API testing is a type of software testing that focuses on verifying the functionality, reliability, performance, and security of Application Programming Interfaces (APIs). An API is basically a set of protocols and tools that allows different applications to communicate with each other.

API testing has been considered the future of software testing thanks to its advantages in the ability to test for core functionality, time effectiveness, language independence, and GUI integration. It becomes a must for software testing projects to ensure product quality.

In recent years, many testers will definitely need to face interview questions when applying for a QA job. The web API testing interview questions below have been collected from the test professionals to help you get ready for a new role. More than just an API interview, this list of content will also benefit both freshers and senior testers who would like to achieve both general and advanced knowledge in web API testing.

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

**1. What is an API?**

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

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

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

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

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

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

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

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

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

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

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

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

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

**6. What is API Testing?**

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

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

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

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

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

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

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

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

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

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

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

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

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

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

1. Validation Testing
2. [Functional Testing](https://katalon.com/resources-center/blog/functional-testing)
3. [UI testing](https://katalon.com/visual-testing)
4. Load testing
5. Runtime/ Error Detection
6. Security testing
7. Penetration testing
8. Fuzz testing
9. Interoperability and WS Compliance testing



[[Text

Description automatically generated](https://katalon.com/open-testops)](https://katalon.com/open-testops)

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

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

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

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

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

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

The following factors should be considered when performing API testing:

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

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

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

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

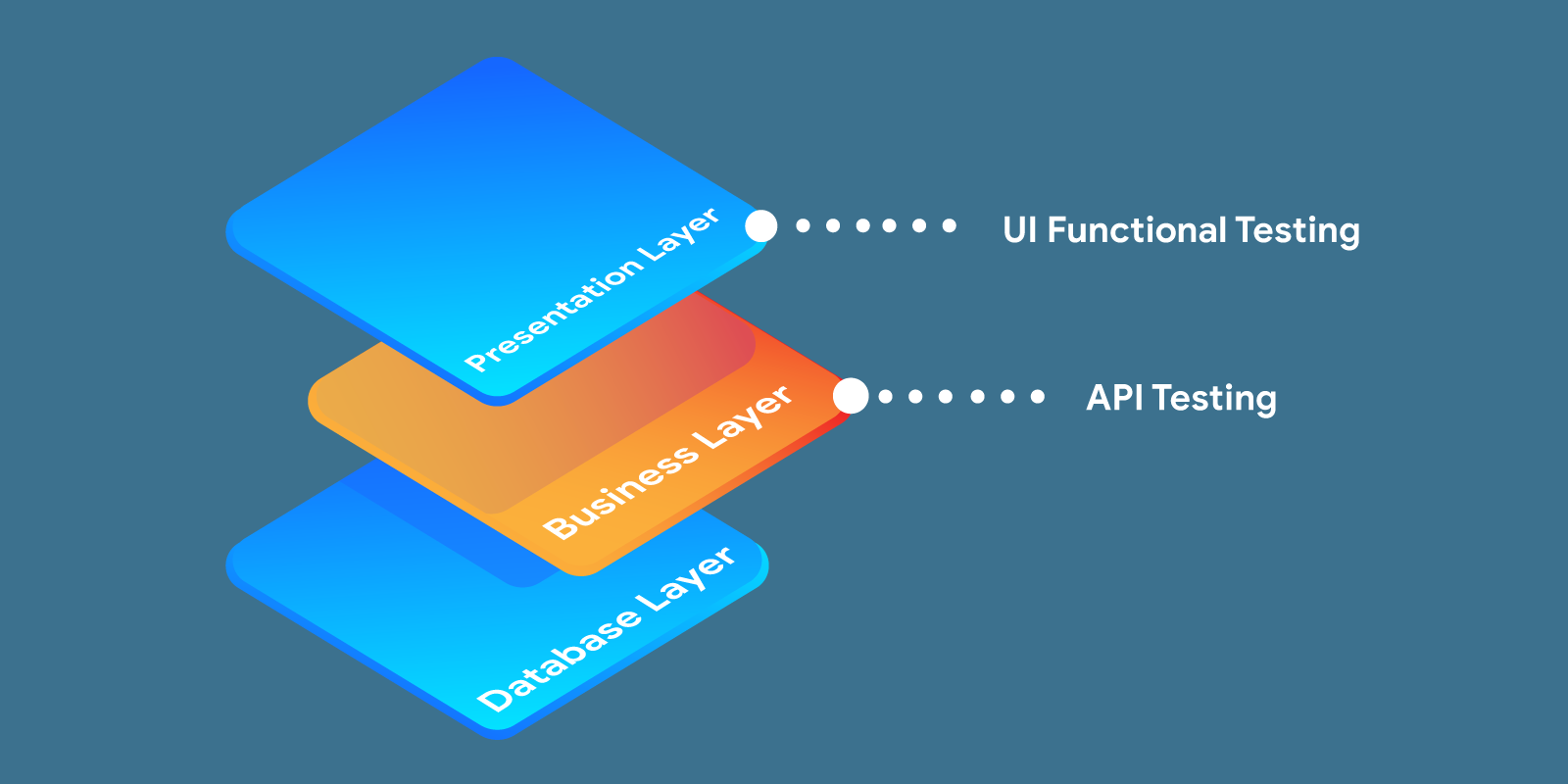
[[Text

Description automatically generated](https://katalon.com/open-testops)](https://katalon.com/open-testops)

