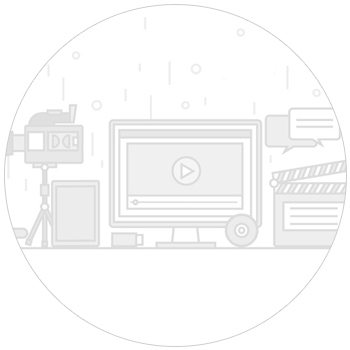
<https://www.softwaretestingmaterial.com/postman-interview-questions/>

<https://www.toolsqa.com/postman/postman-interview-questions/>

**1. What is Postman?**





Get Ready to Answer These Interview Questions During the Pandemic

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.

API

<https://www.javatpoint.com/api-testing-interview-questions>

<https://www.guru99.com/rest-api-interview-question-answers.html>

<https://www.katalon.com/resources-center/blog/web-api-testing-interview-questions/>

<https://www.indeed.com/career-advice/interviewing/api-interview-questions>

## REST Specific Status Codes

#### [200 (OK)](https://restfulapi.net/http-status-200-ok/)

It indicates that the REST API successfully carried out whatever action the client requested and that no more specific code in the 2xx series is appropriate.

Unlike the 204 status code, a 200 response should include a response body. The information returned with the response is dependent on the method used in the request, for example:

* GET an entity corresponding to the requested resource is sent in the response;
* HEAD the entity-header fields corresponding to the requested resource are sent in the response without any message-body;
* POST an entity describing or containing the result of the action;
* TRACE an entity containing the request message as received by the end server.

#### [201 (Created)](https://restfulapi.net/http-status-201-created/)

A REST API responds with the 201 status code whenever a resource is created inside a collection. There may also be times when a new resource is created as a result of some controller action, in which case 201 would also be an appropriate response.

The newly created resource can be referenced by the URI(s) returned in the entity of the response, with the most specific URI for the resource given by a Location header field.

The origin server MUST create the resource before returning the 201 status code. If the action cannot be carried out immediately, the server SHOULD respond with a 202 (Accepted) response instead.

#### [202 (Accepted)](https://restfulapi.net/http-status-202-accepted/)

A 202 response is typically used for actions that take a long while to process. It indicates that the request has been accepted for processing, but the processing has not been completed. The request might or might not be eventually acted upon, or even maybe disallowed when processing occurs.

Its purpose is to allow a server to accept a request for some other process (perhaps a batch-oriented process that is only run once per day) without requiring that the user agent’s connection to the server persist until the process is completed.

The entity returned with this response SHOULD include an indication of the request’s current status and either a pointer to a status monitor (job queue location) or some estimate of when the user can expect the request to be fulfilled.

#### 203 Non-Authoritative Information

#### [204 (No Content)](https://restfulapi.net/http-status-204-no-content/)

The 204 status code is usually sent out in response to a PUT, POST, or DELETE request when the REST API declines to send back any status message or representation in the response message’s body.

An API may also send 204 in conjunction with a GET request to indicate that the requested resource exists, but has no state representation to include in the body.

If the client is a user agent, it SHOULD NOT change its document view from that which caused the request to be sent. This response is primarily intended to allow input for actions to take place without causing a change to the user agent’s active document view. However, any new or updated metainformation SHOULD be applied to the document currently in the user agent’s dynamic view.

The 204 response MUST NOT include a message-body and thus is always terminated by the first empty line after the header fields.

#### 300 Multiple Choices

#### [301 (Moved Permanently)](https://restfulapi.net/http-status-301-moved-permanently/)

The 301 status code indicates that the REST API’s resource model has been significantly redesigned, and a new permanent URI has been assigned to the client’s requested resource. The REST API should specify the new URI in the response’s Location header, and all future requests should be directed to the given URI.

You will hardly use this response code in your API as you can always use the API versioning for new API while retaining the old one.

#### 302 (Found)

The HTTP response status code 302 Found is a common way of performing URL redirection.. An HTTP response with this status code will additionally provide a URL in the Location header field The user agent (e.g., a web browser) is invited by a response with this code to make a second. Otherwise identical, request to the new URL specified in the location field.

Many web browsers implemented this code in a manner that violated this standard, changing the request type of the new request to GET, regardless of the type employed in the original request (e.g., POST). RFC 1945 and RFC 2068 specify that the client is not allowed to change the method on the redirected request. The status codes 303 and 307 have been added for servers that wish to make unambiguously clear which kind of reaction is expected of the client.

#### 303 (See Other)

A 303 response indicates that a controller resource has finished its work, but instead of sending a potentially unwanted response body, it sends the client the URI of a response resource. The response can be the URI of the temporary status message, or the URI to some already existing, more permanent, resource.

Generally speaking, the 303 status code allows a REST API to send a reference to a resource without forcing the client to download its state. Instead, the client may send a GET request to the value of the Location header.

The 303 response MUST NOT be cached, but the response to the second (redirected) request might be cacheable.

#### 304 (Not Modified)

This status code is similar to 204 (“No Content”) in that the response body must be empty. The critical distinction is that 204 is used when there is nothing to send in the body, whereas 304 is used when the resource has not been modified since the version specified by the request headers If-Modified-Since or If-None-Match.

In such a case, there is no need to retransmit the resource since the client still has a previously-downloaded copy.

Using this saves bandwidth and reprocessing on both the server and client, as only the header data must be sent and received in comparison to the entirety of the page being re-processed by the server, then sent again using more bandwidth of the server and client.

#### 307 (Temporary Redirect)

A 307 response indicates that the REST API is not going to process the client’s request. Instead, the client should resubmit the request to the URI specified by the response message’s Location header. However, future requests should still use the original URI.

A REST API can use this status code to assign a temporary URI to the client’s requested resource. For example, a 307 response can be used to shift a client request over to another host.

The temporary URI SHOULD be given by the Location field in the response. Unless the request method was HEAD, the entity of the response SHOULD contain a short hypertext note with a hyperlink to the new URI(s). If the 307 status code is received in response to a request other than GET or HEAD, the user agent MUST NOT automatically redirect the request unless it can be confirmed by the user, since this might change the conditions under which the request was issued.

#### 400 (Bad Request)

400 is the generic client-side error status, used when no other 4xx error code is appropriate. Errors can be like malformed request syntax, invalid request message parameters, or deceptive request routing etc.

The client SHOULD NOT repeat the request without modifications.

#### 401 (Unauthorized)

A 401 error response indicates that the client tried to operate on a protected resource without providing the proper authorization. It may have provided the wrong credentials or none at all. The response must include a WWW-Authenticate header field containing a challenge applicable to the requested resource.

The client MAY repeat the request with a suitable Authorization header field. If the request already included Authorization credentials, then the 401 response indicates that authorization has been refused for those credentials. If the 401 response contains the same challenge as the prior response, and the user agent has already attempted authentication at least once, then the user SHOULD be presented the entity that was given in the response, since that entity might include relevant diagnostic information.

