Whether the user has access to the resource.

We can also check whether the TLS or the SSL certificate used over the HTTPS protocol is valid or not.

**Q32. What are the different types of Input Injection and what are the different ways of stimulating user input?**

**Ans:-** Input injection is a popular act of simulating user inputs and it can be done in popular ways as listed below –

[Read: What is Unit Testing? Unit Testing Tutorial Guide for Beginners](https://www.janbasktraining.com/blog/unit-testing-tutorial/)

* By direct method invocation
* With the help of an accessibility interface
* By simulating low-level input
* By simulating a device driver
* Simulation using a robot

**Q33. How to perform the API testing with Run Scope?**

**Ans:-** Run Scope is a web application that supports backend services and easy to understanding user interface tool for testing APIs.

**Q34.  What are the different principles of API testing design?**

**Ans:-** The major principles include – Setup, Execution, Verification, Reporting, Clean up, etc.

**Q35. What are the tools that should be used for API test automation?**

**Ans:-**The automation tools that are frequently used for API test automation include – JUnit for Java, SOAP UI, HP UFT, or NUnit for .NET, etc.

**Q36. What is caching and how does it work?**

**Ans:-** Caching is a mechanism to improve the performance of applications. It is a process of storing and accessing data from a cache. A cache can be defined as a software or hardware component intended at storing data so that future requests for the same data can be retrieved faster.

e XML, like the SMTP server or POP3 protocol to pass the messages or reply to queries.

**Q37. Name some commonly used  API documentation templates?**

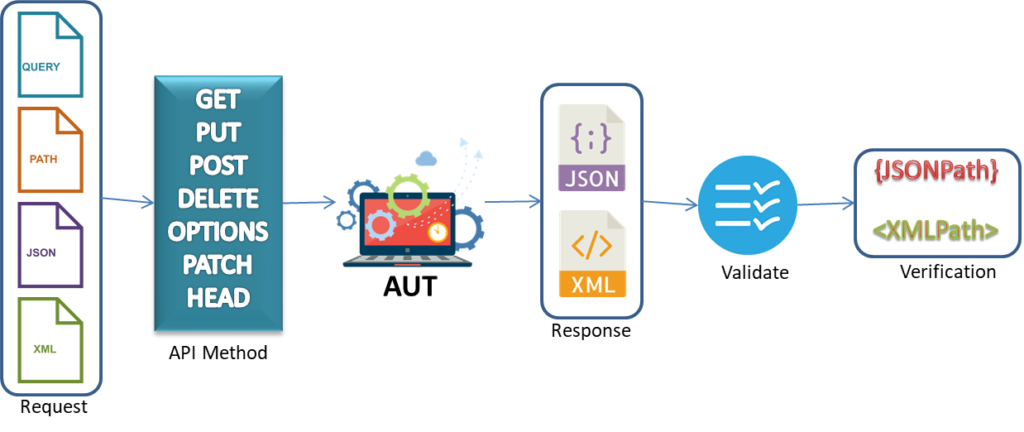
**Ans:-** There are numerous API documentation templates available to make the entire process simple and straightforward.

* Slate
* Miredot
* Swagger
* FlatDoc
* API blueprint
* RestDoc
* API doc
* Web service API specification

Get a complete PDF of API Testing interview questions [here](https://www.janbasktraining.com/schedule-your-call-now/abc)!

**C. Commonly Asked Rest Assured Interview Questions And Answers**

These are 7 rest assured API automation testing interview questions and answers to give you a knowledge of REST assured, RESTful web services, URI, and the types of HTTP methods. You can face such API questions too.



**Q38. What is REST assured?**

**Ans:-**REST Assured is a Java-based library that is used for RESTful APIs testing. It can be used to test applications based on JSON and XML. Also, all methods are entirely supported, including PUT, POST, GET, PATCH, and DELETE.

**Q39. What are RESTful web services?**

**Ans:-** Restful web services are REST Architecture-based Web services. RESTful web services use HTTP as a communication between the server and the client.

**Q40. What is URI?**

**Ans:-**URI (Uniform Resource Identifier) consists of the base URL, path parameter, and query parameter.

**Q41.  Which HTTP methods are used in REST and what do they do?**

**Ans:-**You definitely should remember these HTTP methods before going to appear API testing interview questions.

* **GET:** Retrieves resource representation. Should be cacheable.
* **POST:**Creates a new subordinate resource. Not cacheable
* **PUT:**Updates existing resources. If the resource does not exist, then API may decide whether to create a new one or not.
* **PATCH:**Makes a partial update
* **DELETE** Deletes the resource. Idempotent and not cacheable
* **OPTIONS:** Describe the communication options for the target resource.
* **HEAD:**Asks for a response identical to that of a GET request, but without the response body.

**Q42. Can You explain the payload?**

**Ans.**Payload/body is highly secured input data that is sent to API to process the request. The payload is usually in JSON format in REST API.

**Q43. What is the difference between PUT and POST?**

**Ans.**The main difference between PUT and POST is PUT requests are idempotent. Calling the same PUT request multiple times will always respond with the same result. In contrast, calling a POST request repeatedly have the side effects of creating the same resource multiple times.

**Q44. can we use GET request instead of PUT to create a resource?**

**Ans.**PUT and POST method is used to create a resource. GET is only used to request the resources.

**D. Frequently Asked SOAP Interview Questions and Answers**

Get the Knowledge of SOAP to be able to answer any SOAP-related questions swiftly.

**Q45. What are SOAP Web services?**

**Ans:-** This is one of the fundamental API testing interview questions that you must know the answer to.

The SOAP (Simple Object Access Protocol) is defined as an XML-based method used in web services.  It is both platform and language independent.

It is well known for designing and developing web services. It also enables the communication between applications developed on different platforms using various programming languages over the Internet.

**Q46. How does SOAP work?**

**Ans:-**SOAP provides a user interface that can be accessed by the client’s server object, and the request sent by it goes to the server, which can be accessed using the server object. The user interface creates some files or methods consisting of server objects and the name of the interface to the server object.

 It also contains other information such as the name of the interface and methods. It uses HTTP to send the XML to the server using the POST method, which analyzes the method and sends the result to the client.

The server creates more XML consisting of responses to the request of the user interface using HTTP. The client can use any approach to send the XML, like the SMTP server or POP3 protocol to pass the messages or reply to queries.

**Q47. What are the syntax rules for a SOAP message?**

**Ans:-** These are some basic syntax rules for an SAOP message

* Must use encoded XML
* Envelope namespace must be used
* Encoding namespace must be used
* Must not consist of a DTD reference
* Must not have XML processing instruction

**Q48. How do users utilize the facilities provided by SOAP?**

**Ans:-**

* **PutAddress():** we use it to enter an address in the webpage and has an address instance on the SOAP call.
* **PutListing():** It is utilized to allow the insertion of a complete XML document into the web page. It receives the XML file as an argument and transports the XML file to XML parser liaison, which reads it and inserts it into the SOAP call as a parameter.
* **GetAddress():**It helps in getting a query name and getting the result that best matches a query. The name is sent to the SOAP call in the form of the text character string.
* **GetAllListing():**It helps to return the full list in an XML format

**Q49. Tell the elements of SOAP message structure**

**Ans:-** It is an ordinary XML document that contains the elements as a SOAP message

* **Envelope:**It is a mandatory root element that translates the XML document and defines the start and end of the message
* **Header:** It is an optional element. It stores the information about the message being sent
* **Body:** It contains the XML data comprising the message being sent
* **Fault:** It informs about the errors that occurred while processing the message

**Q50. What are the obstacles users face while using SOAP?**

**Ans:-**The major difficulty faced by users using SOAP is a firewall security mechanism. This locks all the ports leaving some like HTTP port 80 used by SOAP that bypasses the firewall.

The major technical issue with SAOP is, it mixes specifications of message transport and message structure.

## What is an API? How does it work?

**Ans.** API stands for Application Programming Interface. It is a software interface that enables two applications to communicate and exchange data with each other. It is a set of functions that can be executed by another software program.

Let us now understand how an API works. When you use an application on your smartphone, the application connects to the Internet. It sends data to a server. The server will retrieve that data, interpret it, and perform the required actions. Then it sends it back to your phone. The application interprets that data and provides you with the information in a readable way without exposing the internal details.

**Examples of API:**

* Google maps
* Third-party payment processing

***Explore the***[***Top Paying IT Certifications***](https://www.naukri.com/learning/articles/top-paying-certifications/)

## Q2. Define API testing.

**Ans.** API (Application Programming Interface) has predefined sets of routines, protocols, and tools which guide how some software components should interact with any other component. API is a collection of functions that enable the creation of applications that access the features or data of an application or operating system. The testing of these functions is called API testing.

Below are some of the tests which are performed on API:

* Verify what APIs return
* Return value as per the input condition
* Verify what API trigger
* Verify whether API updates the data structure

## Q3. What are the different types of API testing?

**Ans.** API testing types:

* UI Testing
* Validation Testing
* Unit Testing
* Functional Testing
* Load Testing
* Runtime and Error Detection
* Security Testing
* Penetration Testing
* Fuzz Testing
* Interoperability and WS compliance Testing

## Q4. What are the benefits of API testing?

**Ans.** The benefits of API testing are:

**1. Test for Core Functionality**

API testing allows accessing the application without the UI (User Interface). The core functionality of the application is tested before the GUI tests. It helps in identifying the minor issue which can become bigger during the GUI testing.

**2. Time Effective**

API Testing does not need GUI to be ready. It can be performed earlier in the development cycle. API testing is less time-consuming than GUI testing. API tests offer test results quickly. They significantly accelerate development workflows. This helps in speeding up the feedback loop and identifying issues faster.

**3. Reduced Cost**

API testing requires less code and provides better and faster test coverage compare to GUI test automation. You can identify issues at an early stage. It is easier and comparatively inexpensive to fix the issue at the early stages. This reduces the cost of the testing project.

**4. Language-Independent**

The data in API testing is exchanged data using XML or JSON. These transfer modes are completely language-independent. It means that you can select any core language when using automated testing services for the application.

**5. Easy Integration with GUI**

API testing allows highly integrable tests. This is useful if you plan to perform functional GUI tests after API testing. For example, simple integration will allow for the creation of new users within the application before the start of a GUI test.

**Explore**[**Popular Online Courses**](https://www.naukri.com/learning/)

## Q5. What are the tools used for API testing?

**Ans.** Some of the popular API testing tools for 2022 are:

* SoapUI
* PostMan
* Karate DSL
* Rest Assured
* Alertsite API monitoring
* Apigee
* Assertible

**Check out the**[**10 Best Software Development Tools and Frameworks**](https://www.naukri.com/learning/articles/best-software-development-frameworks-and-tools-to-learn/)

## Q6. Explain some protocols used in API Testing.

**Ans.** API protocols facilitate a standardized exchange of information. They provide a collection of defined rules that specifies the accepted data types and commands. Below are some of the common protocols used in API testing:

* HTTP
* JSON-RPC
* REST
* SOAP
* JMS
* UDDI
* XML-RPC

**Also, Explore:**

* Paid and [free online courses](https://www.naukri.com/learning/coursera-courses-certification-training-v475)
* Popular online [courses](https://www.naukri.com/learning/udemy-courses-certification-training-v21)
* Top online [courses](https://www.naukri.com/learning/edx-courses-certification-training-v117)

## Q7. What are the 7 principles of an API test design?

**Ans.** The 7 principles of API test design are the following:

#### **1. Exhaustive Testing**

Exhaustive testing is impossible. An optimal amount of testing based on the risk assessment of the application is needed.

#### **2. Defect Clustering**

Defect Clustering applies the Pareto Principle to software testing. According to this principle, a small number of modules contain most of the defects detected. Approximately 80% of the defect are found in 20% of the modules. By experience, one can identify such risky modules. There are some drawbacks to this approach. If the same tests are repeated again and again, the test case will not find new bugs.

#### **3. Pesticide Paradox**

If the tester conducts the same set of repetitive tests, it will become useless for identifying new defects. Thus, the test cases need to be regularly reviewed and revised. New and different test cases must be added to discover more defects. Testers need not depend on existing techniques. They need to improve the existing method to make testing more effective.

#### **4. Testing shows the presence of defects**

As per this principle, testing talks about the presence of defects not about the absence of a defect. Software testing reduces the number of undiscovered defects which remain hidden in the software. Even if no defects are found, it does not prove that the software or system is 100% defect-free. Teams should accept this concept. They should make efforts to manage client expectations.

#### **5. Absence of error-fallacy**

The software which is 99% defect-free may be still unusable. Such a case may occur if the system is tested for the wrong requirement. Software testing is not about finding the defects but also ensuring that software meets the business needs. The absence of error-fallacy means discovering and fixing defects will not help if the system build is unusable and fails to fulfill the requirements of the user.

#### **6. Early Testing**

Early Testing is crucial in the software lifecycle. Testing should start early in the software development lifecycle. This helps in identifying the defects in the requirement or design phase in the early stages. It is easier and comparatively inexpensive to fix a defect in the early stages of testing.

#### **7. Testing is context-dependent**

It means that the types and methods of testing performed can completely depend on the context of the software or systems. For example, the method of testing an e-commerce site will be different from a testing method of a commercial website.

**Check out the list of**[**Top Universities Offering Free Online Programming Courses**](https://www.naukri.com/learning/articles/top-universities-offering-free-online-courses-for-programmers/)

## Q8. What is the difference between API testing and UI testing?

**Ans.** The differences between API testing and UI testing are:

|  |  |
| --- | --- |
| **API Testing** | **UI Testing** |
| API testing enables the exchange of data and communication between two software systems. | It involves the testing of the graphical user interface (GUI). |
| API testing is the testing between the backend or server-side of the application and the backend of another application. | It is the testing between users and the frontend or client-side of the application. |
| It is focused on the testing of business logic, data responses, and performance challenges. | UI testing concentrates on how users interact with an application and its elements such as fonts, buttons, colors, and layouts. |
| API testing focuses on the business logic layer of the software architecture and not on the look and feel of an application. | It focuses on the look and feel of an application. |

[[](https://www.naukri.com/learning/articles/top-wipro-interview-questions-and-answers/)](https://www.naukri.com/learning/articles/top-wipro-interview-questions-and-answers/)

**[Most Asked Wipro Interview Questions and Answers [2023] -](https://www.naukri.com/learning/articles/top-wipro-interview-questions-and-answers/)**

[Read this article to learn the important Wipro interview questions and answers. These questions contain both general and technical questions.](https://www.naukri.com/learning/articles/top-wipro-interview-questions-and-answers/)

[[](https://www.naukri.com/learning/articles/accenture-interview-questions-and-answers/)](https://www.naukri.com/learning/articles/accenture-interview-questions-and-answers/)

**[Top Accenture Interview Questions and Answers [2023]](https://www.naukri.com/learning/articles/accenture-interview-questions-and-answers/)**

[This article covers Accenture interview questions and answers that are frequently asked. Those who plan to take the interview must](https://www.naukri.com/learning/articles/accenture-interview-questions-and-answers/)**[...read more](https://www.naukri.com/learning/articles/accenture-interview-questions-and-answers/)**

[[](https://www.naukri.com/learning/articles/sap-interview-questions/)](https://www.naukri.com/learning/articles/sap-interview-questions/)

**[Top 10 SAP Interview Questions with Answers](https://www.naukri.com/learning/articles/sap-interview-questions/)**

[These 10 SAP interview questions will help you prepare for your upcoming interview. Prepare these comprehensive answers and find out](https://www.naukri.com/learning/articles/sap-interview-questions/)**[...read more](https://www.naukri.com/learning/articles/sap-interview-questions/)**

## Q9. What is SOAP?

**Ans.** SOAP stands for Simple Object Access Control. It is an XML-based protocol for exchanging information between computers. It is a communication protocol for communicating through the Internet. SOAP is platform-independent and language-independent and can be used for broadcasting a message.

***Explore the list of***[***Top Online IT Courses***](https://www.naukri.com/learning/articles/top-online-courses-for-it-professionals/)

## Q10. What is REST API?

**Ans.** REST stands for Representational State Transfer. REST API is an API that conforms to the principles of the REST architectural style. It allows for interaction with RESTful web services. This is why REST APIs are sometimes referred to as RESTful APIs.

