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

### 1) What is API?

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

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

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

191.2K

Conspiracy/Cryptid Tier List with Matt Pike

### 2) What is API testing?

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

### 3) What are the types of API testing?

API testing involves the following types of testing:

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

### 4) What are the protocols used in API Testing?

Protocols used in API testing are:

* HTTP
* REST
* SOAP
* JMS
* UDDI

### 5) What are the tools used for API Testing?

Tools used for API testing are:

* Parasoft SOAtest
* PostMan
* AlertSite API monitoring

### 6) What is API test environment?

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

API is checked for its proper functioning after installation.

### 7) What is API framework?

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

### 8) What are the limits of API usage?

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

### 9) What are the advantages of API testing?

Advantages of API testing are:

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

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

Here, are the seven principles of API test design.

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

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

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

### 11) What is API framework?

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

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

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

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

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

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

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

### 13) What exactly needs to verify in API testing?

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

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

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

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

### 15) What is API documentation?

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

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

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

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

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

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

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

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

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

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

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

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

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

### 19) What is SOAP?

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

### 20) What is REST API?

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

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

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

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

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

The major challenges faced during the API testing are:

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

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

Difference between API testing and Unit testing are:

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

### 24) What is a RESTFUL web services?

There are two kinds of web services

1. SOAP Web Services
2. RESTFUL Web Services

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

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

### 25) What is Resource in REST?

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

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

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

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

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

### 28) What are the characteristics of REST?

Here, are the two characteristics of REST.

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

### 29) What is messaging in RESTFUL Web Services?

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

### 30) What are the components of an HTTP request?

An HTTP request have five components. These are:

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

### 31) What is the HTTP protocol supported by REST?

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

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

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

POST requests are never cached or bookmark.

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

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

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

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

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

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

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

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

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

### 34) What are SOAP Web Services?

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

### 35) When we can use SOAP API?

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

# 2.KATALON: <https://katalon.com/resources-center/blog/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/api-testing/) is a kind of software testing that determines if the developed APIs meet expectations regarding the functionality, reliability, performance, and security of the application.

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

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

* **Test for Core Functionality:**API testing provides access to the application without a user interface. The core and code-level of functionalities of the application will be tested and evaluated early before the GUI tests. This will help detect 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.

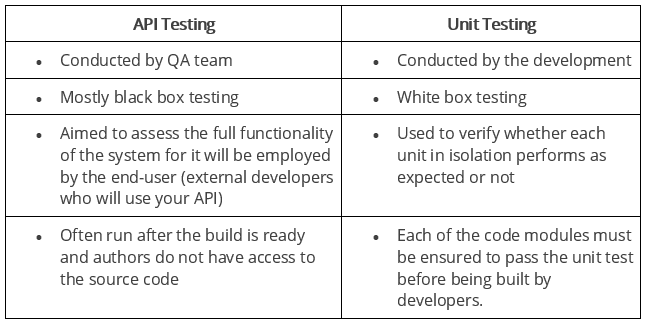
### 15. What are tools could be used for API testing?

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?****

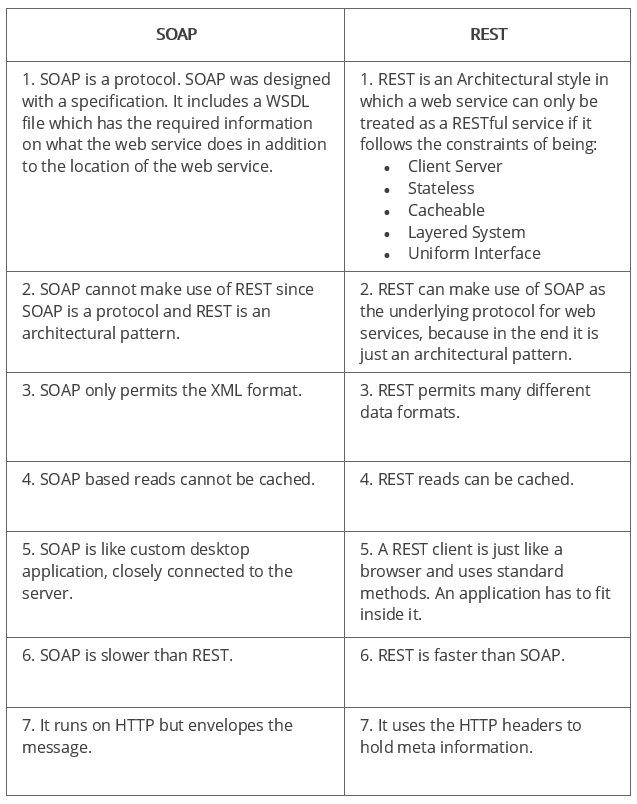
SOAP is typically significantly slower than other types of [middleware](https://searchmicroservices.techtarget.com/definition/middleware) standards, including CORBA, because SOAP uses a detailed XML format. A complete understanding of the performance limitations before building applications around SOAP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

# InterviewBit: <https://www.interviewbit.com/api-testing-interview-questions/>

### 1. What is API testing ?

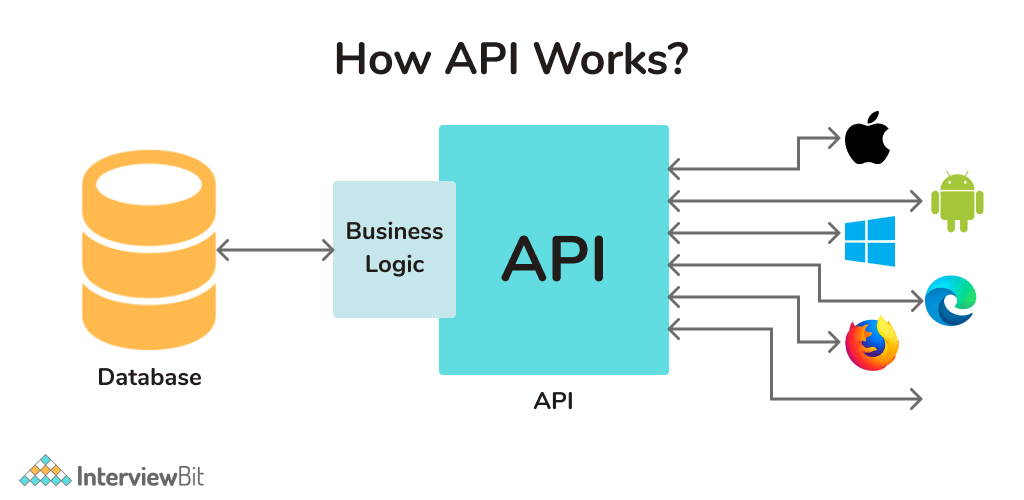
API testing is a category of software integration testing that deals with the testing of Application Programming Interfaces (APIs) directly. It deals with checking if the APIs developed work as expected in terms of reliability, functionality, security, and performance of the business logic covered by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

### 2. What is API?

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

### 3. How do APIs work?

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



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

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

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

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

There are various types of API testing, they are:

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

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

API testing can be used for testing the following protocols:

* HTTP
* REST
* SOAP
* JMS
* UDDI

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

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

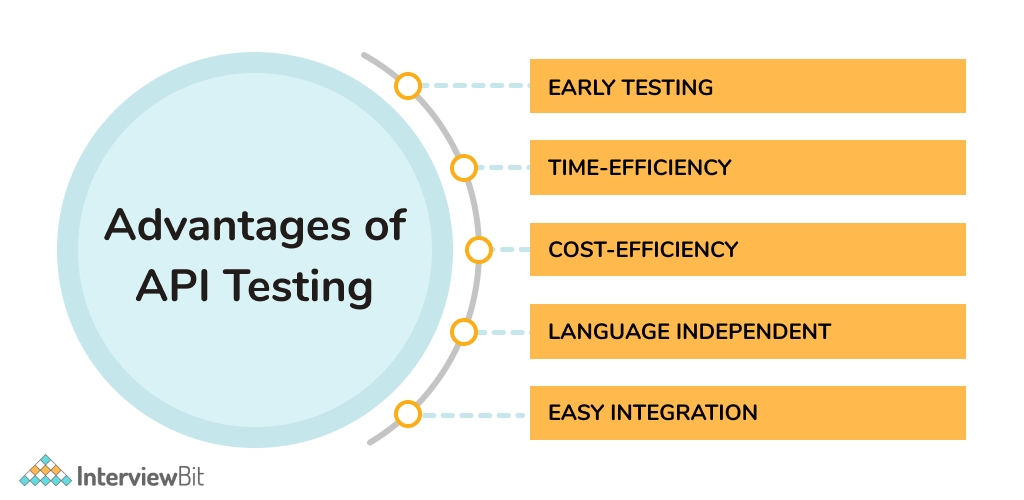
### 7. What are the differences between API Testing and Unit Testing?