#### 403 (Forbidden)

A 403 error response indicates that the client’s request is formed correctly, but the REST API refuses to honor it, i.e., the user does not have the necessary permissions for the resource. A 403 response is not a case of insufficient client credentials; that would be 401 (“Unauthorized”).

Authentication will not help, and the request SHOULD NOT be repeated. Unlike a 401 Unauthorized response, authenticating will make no difference.

"502 bad gateway" and "503 service unavailable" are common errors in your app hosted in [Azure App Service](https://docs.microsoft.com/en-us/azure/app-service/overview). This article helps you troubleshoot these errors.

#### 404 (Not Found)

The 404 error status code indicates that the REST API can’t map the client’s URI to a resource but may be available in the future. Subsequent requests by the client are permissible.

No indication is given of whether the condition is temporary or permanent. The 410 (Gone) status code SHOULD be used if the server knows, through some internally configurable mechanism, that an old resource is permanently unavailable and has no forwarding address. This status code is commonly used when the server does not wish to reveal exactly why the request has been refused, or when no other response is applicable.

#### 405 (Method Not Allowed)

The API responds with a 405 error to indicate that the client tried to use an HTTP method that the resource does not allow. For instance, a read-only resource could support only GET and HEAD, while a controller resource might allow GET and POST, but not PUT or DELETE.

A 405 response must include the Allow header, which lists the HTTP methods that the resource supports. For example:

Allow: GET, POST

#### 406 (Not Acceptable)

The 406 error response indicates that the API is not able to generate any of the client’s preferred media types, as indicated by the Accept request header. For example, a client request for data formatted as application/xml will receive a 406 response if the API is only willing to format data as application/json.

If the response could be unacceptable, a user agent SHOULD temporarily stop receipt of more data and query the user for a decision on further actions.

#### 412 (Precondition Failed)

The 412 error response indicates that the client specified one or more preconditions in its request headers, effectively telling the REST API to carry out its request only if certain conditions were met. A 412 response indicates that those conditions were not met, so instead of carrying out the request, the API sends this status code.

#### 415 (Unsupported Media Type)

The 415 error response indicates that the API is not able to process the client’s supplied media type, as indicated by the Content-Type request header. For example, a client request including data formatted as application/xml will receive a 415 response if the API is only willing to process data formatted as application/json.

For example, the client uploads an image as image/svg+xml, but the server requires that images use a different format.

#### 500 (Internal Server Error)

500 is the generic REST API error response. Most web frameworks automatically respond with this response status code whenever they execute some request handler code that raises an exception.

A 500 error is never the client’s fault, and therefore, it is reasonable for the client to retry the same request that triggered this response and hope to get a different response.

The API response is the generic error message, given when an unexpected condition was encountered and no more specific message is suitable.

#### 501 (Not Implemented)

The server either does not recognize the request method, or it cannot fulfill the request. Usually, this implies future availability (e.g., a new feature of a web-service API).

### ***What is Response?***

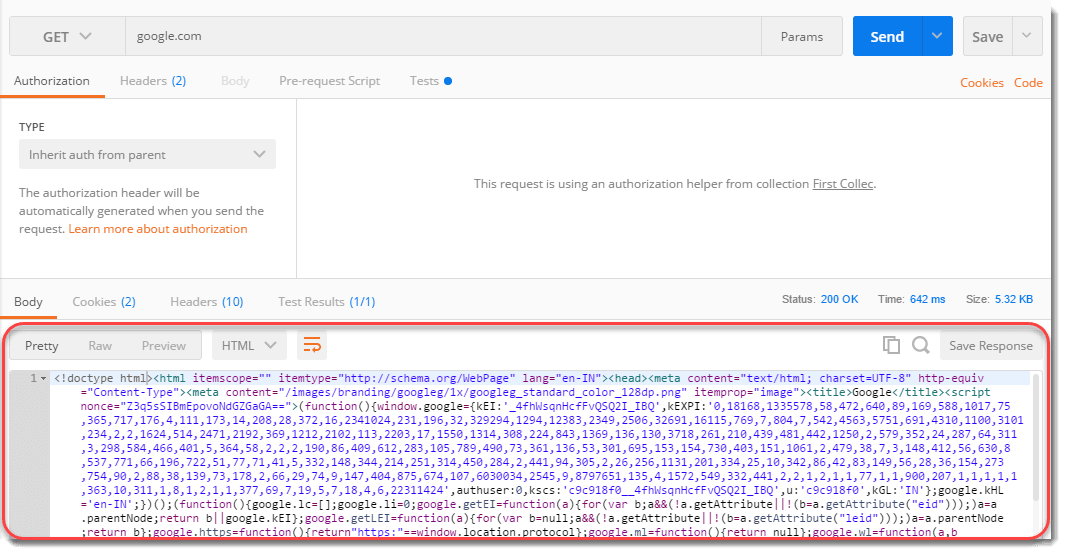
A ***Response*** is a message that is received by the server in return to a ***Request*** that we send. When we request something, the server acts upon the ***Request***and sends back a packet of the requested information. A response depends on the request mainly. Every request has a different kind of response and it is very important that we extract useful information from all of the responses. Postman has a beautiful interface for response and is very user-friendly. We can see a lot of information in the Postman for any response without doing much effort, or any if I might say.

# Understanding Response in Postman

Talking about ***Response in Postman,***the Response user interface contains lots of different things. We will deal with them in detail in this tutorial. The user interface has the following information blocks

* ***Response Status and Information***
* ***Response Body***
* ***Response Cookies***
* ***Response Header***

Let’s start by getting a response for **www.google.com** which looks like this:

[](https://toolsqa.com/wp-content/uploads/sites/1/nggallery/postman/response_with_request.png)

## Response Status and Information

### ***Status Code :***

A ***status code*** tells you the status of the request. There can be a lot of mistakes in the request and without looking at the status code, we might not always get what went wrong to our request. Sometimes, there can be a typing mistake in the URL or there can be a problem at the server side, status code help us know about what went wrong (if something went wrong). There are different status codes and each of them has a different meaning.

You can learn about the complete list of status code [**here**](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes).

[Status_Code_200_2](https://toolsqa.com/wp-content/uploads/sites/1/nggallery/postman/Status_Code_200_2-1.png)

Status code ***200 OK*** means that the request was correct and the desired response has been sent to the client. Now, change the url to **http://restapi.demoqa.com/utilities/weatherfull/city/hyderabd .**Press Send and see the status code now.

[400_Bad_Request](https://toolsqa.com/wp-content/uploads/sites/1/nggallery/postman/400_Bad_Request.png)

It says ***400 BAD REQUEST***. It is so because we have changed the name of the city from ***Hyderabad to Hyderabd***. This means the request was not correct, hence the bad request response. Similarly, you can see other status codes also for different requests.

### ***Time***

***Time*** is the duration which the response took after we sent the request and received the response. This is very important sometimes because many projects have Service Level Agreements(SLA) for the time it should take a web service to return a response, this time can be a used to determine the SLA of the web service endpoint.