## Q11. What is the difference between SOAP and REST API?

**Ans.** The differences between SOAP and REST API are:

|  |  |
| --- | --- |
| **SOAP** | **REST API** |
| SOAP stands for Simple Object Access Protocol. | REST stands for Representational State Transfer. |
| It is a protocol and follows a strict standard to allow communication between the client and the server. | It is an architectural style that follows six constraints – Uniform Interface, Client-Server, Layered System, Stateless, Cacheable, Code on Demand. |
| SOAP only uses XML for exchanging information. | REST is not restricted to XML. It permits different data formats such as Plain text, XML, HTML, and JSON. |
| SOAP cannot use REST. | REST can use SOAP protocol. |
| Difficult to implement. | Easy to implement. |
| Requires more bandwidth. | requires less bandwidth. |

***Also Read:***[***Top Software Testing Interview Questions and Answers***](https://www.naukri.com/learning/articles/top-software-testing-interview-questions-answers/)

## Q12. What are the types of bugs found in API testing?

**Ans.** The different types of bugs found in API testing are:

* Duplicate or missing functionality
* Reliability
* Security
* Unused flags
* Inconsistent error handling
* Performance
* Multi-threading issues

**Explore**[**Free Online Courses with Certificates**](https://www.naukri.com/learning/free-online-courses-certificates)

## Q13. What is the difference between Web Service and API?

**Ans.** Following are some of the differences between web service and API:

1. Every web service is an API. The vice versa may not be true. This means that not every API is a web service.
2. It is not necessary that a web service can perform every task that an API would perform.
3. A web service uses three styles of use including REST, SOAP, and XML-RPC for communication. On the other hand, an API may be exposed to in more ways.
4. Web services always need a network to operate whereas APIs do not need a network for operation.

## Q14. What are the limits on API usage?

**Ans.** Many providers set up limits on API usage. It is important to have an estimate of your usage and understand its impact on the overall cost of the offering. You need to consider the issue may arise depending on how data is leveraged.

## Q15. What are API restrictions?

**Ans.** API restrictions enable you to specify exactly which Google Cloud APIs can be called through the API key. You should add both client and API restrictions to every API key. It is possible to specify one and more services in the API restrictions

### 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.

### 2. 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.

### 3. 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

### 4. 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.

### 5. Name some tools used for API Testing?

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

* Postman
* Tricentis Tosca
* Katalon Studio
* Apigee
* Jmeter
* SoapUI

### 6. 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.

### 7. 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.

### 8. Where are query parameters stored in a GET request?

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

### 9. 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.

### 10. 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.

#### **Prepare Yourself to Answer All Questions!**

Automation Testing Masters Program[EXPLORE PROGRAM](https://www.simplilearn.com/automation-testing-masters-program-certification-training-course?source=GhPreviewCTABanner)



### 11. 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.

### 12. 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.

### 13. 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.

### 14. 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.

### 15. 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.

### 16. Which programming language is used for Postman tests?

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

### 17. What is a Postman Collection?

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

### 18. 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.

### 19. 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.

### 20. 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.

#### **Learn 15+ In-Demand Tools and Skills!**

Automation Testing Masters Program[EXPLORE PROGRAM](https://www.simplilearn.com/automation-testing-masters-program-certification-training-course?source=GhPreviewCTABanner)



### 21. 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"));

### 22. How do you access postman variables?

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

### 23. 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

### 24. 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

### 25. 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.

### 26. 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.

### 27. 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.

### 28. 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.

### 29. 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.

### 30. 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.

### 31. 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.

### 32. 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.

### 33. 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.

### 34. 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.

### 35. 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.

### 36. 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.

### 37. How do you remove local variables?

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

### 38. 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-urlencoded.

#### **Get the Must-Have Skills of a Web Developer**

Caltech Coding Bootcamp[EXPLORE PROGRAM](https://www.simplilearn.com/coding-bootcamp?source=GhPreviewCTABanner)



### 39. 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

### 40. 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.

### 41. 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.

### 42. 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.

### 43. 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

### 44. 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.

### 45. 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);

  });

### 46. 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.

### 47. 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.

### 48. 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.

### 49. 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);

#### **New Course: Full Stack Development for Beginners**

Learn Git Command, Angular, NodeJS, Maven & More[ENROLL NOW](https://www.simplilearn.com/free-online-full-stack-development-course-skillup?utm_source=frs&utm_medium=skillup-course-banner&utm_campaign=frs-skillup-course-promotion)



### 50. 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.

### 51. 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'));

### 52. 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.

### 53. 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)-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.

### 54. 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}}

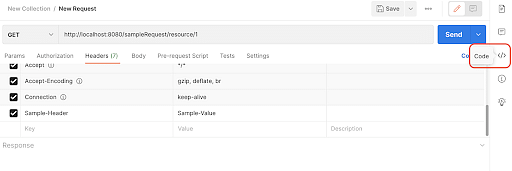
### 55. What do you understand about ScratchPad?

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.

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

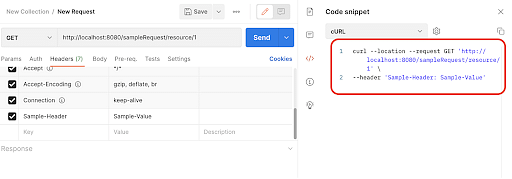
You can use the steps listed below to get the cURL command equivalent:

As indicated below, click the Code icon.



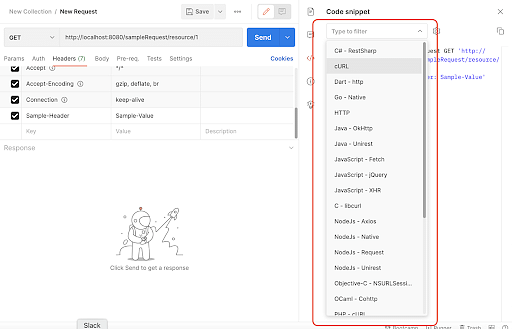
Step 1 - Click on Code

In response to the REST API request, you will receive the cURL command:



Step 2 - REST API request

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.



Step 3 - Select Language

Get prompt resume building and interview preparation assistance to land your dream job. Enroll in our [Post Graduate Program in Full Stack Web Development](https://www.simplilearn.com/pgp-full-stack-web-development-certification-training-course?source=GhPreviewCTAText) today!

**What are the main challenges you face in the API testing in your project?**

Challenges are for example,

***API documentation***

***Access to the database***

***Call sequencing***

**2- What is the difference between the PUT and the POST method?**

This is the most common API testing interview question these days.

* Firstly, the POST request means to create a new object in the database.
* Secondly, a PUT request means to update the existing object in the database with the new value.

**3- What are the most commonly used HTTP methods?**

For example:

|  |  |
| --- | --- |
| GET | Used to retrieve data from the server |
| POST | create a new object in the server |
| PUT | Used to update an existing object in the server |
| DELETE | used to delete data from the server |

**4- List a few authentication techniques used in API testing.**

* Session / Cookies based Authentication
* Basic Authentication
* Digest Authentication
* OAuth

Above all are the few most important authentication techniques.

**5- What is the REST API?**

REST – Representational State Transfer, is a set of functions that helps developers performing requests and receive responses. Interaction is made through HTTP protocol in REST API.

**6- What exactly you verify in API testing?**

***Accuracy of data***

***HTTP status code***

***Response time***

***Error codes if API returns an error***

***Authorization***

***Performance***

***Security***

Above all are the most important verification checks.

**7- Differentiate API testing and UI testing.**

UI (User Interface) testing means testing the graphical user interface. The focus of UI testing is on the look and feel of the application, like how the user interacts with the application elements, such as images, font, layout, etc.

API testing allows communication between two software systems. It determines if the developed APIs meets the expectation regarding functionality, reliability, performance, and security. It works on the backend and also knows and the backend testing.

To clarify, describe some scenarios while answering this question in an interview.

**8- What protocol RESTFUL Web services use?**

RESTFUL web services use the HTTP protocol as the medium of communication between client and server.

**9- Can we use POST instead of PUT to create a resource?**

Yes, we can because POST is the superset of all HTTP requests except GET requests.

**10- What do you understand by payload?**

Payload/body is a secured input data that is sent to API to process the request. The payload is generally constructed in JSON format in REST API.

**11- What kind of bugs that API testing can find?**

***Missing or duplicate functionality***

***Fail to handle error conditions gracefully***

***Stress***

***Reliability***

***Security***

***Unused flags***

***Performance***

***Multithreading issues***

***Improper errors***

Above all are the most frequent bug that API testing can detect.

**12- Describe the term Environment with respect to Postman?**

The environment in Postman is a set of key-value pairs. We can create multiple environments in postman.

There are two types of environment, the global environment, and the local environment. They define the scope of the variable to use it in the requests.

The most common variable we use is URL because the URL is used in every request and changing it in every request can be very time-consuming.  When we create an environment inside Postman, we can change the value of the key-value pairs and the changes are reflected in the requests.

**13- State the common status code you encounter in API testing.**

This is the most common API testing interview question.

**200 (OK)** Defines that the request was correct.

**201 (Created)** The value wrapped with the request has been created in the database.

**204(No Content)** This status code means that the request was correct and received but there is no response to send to the client by the server.

**400 (Bad Request)** A bad request means that the syntax of the request was incorrect. It can happen if you have sent the wrong parameters along with the request URL or in the body of the request.

**401 (Authorized)**  We can incur such a status code when you are not authorized to access the server or you have entered the wrong credentials.

**404 (Not Found)** A response code 404 means that the server was connected but it could not find what was requested. You can normally see this status code when you request a web page that is not available.

**500(Internal Server Error)** A response code 500 means there was some exception at the server level while executing the request.

**502(Bad Gateway)** The server, while acting as a gateway or proxy, received an invalid response from the upstream server it accessed in attempting to fulfill the request.

**503(Service Unavailable)** The server is currently unable to handle the request due to a temporary overloading or maintenance of the server.

**504(Gateway Timeout)** The server, while acting as a gateway or proxy, did not receive a timely response from the upstream server specified by the URI

Above all are the most common status codes.

**14- What is Pre-Request Script in Postman?**

In short, a pre-request script is a script that runs before the execution of a request.

**15- What is the difference between authorization and authentication?**

* Authentication is a process of presenting your credentials to the system and then the system validates your credentials. These credentials tell the system about who you are.
* Authorization is a process of allowing or denying someone from accessing something once authentication is completed.

**16- What is the importance of setNextRequest in Postman?**

setNextRequest is used to define the workflow of API testing. setNextRequest is needed to control the order of the execution of requests.

**17- What are the two types of scripts in Postman?**

* Tests script
* Pre-request script

Above are the types of scripts in postman.

**18- What is REST?**

Representational State Transfer is an architectural style of developing web services. In this architecture, the server provides access to resources and the client presents those resources. Each resource is identified by URI. REST uses different ways to represent a resource like JSON, text and XML. XML and JSON are the most popular one. Resource are accessed by a common interface using HTTP standard methods.

**19- Which is the most popular way to represent a resource in REST?**

JSON is the most popular and important way to represent resources.

**20- What do you understand by messaging in RESTful web services?**

RESTful web services use HTTP as a medium of communication between client and server. The client sends a message in the form of an HTTP request and then the server transmits the HTTP response. This technique of interaction is called messaging. These messages contain both message data and metadata (information about the message itself).

**21- List the core components of an HTTP request?**

* HTTP methods type such as GET, PUT, POST, DELETE
* URI that acts as an identifier for the resource on the server
* HTTP Version
* Request Header, Metadata, Cache Settings, Authentication Parameters
* Request Body or the Payload

Above are the core components of an HTTP request.

**22- What is Rest Assured?**

Rest Assured is a java based library that is used to test the RESTful Web Services. It acts as a headless client to access REST services. REST Assured provides a lot of features, which makes API automation testing very easy. Like it offers friendly DSL-like syntax, XPath-Validation, Specification Reuse, Easy file uploads.

**23- Define what is a URI?**

Uniform Resource Identifier, URI consists of base URL, path parameter, and query parameter

URI= Base URL + Path Parameter + Query Parameter

Example:

URI-   <https://reqres.in/api/users?page=2>

**24- What do you mean by the HTTP status code?**

REST APIs use HTTP status codes to tell what exactly happened when the server processed the request.

 Grouping for HTTP Status Codes will be :

* **1xx** – Informational
* **2xx** – Success e.g. 200 Success, 201 Created
* **3xx** – Redirection e.g. 302 Temporary Redirect
* **4xx** – Client Error e.g. 400 Bad Request, 404 Not Found
* **5xx** – Server Error e.g. 500 Internal Server Error

The type of status code you receive depends on the application you are interacting with. Usually, a 4xx error means that you have done something wrong and a 5xx error means that something has gone wrong with the application server you are interacting with.

**25- Explain the main differences between API and Web Service?**

* All web services are APIs but not all APIs are web services.
* A web service uses only three styles of use: SOAP, REST and XML-RPC for communication whereas API may be exposed to in multiple ways.
* A web service always needs a network to operate while APIs don’t need a network for operation.
* Web services might not contain all the specifications and cannot perform all the tasks that APIs would perform.

**26- Who can use a Web API?**

Clients that support HTTP verbs such as GET, PUT, DELETE, POST. Since Web API services do not require configuration, they can be easily used by any client. Portable devices such as mobile devices can easily use Web API, which is undoubtedly the biggest advantage of API.

**27- List the advantages of API Testing?**

* Compatibility and easy integration with GUI: Simple integration would allow new user accounts to be created within the application before a GUI test started.
* Language-Independent: In API testing, data is exchanged using XML or JSON. These transfer modes are completely language-independent that allows users to select any coding language while adopting automation testing services for the project.
* Time Effective: In comparison to functional GUI testing API testing usually is less time-consuming. The web elements in GUI testing must be polled, which makes the testing process slower. Particularly, API test automation requires less code so it can provide better and faster test coverage compared to GUI test automation.
* Test for Core Functionality: API testing provides the ability to access the application without a user interface. The core and code-level functionalities of the application will be tested and evaluated early before the GUI tests. This helps in detecting the minor issues which can become bigger during the GUI testing.

**28- What are the principles of an API test design?**

Basically, there are the five most important principles of an API test design:

**Setup**: Create objects, start services, initialize data, etc

**Execution**: Steps to apply API or the scenario, including logging

**Verification**: Oracles to evaluate the result of the execution

**Reporting**: Pass, failed, or blocked

**Clean up**: Pre-test state

**29- List the common API testing types?**

***Validation Testing***

***Functional Testing***

***UI testing***

***Load testing***

***Runtime/ Error Detection***

***Security testing***

***Penetration testing***

***Fuzz testing***

***Interoperability testing***

**30- What is the procedure to perform API testing?**

***Create the suite to add the API test case***

***Create the test development mode***

***Demand the development of test cases for the required API methods***

***Configure the control parameters of the application and then test conditions***

***Configure method validation***

***Arrange all API test cases***

***Execute the API test***

***Check test reports***

**31- What must be checked when performing API testing?**

***Accuracy of data***

***Non-functional testing like performance and security testing***

***Implementation of response timeout***

***Schema validation***

***HTTP status codes***

***Data type, validations, order, and completeness***

***Authorization checks***

***Error codes in case API returns***

**32- Difference between API Testing and UI Testing?**

API enables communication between two separate software systems. A software system implementing an API contains functions or subroutines that can be executed by another software system.

On the other hand, UI ( User Interface) testing refers to testing graphical interfaces such as how users interact with the applications, testing application elements like fonts, images, layouts, etc. UI testing basically focuses on the look and feel of an application.

**33- What are the types of Bugs API testing can find?**

The types of Bugs, API will find

***Missing or duplicate functionality***

***Stress***

***Reliability***

***Security***

***Performance***

***Unused flags***

***Not implemented errors***

***Inconsistent error handling***

***Improper errors***

***Fails to handle error conditions gracefully***

***Multi-threading issues***

**34- What is API Automation?**

We often need to automate the test cases which are repeatedly executed in every sprint**.**Like regression cases. Similarly, in the case of API testing, there are some cases that we need to execute before every release and those cases should be automated.

There are many tools for API automation like-

*SOUP UI*

*Katalon studio*

*Postman*

*Jmeter*

*RestAssured*

*CloudQA TruAPI*

**35- How we can segregate the entire system into three layers?**

**Presentation Layer –** This is the user interface (GUI) that is open to end-users. QA performs functional testing at this layer.

**Business Layer –** This is the Application user interface where the logic is written. In technical terms, this is where code/algorithm resides. APIs come into the picture at this layer.

**DataBase Layer –**Where application data is present.

**36- List the main differences between SOAP and REST?**

|  |  |
| --- | --- |
| **SOAP** | **REST** |
| 1. SOAP is a protocol through which two computers communicates by sharing XML document. | 1. Rest is a service architecture and design for network-based software architecture. |
| 2. SOAP supports the only XML format. | 2. It supports many different data formats. |
| 3. SOAP does not support caching. | 3. It supports caching. |
| 4. SOAP is like a custom desktop application, closely connected to the server. | 4. A REST client is just like a browser and uses standard methods. An application has to fit inside it. |
| 5. SOAP is slower than REST. | 5. It is faster than SOAP. |
| 6. It runs on HTTP but envelopes the message. | 6. It uses the HTTP headers to hold meta information. |

**37- What is the caching mechanism?**

Caching is a process in which we store server responses at the client end. It allows the server to save significant time from serving the same resource again and again.