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

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

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

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



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

We follow the below approaches in API Testing:

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

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

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

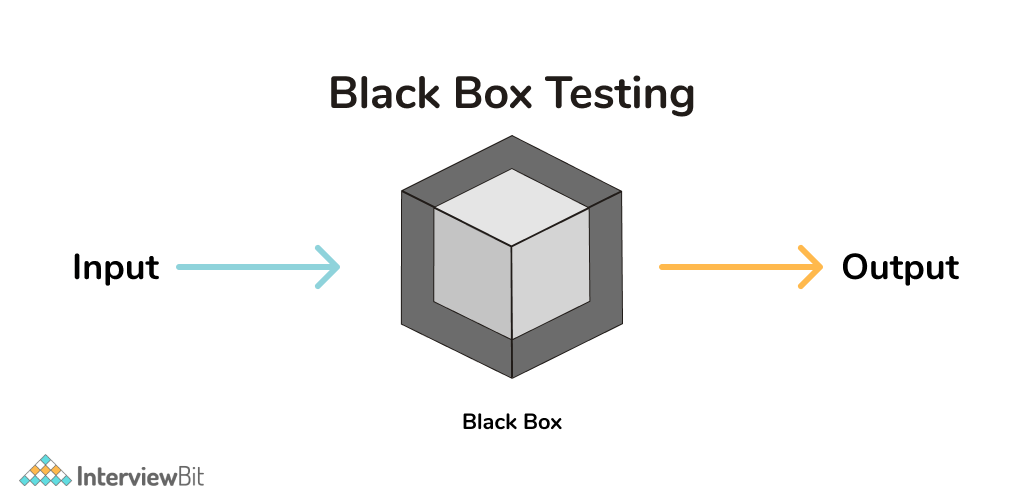
1. Data accuracy of the actual response with the expected response.
2. HTTP status codes of the resultant API.
3. Response time of the API.
4. Error codes if the APIs are expected to throw an error.
5. Authorization Details
6. Test non-functional specifications of the applications like security, performance,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

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

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

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

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



### 13. Define Test Data.

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

### 14. Define test coverage.

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

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

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

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

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

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

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

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

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

## API Testing Interview Questions for Experienced

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

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

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

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

* JavaDoc
* Doxygen
* OpenAPI
* Redoc
* Swagger UI

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

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

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

Some of the challenges faced while doing API testing are:

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

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

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

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

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

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

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

### 24. Define Test API.

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

### 25. What is Payload?

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

### 26. What is Run Scope?

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

# CareerGuru: <https://career.guru99.com/top-20-questions-on-api-testing/>

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

The tools used for various API testing are

* SoapUI Pro
* PostMan
* Alertsite API monitoring

**2) What is API testing?**

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

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

The common tests performed on API’s

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

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

UI ( User Interface) refers to testing graphical interface such as how user interacts with the applications, testing application elements like fonts, images, layouts etc. UI testing basically focuses on look and feel of an application.



While, API enables communication between two separate software systems. A software system implementing an API contains functions or sub-routines that can be executed by another software system

**5) Explain what is SOAP?**

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

**6) Explain what is REST API?**

It is a set of functions to which the developers performs requests and receive responses. In REST API interaction is made via HTTP protocol

REST – stands for Representational State Transfer, it is quickly becoming defacto standard for API creation.

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

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

**8) How to test API’s ?**

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

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

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

The key area to be considered when writing API documents are

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

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

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

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

**11) Explain API framework?**

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

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

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

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

**13) Explain what is TestApi ?**

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

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

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

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

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

The main challenges in API testing is

* Parameter Selection
* Parameter Combination
* Call sequencing

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

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

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

The principle for API test design are

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

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

The types of Bugs, API will find

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

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

While testing Unit and API testing,  both target source code, if an API method is using code  based on .NET then the tool which is supporting should have .NET

Automation tools for API testing can be used are

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

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

API testing steps

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

**21) What are the common protocols that are testing in API tesing ?**

* HTTP
* JMS
* REST
* SOAP
* UDDI

# STH: <https://www.softwaretestinghelp.com/api-testing-interview-questions-and-answers/>

**Q #1) What is API Testing?**

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

**Answer: Best API Testing tools:**

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

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

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

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

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

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

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

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

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

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

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

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

**API documentation tools are:**

* JavaDoc
* Doxygen

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

* Parameter validation warning
* Missing module warning

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

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

# STM: <https://www.softwaretestingmaterial.com/api-testing-interview-questions/>

### ****1. What is an API?****

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

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

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

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

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

API testing typically involves the following practices:

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

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

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

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

* HTTP
* REST
* SOAP
* JMS
* UDDI

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

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

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

### ****What is API test environment?****

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

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

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

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

**Web services:**

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

**APIs:**

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

### ****6. What is Soap?****

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

### ****7. What is Rest API?****

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Some of the API documentation templates are as follows.

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

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

Some of the popular API examples are

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

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

**UNIT TESTING:**

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

**API TESTING:**

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

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

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

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

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

Issues observed when performing API testing are

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

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

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

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

Some of the HTTP methods are

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

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

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

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

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

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

# 7. MindMajix: <https://mindmajix.com/api-testing-interview-questions>

### 1. What does API testing mean?

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

JMS, REST, HTTP, UDDI, and SOAP

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

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

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

### 9. What is SOAP?

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

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

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

1. Make a selection of the suite you like to add the API test case to
2. Now choose the test development mode
3. Next demands the development of test cases for the required API methods
4. After this, you need to configure the control parameters of the application and then test 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!

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

### 11. What is REST API?

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

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

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

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

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

### 13. What is an API framework?

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

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

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

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

### 15. Define API testing with Runscope.

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

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

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

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

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

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

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

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

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

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

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

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

There are various automation tools for API testing:

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

### 20. What is the API documentation?

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

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

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

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

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

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

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

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

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

### 23. Define TestApi?

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

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

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

A wide range of warnings that can form are:

Missing module warning and parameter validation warning

A wide range of errors that can form are:

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

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

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

## API Testing Interview Questions For Experienced

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

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

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

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

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

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

Description: Small description of what a function does

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

Parameters: Functions parameters

Error Messages: Syntax of error messages

Example Code: Small snippet of code

Related Links: Related functions

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

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

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

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

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

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

### 32. How to perform API testing?

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

API testing should perform the following testing methods:

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

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

API testing is considered most suitable for automation testing because:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

### 41. Define the caching mechanism.

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

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

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

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

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

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

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

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

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

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

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

# JANBASK: <https://www.janbasktraining.com/blog/api-testing-interview-questions/>

**Q1. What is an API?**

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

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

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

**Q2. What is API Testing?**

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

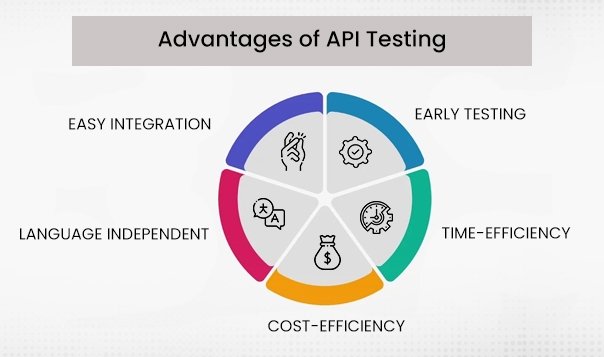
**Ans:** API testing is the testing of the developed so APIs to check their functionality, efficiency, reliability, and security.

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

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

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

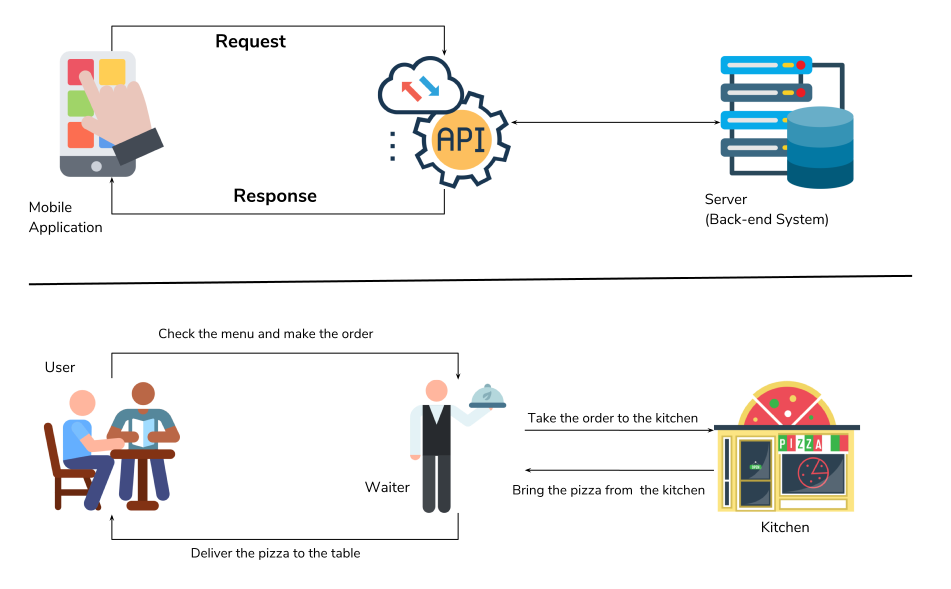
Give this answer in a way to show the interviewer the importance and relevance of API testing and where and how you are going to use it or have already used it.