[Time](https://toolsqa.com/wp-content/uploads/sites/1/nggallery/postman/Time.png)

***NOTE***: The time given here is not the actual time that the request will take. It is just approximate but almost what it would be because there are a lot of things that Postman do after getting a response such as formatting and dividing Headers and cookies separately. As the additional work by Postman can be roughly considered as a constant time ***(WebServiceTime + Constant processing time by Postman***). Therefore, it is an approximate of the time and is proportional to what the actual time will be. So you can consider this as actual time as well.

### ***Size***

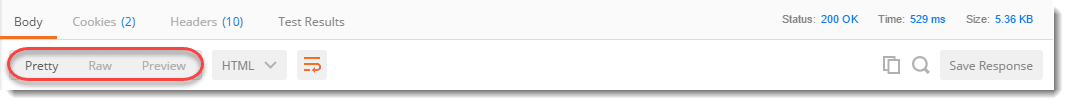
***Size*** is just the response size when it will be saved inside the memory. This response size is the size of complete response and headers and cookies and everything that has been sent along with the response.

[Size_Reponse](https://toolsqa.com/wp-content/uploads/sites/1/nggallery/postman/Size_Reponse.png)

***NOTE***: The response size that is shown in the Postman is approximate response size and not the exact size.

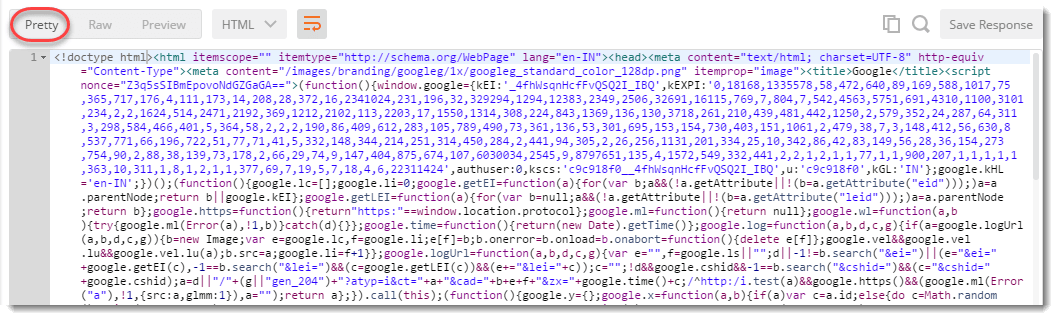
## Response Body

A ***body*** depicts the body of the response, which is the main response content, that has been sent from the server. In this case as you can see it is a web page code being sent to us as a response. Now, there lies three ways ahead of us to look at this response:

[](https://toolsqa.com/wp-content/uploads/sites/1/nggallery/postman/Body_Types.png)

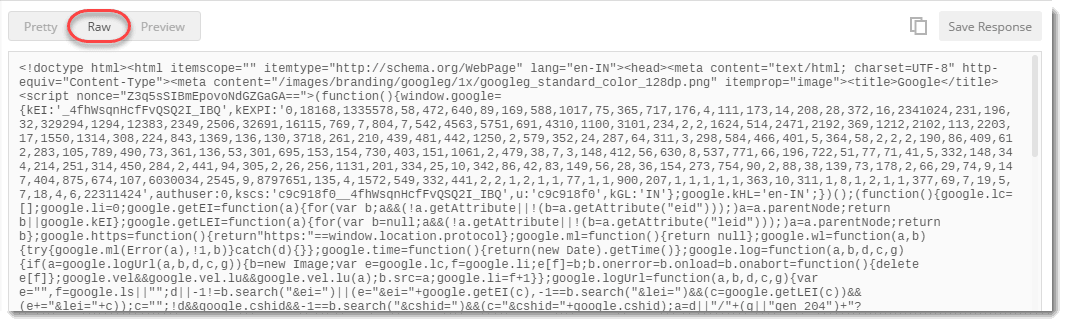
### Pretty

***Pretty*** is a prettier version of the content being sent. The content is prettier as it is more readable. It has coloured key words and different colours have different meanings. This makes a code more readable and look nicer. This formatting is done by the Postman itself after getting the code.

[](https://toolsqa.com/wp-content/uploads/sites/1/nggallery/postman/pretty_response-1.png)

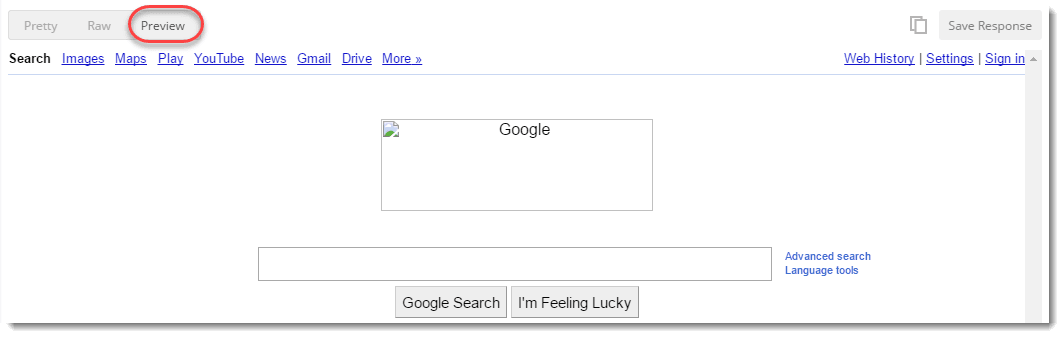
### Raw

Once you click on Preview you will get just the plain view of the content, as received from the server. It is just a raw version of the code without any colourful keywords. By looking at this code you might get why the other code is called “***Pretty***“.

[](https://toolsqa.com/wp-content/uploads/sites/1/nggallery/postman/raw_response.png)

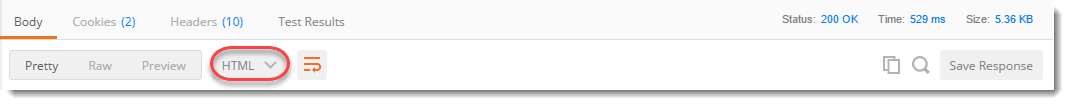
### ***Preview***

Preview of the code will show you the preview of the page, had the page been run inside a browser. Click on preview and you will see the exact page as you would have seen inside a browser. So this would let you know the response preview without visiting the browser.

[](https://toolsqa.com/wp-content/uploads/sites/1/nggallery/postman/preview_response-1.png)

### ***Format Type***

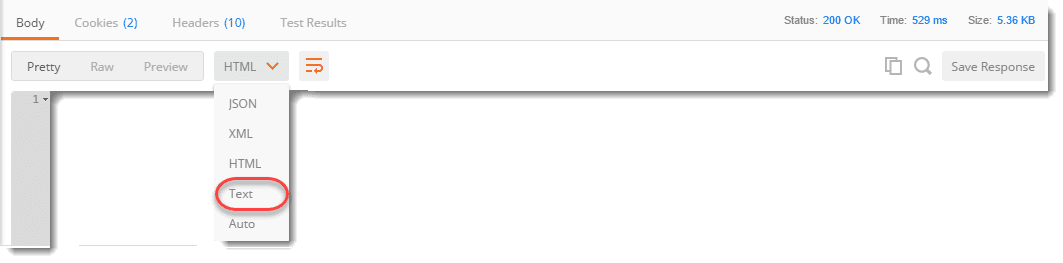
As discussed above, a request has a defined response to it as defined by the ***Content-Type*** header. That response can be in any format. For example, in this case we have the response as a HTML code file.