The server response holds information that leads a client to perform the caching. It helps the client to decide how long to archive the response or whether not to store it at all.

**38- Is there any upper limit for a payload to pass in the POST method?**

Theoretically, one can pass unlimited data as the payload to the POST method. But, while considering a real use case, then sending a POST with a large payload will consume large bandwidth. It will take more time and cause performance challenges to the server.

**39- State some of the API examples which are very well known.**

**Google Maps API:**

These are designed mainly for mobile and desktop use with the help of flash interface and JavaScript.

**Amazon Advertising API:**

Amazon is known for its products and thus their advertising API accesses their product to discover their functionality and thus advertise accordingly.

**Twitter:**

The API for Twitter is usually in two categories, one for accessing data and the other for interacting with Twitter search.

**YouTube:**

This API used for YouTube includes various functionalities including videos, live streaming, player, etc.

**40- What is REST parameters?**

The REST API has four types of parameters:

**Request parameter**s – These are submitted as JSON parameters present in the request.

**Header parameters** – These are present in the request header.

**Query string parameters** – These are provided at the endpoint of the query.

**Path parameters** – These are provided in the endpoint path.

**What is Postman?**

Postman is a free, HTTP Client based software application primarily used to perform API testing. It supports testing of HTTP requests by utilizing GUI (Graphical User Interface) which can be executed and the responses can be validated. It also helps in collaborating among the team members for the development of API by providing a platform to design, develop, test, and document APIs.

**2. What is a collection in Postman?**

A collection in Postman helps to group similar requests. It helps in systematically arranging the requests into folders.

**3. Why do we use Postman?**

We use Postman for the below reasons:

* Firstly, Postman is free software that is useful for API testing.
* It can send HTTP requests of various types (such as GET, POST, PUT, PATCH, etc) and gives the ability to save environments for future use.
* It helps in managing the end-to-end lifecycle of the API - starting from design to mocking to testing and finally maintaining the APIs.
* It provides Runtime Service that helps in managing API collections, environments, work-spaces, and different examples.
* It can be used to easily integrate with CI/CD tools such as Jenkins.
* Has extensive support from the community and provides extensive documentation.

**You can download a PDF version of Postman Interview Questions.**

[**Download PDF**](javascript:void(0))

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

We can log the variable values in Postman in the console by using the command:  
console.log(pm.variables.get("variable\_name"));

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

It can be accessed by using the variable name as:{{variable\_name}}

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

Postman provides the below API request Authorization Options:

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

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

Postman supports the following type of requests:

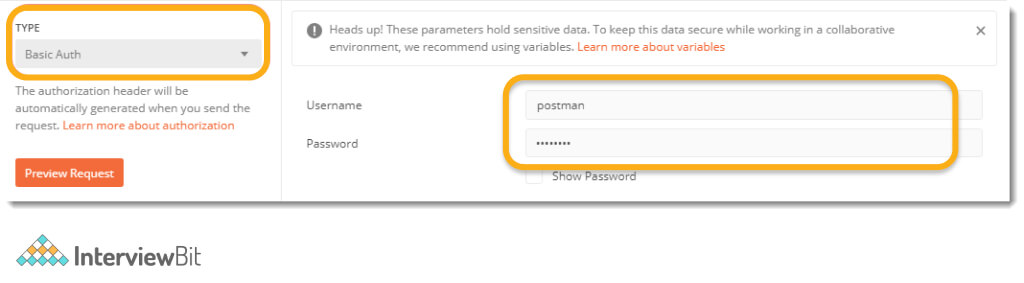
* GET
* POST
* PUT
* PATCH
* DELETE
* COPY
* HEAD
* OPTIONS
* LINK
* UNLINK
* PURGE
* LOCK
* UNLOCK
* PROPFIND
* VIEW

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

Path Variables are used for identifying specific resources and Query Parameters are used for sorting or filtering the resources.

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

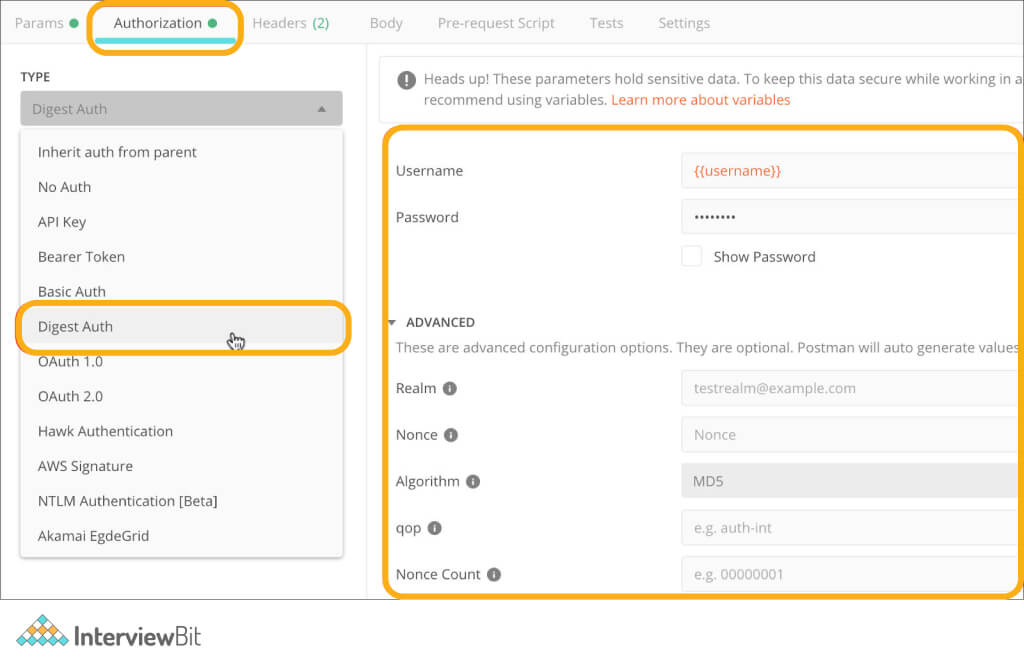
Basic Auth in Postman is a type of authorization technique provided in Postman for HTTP user agents like web browsers. It provides fields to enter **username** and **password** which when entered gets associated with the request.



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

Digest Authorization is one of the authorization techniques provided by Postman. In this technique, the client first sends the request to the API and get responses from the server including a number which is usable only once, a realm value and 401 unauthorized response. We will be then sent back an encrypted data array having both username and password along with the data received from the server earlier. The server uses this data to generate an encrypted data string and compares this with what was sent for authenticating the request.

We can do this by selecting the Authorization tab, then selecting “Digest Auth” from the drop-down list. Postman window presents the fields for both stages of the authentication request. The fields required for the second stage of the request are auto-filled based on the data received from the server. For more information, you can refer [here](https://learning.postman.com/docs/sending-requests/authorization/#digest-auth).



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

Postman accepts authorization credentials in Base64 encoding format only. It is provided in Postman by default. If we do not want to use an inbuilt encoding system, we can refer to third-party websites for converting the credentials in base64 format.

**12. Can we have the same names for global variables in postman?**

The scope of global variables is limited to the workspace and is global. Due to this, variables having global scope cannot have the same names. We can have the same names for local variables but they need to be part of different variables.

**13. What do you know about postman monitor?**

Monitoring is a method of staying in sync with the health and performance of the APIs. Postman provides inbuilt monitoring services that help us be in sync with the API development and performance. The monitors provided by Postman are mainly based on the working of collection runners. They run every request in the collection and analyze the values mentioned in the test scripts. Monitors use the test scripts for validating and monitoring the responses. The reports generated are shared with the developers over emails or alerts in slack, hipchat, etc based on our configuration settings.

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

The binary form is designed to help send data in a format that is not possible to be entered manually. These options are used while sending large files like images, CSV files, etc in the POST request. Binary representation is the easiest representation for sending complex data with the request.

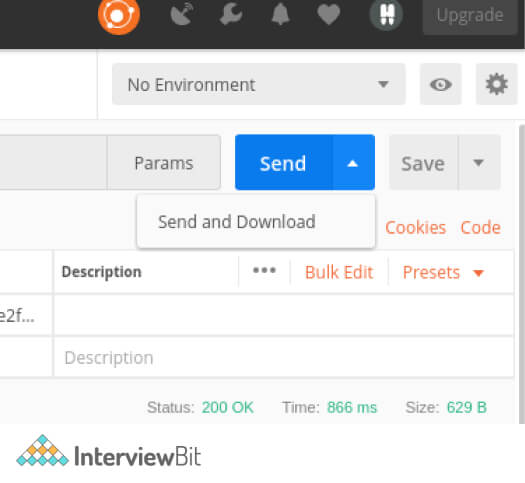
**15. What are the limitations of Postman?**

* Postman is not suitable for processing 1000+ API requests.
* If the project is very large, managing the collections and requests become cumbersome.
* It is not suitable if we want to manage the workspace in the form of code as there would be a lot of code duplication for dynamic API requests.

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

We can do this in two ways:

1. Click on the Download button in the response section.
2. Click on the arrow beside send button - There will be an option to send and download. Clicking on it will prompt Postman to ask the location of saving the response post successful execution of the request.



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

301 status code represents permanent redirect from one website page to another. It tells the search engine that the old page is outdated and the engine has to index the new page URL.

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

The request history can be accessed in the History tab provided on the Postman application. If we sign into the Postman account, then the history will be synced across the devices where you are logged in.

* When you click on any of the requests present in the History tab, the view opens the request that we have saved while we were working on it earlier.
* History also consists of the collection runs that were executed as summarized versions. They are not logged in History as single requests.
* Click on “View More option (…)” on the request, we will see options to save, document, monitor, delete or mock the request.
* Multiple requests can be selected by using Command or Control button and then clicking on the request.

The following image shows the page that shows when you click on View More Option upon a request present in the History tab:

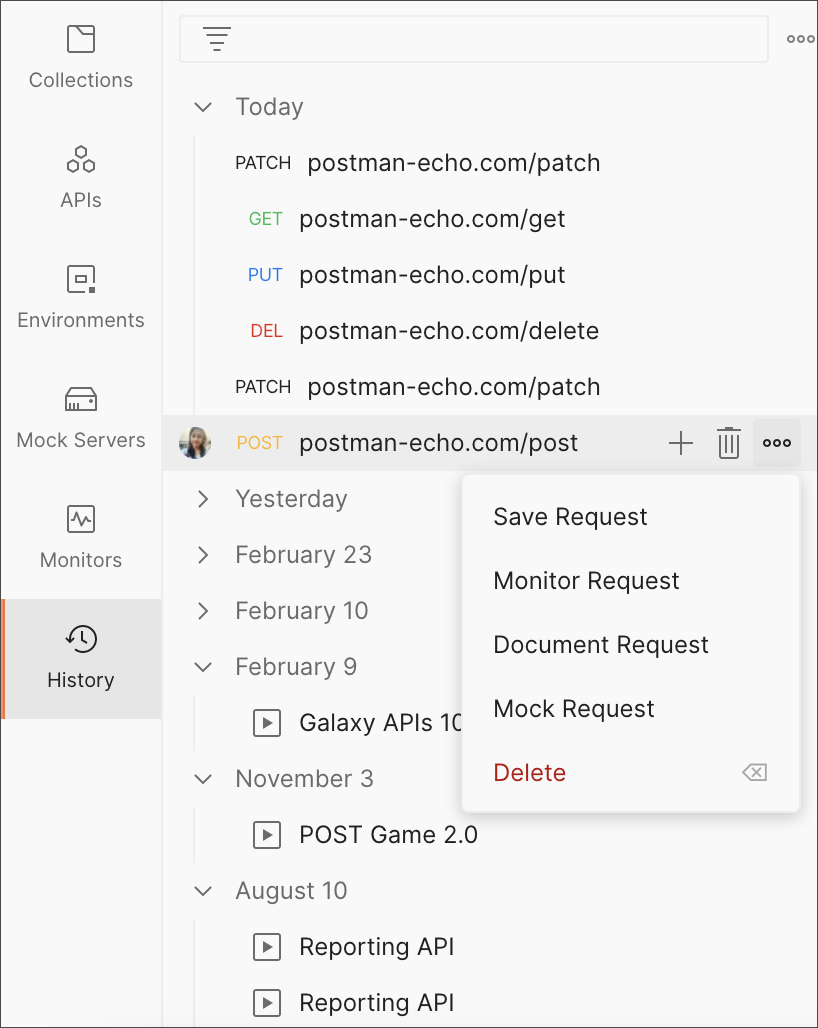


Image Source: Postman Learning Documentation

**Postman Interview Questions for Experienced**

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

Base64 encoding is primarily used because it does the task of data transmission in a textual format that is easier to be sent in the requests in HTML form statistics format. Another reason why we use this is that using identical 64 characters for encoding is heavily reliable in any language we use.

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

304 status code stands for NOT MODIFIED. This is used for diminishing the bandwidth of the network in cases of restricted GET requests. In such cases, the response body should be empty or blank and the headers should have a date, location, signature etc.

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

When working on enterprise-level applications for organizations, it is not preferred to store our work on the Postman cloud because of the required privacy and security. In the Postman cloud, there are chances of security breaches by a skilled hacker.

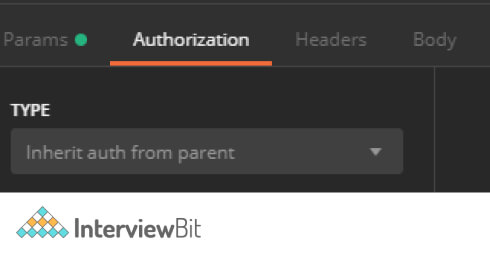
**22. What are the various variable scopes provided by Postman?**

Postman has the following variable scopes:

* **Global Variables:** Global variables allow data access between different collections, requests, and scripts. They are available throughout the workspace.
* **Environment Variables:** These allow us to tailor the requests about different development environments - such as local testing, stage testing, or prod testing.
* **Local Variables:** These are temporary variables that are accessible only within the scope of requests scripts. They are either scoped to a single request or single collection, depending on the requirements. These variables are not available after the completion of the script execution.
* **Collection Variables:** These variables are scoped to be available for all the requests present in a collection. They are independent of the environment.
* **Data Variables:** These variables come from external JSON or CSV files for defining the datasets required to run the collection in Collection Runner or Newman.

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

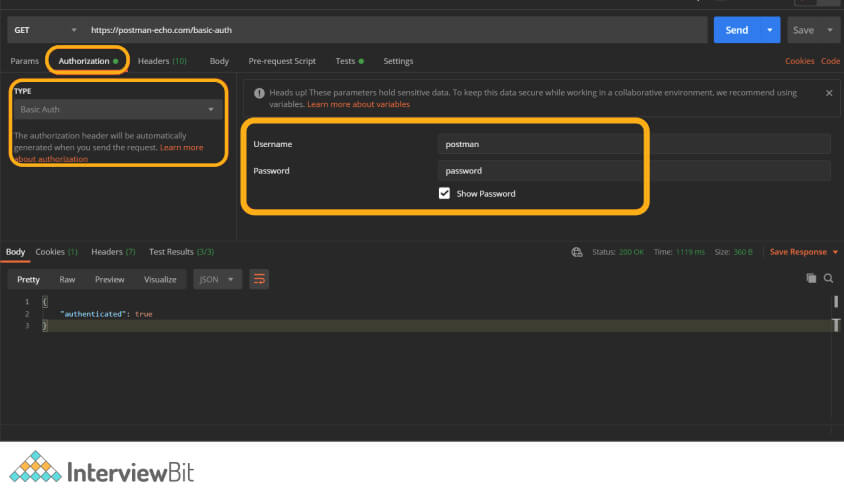
Yes, it is possible. It can be achieved by creating a collection and adding all the requests having the same authentication token to that collection and then assigning the auth token to the collection. This can be applied to the individual requests by selecting the “Inherit auth from parent” option in the Authorization tab.



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

Basic Authentication is one of the authentication techniques provided in Postman that ensures that we can set the username and password along with the API requests. We can do this by first setting the credentials of the API by:

* Navigating to the Authorization tab.
* From the dropdown, select Basic Auth.
* Add the username and password to the API in the input fields given.