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

|  |  |
| --- | --- |
| **API Testing** | **Unit Testing** |
| Conducted by QA Team | Conducted by the development team |
| Mostly black-box testing | White box testing |
| Aimed to assess the full functionality of the system for it will be employed by the end-user (external developers who will use your API) | Used to verify whether each unit in isolation performs as expected or not |
| Often run after the build is ready and authors do not have access to the source code | Each of the code modules must be ensured to pass the unit test before being built by developers |

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



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

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

Background pattern

Description automatically generated with low confidence

If you can overcome the [challenges in API Testing](https://docs.katalon.com/docs/create-tests/introduction-to-test-creation/introduction-to-api-testing-in-katalon-studio#challenges-in-api-testing), you can be confident in the API testing interview too. They are:

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

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

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

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

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

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

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

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

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

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

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

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

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

**23. What is API documentation?**

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

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

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

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

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

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

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

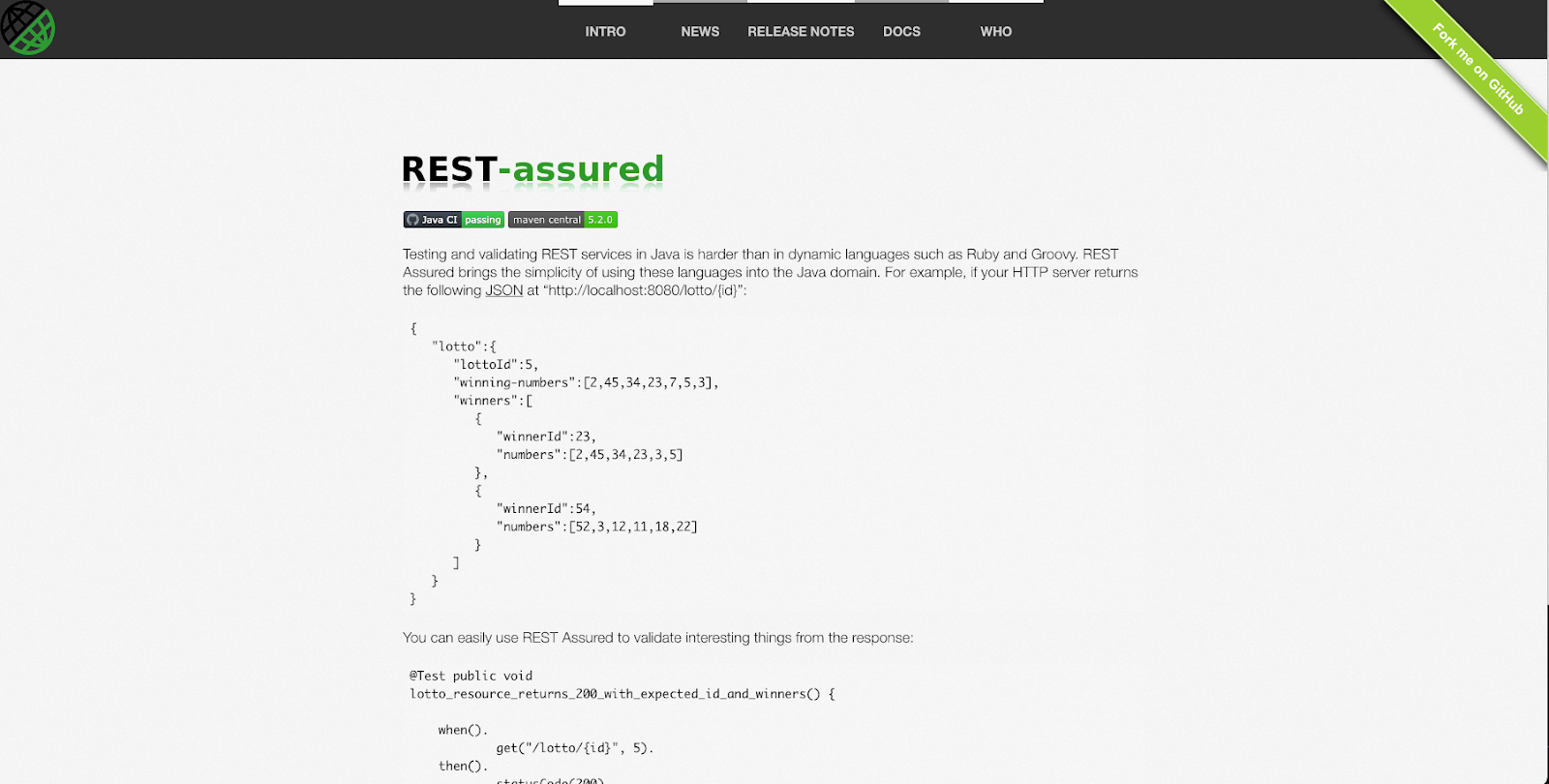
APIs, especially modern RESTful APIs, is a nice creation that can certainly simplify and accelerate integration efforts, which makes it more likely you will benefit from them. But APIs can and do change for various reasons, sometimes abruptly, and hence REST APIs do not differ from traditional integration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

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

**27. What is REST?**

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

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



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

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

**29. What is a “Resource” in REST?**

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

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

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

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

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

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

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

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

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

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

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

An HTTP request contains five key elements:

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

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

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

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

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

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

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

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

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

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

**Final Thoughts**

The aforementioned API Testing interview questions are quite common in interviews for software QA engineers and testers positions. To best prepare for your upcoming interview, make sure to read thoroughly those questions, and try to understand the reasoning and knowledge behind them instead of simply memorizing.