**Ans:-**

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

**Q4. How does API work?**

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

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

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

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

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

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

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

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

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

**Q8.  What to perform API test?**

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

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

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

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

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

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

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

**Q11. Define Test Data.**

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

**Q12. Define test coverage.**

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

**Q13. What is a SOAP web service?**

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

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

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

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

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

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

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

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

**Q17. What is test documentation?**

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

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

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

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

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

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

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

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

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

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

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

**Ans:-**

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

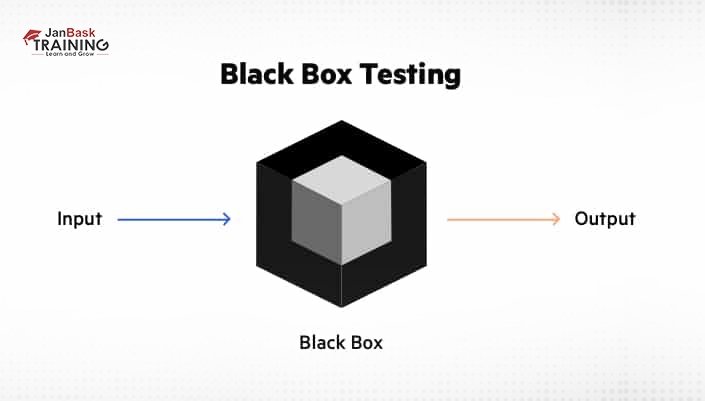
**Ans:-**It is one of the API design interview questions. You can answer it like this:

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

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

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

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



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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

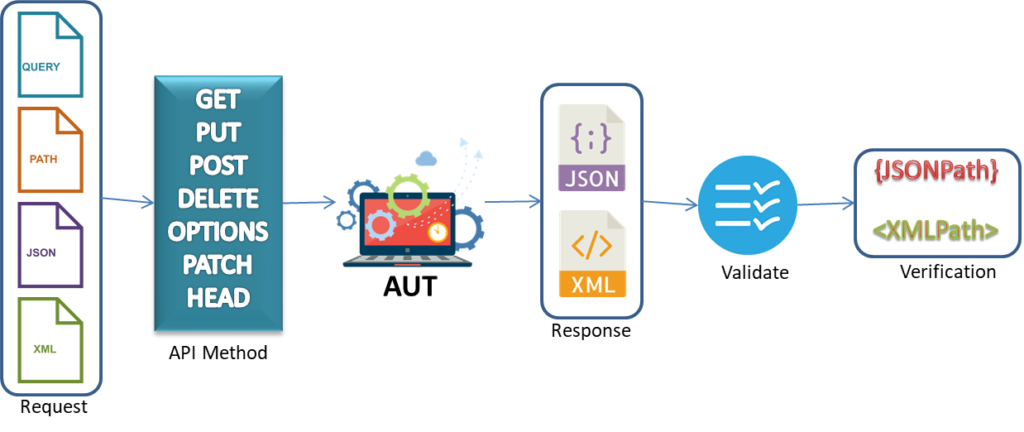
**Ans:-** There are numerous API documentation templates available to make the entire process simple and straightforward.

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

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

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

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



**Q38. What is REST assured?**

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

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

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

**Q40. What is URI?**

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

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

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

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

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

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

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

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

**Q44. can we use GET request instead of PUT to create a resource?**

**Ans.**PUT and POST method is used to create a resource. GET is only used to request the resources.

### D. Frequently Asked SOAP Interview Questions and Answers

Get the Knowledge of SOAP to be able to answer any SOAP-related questions swiftly.

**Q45. What are SOAP Web services?**

**Ans:-** This is one of the fundamental API testing interview questions that you must know the answer to.

The SOAP (Simple Object Access Protocol) is defined as an XML-based method used in web services.  It is both platform and language independent.

It is well known for designing and developing web services. It also enables the communication between applications developed on different platforms using various programming languages over the Internet.

**Q46. How does SOAP work?**

**Ans:-**SOAP provides a user interface that can be accessed by the client’s server object, and the request sent by it goes to the server, which can be accessed using the server object. The user interface creates some files or methods consisting of server objects and the name of the interface to the server object.

 It also contains other information such as the name of the interface and methods. It uses HTTP to send the XML to the server using the POST method, which analyzes the method and sends the result to the client.

The server creates more XML consisting of responses to the request of the user interface using HTTP. The client can use any approach to send the XML, like the SMTP server or POP3 protocol to pass the messages or reply to queries.

**Q47. What are the syntax rules for a SOAP message?**

**Ans:-** These are some basic syntax rules for an SAOP message

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

**Q48. How do users utilize the facilities provided by SOAP?**

**Ans:-**

* **PutAddress():** we use it to enter an address in the webpage and has an address instance on the SOAP call.
* **PutListing():** It is utilized to allow the insertion of a complete XML document into the web page. It receives the XML file as an argument and transports the XML file to XML parser liaison, which reads it and inserts it into the SOAP call as a parameter.
* **GetAddress():**It helps in getting a query name and getting the result that best matches a query. The name is sent to the SOAP call in the form of the text character string.
* **GetAllListing():**It helps to return the full list in an XML format

**Q49. Tell the elements of SOAP message structure**

**Ans:-** It is an ordinary XML document that contains the elements as a SOAP message

* **Envelope:**It is a mandatory root element that translates the XML document and defines the start and end of the message
* **Header:** It is an optional element. It stores the information about the message being sent
* **Body:** It contains the XML data comprising the message being sent
* **Fault:** It informs about the errors that occurred while processing the message

**Q50. What are the obstacles users face while using SOAP?**

**Ans:-**The major difficulty faced by users using SOAP is a firewall security mechanism. This locks all the ports leaving some like HTTP port 80 used by SOAP that bypasses the firewall.

The major technical issue with SAOP is, it mixes specifications of message transport and message structure.

# ALLSTQ: https://allstq.com/api-testing-interview-questions/

### 1- What are the main challenges you face in the API testing in your project?

Challenges are for example,

**API documentation**

**Access to the database**

**Call sequencing**

### 

### 2- What is the difference between the PUT and the POST method?

This is the most common API testing interview question these days.

* Firstly, the POST request means to create a new object in the database.
* Secondly, a PUT request means to update the existing object in the database with the new value.

### 

### 3- What are the most commonly used HTTP methods?

For example:

|  |  |
| --- | --- |
| GET | Used to retrieve data from the server |
| POST | create a new object in the server |
| PUT | Used to update an existing object in the server |
| DELETE | used to delete data from the server |

### 

### 4- List a few authentication techniques used in API testing.

* Session / Cookies based Authentication
* Basic Authentication
* Digest Authentication
* OAuth

Above all are the few most important authentication techniques.

### 

### 5- What is the REST API?

REST – Representational State Transfer, is a set of functions that helps developers performing requests and receive responses. Interaction is made through HTTP protocol in REST API.

### 6- What exactly you verify in API testing?

**Accuracy of data**

**HTTP status code**

**Response time**

**Error codes if API returns an error**

**Authorization**

**Performance**

**Security**

Above all are the most important verification checks.

### 7- Differentiate API testing and UI testing.

UI (User Interface) testing means testing the graphical user interface. The focus of UI testing is on the look and feel of the application, like how the user interacts with the application elements, such as images, font, layout, etc.

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

To clarify, describe some scenarios while answering this question in an interview.

### 8- What protocol RESTFUL Web services use?

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

### 9- Can we use POST instead of PUT to create a resource?

Yes, we can because POST is the superset of all HTTP requests except GET requests.

### 10- What do you understand by payload?

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

### 11- What kind of bugs that API testing can find?

**Missing or duplicate functionality**

**Fail to handle error conditions gracefully**

**Stress**

**Reliability**

**Security**

**Unused flags**

**Performance**

**Multithreading issues**

**Improper errors**

Above all are the most frequent bug that API testing can detect.

### 12- Describe the term Environment with respect to Postman?

The environment in Postman is a set of key-value pairs. We can create multiple environments in postman.

There are two types of environment, the global environment, and the local environment. They define the scope of the variable to use it in the requests.

The most common variable we use is URL because the URL is used in every request and changing it in every request can be very time-consuming.  When we create an environment inside Postman, we can change the value of the key-value pairs and the changes are reflected in the requests.

### 

### 13- State the common status code you encounter in API testing.

This is the most common API testing interview question.

**200 (OK)** Defines that the request was correct.

**201 (Created)** The value wrapped with the request has been created in the database.

**204(No Content)** This status code means that the request was correct and received but there is no response to send to the client by the server.

**400 (Bad Request)** A bad request means that the syntax of the request was incorrect. It can happen if you have sent the wrong parameters along with the request URL or in the body of the request.

**401 (Authorized)**  We can incur such a status code when you are not authorized to access the server or you have entered the wrong credentials.

**404 (Not Found)** A response code 404 means that the server was connected but it could not find what was requested. You can normally see this status code when you request a web page that is not available.

**500(Internal Server Error)** A response code 500 means there was some exception at the server level while executing the request.

**502(Bad Gateway)** The server, while acting as a gateway or proxy, received an invalid response from the upstream server it accessed in attempting to fulfill the request.

**503(Service Unavailable)** The server is currently unable to handle the request due to a temporary overloading or maintenance of the server.

**504(Gateway Timeout)** The server, while acting as a gateway or proxy, did not receive a timely response from the upstream server specified by the URI

Above all are the most common status codes.

### 14- What is Pre-Request Script in Postman?

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

### 15- What is the difference between authorization and authentication?

* Authentication is a process of presenting your credentials to the system and then the system validates your credentials. These credentials tell the system about who you are.
* Authorization is a process of allowing or denying someone from accessing something once authentication is completed.

### 

### 16- What is the importance of setNextRequest in Postman?

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

### 17- What are the two types of scripts in Postman?

* Tests script
* Pre-request script

Above are the types of scripts in postman.

### 18- What is REST?

Representational State Transfer is an architectural style of developing web services. In this architecture, the server provides access to resources and the client presents those resources. Each resource is identified by URI. REST uses different ways to represent a resource like JSON, text and XML. XML and JSON are the most popular one. Resource are accessed by a common interface using HTTP standard methods.