The test cases can be written as follows:

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);

});

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

Postman collections allow using pre-request scripts at the individual request level and the collection level. We can add any script that applies to all requests in the collection in the pre-request scripts. We can do it by following the below steps:

* Right-click on the collection, navigate to the pre-request tab.
* Add the below lines of code in the script to add a request header for all the requests present in the collection.

pm.request.headers.add({

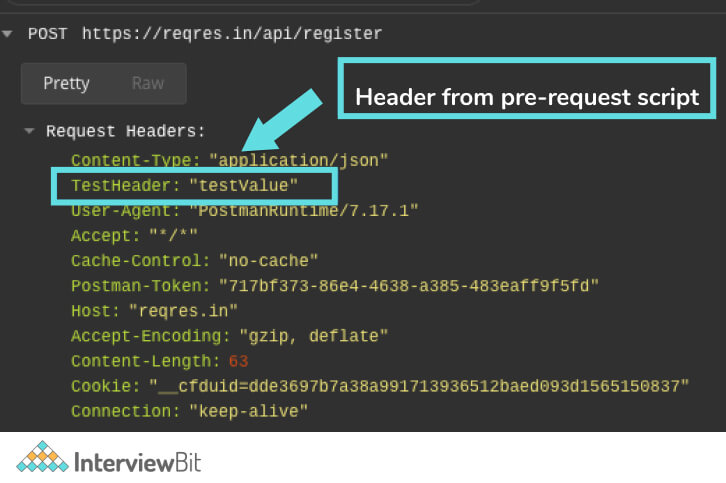
key: 'TestHeader',

value: 'testValue'

});

* Click on Update for saving the script.
* Execute the request in the collection and check if the headers are added to the request in the Postman console.

Header from Pre-Request Script:



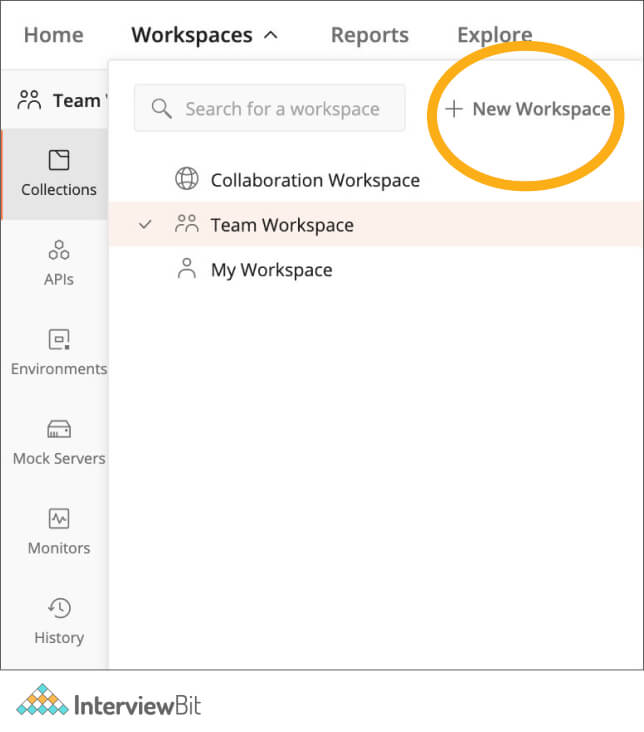
**26. What are workspaces in Postman? What are their uses?**

Workspaces are the areas/space given by Postman for team collaboration to work on a specific or set of collections. It provides a way to logically separate requests or collections that are personal to the developer or the team so that the maintenance of requests is made easy.

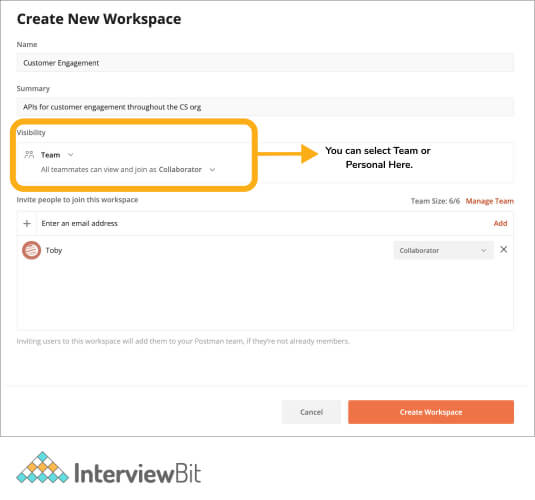
There are two types of workspaces in Postman:

* **Personal Workspace:**
  + These workspaces are useful when we are working simultaneously on multiple projects and we require logical separation between the requests to handle the requests better.
* **Team Workspace:**
  + These are created for team collaboration so that more than one person can be part of testing requests.
  + We can invite new users for collaborating on our collections by sharing the email id of the users. Once the invite is accepted, the new users can start contributing to the workspace by adding or modifying the requests.

We can create a new workspace by clicking on the Workspace icon and then clicking on “Create New”.

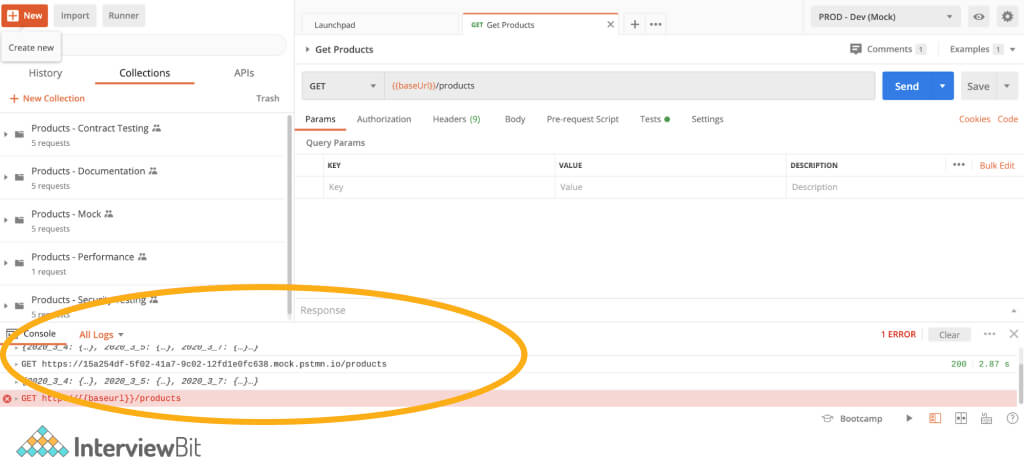


We can select our workspace to be personal or team workspace by configuring the properties in the create window.



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

Postman does allow viewing of requests and response parameters in the software application itself. But it is important to see how the request was sent upon applying the pre-request scripts. In such cases, Postman has an additional tool called “Postman Console” which is used for viewing every request and response detail. We can also log the details in the console by using console.log statements in the scripts.



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

We can use the below code to stop the execution of the next request: pm.setNextRequest(null);

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

Pre-request scripts are those scripts that are used for executing Javascript code before a request is run. It is used for performing pre-processing tasks like setting variables, parameters, headers, body data, etc.

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

Postman provides a lot of built-in tools and libraries that we can use to add in our pre-request or post-request scripts or test cases. Let us take the example of using the moment.js library. It provides a lot of useful functions to format data around time. Consider that we have a POST request that needs to specify the created date to the user which expects the format “DD/MM/YYYY”. We can use the moment library to perform this using a single line of code. In our pre-request script, we need to add the below lines of code to get the correctly formatted data and then store that in an environment variable:

var moment = require('moment');

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

There are a lot of other useful libraries like crypto.js that are useful for converting text to encrypted values which can further be used anywhere in the request body.

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

In such cases, the higher precedence is given to the local variable by overwriting the value of the global variable.

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

Postman provides a command-line tool called Newman using which we can run any Postman collection. It is a NodeJS based package that requires a node environment for executing collections using Newman Collection Runner. It has full parity with Postman’s Collection Runner i.e it provides support for running assertions, pre-request scripts, or other request scripts linked with the requests that belong to the collection.

We can use Newman by following the below steps:

* Install Node
* Install Newman package using npm command as: npm install -g newman
* To run the collection, first export the environment to JSON format in Postman. Then run the below command for running the collection in Newman:

newman run {{path to collection json}} -e {{path to environment json}}

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

Suppose you want to generate numbers between the range 1 to N, then it can be done in the pre-request script as follows:

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

We can then use this variable in the URL as:{{randomNumber}}

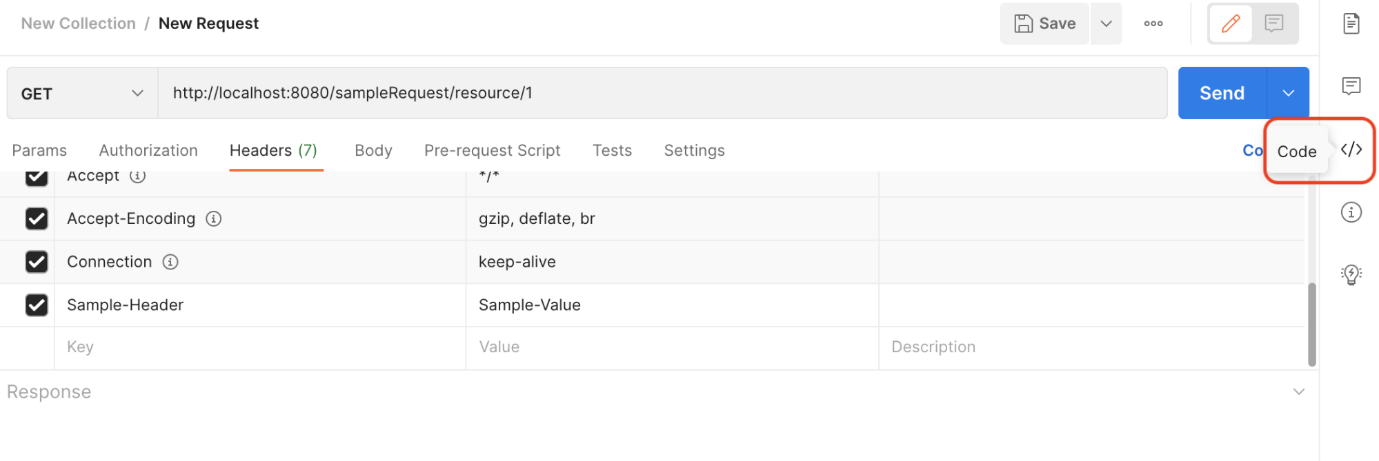
**34. What do you understand by Scratch Pad?**

Scratch Pad is a space provided by Postman that helps us to work without being connected to Postman servers. It provides the flexibility of utilizing some of the features of postman offline. The features include- collection creation, creating requests and the ability to send requests. These are stored locally and once logged in, the work is saved into the workspace.

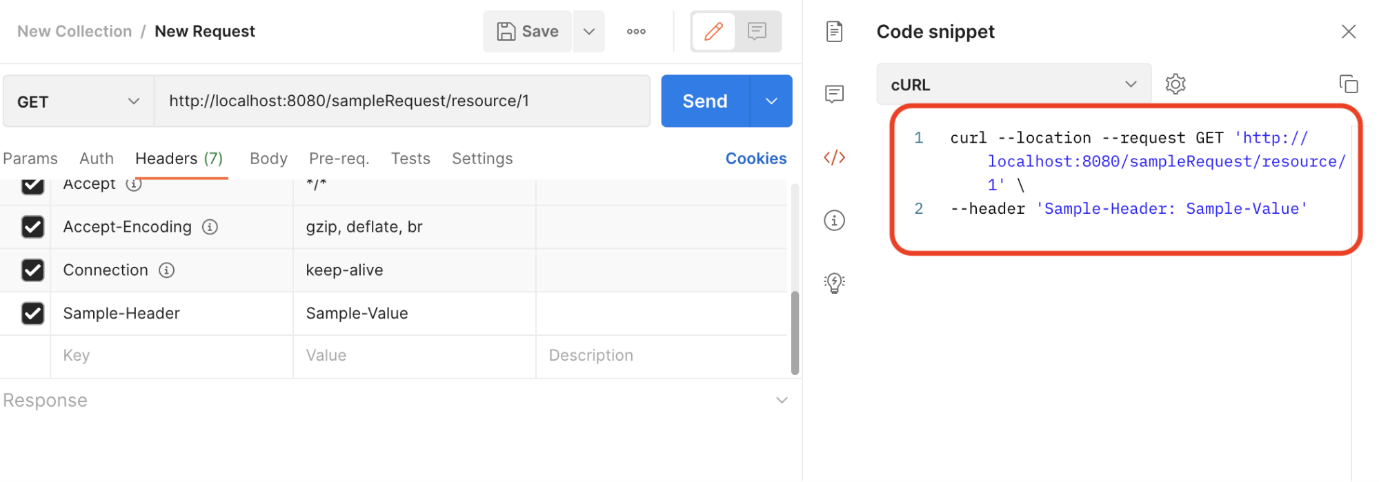
**35. How do you get the cURL command based on the details of REST API obtained from Postman?**

We can obtain the cURL command equivalent by following the below steps:

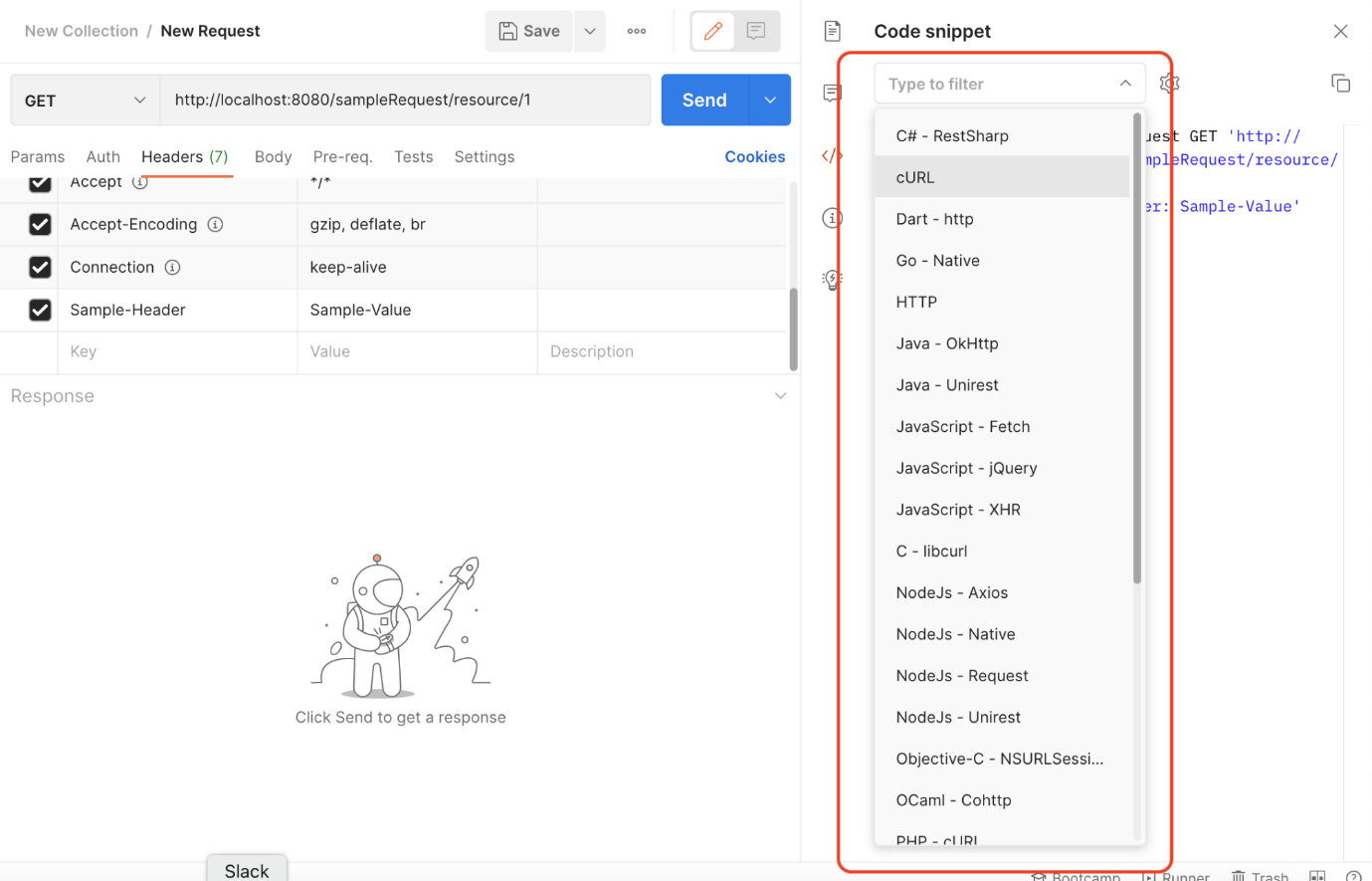
* Click on the Code icon as shown below:



* You will get the cURL command of the REST API request sent as shown in the screenshot below:



* We can also get the command for the request in different languages like C#, Javascript, NodeJs, PHP, etc by selecting the option that we require from the drop-down as shown in the image below:



**Conclusion**

Postman has become the most popular and beloved tool for API testing. Hence, hands-on experience with the Postman tool has become very important for any backend developer.