[](https://toolsqa.com/wp-content/uploads/sites/1/nggallery/postman/Format_Type_HTML.png)

Postman is smart enough to detect the response type and show you in the desired format, but sometimes Postman can also make a mistake. For example, use ***http://restapi.demoqa.com/utilities/weatherfull/city/hyderabad*** to get a response.

You will see that we have received a status code 200 and still there is no response. This is because Postman has failed to recognize the format of the response and is expecting a HTML file as seen in the dropdown.

Select ***Text*** in dropdown and you will be able to see the response now.

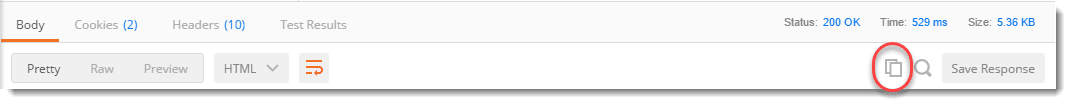
[](https://toolsqa.com/wp-content/uploads/sites/1/nggallery/postman/Format_Type_Text.png)

Sometimes, the server sends the response in two or more different formats. The type of response will be visible to its corresponding format type.

**Note:** Content-Type header defines the format of the response. For e.g. the Content-Type header may say that the response is Json, however the content being sent is XML or a malformed Json. In that case Postman will not be able to do much. Take it as an exercise to understand why Postman is not able to understand the format of response returned by***http://restapi.demoqa.com/utilities/weatherfull/city/hyderabad***

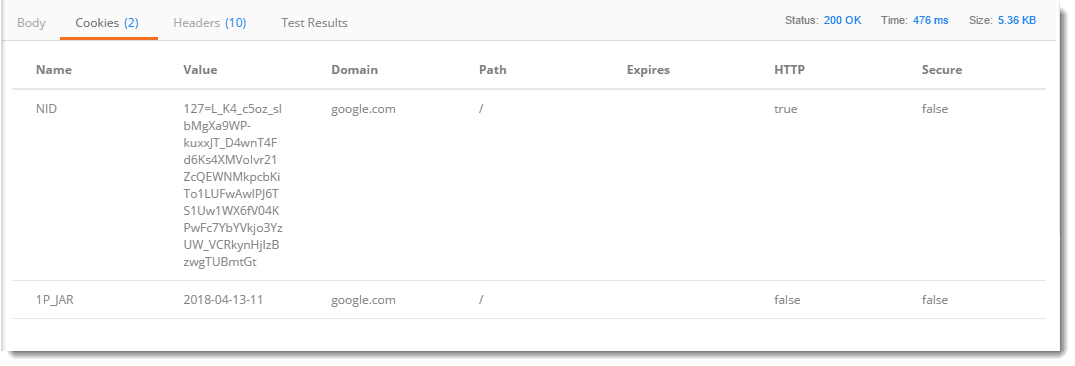
### ***Copy Response***

The icon with two rectangles that you see in the corner is used for copying the complete response to the clipboard which is very handy to send the response to your teammates or using afterwards.

[](https://toolsqa.com/wp-content/uploads/sites/1/nggallery/postman/Copy_Response.png)

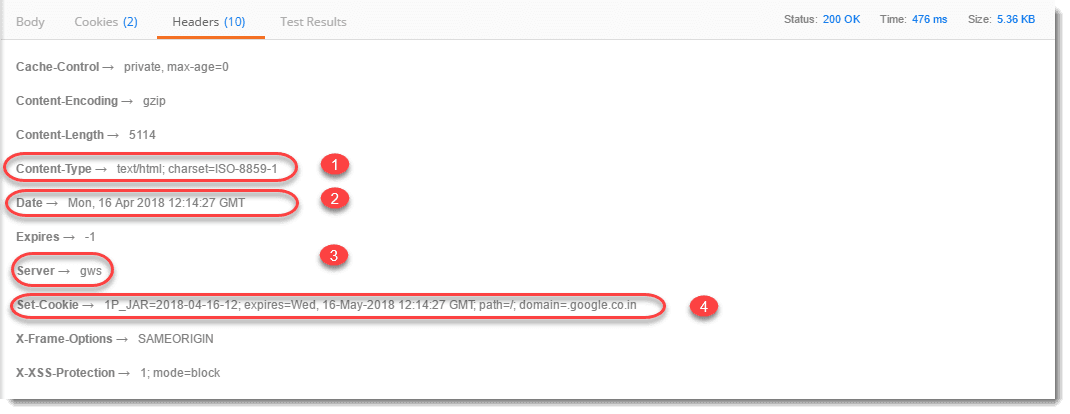
## Cookie

Cookies are the small files which are related to the server files (website pages). Once you visit a website for the first time, a cookie is downloaded on the client’s machine. This cookie contains the information which can be used by the same website when you visit again. This helps the website to get you the specific response and specific information based on your last visit. In postman we can clearly see the cookies that have been sent from the server as a response. This makes it easy for the client to see what cookies are being saved inside his browser. We cannot manipulate this cookies since they are sent from server, Postman is used just to separate it from the response and have a clear view.

[](https://toolsqa.com/wp-content/uploads/sites/1/nggallery/postman/Cookies.png)

## Header

***Headers*** in a HTTP request or response is the additional information that is transferred to the user or the server. In postman, the headers can be seen in the ***Headers*** tab.

[](https://toolsqa.com/wp-content/uploads/sites/1/nggallery/postman/Header_Response.png)

Once you click on header you can see different information such as below. Although, every entry in the Headers tab is a header item we will just take a look at the most important ones.

* ***Content- Type :***This is the content type of the response. In the above example when we used www.google.com the content type is given as ***text/html*** because the response is being sent in the HTML which is one of the options.
* ***Date :***This option shows the date, day and time of the response along with the time zone.
* ***Server :***This option tells the name of the server which has responded to the request. In the above example, the server name is shown as ***gws***which corresponds to ***Google Web Server.***
* ***Cookie expire time :***As the name suggests, this option tells the expire time of the cookie that has been sent along with the response.

### What is an API?

At the very least, offer your interviewer a brief description of what API is but you can also expand the answer to include an example of a time that you’ve previously used it.

**Example:** “An API (Application Programming Interface) is a software that allows two applications to communicate with each other.”

**Related:** [**15 Computer Science Jobs That Pay Well**](https://www.indeed.com/career-advice/finding-a-job/computer-science-jobs-that-pay-well)

### What are some styles for creating a Web API?

To really stand out, do more than just answer the question. Include your preference or a story about a time when you used a specific architectural style.

**Example:** “Common Web API architectural styles are XML/JSON as a formatting language, stateless communication, basic URI as the address for the services and HTTP for communication between the client and server. Personally, I prefer XML over JSON.”