### 19- Which is the most popular way to represent a resource in REST?

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

### 20- What do you understand by messaging in RESTful web services?

RESTful web services use HTTP as a medium of communication between client and server. The client sends a message in the form of an HTTP request and then the server transmits the HTTP response. This technique of interaction is called messaging. These messages contain both message data and metadata (information about the message itself).

### 21- List the core components of an HTTP request?

* HTTP methods type such as GET, PUT, POST, DELETE
* URI that acts as an identifier for the resource on the server
* HTTP Version
* Request Header, Metadata, Cache Settings, Authentication Parameters
* Request Body or the Payload

Above are the core components of an HTTP request.

### 22- What is Rest Assured?

Rest Assured is a java based library that is used to test the RESTful Web Services. It acts as a headless client to access REST services. REST Assured provides a lot of features, which makes API automation testing very easy. Like it offers friendly DSL-like syntax, XPath-Validation, Specification Reuse, Easy file uploads.

### 23- Define what is a URI?

Uniform Resource Identifier, URI consists of base URL, path parameter, and query parameter

URI= Base URL + Path Parameter + Query Parameter

Example:

URI-   <https://reqres.in/api/users?page=2>

### 24- What do you mean by the HTTP status code?

REST APIs use HTTP status codes to tell what exactly happened when the server processed the request.

 Grouping for HTTP Status Codes will be :

* **1xx** – Informational
* **2xx** – Success e.g. 200 Success, 201 Created
* **3xx** – Redirection e.g. 302 Temporary Redirect
* **4xx** – Client Error e.g. 400 Bad Request, 404 Not Found
* **5xx** – Server Error e.g. 500 Internal Server Error

The type of status code you receive depends on the application you are interacting with. Usually, a 4xx error means that you have done something wrong and a 5xx error means that something has gone wrong with the application server you are interacting with.

### 25- Explain the main differences between API and Web Service?

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

### 

### 26- Who can use a Web API?

Clients that support HTTP verbs such as GET, PUT, DELETE, POST. Since Web API services do not require configuration, they can be easily used by any client. Portable devices such as mobile devices can easily use Web API, which is undoubtedly the biggest advantage of API.

### 27- List the advantages of API Testing?

* Compatibility and easy integration with GUI: Simple integration would allow new user accounts to be created within the application before a GUI test started.
* Language-Independent: In API testing, data is exchanged using XML or JSON. These transfer modes are completely language-independent that allows users to select any coding language while adopting automation testing services for the project.
* Time Effective: In comparison to functional GUI testing API testing usually is less time-consuming. The web elements in GUI testing must be polled, which makes the testing process slower. Particularly, API test automation requires less code so it can provide better and faster test coverage compared to GUI test automation.
* Test for Core Functionality: API testing provides the ability to access the application without a user interface. The core and code-level functionalities of the application will be tested and evaluated early before the GUI tests. This helps in detecting the minor issues which can become bigger during the GUI testing.

### 

### 28- What are the principles of an API test design?

Basically, there are the five most important principles of an API test design:

**Setup**: Create objects, start services, initialize data, etc

**Execution**: Steps to apply API or the scenario, including logging

**Verification**: Oracles to evaluate the result of the execution

**Reporting**: Pass, failed, or blocked

**Clean up**: Pre-test state

### 

### 29- List the common API testing types?

**Validation Testing**

**Functional Testing**

**UI testing**

**Load testing**

**Runtime/ Error Detection**

**Security testing**

**Penetration testing**

**Fuzz testing**

**Interoperability testing**

### 

### 30- What is the procedure to perform API testing?

**Create the suite to add the API test case**

**Create the test development mode**

**Demand the development of test cases for the required API methods**

**Configure the control parameters of the application and then test conditions**

**Configure method validation**

**Arrange all API test cases**

**Execute the API test**

**Check test reports**

### 

### 31- What must be checked when performing API testing?

**Accuracy of data**

**Non-functional testing like performance and security testing**

**Implementation of response timeout**

**Schema validation**

**HTTP status codes**

**Data type, validations, order, and completeness**

**Authorization checks**

**Error codes in case API returns**

### 

### 32- Difference between API Testing and UI Testing?

API enables communication between two separate software systems. A software system implementing an API contains functions or subroutines that can be executed by another software system.

On the other hand, UI ( User Interface) testing refers to testing graphical interfaces such as how users interact with the applications, testing application elements like fonts, images, layouts, etc. UI testing basically focuses on the look and feel of an application.

### 

### 33- What are the types of Bugs API testing can find?

The types of Bugs, API will find

**Missing or duplicate functionality**

**Stress**

**Reliability**

**Security**

**Performance**

**Unused flags**

**Not implemented errors**

**Inconsistent error handling**

**Improper errors**

**Fails to handle error conditions gracefully**

***Multi-threading issues***

### 34- What is API Automation?

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

There are many tools for API automation like-

SOUP UI

Katalon studio

Postman

Jmeter

RestAssured

CloudQA TruAPI

### 35- How we can segregate the entire system into three layers?

**Presentation Layer –** This is the user interface (GUI) that is open to end-users. QA performs functional testing at this layer.

**Business Layer –** This is the Application user interface where the logic is written. In technical terms, this is where code/algorithm resides. APIs come into the picture at this layer.

**DataBase Layer –**Where application data is present.

### 36- List the main differences between SOAP and REST?

|  |  |
| --- | --- |
| **SOAP** | **REST** |
| 1. SOAP is a protocol through which two computers communicates by sharing XML document. | 1. Rest is a service architecture and design for network-based software architecture. |
| 2. SOAP supports the only XML format. | 2. It supports many different data formats. |
| 3. SOAP does not support caching. | 3. It supports caching. |
| 4. SOAP is like a custom desktop application, closely connected to the server. | 4. A REST client is just like a browser and uses standard methods. An application has to fit inside it. |
| 5. SOAP is slower than REST. | 5. It is faster than SOAP. |
| 6. It runs on HTTP but envelopes the message. | 6. It uses the HTTP headers to hold meta information. |

### 

### 37- What is the caching mechanism?

Caching is a process in which we store server responses at the client end. It allows the server to save significant time from serving the same resource again and again.

The server response holds information that leads a client to perform the caching. It helps the client to decide how long to archive the response or whether not to store it at all.

### 38- Is there any upper limit for a payload to pass in the POST method?

Theoretically, one can pass unlimited data as the payload to the POST method. But, while considering a real use case, then sending a POST with a large payload will consume large bandwidth. It will take more time and cause performance challenges to the server.

### 39- State some of the API examples which are very well known.

**Google Maps API:**

These are designed mainly for mobile and desktop use with the help of flash interface and JavaScript.

**Amazon Advertising API:**

Amazon is known for its products and thus their advertising API accesses their product to discover their functionality and thus advertise accordingly.

**Twitter:**

The API for Twitter is usually in two categories, one for accessing data and the other for interacting with Twitter search.

**YouTube:**

This API used for YouTube includes various functionalities including videos, live streaming, player, etc.

### 

### 40- What is REST parameters?

The REST API has four types of parameters:

**Request parameter**s – These are submitted as JSON parameters present in the request.

**Header parameters** – These are present in the request header.

**Query string parameters** – These are provided at the endpoint of the query.

**Path parameters** – These are provided in the endpoint path.

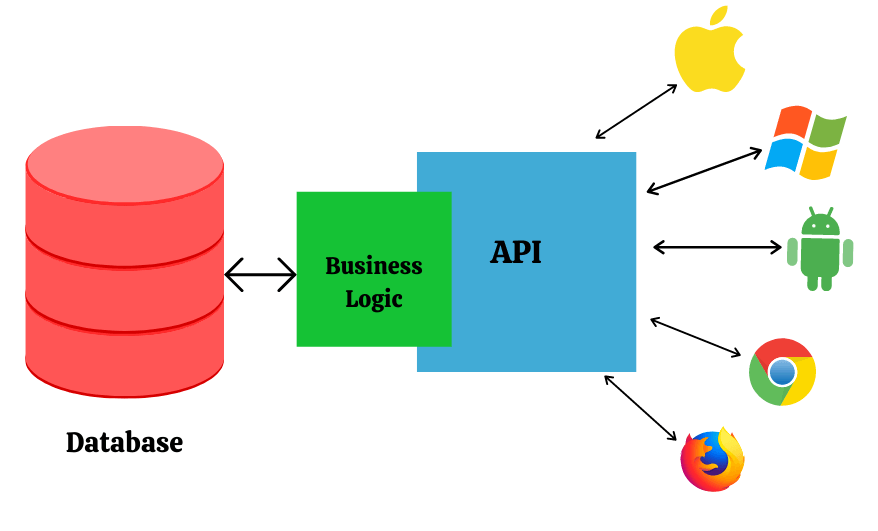
# AutomationsQA: <https://automationtestings.com/api-testing-interview-questions/>

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

API full form-  **A**pplication **P**rogramming **I**nterface. By using this, we can communicate and data exchange between software systems. It acts as an interface between two applications to communicate and share data with each other.

API takes the user’s request (request from the database, request from the source) and return a response to the user without exposing the internal details. API acts as an abstraction.