**Recommended Resources:**

* [API Testing](https://www.interviewbit.com/api-testing-interview-questions/)
* [REST API](https://www.interviewbit.com/rest-api-interview-questions/)
* [Software Testing](https://www.interviewbit.com/software-testing-interview-questions/)
* [Web Services](https://www.interviewbit.com/web-services-interview-questions/)
* [JMeter](https://www.interviewbit.com/jmeter-interview-questions/)

**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 QuestionsPostman 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.

EXPLORE MORELearn Java Programming with Beginners Tutorial08:32Linux Tutorial for Beginners: Introduction to Linux Operating...01:35What is Integration Testing Software Testing Tutorial03:04What is JVM (Java Virtual Machine) with Architecture JAVA...02:24How to write a TEST CASE Software Testing Tutorial01:08Seven Testing Principles Software Testing05:01Linux File Permissions Commands with Examples13:29How to use Text tool in Photoshop CC Tutorial08:32What is NoSQL Database Tutorial02:00Important Linux Commands for Beginners Linux Tutorial15:03

**6) What 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.
* Authorization always comes first, while authentication comes after authorization.
* 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 automatically 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 as it requires sign-in. Therefore, saving your work in the Postman cloud is not advisable.

**31) What are the standard rules of an API test design?**

Here are the key principles of an API test design:

1. **Setup:** Create objects, start services, and initialize data.
2. **Execution:** Apply API or the scenario, including logging
3. **Verification:** It is use for evaluating the result of the execution
4. **Reporting:**Indicates Pass, failed, or blocked status
5. **Clean up:** Pre-test state

**32) What is the Team workspace in Postman?**

A Team workspace is a synergetic environment for users where many programmers may develop APIs, provides feedback, and collaborate on the same pools of requests. It also helps to synchronize and collaborates all the team’s work in one place.

**33) What is the 301-status code?**

301 status code represents a permanent redirect from one website page to another. It tells the search engines that the old page has become outdated, and the engine has to index the new page or URL.

**34) What is status code 201?**

Status code 201 is created only when a resource is successfully created using a PUT or POST request. It returns a link to a newly created one with the help of the location header.

**35) What is the procedure to remove local variables?**

The local variables can be automatically removed when you have executed and completed the tests.

**36) What is Pre-Request Script in Postman?**

In Postman, a pre-request script is a set of rules that runs before executing a request.

**37) How are Query Parameters different from Path Variables?**

In Postman, Path Variables are used to identify specific resources, and Query Parameters are used to sort or filter the resources.

**38) What are the main drawbacks of Postman?**

Here are some cons/ disadvantages of using Postman:

* Postman cannot process more than 1000 API requests.
* It is not easy to manage the collections and requests in a massive size project.
* Postman is not an ideal API tool for workspace management in the form of code as there can be lots of code duplication while handing the dynamic API requests.

**39) What are some of the JS libraries available in Postman?**

Some JS libraries available in Postman are

1) Lodash  
2) Moment  
3) GUID

**40) What is GUID?**

GUID is short for Global Unique Identifier. It is hexadecimal digits that are separated by hyphens. This Postman identifier GUID solves the purpose of uniqueness.

**41) How can you view log requests and responses in Postman?**

You can use the Postman Console window to view request logs and response logs.

**42) What is the importance of setNextRequest in Postman?**

setNextRequest helps you to define the workflow. It is needed to change the order of the requests being executed.

**43) What test code allows you to check whether the response status is 200 or not?**

Following is a test code to check whether the response status is 200 or not:

tests[“Status Code is 200”] = responseCode.code === 200;

**44) What do you understand by ScratchPad?**

Scratch Pad is a place which is provided by Postman that helps you work without the need to connect with Postman servers.

**45) How can you iterate a request 100 times in Postman?**

You can iterate a request 100 times in Postman by using Collection Runner.

**46) What would happen if {{$randomInt}} dynamic variable is added?**

You need to add a random integer between 0 and 1000.

**47) How do you access variable values from a file inside pre-request and test scripts?**

data.var\_name

data['var\_name']

**48) Can you read the Postman Chrome application to read and write cookies?**

No, it is impossible to read and write cookies using the app.

**49) Which programming language is used for Postman tests?**

JavaScript is used for Postman tests.

**50) Which tool can be used to run Postman Collections in Jenkins?**

Newman can be used to run Postman Collection in Jenkins.

What is Postman?

Postman is a free HTTP client-based software application and a collaboration platform for API development. It is mainly used to perform API testing. It is a very popular API client which facilitates developers and provides a platform to design, build, share, test, and document APIs. Postman supports testing of HTTP requests by utilizing GUI (Graphical User Interface), which later we can execute and validate the responses.

The Postman tool also facilitates us to send HTTP/s requests to a service and get their responses. We can ensure that the service is up and running by using this.

2) What was the Postman originally developed?

When the Postman was developed, it was originally a Chrome browser plugin. It extends its solution with the native version for both Mac and Windows.

3) Why do we use Postman?

The Postman tool has become a choice of more than 8 million users. Following are the main reasons for using Postman:

Play Video

* **It is free:** Postman is free software that we can use for API testing. It is free to download and use for teams of any size.
* **It is easy to use:** Postman is an easy-to-use software tool. We can send HTTP requests of various types (such as GET, POST, PUT, PATCH, etc.). We have to download it, and we can send our first request in minutes. It also gives us the ability to save environments for future use.
* **Community & Support:** It has a huge community forum for customer support and extensive documentation.
* **It is extensible:** Postman facilitates us customizing it according to our needs with the Postman API.
* **APIs Support:** It facilitates us to make any API call (REST, SOAP, or plain HTTP) and easily inspect even the largest responses. It also helps manage the end-to-end lifecycle of the API - starting from design to mocking to testing and finally maintaining the APIs.
* **Runtime Services:** Postman provides Runtime Services that help us manage API collections, environments, work-spaces, and different examples.
* **Integration:** Postman facilitates us to easily integrate test suites into our preferred CI/CD tools and services, such as Jenkins with Newman (command-line collection runner).

4) What is an API?

API is an acronym that stands for "Application Programming Interface". It is a set of routines, protocols, and tools used for building Software applications. API is an interface, so it specifies how one software program should interact with other software programs.

In other words, we can say that API is an Application Programming Interface that acts as an interface between two software applications and allows these two software applications to communicate with each other. API is a collection of software functions that another software program can execute.

5) What are the various authorization methods provided by Postman?

Postman provides the following API request authorization options:

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

6) What is the use of the collection in Postman?

In Postman, a collection is used to group similar requests. It systematically arranges the requests into folders.

7) What are the various tools used for API Testing?

Following is a list of some tools that are used for API Testing:

* Postman
* SoapUI
* Katalon Studio
* Tricentis Tosca
* Apigee
* Jmeter etc.

8) How can you access the Postman variables?

We can access the Postman variables by using the variable name as:{{variable\_name}}

9) What are the different types of API requests supported in Postman?

Following is a list of the different types of API requests supported in Postman:

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

10) What is an HTTP request? Give an example of an HTTP request.

An HTTP request is a programming request made by the client to a named host located on a server. HTTP works as a communication interface or a request-response protocol between a client and server. The main aim of the HTTP request is to access a resource on the server. To make the HTTP request, the client uses components of a URL (Uniform Resource Locator), which includes the information needed to access the resource.

**An example of the HTTP request:**

A client (browser) sends an HTTP request to the server; then, the server responds to the client. The response sent by the server contains the status information about the request, and it can also contain the requested content.

11) What are the core components of an HTTP request?

Following are the five core components of an HTTP request:

* **HTTP methods:** It is a set of request methods used to perform needed action for a given resource (GET, PUT, POST, and DELETE).
* **Uniform Resource Identifier (URI):** It is a kind of address that describes the resource.
* **HTTP Version:** It specifies the version of the HTTP. For example HTTP v1.1
* **Request Headers:** It specifies the content type and content length of the request. For example: **Content-type:** application/ JSON, Content-Length: 511
* **Payload:** It is used to specify the Request Body that includes message content.

12) Why does Postman accept Base64 encoding only?

Postman accepts Base64 encoding only because it transmits the data into the textual form and sends it in an easier form, such as HTML form data. This is also preferred because we can rely on the same 64 characters in any encoding language we want to use.

13) Which type of encoding does Postman accept authorization credentials and why?

Postman accepts authorization credentials in Base64 encoding only. This is inbuilt in Postman, or else you can refer to a third-party website to convert the credentials in base64. The Base64 authorization credentials are generally used because they transmit the data into a textual form and send it in an easier form, such as HTML form data.

14) What are the core components of an HTTP Response?

Following are the four core components of an HTTP Response:

* Status/Response Code: The server generates the status or response codes when the client makes a request. For example, 404 means Page Not Found, and 200 means Response is OK.
* HTTP Version: It describes the HTTP version. For example: HTTP v1.1, HTTP v1.3 etc.
* Response Header: The Response Header includes the information for the HTTP response message. For example, Content-type, Content-length, date, status, and server type.
* Response Body: The Response Body contains the client's data requested from the server.

15) What is the meaning of the term environment in Postman?

In Postman, the term environment is a set of key-value pairs. You can create multiple environments in Postman and switch among them quickly by pressing a button. There are 2 types of environment, global and local.

16) Can we have two global scope variables with the same name in Postman?

The global variables are global, so we cannot set duplicate names for them without any environment as it creates confusion for the software. On the other hand, local variables can have the same name but in different environments.

Learn more

17) Which one is preferred in Postman, a global or local variable?

In Postman, if 2 variables have the same name (one being local, the other global), then the higher priority is of the local variable. The local variable will overwrite the global variable.

18) How can you log variable values in Postman?

In Postman, we can log the variable values in the console by using the following command:

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

19) What is the use of Postman monitor?

The Postman monitor is used for running collections. Collections run till the specified time defined by the users. It requires the users to be logged in, and the users share the Monitor reports over an email on a daily or monthly basis.

20) What do you understand by team workspace in Postman?

As we know, a workspace is a collaborative environment for users to develop and test APIs. In the same way, a team workspace is a workspace that is shared by the whole team working on the same collections of requests. Usually, it is time-consuming and hard to share the collections through external drives or other sharing; the team workspace synchronizes and collaborates all the team's work in one place.

21) What is the difference between Query Params and Path Variables?

Query Params or Query Parameters are used for sorting or filtering the resources. On the other hand, Path Variables are used for identifying specific resources.

22) What is the use of Postman Collection runners?

The Postman Collection runner is used to perform Data-driven testing. It runs a group of API requests for multiple iterations with different data sets.

23) What is Basic Auth in Postman?

In Postman, Basic Auth is an authorization technique provided for HTTP user agents like web browsers to enter username and password. After entering the username and password, it gets associated with the request.

24) Can we import local variables in Postman Monitors?

Yes, we can import local variables in Postman Monitors, but it is not allowed to import global variables in Postman Monitors.

25) What are the main limitations of Postman?

Following is the list of key limitations of Postman:

Pause

Unmute

* Postman cannot process 1000+ API requests.
* In the case of huge projects, it isn't easy to manage the collections and requests.
* Postman is not suitable for managing the workspace in the form of code. This is because there would be a lot of code duplication for dynamic API requests.

26) What is a binary form in POST methods? What is its usage?

In POST methods, the binary form is designed to send data easily in a format it is impossible to enter data manually. This is mainly used when sending large files like images, CSV files, etc., in the POST request. The binary representation is one of the easiest representations used for sending complex data with the request.

27) What is the use of Postman cloud when we are working in a company?

The Postman cloud is a common repository of companies to access Postman collections. That is why we use it when we are working in a company. We can save the work instantly in the Postman cloud after logging in. It facilitates the team members to access data/collections from anywhere.

28) What is digest auth in Postman?

The digest auth or digest authorization is one of the authorization techniques provided by Postman. This technique lets the clients send the request first to the API and get responses from the server, including a number that can be used only once a real value and 401 unauthorized responses. After that, the client can send back an encrypted data array with both username and password and the data received from the server earlier. Now, the server uses this data to generate an encrypted data string and compares this with what was sent for authenticating the request.

29) Why is it not suggested to save work in Postman cloud?

It is not suggested and is not preferred to save your work in the Postman cloud as the company's work is not allowed to be leaked and remain confidential. While saving your work on Postman cloud, it may cause a security breach as it requires sign-in. That's why the Postman Cloud is not suggested for saving work, and team workspace is highly preferred.

30) What do you understand by status code 201?

The status code 201 means created when you have successfully created a resource using POST or PUT request. It returns a link to a newly created resource using the location header.

31) What is the procedure to remove local variables?

The local variables are automatically removed once you have executed and completed the tests.

32) What is the procedure to save the responses of API to a file in Postman?

In Postman, there are two ways to save the responses of an API to a file:

* First, click on the download button in the response section.
* Second, click on the arrow next to the send button. Here, you will see an option to send and download. After clicking on it, you will get a prompt Postman asking for the location of saving the response post successful request execution.

33) What is the meaning of the status code 304?

The status code 304 means NOT MODIFIED. It is used to minimize the network bandwidth usage in conditional GET requests. The response body should be empty. Headers should have a date, and location, etc.

34) What is the use of the 301 status code in Postman?

In Postman, the 301 status code is used to specify that the page has been permanently redirected from one website page to another. It tells the search engine that the old page is outdated, and the search engine has to index the new page URL.

35) What is the method to organize requests in Postman?

We can organize requests in Postman with the Collections.

36) What are the various variable scopes provided by Postman?

Following is a list of several variable scopes provided by Postman:

* **Global Variables:** Global variables allow data access between different collections, requests, and scripts. They are available throughout the workspace.
* **Local Variables:** Local variables are the temporary variables that can be accessed only within the scope of requests scripts. Depending on the requirements, these variables are either scoped to a single request or single collection. These variables are not available once the script execution is completed.
* **Environment Variables:** The Environment variables allow us to tailor the requests about different development environments such as local testing, stage testing, or prod testing.
* **Collection Variables:** The Collection variables are independent of the environment and scoped to be available for all the requests present within the collection.
* **Data Variables:** The Data variables come from external JSON or CSV files and define the datasets required to run the collection in Collection Runner or Newman.

37) What is the difference between the form data and x-www-form-urlencoded?

The key difference between the form data and x-www-form-urlencoded is that the URL will be prearranged when sent through x-www-form-urlencoded.

38) In which conditions should we use global, collection, and local variables?

Global variables are the general-purpose variables. They are used while passing data to other requests and are ideal for quick results and prototyping.

Collection variables are generally used for storing some constants that do not change during the execution of the collection. These variables are used for constants that do not change during the execution and URLs / authentication credentials if only one environment exists.

Local variables are the temporary variables and are only available within the request that has set them or when using Newman/Collection runner during the script execution. These variables are used whenever we have to override all other variable scopes.

39) How can we access the history of requests in Postman?

In Postman, we can access the history of requests in the History tab provided on the Postman application. When we sign into the Postman account, the history will be synced across the devices where you are logged in. After clicking on any of the requests present in the History tab, the view will open the request that we saved while working on it earlier. History also contains the collection runs that were executed as summarized versions.

When you click on the "View More option" on the request, you will see options to save, document, monitor, delete or mock the request. You can also select multiple requests by using Command or Control button and then clicking on the request.

40) How can we stop executing requests or stop the collection run?

We can stop executing requests or stop the collection run by using the following command:

1. postman.setNextRequest(null);

41) Is it possible to reuse the authentication token for multiple requests?

Yes, it is possible to reuse the authentication token for multiple requests. We can achieve it by creating a collection and adding all the requests having the same authentication token to that collection and then assigning the auth token to the same collection. We can apply it to the individual requests by selecting the "Inherit auth from parent" option in the Authorization tab.

42) Where are the query parameters stored in a GET request in Postman?

In the GET request, the query parameters are stored in the URL in Postman.

43) What is executed first in a collection run?

The pre-request scripts at the compilation level are executed first in a collection run.

44) How can we access a Postman variable?

We can access a Postman variable by entering the variable name as {{var}}.

45) Which tool can be used to run Postman collections in Jenkins?

We can use the Newman tool to run Postman collections in Jenkins.

46) What are the most commonly seen status codes in Postman?

Following is the list of most commonly seen status codes in Postman:

* **200 (OK):** Status code 200 specifies that the request was correct.
* **201 (Created):** Status code 201 specifies that the value wrapped with the request has been created in the database. Here, it is obvious that the request was correct.
* **204 (No Content):** Status code 204 specifies that the request was correct and received, but there is no content to send. It means there is no response to send to the client by the server.
* **400 (Bad Request):** Status code 400 specifies a bad request. A bad request means that the syntax of the request was incorrect. It appears when we have sent the wrong parameters along with the request URL or in the request's body.
* **401 (Unauthorized request):** Status code 401 specifies an unauthorized request. An unauthorized request is a request for which you are not authorized. This status code appears when we are not authorized to access the server or enter the wrong credentials.
* **404 (Not Found):** Status code 404 specifies that the server was connected, but it could not find what was requested. It simply means "request not found". This status code normally appears when we request a web page not available on the server.