### What is API testing?

If an interviewer asks you this question, they may be looking to determine if you can adequately articulate your response. Provide a clear definition as well as some examples.

**Example:** “API testing is a type of software testing that determines if the developed APIs are functional, reliable and secure. Some of the common API testing types are validation, security, UI, functional, load, penetration, runtime/error detection, fuzz and interoperability and WS Compliance.”

### What are the advantages of API Testing?

Make sure that your answer reflects a thorough comprehension of the advantages of API testing. You should show the interviewer that you know how and when to use it in your work.

**Example:** “API testing gives access to the application without needing a user interface. This allows you to detect the minor issues before they become big problems during GUI testing.

Also, API testing is typically less time consuming than GUI testing because it uses less code. As a result, it offers a more effective and efficient test coverage.

Another benefit is that the data is transferred using XML or JSON. These modes of exchange are language-independent, allowing users to select any coding language when choosing automation testing services. Additionally, API testing is easily integrated with GUI testing.”

**Related:** [**Job Search Guide: Product Management and Software Engineering**](https://www.indeed.com/career-advice/finding-a-job/job-search-guide-product-management-and-software-engineering)

### What is the procedure for performing API testing?

Show the interviewer the steps you would take when performing API testing. You’re proving that you actually know what you’re talking about, so take your time and be sure to list each action.

**Example:** “When performing API testing, you’ll first choose the suite where you’d like to add the API case that you wish to test, then you choose the test development mode. After that, you create test cases for the desired API methods, configure the control parameters and test conditions of the application as well as the method of validation. Then you can perform the API test. Once the test is complete, you check the test reports, filter and sequence all of the API test cases.”

Image description

### What are the primary challenges of API testing?

Be honest about the aspects of API testing that you find most challenging. Your answer will convey an intimate knowledge of the software.

**Example:** “I find selecting and combining parameters and sequencing calls to be the most challenging parts of API testing.”

### What are some tools used for API testing?

Though they’re asking for a list of tools, what they really want is your input about them. Make sure to mention your opinion to prove that you have an understanding of what makes for effective API testing.

**Example:** “A few popular tools are Katalon Studio, Postman, SoapUi Pro, Tricentis Tosca and Apigee. Personally, I prefer SoapUi’s interface. It’s quick and really easy to use.”

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

Use this answer to prove that you have more than just a theoretical knowledge of API. Your experience working with this software can be just as valuable as the technical training to a potential employer.

**Example:** “I have often used API testing to find several different issues, such as missing or duplicate functionality, failure to handle errors effectively and seamlessly as well as any performance, stress, multi-threading, reliability or security issues. However, unimplemented and improper errors, unused flags and inconsistent error handling are some of the other errors that can be found through API testing.”

### What is the difference between API and Web services?

API and Web services serve different functions. Your answer will convey that you recognize when it’s appropriate to use each. Consider listing the basic definition of both as well as their key differences.

**Example:** “Web services must be exposed over the web and have three styles of communication: SOAP, REST and XML-RPC. They always need a network to operate. However, APIs have multiple methods of communication. A network is unnecessary for their operation, and they don’t have to be exposed over the Web.”

**Related:**[**Software Engineer Cover Letter Sample**](https://www.indeed.com/career-advice/cover-letter-samples/software-engineer)

### What is SOAP?

In this answer, state the definition for “SOAP” as well as why you’d want to use it.

**Example:** "SOAP, also known as Simple Object Access Protocol, is an XML-based messaging protocol. It aids in the exchanging of information between computers. You utilize SOAP API to make, find, delete or update records. In instances where there are more than 20 different calls, SOAP API can be utilized to do searches and manage passwords by adapting the protocol to whatever language supports web services.”

### What is REST API?

Be sure to include your reasoning for using REST along with its definition.

**Example:** “REST, or Representational State Transfer, is a set of functions that help developers perform requests and receive responses. Interaction is performed through HTTP Protocol. REST is stateless, so the server has no status or session data. With an effectively-applied REST API, you can restart the server in between two calls. Web services typically use the POST method to perform operations. REST, however, uses GET to access resources.”

### What is the difference between SOAP and REST?

Successfully answering this question will show your interviewer that you can weigh several options simultaneously, a trait that is especially useful when troubleshooting.

**Example:** “There are several differences between SOAP and REST. First, SOAP is a protocol through which two computers can communicate by sharing XML, while REST is a service designed for network-based software architecture. Additionally, SOAP supports only XML format, and REST supports a lot of different data formats. SOAP is unable to support caching, and REST can.

SOAP is also less quick than REST and is similar to a desktop application, where it is closely connected to the server. REST acts like a browser and uses standard methods. An application has to fit inside it. Lastly, SOAP runs on HTTP but envelopes the message, while REST uses the HTTP headers to hold meta information.”

### What factors help inform your decision on which style of Web services—SOAP or REST—to use?

This question will provide an opportunity for you to prove your qualifications and understanding of API. If applicable, mention an instance when this decision was particularly difficult or important.

**Example:** “REST is usually preferred because of its simplicity, performance, scalability and support across many data formats. However, SOAP is a viable choice when service requires an advanced level of security and reliability.”

### What tests can be performed on APIs?

Demonstrate your knowledge and understanding of API by addressing the question before supplying your answer.

**Example:** “Tests can and should be performed on APIs for several reasons, including testing the return values or inputting conditions.”

### What kind of testing environment is needed for API?

When answering this question, provide some personality. Setting up the testing environment is difficult, so feel free to share your opinion to communicate your intimacy with the process.

**Example:** “Setting up the API testing environment can be difficult because you have to configure both the server and the database without the use of GUI.”

### What’s the difference between UI and API testing?

This question checks that you are familiar with the different types of software testing and aware of when each should be employed.

**Example:** “UI, or user interface, testing is focused on examining the graphical interface of an application, such as how the user can interact with its elements. API testing, on the other hand, sets up a mode of communication between two software systems, allowing them to share sub-routines and functions.”

### What is input injection?

This is a rather fundamental aspect of API, so keep your answer concise. A drawn-out response could signal an incomplete or unsure understanding of the process.

**Example:** “Input injection refers to the simulation of user input.”

### What are some ways that you can simulate user input?

There are several ways to employ input injection, but just share a few examples with your interviewer. They know the information. They want to make sure that you do too.

**Example:** “You can accomplish input injection by utilizing a robot, a device driver or low-level input, just to name a few.”

### What is Runscope?

Be sure to explain what the application is as well as what it provides to the API testing process.

**Example:** “Runscope is a Web application used to test APIs by supplying an accessible interface and backend services.”

### Explain API documentation

Documentation is key when performing API testing. Make sure that you communicate the necessity of the process while describing it.

**Example:** “Good API documentation is vital to the process. It supplies a quick reference while working within a program. It provides the plan, delivery layout, sources the content and details every function within the system.”

### What is Unit testing?

Unit testing and API testing are closely related, so make sure that your answer highlights their key differences.