**Example**: Google Map API, Amazon advertising API, Twitter etc.

Application Programming Interface

### ****What are the different types of API testing?****

API testing uses the following types of testing:

1. Unit testing
2. Load testing
3. Functional testing
4. Runtime/Error detection
5. UI testing
6. Security testing
7. Penetration testing
8. Fuzz testing

### ****What are the different protocols used in API testing?****

Some of the standard protocols used in API testing are as follows:

HTTP

REST

SOAP

JMS

UDDI

### ****What are the tools used in API testing?****

#### ****Best tools for API testing:****

SOAPUI

Katalon Studio

POSTMAN

Apigee

Rest-Assured

### ****Is JMeter used for API testing?****

Yes, [JMeter](https://jmeter.apache.org/" \t "_blank) used for API testing.

It is an open-source tool—lots of plugins and extensions. It supports Load testing, Fast API testing, Stress testing.

### ****What is a better tool? POSTMAN or SoapUI?****

Both tools have different advantages. POSTMAN tool uses to perform manual API testing, whereas SoapUI uses for automation in API testing.

### ****List some most used templates for API documentation?****

Some template which makes API documentation much easier and simple. Below is mentioned

* Swagger
* API blueprint
* Restdoc
* Miredot
* Web Services API Specification

### ****What is the test environment of API?****

The easiest way to test an API to send the request to the server where API service is configured.

The test environment is a localized version of the production where QA can perform testing.

For example, in a test environment, you may have-

A test database

A localized load balancer

A localized API gateway

A localized API server

#### ****Kinds of test environments –****

Test server

Testing on local builds

Testing Sample apps

Testing hardware products

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

There are various things that we need to be validating in API testing.

1.**Requirement document of API testing-**

For better result, we need to write every requirement. So we need to prefer a requirement document.

2. **Set the output result of API testing-**

Once the requirement document is ready, we need to finalize the output of the API tests.

We need to verify the response status code which could be pass/fail/invalid etc.

3. **Need to focus on small API functions-**

**API**testing is different from another testing type so we cannot jump directly to the more significant test case. The small test case in API testing is less chance of uncertainty in the API tests.

4. **Organize endpoints-**

In testing, the project may have few or even hundreds of APIs for testing. We highly suggest that we need to organize then into categories.

5. **Create positive and negative tests**–

**Positive Test –**Send valid input/request to API and receive the expected result.

**Negative Test –**Send an invalid request to API and receive the expected result and see expected behavior.

### ****What is Web Services?****

Web Services is used to interact with the server without open web pages on the browser.

Let’s relate web services to the web application. In the web application, we have a server on which application is deployed. In this we have a URL, we are putting the URL on the browser so that a web page will be open, here we are giving some input and get output. So this is web application.

But in Web Service, application deployed to the server, but we don’t have any URL for access that application on any browser. So web services allow a program (write the program in any programming languages) to interact with the server, sending input to the server and receiving output from the server. So web service does not provide any GUI to interact with the server.

**Note:** If I am providing a URL or UI of my application, the user can access my application through only one UI. But in the case of web service, we write the business logic.

Any other application written in any programming language could be a web application or a mobile application to interact with me because I interact with any programming language because I am just providing the programming interface.

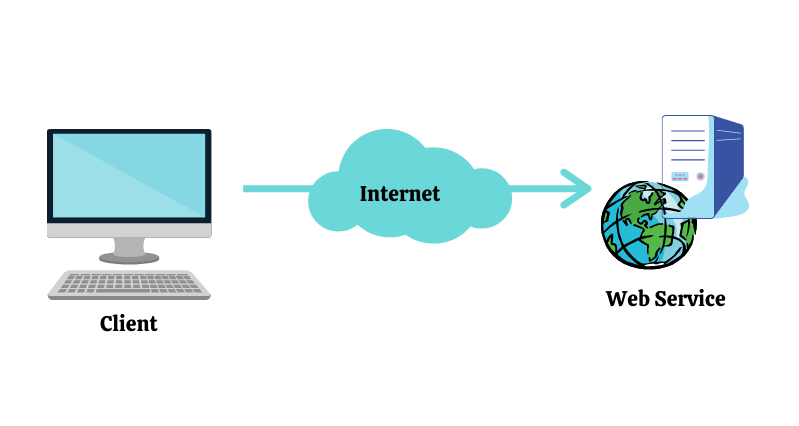
**Example:**

**Google Map web service:** Google developed a web service and provided a UI as well, so we can search the location by typing the address in this application.

So if I want to develop an application where I need the map to find the location, I have two choices either I can develop a whole mapping program from scratch or call a Google web service that Google map provides.

Like Ola cabs developed its UI and purchased the web services of Google mapping, So Ola application behind uses Google web services.

The logic is that when searching any cab on Ola application (it means giving request to the server) then by Google web service we get the result of pick up and drop location which shows on Ola cab application’s UI.

                                                                                            Web Service

**Must Read** – **[Web Services Interview Questions](https://automationtestings.com/web-services-interview-questions/)**

### ****Difference between API and Web Services?****

**API**and **Web Services**both are tech terms that regularly get confused, but there are some differences in both term.

**API (Application Programming Interface):**

It is a set of routines, data structures, classes which are developed by any developer. In other words, it defines methods for one software program to interact with other programs.

APIs don’t need a network to perform any operation.

All APIs are not Web Services.

**Web Services:**

Web Services is used to interact with the server without open web pages on the browser.

It uses the machine-processable format WSDL file to describe the web services.

|  |  |
| --- | --- |
| **API** | **Web Services** |
| It doesn’t require network for operation. | It requires network for operation |
| All API are not a Web Services | All Web services are API |
| It is an open-source but can be used by who know XML or JSON | It is not an open source and can be used by who know XML |
| It supports only HTTP/HTTPs protocol | It supports only HTTP protocol |
| It is used any style of the communication. | It is used for REST, SOAP and XML-RPC for communication. |

**Enlist some API testing tools.**

Here some of the tools used for API testing are as follows:

POSTMAN

SoapUI

Apigee

JMeter

Rest-Assured

Parasoft

HP QTP(UFT)

Ping API

SOAP Sonar

Unirest

Mockbin

Citrus Framework

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

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

**What exactly needs to test in API testing?**

API testing is the part of the integration testing process.

It is testing both request and response needs to be tested.

**API request:**

URL

Request parameters

Request format

**API response:**

Response code (200/2xx/3xx etc)

Response format

Validity of response data url/image/text

### ****Does API testing need coding knowledge?****

In API testing, we do perform both manual and automation testing.

**Manual API testing-**

In manual testing, we don’t require any type of coding knowledge. We require API path, body and header etc.

For manual API testing, there are lots of tools available in the market like Swagger, POSTMAN, RESTClient and Insomnia.

**Automation API testing-**

For API automation testing, you should have knowledge of coding for using tools like REST assured and SoapUI.

### ****What are the types of bugs can be found in API testing?****

API testing can detect different types of bugs which could make product bad. Here are some common types of defects.

Improper messaging

Error handling

Performance issue

Security Issue

Multi-threaded issues

### ****What is SOAP?****

**SOAP**stands for **S**imple **O**bject **A**ccess **P**rotocol.

It is a communication protocol designed to exchange information via the internet. It supports the XML message format.

### ****How is the UI level testing different from API testing?****

In UI testing, we test the user interface level is to test the graphical interface of the application, which includes text type, font size, layout etc.

Whereas, in API testing, we test the communication between different software, and it mainly resides in the business logic layer. It never looks on to the application.

**[Karate Framework Tutorial](https://automationtestings.com/karate-framework/)**

### ****What is TestAPI?****

TestApi is a library of test and utility API which are essential for developers and testers for creating testing tools and automation tests.TestApi provides common types of data structure and algorithms.

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

**UNIT TESTING:**

Unit testing is done by developer

Unit testing comes under white-box testing

Unit testing is done before prior to the build deployment.

Only basic functionalities testing performs in unit testing

The Source code is involved in unit testing.

**API TESTING:**

API testing is done by QA

API testing comes under black-box testing

API testing performed after build deployment

Scope of testing is wide. The Source code is not involved in this testing.

### ****How do you test API security Testing?****

In API security testing, mainly we perform two operation-

**Authentication –** It determines the identity of an end-user.

**Authorization**– It determines what resources a user can access. Also, we usedSSL/TLS certificate and used over HTTPS

### ****Can API be hacked?****

Yes, because we are sending data over the internet.

We use mostly HTTP protocol, which goes to REST and SOAP and it is a text-based protocol which therefore is fortunately easy to read.

## ****REST API Interview Questions |****

## API Testing Interview Questions For Experienced

### ****What is Rest API testing and why it is used?****

In Rest API testing, we test Restful APIs for web applications. It is used because Restful web services used less bandwidth. It also supports different file format like XML, JSON etc.

### ****What is a resource in REST?****

Every content in REST consider as a resource like text file, HTML pages, Images, video etc.

We can access each resources using URI (Uniform resources identifier).

### ****What is the way to represent a resource in REST?****

**REST**uses a different way to represent the resources like text, XML and JSON.

### ****Which protocol is used for RESTful web services?****

RESTful web services use the HTTP protocol. It is used for communication between client and server.

### ****What is messaging in RESTful web services?****

For the communication, the client sends the request in the form of HTTP request and server send a response in the form of the HTTP response. This process of sending the request and receiving the response is called massaging.

### ****What are the methods used in REST?****

There are following methods used in REST-based architecture.

**GET** – By using this method, we can only access the resources. We can’t modify any resources.

**POST** – To create new resources on the server.

**DELETE** – By using this method, we can delete the resources which are identified by a URI.

**PUT** – By using this method we can update the resources (If resources do not exist then API decide to create a new resource or not)

**PATCH**– It is similar to the POST and PUT. The difference is that it is used to a partial modification to the resource.