47) What is Scratch Pad?

Postman provides a space known as Scratch Pad. The Scratch Pad facilitates us to work without being connected to Postman servers and also provides the flexibility to utilize some of the features of Postman offline. These features include collection creation, creating requests, and the ability to send requests. The Scratch Pads are stored locally, and once logged in; the work is saved into the workspace.

48) What are the different Request Method types in Postman?

The different Request Method types in Postman are as follows:

Pause

Unmute

* Get
* Post
* Put
* Delete
* Patch
* Head
* Delete

49) How can you iterate a request 100 times in Postman?

We can iterate a request 100 times in Postman using Collection Runner.

50) What are some examples of the JS libraries available in Postman?

Some examples of the JS libraries available in Postman are Lodash, Moment, GUID, etc.

51) What is the full form of GUID? Where is it used?

GUID is an acronym that stands for Global Unique Identifier. It is a set of hexadecimal digits separated by hyphens and solves the purpose of uniqueness.

In Postman, it is used to generate and send a random value to APIs.

**Example:**

1. {
2. "id": "{{$guid}}",
3. }

52) Which programming language is used for Postman tests?

JavaScript is used for Postman tests.

53) How can you generate random numbers of a given range in Postman?

In Postman, we can generate random numbers of a given range using some script. Suppose you want to generate numbers between the range of 1 to N, then use the following pre-request script:

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

Now, use this variable in the URL as {{randomNumber}}

54) How can we view request logs and response logs in Postman?

We can view request logs and response logs in Postman through the Postman Console window.

55) What is the binary form in the Post method in Postman?

In Postman, the binary form is designed to send the information in a specific format that cannot be entered manually. For example, images, files, etc. Everything in a computer is converted to binary.

56) How can you organize requests in Postman?

In Postman, we can use Collections to organize requests.

57) Which method is preferred to write the tests?/ JavaScript or Functional?

The functional method is advised and recommended while writing tests in Postman. Although we can use JavaScript methods, there has been no notice of ending the support for the JavaScript method.

**What is Postman?**

Postman is a collaboration platform for API development. It is a popular API client and it enables you to design, build, share, test, and document APIs.

Using the Postman tool, we can send HTTP/s requests to a service, as well as get their responses. By doing this we can make sure that the service is up and running.

Being originally a Chrome browser plugin, Postman now extends its solution with the native version for both Mac and Windows.  
  
**2. Why Postman?**

Postman has become a tool of choice for over 8 million users.

* **Free:** It is free to download and use for teams of any size.
* **Easy:** Just download it and send your first request in minutes.
* **APIs Support:** You can make any kind of API call (REST, SOAP, or plain HTTP) and easily inspect even the largest responses.
* **Extensible:** You can customize it for your needs with the Postman API.
* **Integration:** You can easily integrate test suites into your preferred CI/CD service with Newman (command line collection runner)
* **Community & Support:** It has a huge community forum

**3. What is an API?**

API is an acronym and it stands for Application Programming Interface. API is a set of routines, protocols, and tools for building Software Applications. APIs specify how one software program should interact with other software program.

In simple words, API stands for Application Programming Interface. API acts as an interface between two software applications and allows the two software applications to communicate with each other. API is a collection of software functions which can be executed by another software program.

***Must Read:***[***API Testing Complete Tutorial***](https://www.softwaretestingmaterial.com/api-testing/)

**4. Name some tools used for API Testing?**

Some of the tools used to do API Testing are as follows

* Postman
* Katalon Studio
* SoapUI
* Tricentis Tosca
* Apigee
* Jmeter

**5. What are the core components of an HTTP request?**

An HTTP request includes five key elements:

* HTTP methods – Set of request methods to perform desired action for a given resource (GET, PUT, POST, DELETE)
* Uniform Resource Identifier (URI) – Describes the resource
* HTTP Version, (example- HTTP v1.1)
* Request Headers, (example- Content-type : application/json, Content-Length : 511)
* Payload – It is basically a Request Body which includes message content.

**6. State The Core Components of an HTTP Response?**

Every HTTP response contains four key elements.

* Status/Response Code – These are response codes issued by a server to a client’s request. For example, 404 means Page Not Found, and 200 means Response is OK.
* HTTP Version – describes HTTP version, for example-HTTP v1.1.
* Response Header – Includes information for the HTTP response message. For example, Content-type, Content-length, date, status and server type.
* Response Body – It contains the data that was requested by a client to server.

**7. What API information is exposed in Web Developer tools?**

Request headers, Response body, Response cookies  
  
**8. What can we use to get API information from web developer tools into Postman?**

Copy as cURL can get API information from web developer tools into Postman.

**9. In which type of encoding does postman accept authorization credentials?**

Postman accepts Base64 encoding only. This is provided inbuilt in postman or else you can also refer 3rd party websites to convert the credentials in base64.

**10. Why does Postman accept Base64 encoding only?**

We use base64 particularly because it transmits the data into the textual form and sends it in easier form such as HTML form data. Also, we can rely on the same 64 characters in any encoding language that we use.

**11. What is meant by the term environment in postman?**

An environment in postman is a set of key value pairs. You can create multiple environments in postman which can be switched quickly with a press of a button. There are 2 types of environment, global and local.

**12. Can global scope variables have duplicate names in postman?**

Since global variables are global i.e. without any environment, global variables cannot have duplicate names. Local variables can have the same name but in different environments.

**13. Which one will be preferred in postman- a global variable or a local variable?**

In postman, if 2 variables have the same name( one being local, other global) then the higher priority is of the local variable. it will overwrite the global variable.

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

A Postman Collection lets us group individual requests together. Simply it allows us to organize the requests into folders.

**15. What do you mean by postman monitors?**

The postman monitor is used for running collections. Collections are run till specified time defined by the user. Postman Monitor requires the user to be logged in. Monitor reports are shared by users over email on a daily/monthly basis.

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

A postman collection runner is used to perform Data-driven testing. The group of API requests are run in a collection for the multiple iterations with different sets of data.

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

Yes. Postman monitors allow to import local variables but it does not allow to import global variables.

**18. What is the purpose of Postman cloud if we are working in a company? Why?**

A Postman cloud is a common repository of companies to access Postman collections. In Postman cloud, work can be saved instantly after logging in. Anyone from the team can access data/collections from anywhere.

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

It is not preferred to save your work in Postman cloud as company’s work is not allowed to be leaked and remain confidential. Security breaches can be experienced if Postman cloud is used as Postman cloud requires sign in. Therefore Postman Cloud is discouraged for saving work and team workspace is highly encouraged.

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

It means NOT MODIFIED. It is used to reduce network bandwidth usage in case of conditional GET requests. Response body should be empty. Headers should have date, location etc.

**21. Define status code 201?**

It means created, when a resource is successfully created using POST or PUT request. It returns a link to a newly created resource using the location header.

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

**Global variables** are general purpose variables, ideal for quick results, and prototyping. They are used while passing data to other requests.

**Collection variables** can be mostly used for storing some constants that do not change during the execution of the collection. They are used for constants that do not change during the execution and also for URLs / authentication credentials if only one environment exists.

**Local variables** are only available within the request that has set them or when using Newman/Collection runner during the entire execution. They are used whenever you would like to override all other variable scopes.

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

Local variables are automatically removed once the tests have been executed.

**24. How can we stop executing requests or stop the collection run?**

postman.setNextRequest(null);

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

The difference between the form data and x-www-form-urlencoded is that the url will be encoded when sent through x-www-form-urlencoded.

**26. Where are query parameters stored in a GET request?**

Query parameters are stored in the URL in a GET request.

**27. How can we access a Postman variable?**

We can access a Postman variable by entering the variable name as {{var}}

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

400 Bad Request is an ideal response code for request with incorrect parameters.

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

By using Collection Runner

**30. How can we organize requests in Postman?**

We can organize requests in Postman with the Collections.

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

JavaScript

**32. What will execute first in a Collection Run?**

Pre-request scripts at the Collection level are executed first in a Collection run.

**33. What are some of the JS libraries available in Postman?**

Lodash, Moment, GUID

**34. Which tool can be used to run Postman Collections in Jenkins?**

Newman can be used.

**35. How can we log requests and responses in Postman?**

We can view requests logs and response logs through the Postman Console window.

**36. What is GUID?**

GUID stands for Global Unique Identifier. It is basically hexadecimal digits separated by hyphens. GUID solves the purpose of uniqueness.

In Postman, we use this to generate and send a random value to APIs.

**#1) How can you set headers for all the requests that are in a particular Postman collection?**

**Answer:**Postman collections allow adding pre-request scripts at both the collection as well as individual request level. To add any script that applies to all the requests that are present in the collection, we will need to add a pre-request script at the collection level.

**Please follow the below steps to** **add a collection level pre-request script for adding a header to all the requests.**

**a)** Open collection options by right-clicking the collection and navigate the pre-request script tab.  
**b)** Now add the below script for adding a request header for all the requests.

pm.request.headers.add({

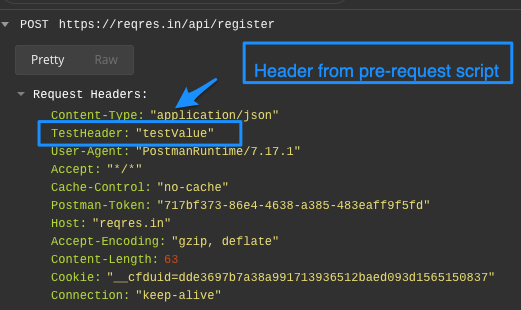
    key: 'TestHeader',

    value: 'testValue'

});

**c)** Click **Update** to save the collection level pre-request script.

**d)** Now execute any request in the collection (directly or through collection runner) and view the request details in the Postman console debugger to validate if the pre-request script is working fine and adding the specified header.

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2019/10/postman-preRequestScriptCollection.png)

**Q #2) What’s the use of Workspaces in Postman?**

**Answer:**Postman workspaces are nothing but collaboration areas or space for one or many people to work on the same collection or set of collections. It’s a way to logically separate the collections or requests from each other.

In other words, it is simply an abstraction in terms of logical separation of requests.

**2 types of Workspaces are supported by Postman i.e. Team, and Personal.**

**#1)** **Team Workspaces** are created for collaborating with multiple people that are a part of the same team. Look at it from the perspective of a common shared repository in git, where anyone can pull the repository code and contribute.

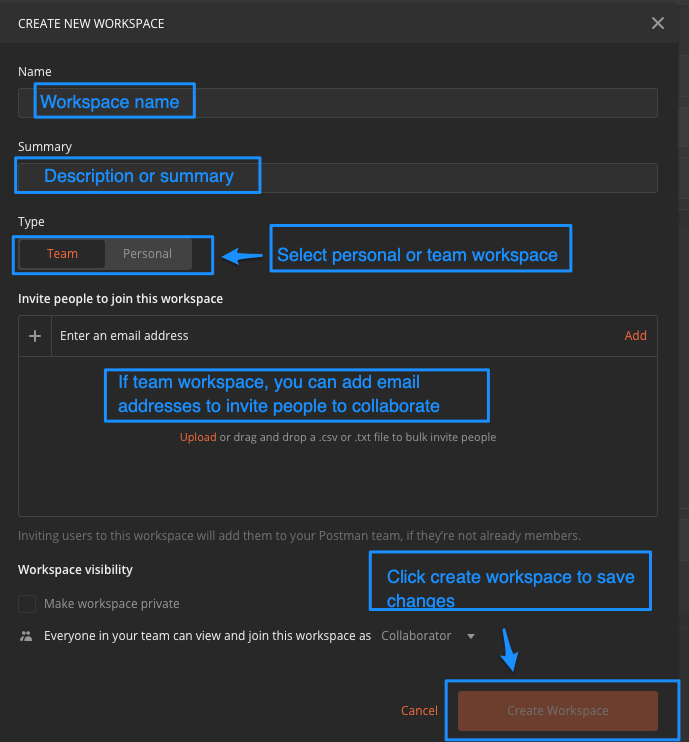
Similarly, for all the people who are part of the team, the workspace gets shared and everyone can contribute. You can also invite new users to collaborate with your collection by sharing their email id and when someone joins or accepts that invite they will be able to collaborate with that collection.

**#2)** **Personal Workspaces** are a way to logically separate collections (or projects) from one another. These are useful when you are working with multiple projects and you wish to separate the associated requests/collections from each other. then you can create separate workspaces for both the projects.

To create a new workspace (either team or personal), simply click the workspace icon and then click “Create New”.

Once the workspace properties window opens, select whether you want to create a personal or team workspace. For team workspace, you can choose to invite people with their email addresses by asking them to collaborate on the workspace.

**This is how the workspace properties window will look like.**

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2019/10/postman-workspaces.png)

**Q #3) How can Postman collections run through the command line?**

**Answer:**Postman has a command-line integration tool called Newman with which you can run any existing Postman collection.

Newman is a nodejs based package, which requires just a node environment to execute the collection and has full parity with the Postman collection runner i.e. the Newman collection runner supports the Postman capabilities like Running assertions, Pre-request scripts or any other scripts that are associated with the requests that are a part of the collection.

**To use Newman:**

* You need to have node installed.
* Now the Newman package needs to be installed through npm using the command.

npm install -g newman

* The collection needs to be executed and the associated environment configuration should be first exported to its JSON form through the Postman application
* Now run the below command to run the Postman collection through Newman.

newman run {{path to collection json}} -e {{path to environment json if any}}

**Q #4) How can you generate HTML based reports running tests through the Postman?**

**Answer:**Newman uses the concept of reporters and templates to generate HTML reports for the executed collection.

Hence, to generate HTML reports, you first need to install a reporter. You can install any of the available HTML reporters like [Newman-reporter-html](https://www.npmjs.com/package/newman-reporter-html) as a node package through the below command.

npm install -g newman-reporter-html

Once the HTML reporter is installed, we can use the Newman command to run the collection with -r flag i.e. the reporter flag and specify the reporter name as HTML.

**The below command is used:**

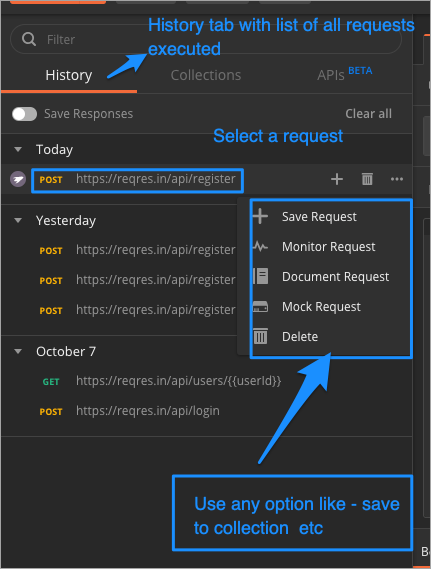
newman run {{path to collection json}} -e {{path to environment json if any}} -r html

Please note that as we have not mentioned the name or folder where we want the reports to get generated, by default the reports will be generated in a folder named “Newman” that gets created in the same directory where the Newman command is executed from.

**Q #5) How can we use Postman history and save requests from the Postman history to the existing or new collections?**

**Answer:**Any request that gets executed through the Postman application, is available for reference in the History section of the application. So in case, the request was not saved to a collection before it was executed, we can always go back to the history section to fetch the executed request and save it to the collection.

**Refer to the below screenshot for more details.**

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2019/10/postman-collectionhistory.png)

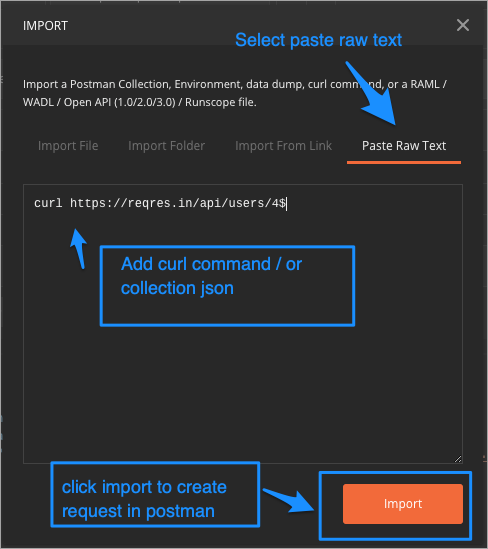
**Q #6) How can you import requests in formats other than cURL into Postman?**

**Answer:** Postman supports a lot of common request formats to export requests to. **Example.** Java, C#, Python, PHP, etc. It supports almost all the commonly used libraries and language bindings.