**Example:** “Unit testing, unlike API, is completed before adding code. It’s a type of white box testing that draws the source code into the form and tests the basic functionalities of a system separately.”

### Describe TestAPI

TestAPI is an important asset for testers. As such, an interviewee should be familiar with what it is and how it aids in the testing process.

**Example:** “TestAPI is a sort of library that provides the building blocks for creating automated test suites and tools for testing.”

### What is the protocol for REST Web services?

Make sure that you provide the function of the protocol when answering this question.

**Example:** “HTTP is the protocol used in REST Web services. It facilitates communication between the server and the client.”

### What is URI?

URI is utilized in Web services that are REST-based. Be prepared to answer questions about its format as well as its function.

**Example:** “URI, or Uniform Resource Identifier, is a way to facilitate unambiguous identification of resources on a Web service’s hosting server through a string of characters.”

### What is caching?

Highlight the usefulness of caching by offering its key benefits when answering.

**Example:** “Through caching, you’re able to temporarily store and retrieve data from your system’s memory. Caching mechanisms are extremely effective and efficient because of their ability to improve the speed of delivery.”

**Rate this article**



## Definition & Functions of an API (Common Web API Testing interview questions)

### 1. 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 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.

### 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, 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://www.katalon.com/resources-center/tutorials/introduction-api-testing/) is a kind of software testing which 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 the 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 the 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 code 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 the 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 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 these following nine categories that you should remember before attending in 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

### 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.

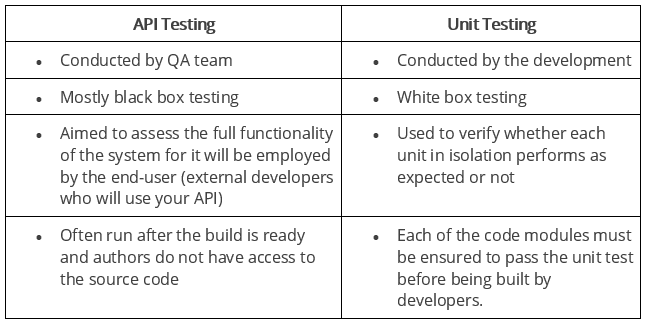
### 15. What are tools could be used for API testing?

There are myriad of different [**API testing tools**](https://www.katalon.com/resources-center/blog/top-5-free-api-testing-tools/) available. A few of common tools are Katalon Studio, Postman, SoapUi Pro, Apigee, etc.  While doing Unit and API testing, both targets 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://www.katalon.com/)**

### 16. What are differences between API Testing and Unit Testing?



### 17. What are differences 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 interface such as how users interact with the applications, testing application elements like fonts, images, layouts etc. UI testing basically focuses on look and feel of an application.

### 18. What are 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 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 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 for 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 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, are 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 which exploit the ubiquity of HTTP protocol and uses HTTP method to define actions. It revolves around resource where every component being 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 Services?

Mostly, there are two kinds of Web Services which 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, 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, 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 to 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 operation are quite similar, except 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 server.

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

Scenario: Let’s say we are designing a network application. Let’s list down few URIs and their purpose to get to know when to use POST and when to use PUT operations.  
  
GET /device-management/devices : Get all devices  
POST /device-management/devices : Create a new device  
  
GET /device-management/devices/{id} : Get the device information identified by “id”  
PUT /device-management/devices/{id} : Update the device information identified by “id”  
DELETE /device-management/devices/{id} : Delete device by “id”

### 38. 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.

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

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>.

### 40. What is payload in RESTFul Web services?

The “payload” is the data you are interested in transporting. This is differentiated from the things that wrap the data for transport like the HTTP/S Request/Response headers, authentication, etc.

### 41. 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 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.

### 42. What is the caching mechanism?

Caching is just the practice of storing data in temporarily and retrieving data from a high-performance store (usually memory) either explicitly or implicitly.

When a caching mechanism is in place, it helps improve delivery speed by storing a copy of the asset you requested and later accessing the cached copy instead of the original.

## SOAP (Common Web API Testing interview questions)

### **43. What are SOAP Web services?**

This is one of the fundamental Web services testing questions that you must know the answer. The SOAP (Simple Object Access Protocol) is defined as an XML-based protocol. It is known for designing and developing web services as well as enabling communication between applications developed on different platforms using various programming languages over the Internet. It is both platform and language independent.

### **44. How does SOAP work?**

SOAP is used to provide a user interface that can be accessed by the client object, and the request that it sends goes to the server, which can be accessed using the server object. The user interface creates some files or methods consisting of server object 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 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.

### **45. When to use SOAP API?**

Use the SOAP API to create, retrieve, update or delete records, like accounts, leads, and user-defined objects. With more than 20 different calls, you can also use the SOAP API to manage passwords, perform searches, etc. by using the SOAP API in any language that supports web services.

### **46. How users utilize the facilities provided by SOAP?**

* PutAddress(): It is used to enter an address in the webpage and has an address instance on the SOAP call.
* PutListing(): It is used 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 is used to get a query name and gets the result that best matches a query. The name is sent to the SOAP call in the form of text character string.
* GetAllListing(): It is used to return the full list in an XML format.

### **47. What is the major obstacle users faced when using SOAP?**

When using SOAP, users often see the firewall security mechanism as the biggest obstacle. This block all the ports leaving 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.

### **48. What are the various approaches available for developing**SOAP based**web services?**

There are two different methods available for developing SOAP-based web services, which are explained below:

* Contract-first approach: the contract is first defined by XML and WSDL, and then Java classes are derived from the contract.
* Contract-last approach: Java classes are first defined, and then the contract is generated, which is normally the WSDL file from the Java class.

“Contract-first” method is the most popular approach.

### **49. What are the elements of a SOAP message structure?**

It is a common XML document that contains the elements as a SOAP message

Envelope: It is an obligatory root element that translates the XML document and defines the beginning and end of the message.

Header: It is an optional item which contains information about the message being sent.

Body: It contains the XML data comprising the message being sent.

Fault: It provides the information on errors that occurred while during message processing.

### **50. What are the syntax rules for a SOAP 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

### **51. What is the transport method in SOAP?**

Application layer and transport layers of a network are used by SOAP; HTTP and SMTP are the valid protocol of the application layer used as the transport for SOAP. HTTP is more preferable, since it works well with the current Internet infrastructure, in particular with firewalls.  
The SOAP requests can be sent using an HTTP GET method while the specification only contains details about HTTP POST.

### **52. What are some important characteristics of a SOAP envelope element?**

* SOAP message has a root Envelope element
* Envelope is an obligatory part of the SOAP message.
* If an envelope includes a header element, it should not contain more than one.
* Envelope version will change if the SOAP version changes.
* The SOAP envelope is indicated by the prefix ENV and the envelope element.
* The optional SOAP encoding is also specified using a namespace and the optional encoding style element.