**HEAD**– It is similar to the GET, but except without the response body. In other words, if GET returns the list of users in response body whereas HEAD can make the same request but wouldn’t return a list of users.

**OPTIONS**– It returns data describing what other methods supports at given URL.

### ****Can we use GET method to create new resource instead of PUT?****

GET has only read-only access. It can only use to get the data from the server.

PUT is used to modification of the resources on the server.

### ****What is addressing in RESTful web services?****

Addressing is used to locating a resource or multiple resources lying on the server.

Using URI (Uniform Resource Identifier), it locates the resources on the server.

**Format of URI:**

|  |  |
| --- | --- |
| 1 | <protocol>://<service\_name>/<ResourceType>/<ResourceID> |

### ****What is RESTful Statelessness?****

Statelessness means state of any client request will not store on the server.

RESTful web services should not keep the client state on the server. That is why RESTful is a statelessness web service.

### ****What is RESTful web services Caching?****

Caching means storing the accessed data into several places. RESTful web services use caching to storing the server response in the client itself. Also, it improves the performance of RESTful.

It also reduces network bandwidth requirements.

### ****What is the Security in RESTful web services?****

Some important points are given below for designing security of RESTful web services.

1. Use always HTTPS protocol to encrypt the communication data
2. Never use sensitive information into URL
3. Always validate input data against SQL or NoSQL injection attacks
4. Encryption level always should be high.
5. Session-based authentication required for the user.
6. Restriction on method (GET, POST, DELETE etc) execution.

### ****What are the main components of the HTTP request?****

An HTTP request contains different key elements:

1. HTTP methods like GET, PUT, POST, DELETE.

2. Uniform Resource Identifier (URI), which is the identifier for the different resources on the server.

3. HTTP Version, which indicates HTTP version, for example-HTTP v1.1.

4. Request Header, which contains metadata (as key-value pairs) for the HTTP Request message. Metadata can 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.

### ****What is the payload in RESTFul Web services?****

“Payload” is the data you are interested in transporting. This is differentiated from the content that wraps the data for transport like the HTTP/S Request/Response headers, authentication, etc.

### ****What are the different status codes and their description?****

The Rest API responds of each request with an HTTP response code. Below I have mentioned the response codes and their description.

|  |  |
| --- | --- |
| **Response Codes** | **Description** |
| 200 Ok | Request accepted. |
| 201 Created | This response code is returned from PUT or POST, and  indicates that a new resource was created. |
| 204 No Content | Successful deletion of a queue. |
| 400 Bad Request | Given URL is not in correct format. |
| 403 Forbbiden | The requester is not authorized to invoke the request. |
| 404 Not Found | The object is not exist on given path. |
| 405 Method Not Allowed | Requested HTTP method does not supported. |
| 409 Conflict | An attempt  made to create an object that already exists. |
| 500 Internal Server Error | An internal error occurred in the server. This might indicate a problem in the server side code. |

These are the Response codes and their error message which we are receiving during sending HTTP request to the server.

### ****Which is better REST and SOAP?****

REST is better than SOAP. It allows different file format (XML, JSON) to process request whereas SOAP only allow XML.

### ****Can SOAP use JSON?****

No, it cannot use JSON.

### ****Is SOAP stateful or stateless?****

SOAP is a **stateless** web service. We can make it stateful by changing code on server.

### ****Is HTTP stateful or stateless?****

HTTP is a stateless protocol layered on top of TCP.

**[API automation testing interview questions]**

### ****What is Parasoft SOA****?

Parasoft SOAtest is an automated API testing tool or tool for Service Oriented Architecture (SOA) that allows developers and testers to perform functional testing, end-to-end testing, security testing, performance testing etc. It also supports 100 number of protocols such as Web services, JSON, MQ, JMS, HTTP, XML and EDI etc.

**Is Parasoft Open Source tool?**

This tool offers free supports open source development community with free access to entire Parasoft suite.

**What is SOA (Service Oriented Architecture)?**

In SOA, a number of services communicate to each other. It is a method of integrating the business applications and processes it together to meet the business needs.

In SOA, Developers either develop or buy the services to use it into SOA.

## POSTMAN API Testing Interview Questions

Postman is also used for manual API testing. Here we will see few Questions on POSTMAN.

**What is POSTMAN?**

POSTMAN is a platform for API development that started as Chrome extension but now it is available in native app also. By using this we can perform API testing using different HTTP methods like Get, Post, Put and Delete.

Using this we can  send HTTP requests to the server and receive responses.

**[Download Postman](https://www.postman.com/downloads/" \t "_blank)**

**Is POSTMAN good for API testing?**

POSTMAN is easy to use. Any non-technical person can use this. It provides the number of API calls for testing APIs.

Give URL, set header, parameters and body of the API and you are ready to perform API testing.

**How Can I check API in POSTMAN?**

Firstly we need to setup the authorization in POSTMAN with the help of below steps.

1. Create a new OAuth client id with the Code Authorization and/or Client Credentials grant type and a callback url of https://www.getpostman.com/oauth2/callback for Code Authorization.   
   2. In Postman, select any API method.  
   3. Click on the Authorization tab.  
   4. Choose OAuth 2.0 and add all details like Authorization URL, Access token, Client ID, Client secret, grant type etc.  
   5. Click on Get access token button.  
   6. Postman starts the authentication flow and prompts you to save the access token.  
   7. Select Add token to header.  
   8. Click the name of your token so Postman will add the token to the authorization header and click Send to make your request.  
   9. If authentication is successful, the API shows a 200/OK response.

**What should be verified in API testing?**

The most important thing to test the HTTP response status code. Verify status is 200 or not to decide test case is passed or failed. After this you can verify response data, response time etc.

**What is the best tool for API testing?**

There are so many tools are available in the market but few are best and popular in the market like-

* SOAP UI
* Postman
* Katalon Studio
* Rest Assured

**Which type of encoding does POSTMAN accept in authorization credentials?**

It accepts base64 encoding only. This is already inbuild in Postman.

**What is Collection in POSTMAN?**

A collection in Postman is like a folder in your system. It groups individual request together. It organize similar request into a folder.

## ****Capgemini API testing interview questions****

**What are the HTTP methods?**

There are different http methods which are mentioned below.

* GET
* POST
* PUT
* PATCH
* DELETE

**What is the difference between PUT and PATCH?**

Both are work same but there are minor differences.

**PUT:** This method is used to update the resource on the server. In this method we need to send the entire resource on the server.

**PATCH:** This method is also used to update the resources on the server but in this method, we need to send only those data that you need to update.

**What is difference between Path parameter and Query Parameter?**

**Path Parameter:**Path parameter is used to identify the specific resources. It is placed before question mark in URL.

**Query Parameter:** Query parameter is used to query or filter the resources based on some conditions. It is placed after the question mark in URL.

## ****SOAP API testing interview questions****

**What are the SOAP UI assertions?**

Assertions are used to validate the test result. There are different types of assertions are used in SOAP UI.

1. Property Content
2. Script Assertion
3. Compliance Status
4. JMS
5. SLA etc.

# GCReddy: <https://www.gcreddy.com/2022/01/api-testing-interview-questions.html>

##### **1. What are Web Services?**

Web services are web components that transfer data between client and server. The client sends a web request to the server and the server then responds to the client. The response and request are related and different requests evoke the corresponding response.

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

REpresentational State Transfer (REST) is an architectural style that defines a set of constraints to be used for creating web services. REST API is a way of accessing web services in a simple and flexible way without having any processing.

##### **3. What is WSDL?**

WSDL stands for Web Service Description Language and is a document written in XML. It uses XML to define the service layer document which consists of the origin of the web service, headers, port types, and request and response data. This one can provide information about web methods and web services.

**It describes:**  
Origin of the web service  
Header information  
Port type  
Input and output messages

##### **4. What is the primary challenge when testing Web Services?**

The majority of the functional testing is carried out via the GUI; the biggest challenge of Web Services is that they do not have a UI.

##### **5. What are the communication channels available for Web Services?**

In general, web service is combined with the following protocols. They are, HTTP / POST  
HTTP / GET SOAP

While exposing the web services, these channels will be used for communication with the clients. Here HTTP / POST protocol transfers the information between the clients with secure mode. HTTP / GET protocol allows the clients to view transferred data partially at the browser’s address bar. SOAP is used for transferring confidential data safely.

##### **6. What are the different components that can be used in Web Services?**

There are four components are used in web services. They are,

WSDL – Web Service Description Language  
SOAP – Simple Access Object Protocol  
UDDI – Universal Description, Discovery and Integration RDF – Resource Description Framework  
XML – eXtensible Markup Language

##### **7. What are the tools used for testing Web Services?**

To perform functional testing for web services, we can use the following tools. SoapUI  
RESTClient – This is a Firefox plug-in

JMeter – Specially made it for performance testing tool and also we can do functional testing the web services.

##### **8. What is the role of the WSDL document in Web Service testing?**

Validating web services is only possible with WSDL document because to configure web services in SoapUI, WSDL document is mandatory. If the WSDL document is not valid, SoapUI will throw an exception immediately.

##### **9. What is UDDI?**

Universal Description, Discovery, and Integration- a directory or global repository where all the web services can be found. A new Webservice can also be registered through this. This is also the place where WSDL detailed definitions are found.