For importing requests, it supports cURL for now. i.e. you can paste a curl command in request import and it gets converted to Postman requests, but the same cannot be done using any other language bindings like Java, Python, etc.

The other way to import multiple requests at once is to import the entire collection directly through a file or collection JSON pasted as raw text in the import window.

**Given below is a screenshot of how the import raw text section of the import options will look like.**

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2019/10/postman-import.png)

**Q #7) Is it possible to Log Requests and Responses in Postman?**

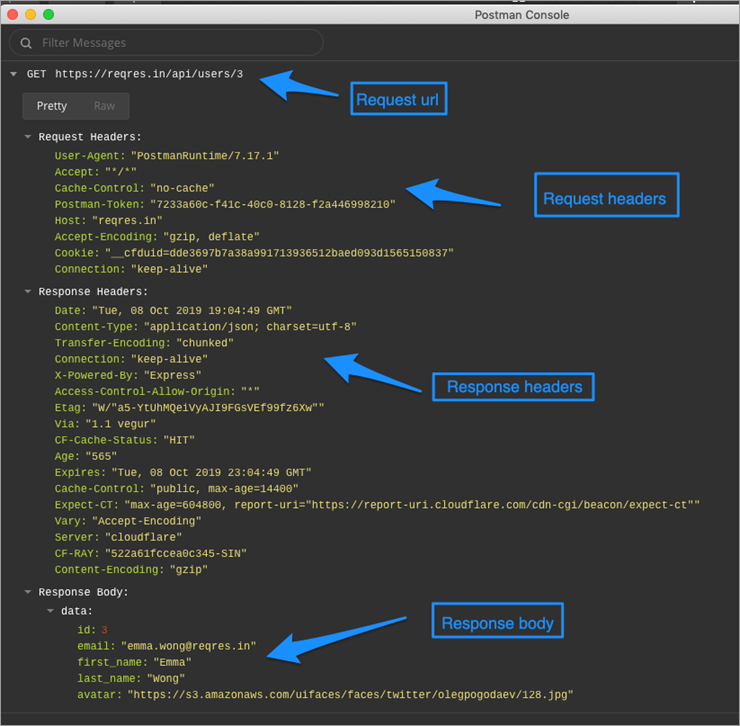
**Answer:** Postman allows viewing the response body and other request parameters in the application itself.

But there are times when we have applied pre-request scripts and as we are unable to see details about the request URLs and headers that were used while executing the request, and it’s always important to see how the actual request looked like.

To view complete Requests and Responses for the executed collection or individual request, Postman provides an additional tool console called “Postman Console” and it can be used to view all the requests/response details.

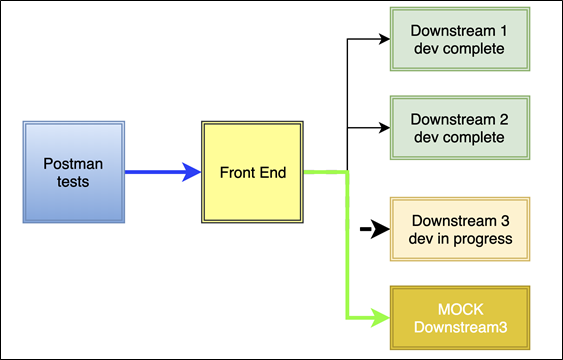
It’s also useful to see the output of any console.log statements that are a part of the pre-request scripts or tests.

**Given below is the screenshot of the Postman console window.**

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2019/10/postman-console-1.png)

**Q #8) How can Postman be used to create Mock Servers?**

**Answer:** Postman allows users to simulate backend servers or any API endpoints that are still under active development and to run an integration test or end to end test, you still need to get some pre-defined response through those endpoints.

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2020/01/postman-to-create-Mock-servers.png)

Refer to the above diagram, where a front end server/API has few downstream dependencies, of which one dependency is still a work in progress. To reduce the dependency of the front end being able to use the downstream until its complete, we can create a mock for the downstream and use it till the time the downstream dependency is not complete.

Thus mock servers are nothing but a fake implementation for the backend. To create/use mock servers, a user should be registered with Postman at least for a free account (Postman allows users to register for a free account through the user’s email).

Also, please note that for a free account the no. of calls to a mock server is limited to 1000 (This limit can be increased by buying an enterprise plan or purchasing an additional quota from the Postman account usage page).

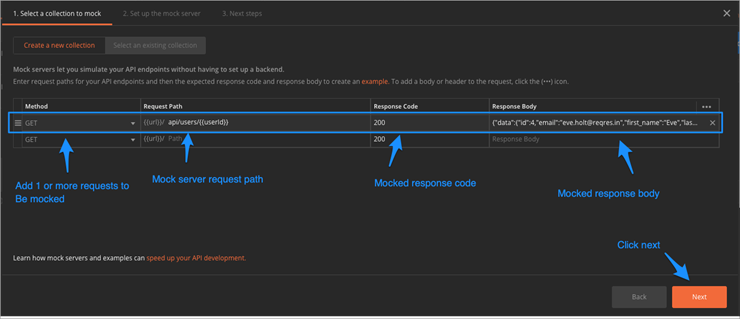
**Here is a Video Tutorial:**

To create a mock server, you can use an existing collection i.e if you want to create a mock for your entire collection or add requests when creating a mock server.

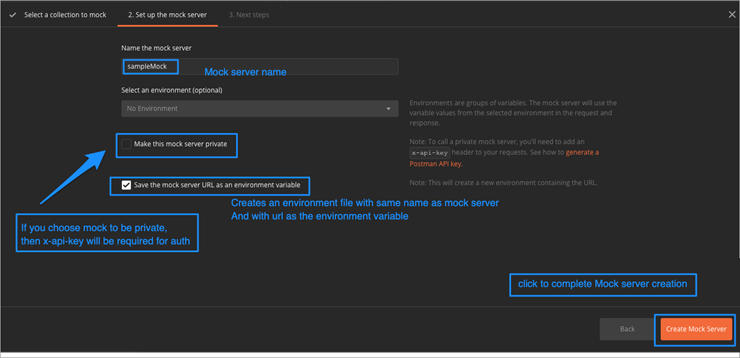
**Follow the below steps to create a mock server:**

**a)** Click New and select “Mock Server”.

**b)** Add request method(s) to be mocked and add the response code and response body to be returned while the particular API endpoint is called.

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2019/10/postman-mockServerCreate_1.png)

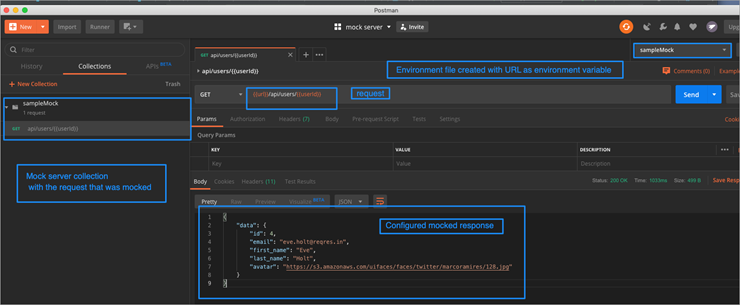
**c)** Click Next and choose the mock server name (If you want this mock server to be private, then an authorization header named x-api-key which will be generated for the user profile through which Postman is signed in will be required).

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2019/10/postman-mockServerCreate_2.png)

**d)** Click “Create Mock Server”. Essentially this will host your API endpoint on some Postman server and will return the set response whenever the particular endpoint is called.

**e)** It will also create a new environment file (that was set during the mock server setup) and set the URL of the mocked API endpoint as an environment variable.

**f)** You are all done and now, you can use this mocked endpoint to send requests to. This mocked implementation can be used for dependent services in the actual code if the real services are still under deployment.

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2019/10/postman-mockedCollection.png)

**Q #9) How can we use Custom Javascript Libraries with Postman Pre-request Scripts or Tests?**

**Answer:** Postman sandbox provides a lot of libraries that are built-in and are available for usage. For a complete list of such libraries, refer [here](https://learning.getpostman.com/docs/postman/scripts/postman_sandbox_api_reference/) to use these libraries, and you will need to add them in pre-request scripts or tests using ‘require’.

**Here is a Video Tutorial:**

Let’s see one such example using moment.js and this library provides a lot of helpful functions to perform formatting around time.

Let’s say, there is a POST request which has to say, created date for a user and it expects the date format **YYYY-MM-DD**. Though it could be achieved using plain javascript as well, moment.js can do this with one line of code.

Let’s see this in action now. In the pre-request script, just add the following line of code, to get the formatted data, stored in an environment variable.

var moment = require('moment');

pm.environment.set('formattedDate',moment().format('YYYY-MM-DD'));

Another example of the moment could be to add a particular value to the current date and use it in the request body. **For example,** you want to set a field like an expiry date, to current date + 2 days, as well as with formatting to ‘YYYY-MM-DD’, and you can simply use the script as below.

pm.environment.set('expiryDate',moment().add(2,'days').format('YYYY-MM-DD'));

In the above script, we can see that we’ve added or included ‘moment.js’ library and used the object as a simple Javascript code. Similar to pre-request scripts, these libraries or modules can be used in the post-request scripts or tests as well to do similar stuff.

Other libraries are available like crypto js which could be useful to convert a text to encrypted value like Base 64 or encoded hash and could be used as a part of the request body.

**Q #10) What are Postman Monitors?**

**Answer:** Postman monitors are nothing but collection monitors that are set up and are executed as per the configured frequency. These are generally used when someone wants their collection to run at a particular frequency and the results are required to be monitored with failures being notified through email or slack integration.

Generally, teams with their infrastructure like CI and own cloud servers would not prefer to use Postman defined monitors as it would run only on published or public endpoints or on mocked endpoints (if configured through mock servers).

**Conclusion**

In this tutorial, we covered questions related to some common Postman concepts that are usually asked in interviews.

Postman being a very extensively and commonly used tool for all kinds of API testing and with the recent support for GraphQL, it’s even more extensible and useful. Along with other powerful features like Assertions & Workflows, it allows us to perform end to end integration testing for almost all types of REST API endpoints.

For any backend developer as well as QA, Postman is one of the main tools for performing all kinds of integration validations.

**What do you understand about collections in Postman?**

Collections in Postman are groups of similar requests and essentially help to organize requests in dedicated folders.

**Q2. What are some core benefits of using the Postman API platform?**

This is one of the most common Postman interview questions for freshers. Some benefits of using the Postman API platform include:

* Postman is equipped to send different types of API requests (such as POST, PUT, PATCH, GET)  based on the requirement.
* Postman helps to save API environments for future deployment.
* It can be integrated with other tools that are used for Continuous Integration and Continuous Delivery.
* Postman helps to manage and handle the end-end API life-cycle.
* Postman offers a unique runtime service that specifically helps with managing environments, workspaces, and API collections.

**Q3. Which command will you use to log values of variables in Postman?**

To log variable values in the Postman console, the command that’s used is:

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

**Q4. What are some common authorization methods associated with Postman?**

Some authorization methods associated with Postman include:

* Bearer Key
* API Token
* OAuth
* AWS Signature
* Hawk Authentication
* Digest Authentication
* Basic Authentication

**Q5. What is Basic Authorization used for in Postman?**

Again, this is one of the most popular Postman interview questions asked at tech interviews. Basic authorization in Postman is an authorization technique for HTTP agents, such as web browsers.

**Sample Postman Interview Questions for Freshers**

Let’s look at some sample Postman interview questions for freshers.

1. What do you understand by Digest Authorization in Postman?
2. What are some limitations of Postman?
3. What do you understand about the Postman Monitor?
4. What are Query Params in Postman?
5. What are Path variables in Postman?
6. How do you save API responses to a file?
7. What do you understand about the 301 Status code?
8. Which are the API requests that are supported by Postman?
9. How do you access specific variables in Postman?
10. What do you understand about the term environment in Postman?

**Postman Interview Questions for Experienced Programmers and Developers**

Experienced programmers generally go through multiple systems design rounds at technical interviews. This means that there is a greater emphasis on API modeling and API design for senior or experienced programmers. To help you understand what type of Postman interview questions are asked for experienced developers, we’ve compiled this list of the most common questions to expect.

This section covers Postman API testing interview questions and Postman rest API interview questions, among other core concepts pertaining to the software application. Let’s take a look.

1. What do you understand about the 304 status code in Postman?
2. Explain the different variable scopes in Postman
3. Explain the different Request methods associated with Postman
4. What is the use of GUID in Postman?
5. Explain the process of writing test cases with basic auth in Postman
6. Explain workspaces in Postman.
7. What is the Pre-request script in Postman?
8. What is a Scratch Pad in Postman? Why is it used?
9. How would you go about using custom JavaScript libraries in Postman?
10. Explain how to generate random numbers within a given range in Postman
11. How do you view Response and Request logs in Postman?
12. What are the primary components of an HTTP response?
13. What is Status Code 201 in Postman?
14. What command will you use to halt execution requests in Postman?
15. Explain some common Status codes in Postman.

These Postman interview questions for freshers and experienced programmers will give you a clear idea of what to expect at your next technical interview.

**FAQs on Postman Interview Questions**

**Q1. What is Postman?**

Postman is a software application developed to design, build and test APIs. It can be easily customized to business needs and integrated with popular JavaScript framework libraries.

**Q2. What are the core concepts around which Postman interview questions are asked at technical interviews?**

Postman interview questions asked at technical interviews are typically around rest APIs, API testing using Postman, API design, authorization techniques, status codes, and commands.

**Q3. Are Postman interview questions asked at software developer interviews?**

Yes, Postman interview questions are asked at systems design rounds of software engineering interviews as Postman is a widely used software application to build and design APIs.

**Q4. In what language is Postman written and developed?**

Postman is written and developed using Electron - an open-source JavaScript framework that was originally maintained and developed by Github.

**Q5. What are some popular tools (aside from Postman) used for API testing?**

Some tools used for API testing include - SoapUI, Apigee, Katalon Studio, and Jmeter.

## Q: What is Postman? Ans:

*The Postman is a very Popular****API****third-party tool that helps the development team for****creating, share, and test documents of API****. POSTMAN provides the flawless user experience of striking the endpoints of API by quickly creating the request as per the requirements of API and not selecting the various parameter responses like Header, Status Code, and the actual body itself a response.*

These features are used for **continuous automated testing**. POSTMAN is an associate tool that is used to develop, document, and test against an API.

POSTMAN gives users an **interface** via which they can make REST API server requests and see the responses.

## Q: What is an API? Ans:

*The Full name of API is****Application Programming Interface****. For building software applications API's set of tools is used. API is an interface, it works as one software program interacts with other software programs.*

In simple words, API works as a **mediator** between two software applications and allowing to communicate with each other. API can be known as the collection of software functions where other programs can be carried out.

Powered By

##### **Take a look at our suggested post :**

[REST API Interview Questions and Answers](https://www.techgeeknext.com/rest-api-interview-questions)

[Rest Assured Interview Questions and Answers](https://www.techgeeknext.com/rest-assured-interview-questions)

[Swagger Assured Interview Questions](https://www.techgeeknext.com/swagger-interview-questions)

[Robot Framework Interview Questions](https://www.techgeeknext.com/tools/robot-framework-interview-questions)

[Nagios Interview Questions](https://www.techgeeknext.com/tools/nagios-interview-questions)

[Spring Boot Interview Questions](https://www.techgeeknext.com/java/spring-boot-interview-questions)

[Java 8 Interview Questions](https://www.techgeeknext.com/java/java8-interview-questions)

[Java 8 Programming Interview Questions](https://www.techgeeknext.com/java/java8-programming-interview-questions-and-answers)

## Q: Why use Postman? Ans:

*The client-server model defines how a server gives resources and services to one or more clients. Web servers, mail servers, and file servers are examples of servers. So, the Client requests something, and the Server fulfils the request.*

## Q: What is REST? Ans:

The use of Postman is become very much popular as a tool because of the following reasons:

1. **Free:**Postman download is free. It can be used for any size of the team.
2. **Easy:**Postman is very easy to use. Just download the Postman app, then send the first request and get the Test result in the minutes. For future use, Postman helps to save the environment.
3. **API Support:**It supports by making any kind of API call such as REST, SOAP, HTTP, and checking the largest responses. The End to End lifecycle of the API is managed by the Postman from design to mocking to testing and finally maintaining the APIs.
4. **Extensible:**According to our needs Postman API can be customized with the help of postman.
5. **Integration:**Postman helps to easily integrate test suites into our preferred CI/CD tools and services, such as Jenkins with Newman (command-line collection runner).
6. **Community & Support:**Postman has a huge community forum for customer support and documentation.
7. **Runtime Services:**API collections, environments, workspaces are managed by the Runtime Services.