### **53. What are the major functionalities provided by the SOAP protocol class?**

The SOAP protocol is used to provide simple access methods for all the applications available on the Internet, providing the following functionalities:

* Call: A class which provides the main functionality for a remote method for which a call is needed. It is used to create the call() and to specify the encoding style of the registry that will be assigned when if necessary. This call() function is used by the RPC call, which represents the options of the call object.
* Deployment Descriptor: A class used to provide the information about the SOAP services. It enables easy deployment without the need for other approaches.
* ***DOM2 Writer***: A class that serializes and uses DOM node as XML string to provide more functionalities.
* ***RPC Message***: A class used as the base class that calls and replies to the request submitted to the server.
* Service Manager: A class that provides, lists and then outputs all SOAP services.

### **54. What are the web relation functionalities provided by SOAP protocol?**

* ***HTTPUtils***: This provides the functionality of the POST method to safely meet the requirements.
* ***Parameter***: It is an argument for an RPC call used by both the client and the server.
* ***Response***: It is an object that represents an RPC reply from both client and server, but the result will not be displayed until after the method call.
* ***TCPTunnel***: It is an object that provides the ability to listen on a specific port and to forward all the host and port names.
* ***TypeConverter***: It helps to convert an object of one type into another type and this is called using the class in the form object.

### **55. How does the message security model allow the creation of SOAP more secure to use?**

The security model includes the given security tokens. These tokens comprise digital signatures for protection and authentication of SOAP messages. Security tokens can be used to provide the bond between authentication secrets or keys and security identities. Security token uses the authentication protocols and an X.509 certificate to define the relationship between the public key and identity key. The signatures are used to verify the messages and their origin, generate knowledge to confirm the security tokens to bind the identity of a person to the identity of the originator. Security model prevents different attacks and can be used to protect the SOAP architecture.

### **56. What is the difference between top down & bottom up approach in SOAP Web services?**

* Top down SOAP Web services include creating WSDL document to create a contract between the web service and the client, with a required code as an option. This is also known as Contract-first approach. The top-down approach is difficult to implement because classes must be written to confirm the contract defined in WSDL. One of the benefits of this method is that both client and server code can be written in parallel.
* Bottom up SOAP web services require the code to be written first and then WSDL is generated. It is also known as Contract-last approach. Since WSDL is created based on the code, bottom-up approach is easy to implement and client codes must wait for WSDL from the server side to start working.

### **57. What are advantages of SOAP?**

* SOAP is both platform and language independent.
* SOAP separates the encoding and communications protocol from the runtime environment.
* Web service can retrieve or receive a SOAP user data from a remote service, and the source’s platform information is completely independent of each other.
* Everything can generate XML, from Perl scripts through C++ code to J2EE app servers.
* It uses XML to send and receive messages.
* It uses standard internet HTTP protocol.
* SOAP runs over HTTP; it eliminates firewall problems. When protocol HTTP is used as the protocol binding, an RPC call will be automatically assigned to an HTTP request, and the RPC response will be assigned to an HTTP reply.
* Compared to RMI, CORBA and DCOM, SOAP is very easy to use.
* SOAP acts as a protocol to move information in a distributed and decentralized environment.
* SOAP is independent of the transport protocol and can be used to coordinate different protocols.

### **58. What are disadvantages of SOAP?**

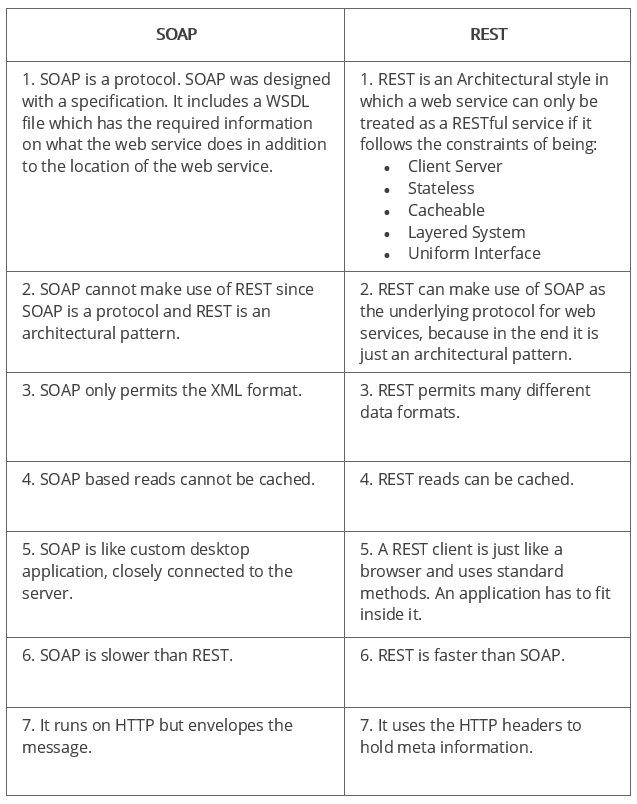
SOAP is typically significantly slower than other types of [**middleware**](https://searchmicroservices.techtarget.com/definition/middleware) standards, including CORBA, because SOAP uses a detailed XML format. A complete understanding of the performance limitations before building applications around SOAP is hence required.

SOAP is usually limited to pooling and not to event notifications when HTTP is used for the transport. In addition, only one client can use the services of one server in typical situations.

If HTTP is used as the transport protocol, firewall latency usually occurs since the firewall analyzes the HTTP transport. This is because HTTP is also leveraged for Web browsing, and so many firewalls do not understand the difference between using HTTP within a web browser and using HTTP within SOAP.

SOAP has different support levels, depending on the supported programming language. For instance, SOAP supported in [**Python**](https://whatis.techtarget.com/definition/Python) and PHP is not as powerful as it is in Java and .NET

### 59. What are the differences between SOAP and REST?



### 60. SOAP or Rest APIs, which method to use?

SOAP is the heavyweight choice for Web service access. It provides the following advantages when compared to REST:

* SOAP is not very easy to implement and requires more bandwidth and resources.
* SOAP message request is processed slower as compared to REST and it does not use web caching mechanism.
* WS-Security: While SOAP supports SSL (just like REST) it also supports WS-Security which adds some enterprise security features.
* WS-AtomicTransaction: Need ACID Transactions over a service, you’re going to need SOAP.
* WS-ReliableMessaging: If your application needs Asynchronous processing and a guaranteed level of reliability and security. Rest doesn’t have a standard messaging system and expects clients to deal with communication failures by retrying.
* If the security is a major concern and the resources are not limited then we should use SOAP web services. Like if we are creating a web service for payment gateways, financial and telecommunication related work, then we should go with SOAP as here high security is needed.

REST is easier to use for the most part and is more flexible. It has the following advantages when compared to SOAP:

* Since REST uses standard HTTP, it is much simpler.
* REST is easier to implement, requires less bandwidth and resources.
* REST permits many different data formats whereas SOAP only permits XML.
* REST allows better support for browser clients due to its support for JSON.
* REST has better performance and scalability. REST reads can be cached, SOAP based reads cannot be cached.
* If security is not a major concern and we have limited resources. Or we want to create an API that will be easily used by other developers publicly then we should go with REST.
* If we need Stateless CRUD operations then go with REST.
* REST is commonly used in social media, web chat, mobile services and Public APIs like Google Maps.
* RESTful service returns various MediaTypes for the same resource, depending on the request header parameter “Accept” as application/xml or application/json for POST and /user/1234.json or GET /user/1234.xml for GET.
* REST services are meant to be called by the client-side application and not the end user directly.
* ST in REST comes from State Transfer. You transfer the state around instead of having the server store it, this makes REST services scalable.