##### **10. What is SOAP?**

Simple Object Access protocol that uses XML to interact with web applications. It uses XML-based content to communicate between two client machines across any network.

##### **11. What would be the message format of the SOAP protocol?**

Generally, all the SOAP-based web services are written by using XML language which uses standard message format that is accepted across the universe. In this format, it is easy to read, identify the errors, avoids interoperability problems etc.

##### **12. What are the advantages of SOAP?**

Since its XML based, it is platform and programming language independent. RPC (Remote procedure calls) are sometimes blocked by firewalls and proxy servers- Soap overcomes that.

##### **13. What does a SOAP document contain?**

Envelope element is the top most tag which identifies the XML document as a SOAP message. Followed by Envelope element, you see the header element that has header information. The Body element specifies the call and response information. Finally, you have a Fault element which contains errors and status information.

##### **14. What is meant by Protocols and what are the major types are used in**[Web Services](https://youtu.be/g2z-kkScn2c)**?**

A protocol is a set of standard rules that helps to communicate the hardware devices through the software applications. There are different types of protocols used in the Internet and Intranet applications.  
They are:

TCP which stands for Transmission Control Protocol. It has the rules to exchange the messages between two different Internet applications.

Internet Protocol uses the rules for sending and receiving the information between two different Internet addresses.

Similarly, HTTP, FTP and DHCP protocols are used the set of rules to transfer the data other than Internet applications.

##### **15. What is XML?**

XML (eXtensible Markup Language) is a mark-up language that is used for storing, sharing and formatting data. In general, an XML document is built by the tags. For more info and examples

##### **16. What is SoapUI?**

SOAPUI is the leading open source cross-platform API/Web Services Testing tool.

SOAPUI allows testers to execute automated functional, regression, compliance, and load tests on different Web API.

SOAPUI supports all the standard protocols and technologies to test all kinds of API’s.

SOAPUI interface is simple that enables both technical and non-technical users to use seamlessly.

##### **17. What we can do with the help of SoapUI?**

SoapUI offers us to perform automation testing which includes functional testing, load testing and data driven testing.  
It also provides in build reporting tool and export test results log provision We assert our services using various types of assertions

##### **17. What is Mocking?**

Mock services are a great way of getting testing early into the picture of a service- oriented project. Once the WSDL of the web service is ready, you can simulate the service implementation and start testing the consumer applications.

##### **18. What is Property Transfer Step in SoapUI?**

Property Transfer enables to transfer the values from one API Response to other API Request

##### **19. How many kind of Assertions we have in SoapUI Tool?**

Xpath Match , Contains, Script Assertion, Xquery Match,Http Status codes etc

##### **20. What hierarchy SoapUI follows to build a proper testing project?**

In a SoapUI project, the following order should be maintained.

TestSuite – This is combination of functional tests and logical blocks

Testcase – Its a group that contains several test steps for the specific aspects of the service.

Teststep – it contains the set of functional tests.

##### **21. What is the basic method to automate Web Services in SoapUI?**

Create a project and add the WSDL file  
Add test suites, Test cases and Test cases- in that order  
Include custom programming/validation using by adding Groovy steps  
Call external data sources if using  
Add assertions if necessary  
Then RUN.

##### **22. What are SoapUI Assertions?**

Assertions compare the parts/all of the response message to the expected outcome.

##### **23. What are the major types of assertions available in SoapUI?**

Assertions are the one of the major feature in SoapUI. It offers the following types of assertions.  
Simple contains  
Schema compliance  
Simple not contains  
Soap Faults  
Response SLA  
XPath Match  
XQuery Match  
WS security status  
Script Assertion  
WS- Addressing Request or Response Assertion

Additionally Equals assertion is introduced in SoapUI NG Pro version.

##### **24. Explain about XPath Assertion in SoapUI**

In SoapUI, XPath assertion is used for asserting the web service response value by specifying the absolute path. If the absolute path is matched with the response value, then the test case or test suite will be considered as PASS otherwise it will be notified as FAILED. We can see the results of assertion at bottom of the screen where the Assertion tab will have resultant information.

##### **25. What is Data Driven Testing?**

Data driven testing means to store our test data which includes input and expected output in an external data source called Excel / Database / XML file. Later, we need to iterate the data source using respective component. In SoapUI, Datasource and Datasource Loop test steps are used for performing data driven testing.

##### **26. What are the different types of assertions used in SoapUI?**

***The following are the different types of assertions:***

Contains & Not Contains  
XPath match  
XQuery match  
Schema compliance  
Soap Faults  
Response SLA  
WS security Status  
Script Assertion  
WS- Addressing Request or Response Assertion

##### **27. How can assertions be added in SoapUI?**

Receive a response to a request as you normally would follow the below steps:  
Create a project, add WSDL  
Add Test suite, Test case and Test steps  
Run the request  
To add assertions:  
Click on the Add Assertions at the top of log tabs.  
Configure the assertions as per the type and data required.

##### **28. What is Groovy script and where can it be used?**

Groovy is a scripting language which internally includes all the java libraries – it helps us to customize and add custom validations to SoapUI tests

##### **29. Can custom code be added to SoapUI? If yes, what can be added?**

Yes -Groovy steps and Javascript steps.

##### **30. How to group tests?**

The basic Test suite is a way for us to group tests in SoapUI. When you need a different set of tests, you just have to create a new test suite and create tests as required under it as test cases.

# RAHULSHETTY: <https://rahulshettyacademy.com/blog/index.php/2021/07/31/top-30-api-testing-interview-questions-and-answers/>

### 1. What is an API?

***API (Application Programming Interface)***is a software intermediary that enables two applications to communicate with each other. HTTP-based API is often called a Web API since they are used to access Web Applications which are deployed to Servers accessible over the Internet or network. Applications that are accessed via HTTP APIs are often called Web Services.

Mobile Applications often use Web Services and REST APIs to communicate with servers to implement their functionality. The Mobile Application processes the message returned from the Web Service and displays it to the User in the application GUI. So again, the user is unaware that HTTP requests are being made or of the format of the requests and responses.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### 2. Define  API TESTING with its types ,examples and some common advantages?

**API Testing**- It is a kind of software testing to determines and check if the developed APIs meet expectations regarding the functionality, reliability, performance, and security of the application or not

**Common protocols that are testing in API testing are** -HTTP, JMS, REST, SOAP, UDDI

**Types-**Validation Testing, Functional Testing, UI testing, Load testing, Runtime/ Error Detection, Security testing, Penetration testing, Fuzz testin, Interoperability testing

**Example of API Testing:**

**Twitter**: The API for Twitter is usually in two categories, one for accessing data and the other for interacting with the Twitter search.

**YouTube**: This API used for YouTube includes various functionalities including videos, live streaming, player, etc.

**Google Maps AP**I: These are designed mainly for mobile and desktop use with the help of a flash interface and JavaScript.

**Amazon Advertising API**: Amazon is known for their products and thus their advertising API accesses their product to discover their functionality and thus advertise accordingly

**ADVANTAGES of API Testing:**

**Compatibility and easy integration with GUI**: Simple integration would allow new user accounts to be created within the application before a GUI test started.

**Language-Independent**: In API testing, data is exchanged using XML or JSON. These transfer modes are completely language-independent that allows users to select any coding language while adopting automation testing services for the project.

**Time Effective**: In comparison to functional GUI testing API testing usually is less time-consuming. The web elements in GUI testing must be polled, which makes the testing process slower. Particularly, API test automation requires less code so it can provide better and faster test coverage compared to GUI test automation.

**Test for Core Functionality**: API testing provides the ability to access 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 helps in detecting the minor issues which can become bigger during the GUI testing

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### 3. What are the main differences between API and Web Service?

**Main differences are given below:**  
a) All web services are APIs but not all APIs are web services.

b) 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.

c) A web service always needs a network to operate while APIs don’t need a network for operation.

d) Web services might not contain all the specifications and cannot perform all the tasks that APIs would perform.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### 4. What are the differences between API Testing and UI Testing?

API enables the communication between two separate software systems. A software system implementing an API contains functions or subroutines that can be executed by another software system.

On the other hand, UI ( User Interface) testing refers to testing graphical interface such as how users interact with the applications, testing application elements like fonts, images, layouts, etc. UI testing basically focuses on the look and feel of an application.  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### 5.  What are the principles of an API test design?

Basically, there are five most important principles of an API test design:

**Setup**: Create objects, start services, initialize data, etc

**Execution:** Steps to apply API or the scenario, including logging

**Verification**: Oracles to evaluate the result of the execution

**Reporting**: Pass, failed or blocked

**Clean up**: Pre-test state

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### 6.  What is the procedure to perform API testing?

a) Create the suite to add the API test case

b) Create the test development mode

c) Demand the development of test cases for the required API methods

d) Configure the control parameters of the application and then test conditions

e) Configure method validation

f) Arrange all API test cases

g) Execute the API test

h) Check test reports

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### 7. What must be checked when performing API testing?

a) Accuracy of data

b) Non-functional testing like performance and security testing

c) Implementation of response timeout

d) Schema validation

f) HTTP status codes

g) Data type, validations, order, and completeness

h) Authorization checks

I) Error codes in case API returns

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

A few common tools are Katalon Studio, Postman, SoapUi Pro, Apigee, automated API testing using REST Assured and CURL etc.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### 9. What are the types of Bugs API testing can find?

a) Missing or duplicate functionality

b) Stress

c) Reliability

d) Security

e) Performance

f) Unused flags

g) Not implemented errors

h) Inconsistent error handling

I) Improper errors

j) Fails to handle error conditions gracefully

k) Multi-threading issues

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### 10. Difference API and Unit Testing?