## Q: Which are the various authorization methods provided by Postman? Ans:

Postman provides the following API request authorization options:

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

[ⓘ](https://www.pixfuture.com/advertisers/?id0348293521d)

## Q: Why the collection is used in postman? Ans:

*The Postman collection helps together with the individual request in a group and allows to organize the requests into folders.*

## Q: What tools are used for API testing? Ans:

The tools used for API Testing are:

* Postman
* Katalon Studio
* SoapUI
* Tricentis Tosca
* Apigee
* Jmeter

## Q: How the Postman variables are accessed? Ans:

*The Postman variables are accessed by using the variable name as: {{variable\_name}}.*

## Q: Which are the different types of API requests supported in Postman? Ans:

The different types of API requests supported in the Postman are:

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

## Q: What is an HTTP request? Give an example of it? Ans:

*An HTTP (Hypertext Transfer Protocol) means a request made by the client to a named host located on a server and works as a communicator interface or a request-response transmission between a client and server. HTTP request's main aim is to access a resource on the server. The HTTP request is made by the client using the components of a URL (Uniform Resource Locator), which includes the information needed to access the resource.*

Example of the HTTP request: A browser sends an HTTP request to the server and then, the server responds to the browser. The status information about the request and its content is content by the response sent by the server.

## Q: What are the HTTP request core components? Ans:

The core components of an HTTP request includes five key elements:

* **HTTP methods:**HTTP methods is a set of request methods that are used to perform needed action for a given resource such as GET, PUT, POST, and DELETE.
* **URI:**It is a type of address in which the resources are described.
* **Version:**The HTTP version is specified. For example HTTP v1.1
* **Headers Request:**The content type and content length of the request are specified.

For example: Content-type: application/ JSON, Content-Length: 511

* **HTTP Payload:**It contains the Request Body that includes message content.

## Q: Why does Postman accept Base64 encoding only? Ans:

*Postman accepts Base64 encoding only because the same 64 characters in encoding on which we depend can be used as per our want. The data which is transmitted is in the textual form and sends the data in the form such as HTML form which is easier.*

## Q: In which type of encoding does the postman accept authorization credentials? Why? Ans:

*The accepted authorization credentials by Postman are encoding Base64. In Postman it is inbuilt or another option is to consult the third party for converting the credentials in Base64. It is mainly used for the data which is transmitted in the textual form and sends the data in the form such as HTML form which is easier.*

## Q: What is the meaning of the term environment in Postman? Ans:

*The term environment in the Postman means a collection of Key-Value Pairs. Multiple environments can be created in the Postman and by pressing a button switching among them becomes quick. There are mainly two environments****Global and Local****.*

## Q: Can we have two global variables with the same name in Postman? Ans:

*Because global variables in Postman are global, two variables with the same name in the absence of an environment****can cause confusion****for the software.****Local variables might have the same name in different environments****.*

## Q: In Postman which same name variable is preferred local or global? Ans:

*In Postman having the same variable names for local and global, most probably the local variable will be preferred as the local variable will overwrite the global variable.*

## Q: How can you log variable values in Postman? Ans:

In Postman, log variable values are in the console by using the following code

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

## Q: What is the use of Postman monitor? Ans:

*The collections are run by using a Postman monitor. The users have defined a specific time till that time only the collections are run. The users are required to log in and share the Monitor reports over an email on a daily or monthly basis.*

## Q: What is the use of the Postman Collection runner? Ans:

*The Postman collection runner is used for performing DataDriven Testing. The group of API requests is run in a collection for multiple iterations with different sets of data.*

## Q: What is the difference between a Collection monitor and a Collection runner? Ans:

|  |  |  |
| --- | --- | --- |
| **Difference** | **Collection monitor** | **Collection runner** |
| Definition | Collection **monitor is a web client that allows us to send scheduled API calls to the deployed APIs that we have created**. | The Collection Runner **allows to run sets of requests in a specified chronology**. |
| Importance | The Monitor allows to obtain statistics of APIs on the performance. | The Collection Runner will log request test results, and scripts can pass data between requests as well as altering the request workflow. |
| Use | The collections are run by using a Postman monitor. The users have defined a specific time till that time only the collections are run. The users are required to log in and share the Monitor reports over an email on a daily or monthly basis. | The Postman collection runner is used for performing DataDriven Testing. The group of API requests is run in a collection for multiple iterations with different sets of data. |

## Q: What are the limitations of the Postman? Ans:

The limitations of the Postman are as follow:

* Thousands plus API cannot be processed by the Postman.
* In Postman it is not easy to manage collections and requests in huge projects.
* he Postman is not suitable because a lot of code duplication for dynamic API requests could happen and managing the workspace in the form of code.

## Q: What is the team workspace in Postman? Ans:

*In Postman the workspace is a synergetic environment for users to develop and test API'S. Similarly, the team workspace means, a workspace that is shared by the whole team working on the same collections of requests. The team workspace is accompanied and associates all the team's work in one place.*

## Q: What is the difference between authorization and authentication? Ans:

|  |  |  |
| --- | --- | --- |
| **Difference** | **Authorization** | **Authentication** |
| Use | The Authorization is used to access the resource. | The Authentication enables to ensure and confirm a user's identity to the system. |
| Definition | Once authentication is done, the technique of authorization process allows or denies someone from accessing. | Authentication is a techinque of presenting your credentials to the system and the system validating your credentials. These credentials tells about who are you to the system. |
| Importance | The importance of Authorization is that the security is maintained. | Similarly like Authorization, Authenication also helps in maintaining security. |

## Q: What is the difference between Query Parameters and Path Variables? Ans:

Query Parameters and Path variables difference is :

* **Query Parameters:**The Query Parameters are used for sorting the resources.
* **Path Variables:**The Path variables are used for identifying specific resources.

## Q: What is Basic Auth in Postman? Ans:

*The term Basic Auth means the authorization process provided in which HTTP users have to log in****username and password****for web browsing in Postman.*

## Q: Why do we group requests under collections when the collection is already a grouping of requests? Ans:

*The collection may have thousands of requests under it. So the use of a folder should be done to subcategorize the requests so that it becomes easy to find, edit, and modify them. The collection may consist of many folders in it and many requests in that folders. The evaluation of the requests can be done by the requests which are already evaluated.*

## Q: Which are the two ways in which tests can be written in Postman? Ans:

There are two ways of writing Tests in the Postman. They are:

##### **Functional Method**

The syntax in the functional method is different but the language which is used is JavaScript. In Postman it is officially recommended method. The inbuilt snippets in the Postman are part of the functional method.

##### **JavaScript Method**

The Postman API request scripts are written the JavaScript. The test scripts are written in the JavaScript method.

## Q: Can Postman Monitors can import local variables? Ans:

*In the Post Monitors,****local variables****can be imported but global variables cannot be imported.*

## Q: What is a binary form and its usage in POST methods? Ans:

*To send data easily in Post methods in a format binary form is used. In Post request, it is important to use for sending large files such as CSV, images. The easiest way to send complex data with a request becomes easier with the binary presentation.*

## Q: Why do we use Postman cloud in an organizations? Ans:

*We use the Postman cloud which is a common repository of companies to access Postman collections. After logging in work can be saved immediately in the Postman Cloud. The team can access data from anywhere.*

##### Converting a POSTMAN request to Curl?

You can see the button **Code** on the right-top corner in postman, press it and you can get your code in many different languages including cURL  
  
  
**Then select the Curl option from the dropdown**

##### How to handle a self-signed certificate with Postman?

You can handle this by disabling SSL certificate verification in Postman. **Settings->SSL certificate verification**

##### How to run a single request from the collection in Newman?

You can run a specific request in a collection runner in Newman CLI with the option **--folder**.

newman run "postman-collection-API" --environment "postman-environment-API" --folder request-name

##### Can I save JSON objects to environment variables so as to chain them for another request?

You can save and use it;  
  
You should use **JSON.stringify()** before setting it to the environment variable., use **postman.setEnvironmentVariable()** method in which you can set simple numbers and string values.  
  
Then after you call it in the next request, use **JSON.parse()** to recreate the object.

##### *How to add a timestamp in Postman request?*

You can use the moment module to get the current date and time

// add in global variables

var moment = require('moment')

pm.globals.set("CurrentDatetime", moment().format("YYYY/MM/DD"))

// on request

"ReqDate": {{CurrentDatetime}}",

##### How do you save .zip file from the rest web service using Postman?

You can use the **Send and Download** option in order to download the file, though, you will get a dialogue asking where to save the file.

##### How to generate random numbers in a specific range using POSTMAN?

In your Pre-request script define your variable with:

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

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

{{randomNumber}}

##### Is there a way to Collapse All JSON fields in Postman?

First, click on a level which you want to collapse, then press the buttons

* **Alt-0** to collapse all (zero not o)
* **Alt-Shift-0** to uncollapse (zero not o)
* On Mac, it's **Command-Option-0** to collapse all and **Command-Option-Shift-0** to uncollapse.

##### How to check if responseBody does not contain string in postman tests?

You can use the fluent style in the desktop postman application.

pm.test("Body matches string", function ()

{

pm.expect(pm.response.text()).to.not.include("string\_you\_want\_to\_search");

});

##### How to check whether the field is returning null in the Postman automation?

pm.test("To Check if Value is Null", function() {

var jsonData = pm.response.json();

pm.expect(jsonData.<yourfield>).not.eq(undefined);

)};

##### How do you add a header to every request in Postman in the pre-request script?

To add a header to every request, go to your collection, right-click it, select Edit, and go to the **Pre-request Scripts** tab, where you can add the snippet

// example

pm.request.headers.add({

key: 'X-HEADER-TEST',

value: '1'

});

##### How do I extract a variable from XML using Postman?

For example, in the following XML response

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<QuotesPrivateResponse>

<lease-service>

<duration-in-months>24</duration-in-months>

</lease-service>

</QuotesPrivateResponse>

To retrieve the value duration-in-months:

var response = xml2Json(responseBody);

var duration = response["QuotesPrivateResponse"]["lease-service"]["duration-in-months"];

pm.environment.set("duration", duration);

##### How do you delete cookies in the pre-request script in POSTMAN?

You can use the code to clear all the cookies

const jar = pm.cookies.jar();

jar.clear(pm.request.url, function (error) {

// error - <Error>

});

##### How to send a base64 image in postman?

You could find an online base64 image encoder. They encode an image to a string. The example of a raw body in JSON format in the POSTMAN:

"profile": { "first\_name": "John",

"last\_name": "Dow",

"photo": "iVBORwAAAYII= I have shorted it because lenthy"

}

##### How do you save API response in Postman to a file?

There are 2 ways of saving the response to a file:

1. Click on the small down arrow beside the **Send** button, this will show the **Send and Download** button. Click on it and the postman will ask you where to save the response when the request is done.  
   send and download button
2. There is a **Download** button in the response section of the window.

##### Postman Chrome: What is the difference between form-data, x-www-form-urlencoded and raw ?

These are different Form content types defined by W3C. If you want to send simple text/ ASCII data, then **x-www-form-urlencoded** will work. This is the default.  
  
But if you have to send a non-ASCII text or large binary data, the **form-data** is for that.  
  
You can use **Raw** if you want to send plain text or JSON or any other kind of string. As the name suggests, Postman sends your raw string data as it is without modifications. The type of data that you are sending can be set by using the content-type header from the drop-down.  
  
**Binary** can be used when you want to attach non-textual data to the request, e.g. a video/audio file, images, or any other binary data file.

##### What is the difference between PUT and POST methods?

A POST request creates a new object on the server whereas, a PUT request replaces an object with another object.  
  
POST should be used when the client sends the page to the server and then the server lets the client know where it put it. PUT should be used when the client specifies the location of the page

##### What are the most commonly used HTTP methods supported by REST?

* GET is only used to request data from a specified resource. Get requests can be cached and bookmarked. It remains in the browser history and haS length restrictions. GET requests should never be used when dealing with sensitive data.
* POST is used to send data to a server to create/update a resource. POST requests are never cached and bookmarked and do not remain in the browser history.
* PUT replaces all current representations of the target resource with the request payload.
* DELETE removes the specified resource.
* OPTIONS is used to describe the communication options for the target resource.
* HEAD asks for a response identical to that of a GET request, but without the response body.

##### What are the core components of an HTTP response?

Every HTTP response includes four key elements.

* Status/Response Code – Indicates Server status for the resource present in the HTTP request. For example, 404 means resource not found, and 200 means response is ok.
* HTTP Version – Indicates HTTP version, for example-HTTP v1.1.
* Response Header – Contains metadata for the HTTP response message stored in the form of key-value pairs. For example, content length, content type, response date, and server type.
* Response Body – Indicates response message content or resource representation.

##### In which type of encoding does the postman accept authorization credentials? Why?

Postman accepts authorization in Base64 encoding only. This is provided inbuilt in Postman or else you can also refer to third-party websites to convert the credentials in base64.  
  
We use base64 particularly because it transmits the data into a textual form and sends it in an easier form such as HTML form data. We use Base64 particularly because we can rely on the same 64 characters in any encoding language that we use.

##### What are the common protocols that are used in API testing?

* + HTTP
  + JMS
  + REST
  + SOAP
  + UDDI

##### What are Collections in Postman? And what can they contain?

Collections are lists of predefined HTTP requests. When you create an HTTP request, it always belongs to some collection.  
  
Once you start having many HTTP requests in your collection, it becomes chaotic. So to allow for further categorization of your requests, you can create folders and put your requests to a folder structure.  
  
So a collection can contain

* HTTP requests
* Folders.

##### What are the main differences between API and Web Service?

* + All Web services are APIs but not all APIs are Web services.
  + Web services might not contain all the specifications and cannot perform all the tasks that APIs would perform.
  + A Web service uses only three styles of use: SOAP, REST, and XML-RPC for communication whereas API may be exposed to in multiple ways.
  + A Web service always needs a network to operate while APIs don’t need a network for operation.

##### What are the core components of an HTTP request?

An HTTP request contains five key elements:

* An action showing HTTP methods like GET, PUT, POST, DELETE.
* Uniform Resource Identifier (URI), which is the identifier for the resource on the server.
* HTTP Version, which indicates HTTP version, for example-HTTP v1.1.
* Request Header, which carries metadata (as key-value pairs) for the HTTP Request message. Metadata could be a client (or browser) type, format supported by the client, format of a message body format, cache settings, and so on.
* Request Body, which indicates the message content or resource representation.

##### Mention, whether you can use GET request instead of PUT, to create a resource?

Ans. No, you shouldn’t use a PUT or POST method. Instead, apply the GET operation which has view-only rights.

##### Can we have two global scope variables with the same name in Postman?

Since global variables are global i.e. without any environment, they cannot have duplicate names as it creates confusion for the software. Local variables can have the same name but in different environments.

##### What are the principles of an API test design?

The five most important principles of an API test design are:

* **Setup :** Create objects, start services, initialize data, etc
* **Execution :** Steps to apply API or the scenario, including logging
* **Verification :** Oracles to evaluate the result of the execution
* **Reporting :** Pass, failed, or blocked
* **Clean up :** Pre-test state

##### Which purpose does the OPTIONS method serve for the RESTful Web services?

The OPTIONS Method lists down all the operations of a web service supports. It creates read-only requests to the server.

##### What are the areas that need to be taken care of while writing an API document?

The main areas that need your concentration while writing the API documents are as follows:

* You need to check the source of the content
* Plan or sketch of your document
* The delivery layout of the same
* Information needed for each of the functions available in the document
* Lastly, automatic document creation programs

##### Can we import local variables in Postman Monitors?

Yes. Postman monitors allow importing local variables but it does not allow importing global variables.

##### Difference between API testing and Unit Testing?

**UNIT testing :**

* Unit testing is conducted by the Development Team
* Unit testing is a form of White box testing
* Unit testing is conducted prior to the process of including the code in the build
* Source code is involved in Unit testing
* In unit testing, the scope of testing is limited, so only basic functionalities are considered for testing

**API testing :**

* API testing is conducted by QA Team
* API testing is a form of Black box testing
* API testing is conducted after the build is ready for testing
* Source code is not involved in API testing
* In API testing, the scope of testing is wide, so all the issues that are functional are considered for testing

##### What is status code 401 and when does it occur?

Status code 401 is referred for an unauthorized request. An unauthorized request is a request for which you are not authorized. We can incur such a status code when you are not authorized to access the server or you have entered the wrong credentials.

##### Write test code to check whether the response status is 200 or not?

A test code to check whether the response status is 200 or not is as follows

tests["Status Code is 200"] = responseCode.code === 200;

##### What are the major areas to focus on while writing API documents?

* Focus on the content source
* Sketch or document the plan well.
* Layout delivery
* Detailed information about each of the functions
* Automatic document creation programs

##### What kind of common tests are performed on the APIs?

* You should verify the API first and check either it is updating any data structure or not.
* You need to check either API returns anything.
* As per the given parameters or values, the values returned by APIs need to check.
* Verify the API either it triggers any other event or calling another API.