### 61. What are the factors that help to decide which style of Web services – SOAP or REST – to use?

Generally, REST is preferred due to its simplicity, performance, scalability, and support for multiple data formats.

However, SOAP is favorable to use where service requires an advanced level of security and transactional reliability.

But you can read the following facts before opting for any of the styles.

* **Does the service expose data or business logic?** REST is commonly used for exposing data while SOAP for logic.
* **The requirement from clients or providers for a formal contract**. SOAP can provide contract via WSDL.
* **Support multiple data formats**.
* **Support for AJAX calls.** REST can apply the XMLHttpRequest.
* **Synchronous and asynchronous calls.**SOAP enables both synchronous/ asynchronous operations whereas REST has built-in support for synchronous.
* **Stateless or Stateful calls.** REST is suited for stateless operations.
* **Security.** SOAP provides a high level of security.
* **Transaction support.** SOAP is good at transaction management.
* **Limited bandwidth**. SOAP has a lot of overhead when sending/receiving packets since it’s XML based, requires a SOAP header. However, REST requires less bandwidth to send requests to the server. Its messages are mostly built using JSON.
* **Ease of use**. REST based application is easy to implement, test, and maintain.

## API Testing Interview Questions And Answers:

**Types of API Testing:**

|  |  |
| --- | --- |
| **Testing Type** | **Description** |
| Unit testing | Individual operation functionality testing. |
| Functional testing | Block of unit test results tested together. |
| Load testing | Analyzing performance and functional ability when the load is applied. |
| Runtime error detection | Manual and automated executions to detect any sort of errors or data leaks. |
| Security testing | To verify the ability to safeguard access and data. |
| Web UI testing | GUI test to identify proper executions of any transactions. |
| Interoperability testing | Validate performance with Interoperability Profiles. |

**Q1. What does API testing mean?**

Ans: 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 andfunctions 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.

**For More Info:** [What is API Testing?](https://mindmajix.com/what-is-api-testing)

**Q2. What are the names of tests executed on APIs?**

**Ans:**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

**Q3. What is the procedure to perform API testing and what exactly needs to be checked?**

**Ans:**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

### Q4. What is the test environment of API?

**Ans:** 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 (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

### Q5. What approach should be followed for the API testing?

**Ans:** 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.

**Q6. Define the basic difference between API testing and UI level testing?**

**Ans:** 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 is 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.

**Q7. Name the common protocols used in API testing.**

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

JMS, REST, HTTP, UDDI, and SOAP

### Q8. Name different tools used for API testing.

**Ans:** 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 Alertsite API monitoring

### Q9. What is SOAP?

**Ans:** The term SOAP refers to Simple Object Access Control. In simple terms, it is an XML based protocol that helps in exchanging information among computers.

### Q10. What’s the procedure to test API’s?

**Ans:** 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 conditions

5. Once done with all the previous steps, configure method validation

6. Now is the time for 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!

### Q11. What is REST API?

**Ans:** 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.

### Q12. What are the areas that need to be taken care of while writing API document?

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

You need to check the source of the content

#### Subscribe to our youtube channel to get new updates..!

2. Plan or sketch of your document

3. Delivery layout of the same

4. Information needed for each of the function available in the document

5. Lastly, automatic document creation programs

### Q13. What is an API framework?

**Ans:**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.

### Q14. What do you mean by input injection? Explain different ways of doing it.

**Ans:** 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

Q15. Define API testing with Runscope.

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

### Q16. Explain the major challenges that come while API testing.

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

1. Parameter Combination

2. Parameter Selection and

3. Call Sequencing

### Q17. What are the main principles of API test design?

**Ans:** 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

### Q18. Explain the types of bugs that can be found using API testing?

**Ans:** 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. Performance

7. Incompatible error handling

8. Multi-threaded issues, and

9. Improper errors

### Q19. Name various tools used for API test automation.

### [API Testing Certification Training!](https://mindmajix.com/api-testing-training" \l "curriculum" \t "_blank)

[Explore Curriculum](https://mindmajix.com/api-testing-training" \l "curriculum" \t "_blank)

**Ans:** 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](https://mindmajix.com/net-training)

There are various automation tools for API testing:

1. HP UFT

2. Soap UI

3. JUnit for Java

4. NUnit for .Net

### Q20. What is the API documentation?

**Ans:** 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 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](https://en.wikipedia.org/wiki/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

### Q21. List some templates for API documentation that are most used.

**Ans:** 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

### Q22. Explain the difference between API testing and Unit Testing.

**Ans:**

* 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.

### Q23. Define TestApi?

**Ans:** 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.

### Q24. Explain everything about warnings and API errors.

Ans: 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 for 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:

### Q25. Explain the working of API Builder.

**Ans:**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 for 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 as output\_script\_.sq.

## ****Basic API Interview Questions****

### Q.1 What are the benefits of API testing?

**Ans**:

* 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

### Q.2 What are the challenges faced in API testing?

**Ans.** 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.

### Q.3 Explain how to document each function in the API document?

**Ans:**

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

### Q.4 What are the tools used for API documentation?

**Ans:**The free tools used for API documentation are ReDoc, Swagger UI, and DapperDox.

### Q.5 What’s the difference between API and Web Service?

**Ans**.

|  |  |
| --- | --- |
| **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. |

### Q.6 What are the architectural styles used for creating a Web API?

**Ans:**

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

### Q.7 How to perform API testing?

**Ans:** 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.

### Q.8 Why is API testing the most preferred for Automation testing?

**Ans:** 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.

### Q.9 Name a few API documentation templates?

**Ans.** 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

### Q.10 What to be considered to create great API documentation?

**Ans:**

* 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

### Q.11 What are the differences between SOAP and REST API?

**Ans:**

|  |  |
| --- | --- |
| **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. |

### Q.12 What is messaging in RESTFUL Web Services?

**Ans:**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.

### Q.13 What are the main components of an HTTP request?

**Ans:**

* 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.

### Q.14 Which HTTP protocols are supported by REST?

**Ans:**

* 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 which is similar to GET requests, but without the response body.

### Q.15 What is URI? What is the purpose of a web-based service, and what is its format?

**Ans:**

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 :////.

### Q.16 Define the caching mechanism.

**Ans:**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.

### Q.17 What’s the difference between PUT and POST operations in Rest API?

**Ans.**

|  |  |
| --- | --- |
| **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, results will be different. |
| PUT works as specific. | POST work as abstract. |

### Q.18 Can we use GET requests instead of PUT to create a resource?

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

### Q.19 What are the commonly used HTTP methods for RESTful services?

**Ans:**

* 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

### Q.20 What is Payload in REST API?

**Ans.**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.