API is owned by QA team where as  Unit testing is owned by the development team

API is mostly black box testing where as Unit testing is white box testing

Full functionality of the system is considered in API testing as it will be used by the end-user (external developers who will use your API )where as Unit testing is done to verify whether each unit in isolation performs as expected or not

API test are often run after the build is ready and authors do not have access to the source code where as For each of their module, the developers are expected to build unit tests for each of their code modules and have to ensure that each module pass unit test before the code is included in a build.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### 11. What are the main challenges of API testing?

1) Parameter Selection

2) Parameter Combination

3) Call sequencing

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### 12. 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.

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### 13. Explain what is an HTTP request?

Hypertext Transfer Protocol is a way of sending messages to software on another computer over the internet or over a network.

**An HTTP request is sent to a specific URL and consists of:**

A) A VERB specifying the type of request e.g. GET, POST, PUT, DELETE

B) A set of HTTP Headers. The headers specify information such as the type of browser, type of content in the message, and what type of response is accepted in return.

C) A body, or payload in the request, representing the information sent to, or from, the Web Application. Not all HTTP messages can have payloads: POST and PUT can have payloads, GET and DELETE do not.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### 14. What is a URL?

URL is a Uniform Resource Locator and is the address we use to access websites and web applications. When working with APIs you will often see this referred to as a URI (Uniform Resource Identifier). We can think of a URI as the generic name for a URL. When we have to call an HTTP API we need the URL for the endpoint we want to call Example-

<https://www.thetravel.com/10-best-indian-foods/>

This is the locator that says “I want to call the “10 best indian foods” resource located at “thetravel.com” using the http protocol”.

**The above URL can be broken down into the form:**

**scheme://host/resource**

• **scheme**-HTTP

• **host**-thetravel.com

• **resource**-10-best-indian-foods

**A larger form for a URL is:**

scheme://host/resource?query#fragment

The query is a way of passing parameters in the URL to the endpoint e.g. Google uses query parameters to define the search term and the page:

<https://www.google.co.uk/?q=test&start=10#q=test>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### 15. Describe what are HTTP verbs?

A web browser will usually make **GET requests** and **POST requests.**

**GET**requests to ask to read information from the server e.g. clicking on a link.

**POST**requests supply information to the server e.g. submitting a form.

**GET**requests do not have a body, and just consist of the Verb, URL, and the Headers.

**POST**requests can have a payload body.

***When working with a Web Application or HTTP API the typical HTTP Verbs used are:***

**GET**, to read information.

**POST**, to create information.

**PUT**, to amend or create information.

**DELETE**, to delete information.

# ACTE: <https://www.acte.in/api-testing-interview-questions-and-answers>

**1. What is an API?**

**Ans:**

* API (Application Programming Interface) is a software intermediary that enables two applications to communicate with each other. It comprises a number of subroutine definitions, logs, and tools for creating application software.
* In an API testing interview, you could be asked to give some API examples, here are the well-known ones: Google Maps API, Amazon Advertising API, Twitter API, YouTube API, etc.

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

**Ans:**

* 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 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?**

**Ans:**

 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.

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

**Ans:**

 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?**

**Ans:**

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.

**6. What is API Testing?**

**Ans:**

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?**

**Ans:**

 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?**

**Ans:**

 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?**

**Ans:**

* 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?**

**Ans:**

**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?**

**Ans:**

 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?**

**Ans:**

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?**

**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. 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?**

**Ans:**

**The following factors should be considered when performing API testing:**

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

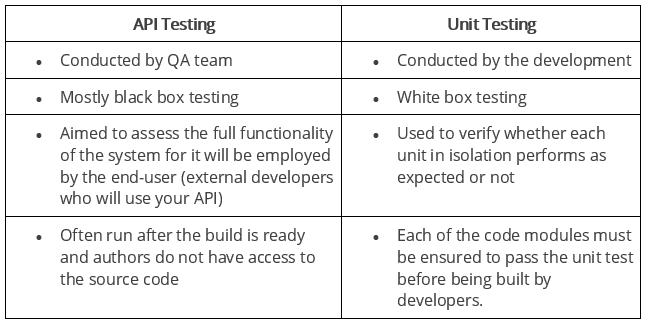
**15. What tools could be used for API testing?**

**Ans:**

There are a myriad of different 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.

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

**Ans:**



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

**Ans:**

* 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?**

**Ans:**

**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?**

**Ans:**

**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?**

**Ans:**

**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.

Top of Form

##### Subscribe For Free Demo



SUBSCRIBE

Bottom of Form

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

**Ans:**

Not only API fundamental questions, the interviewer also determines 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?**

**Ans:**

* 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

**23. What is API documentation?**

**Ans:**

 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?**

**Ans:**

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 documents, what must be considered?**

**Ans:**

* 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?**

**Ans:**

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.

**27. What is REST?**

**Ans:**

* 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 a 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?**

**Ans:**

**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?**

**Ans:**

* 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?**

**Ans:**

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



### Get Best API Testing Training from Top-Rated Software Experts

* *Instructor-led Sessions*

* *Real-life Case Studies*
* *Assignments*

[**Explore Curriculum**](https://www.acte.in/rest-api-automation-testing-rest-assured)

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

**Ans:**

 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?**

**Ans:**

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?**

**Ans:**

 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?**

**Ans:**

**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?**

**Ans:**

* 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?**

**Ans:**

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?**

**Ans:**

* 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 the 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 a 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?**

**Ans:**

 The OPTIONS Method lists down all the operations 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?**

**Ans:**

* 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?**

**Ans:**

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?**

**Ans:**

* <GET> appends data to the service URL. But, its size shouldn’t exceed the maximum URL length. However, <POST> doesn’t have any such limit.
* So, theoretically, a user can pass unlimited data as the payload to POST method. But, if we consider a real use case, then sending POST with a large payload will consume more bandwidth. It’ll take more time and present performance challenges to your server. Hence, a user should take action accordingly.

**42. What is the caching mechanism?**

**Ans:**

* 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.

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

**Ans:**

This is one of the fundamental Web services testing questions that you must know the answer to. 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?**

**Ans:**

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 the user interface using HTTP. The client can use any approach to send the XML, like the SMTP server or POP3 protocol to pass the messages or reply to queries.

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

**Ans:**

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.



### Learn Hands-On Practical API Testing Course to Enhance Your Career

Weekday / Weekend Batches[**See Batch Details**](https://www.acte.in/rest-api-automation-testing-rest-assured)

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

**Ans:**

* 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 a text character string.
* GetAllListing(): It is used to return the full list in an XML format.

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

**Ans:**

 When using SOAP, users often see the firewall security mechanism as the biggest obstacle. This blocks 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?**

**Ans:**

**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?**

**Ans:**

* 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?**

**Ans:**

* 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?**

**Ans:**

* 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?**

**Ans:**

* 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?**

**Ans:**

 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 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?**

**Ans:**

* **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?**

**Ans:**

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.

**API Sample Resumes! Download & Edit, Get Noticed by Top Employers!**[DOWNLOAD](https://www.acte.in/api-testing-sample-resumes)

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

**Ans:**

* Top down SOAP Web services include creating WSDL documents 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, the bottom-up approach is easy to implement and client codes must wait for WSDL from the server side to start working.

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

**Ans:**

* SOAP is both platform and language independent.
* SOAP separates the encoding and communications protocol from the runtime environment.
* Web services can retrieve or receive 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.

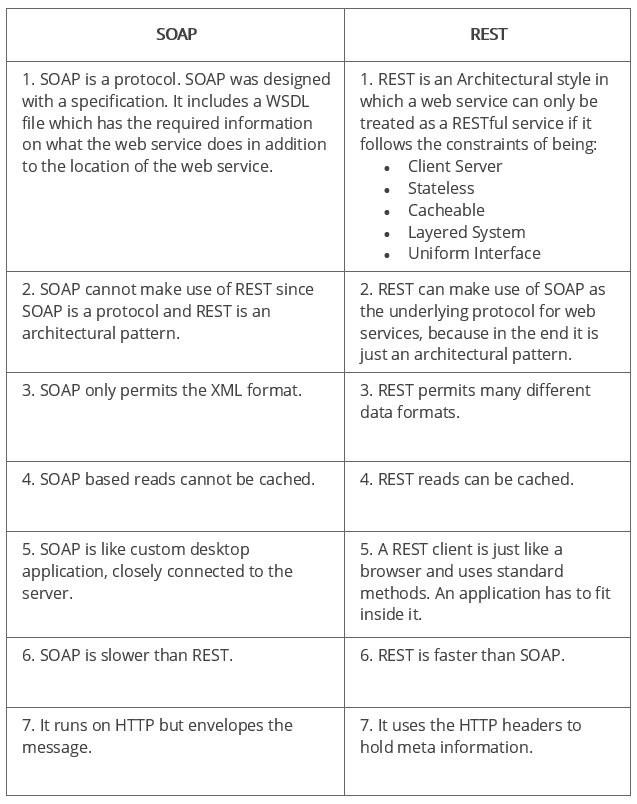
**58. What are the disadvantages of SOAP?**

**Ans:**

* SOAP is typically significantly slower than other types of 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 and PHP is not as powerful as it is in Java and .NET

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

**Ans:**



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

**Ans:**

1. **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 a 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.

1. **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?**

**Ans:**

* 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.