**[Top 30 API Testing Interview Questions and Answers – Part 1](https://rahulshettyacademy.com/blog/index.php/2021/07/31/top-30-api-testing-interview-questions-and-answers/" \o "Top 30 API Testing Interview Questions and Answers – Part 1)**

1. **What is an API?**

API stands for Application Programming Interface. API acts as an interface between two software applications and allows the two software applications to communicate with each other. API is a collection of software functions that can be executed by another software program.

1. **How does an API work?**

An API works as an intermediary that allows two applications to communicate with each other. It acts as a bridge that defines the methods and data formats necessary for interaction, ensuring consistency and efficiency. When an API request is made, the client sends specific parameters or data to the server through the API endpoint. The server processes this request, performs the necessary actions, and responds with the requested data or an appropriate status message.

**For example,** consider a weather application on your smartphone. When you input your location to check the weather forecast, the application sends a request to a weather API with your location data. The API processes this request by querying the backend database or external data sources and then returns structured weather information, such as temperature, humidity, and forecasts, in a specified format like JSON. This seamless process allows the weather application to display the requested data to you without requiring you to interact with the underlying systems directly. APIs, therefore, enable streamlined communication between various software systems, enhancing productivity and integration.

1. **What are the types of APIs?**

When discussing the types of APIs, it’s essential to understand the various categories that serve different purposes in software development. Here are some of the main types of APIs:

Web APIs: These are interfaces that allow interaction between web servers and clients. They commonly use HTTP protocols to facilitate communication, enabling applications to access and exchange data over the internet. Examples include RESTful APIs and GraphQL.

Operating System APIs: These APIs provide functionalities for interacting with the operating system resources. They allow software applications to perform tasks such as managing files, handling memory, and managing processes. Examples include the Windows API and POSIX.

Library APIs: These consist of a set of routines, protocols, and tools for building software and applications. Library APIs allow developers to use predefined functions and processes without having to reinvent the wheel, thus speeding up the development process. Examples include the C Standard Library and Java’s JavaFX API.

Hardware APIs: These interfaces allow software to communicate directly with hardware components. They provide the necessary commands and controls to interact with physical devices, like printers or graphics cards. Examples include the JavaScript Web API for accessing device capabilities and the Android Open Accessory Protocol.

Each type of API plays a crucial role in enabling developers to interact with different layers of technology, facilitating a smoother integration and development experience.

1. **What are the different ways to authenticate an API?**

There are several methods to authenticate an API to ensure secure communication and access control:

1. **API Key Authentication** – This involves providing a unique key, generated by the API provider, which the client includes in the request header or URL. API keys help identify the client but should be used over secure protocols like HTTPS to prevent interception.
2. **Basic Authentication** – This is a simple method where a username and password are encoded in base64 format and included in the request header. It is straightforward but less secure unless used alongside HTTPS.
3. **OAuth (OAuth 1.0, 2.0)** – OAuth is a robust framework that allows third-party authentication without passing sensitive credentials. OAuth 2.0 is particularly popular, offering tokens for secure communication between server and client.
4. **Bearer Token Authentication** – Commonly used with OAuth 2.0, it involves sending a token in an HTTP Authorization header. The bearer token grants access to specific resources based on the token’s scope and validity.
5. **JWT (JSON Web Tokens)** – These tokens are compact, JSON-based, and include claims about the user or client. JWTs are signed and optionally encrypted, providing secure, stateless authentication.
6. **Digest Authentication** – Unlike basic authentication, digest authentication uses a hashing technique to secure credentials, making it more secure over non-encrypted channels.
7. **Mutual TLS (mTLS)** – This method uses client certificates in addition to server certificates during the TLS handshake to verify both parties before proceeding with communication.

Each method has its own use cases and security considerations, making it essential to choose the one best suited for your application’s needs.

1. **What is the role of headers in API testing?**

Headers play a crucial role in API testing as they convey critical information between the client and server. They include metadata such as authentication tokens, content type, and caching directives, ensuring proper communication and data handling. Proper validation of headers is essential to ensure API functionality, security, and compliance with standards.

1. **Can you discuss the different API protocols?**

There are several API protocols that facilitate communication between systems, each with its own use cases and characteristics.

1. **REST (Representational State Transfer)** – REST is one of the most widely used protocols for building APIs. It leverages standard HTTP methods like GET, POST, PUT, and DELETE and works with a stateless communication model. REST APIs are highly scalable, flexible, and easy to implement, making them a popular choice.
2. **SOAP (Simple Object Access Protocol)** – SOAP is a protocol that relies on XML-based messaging. It is designed for exchanging structured information in web services, emphasizing security and reliability. SOAP is often used in enterprise applications where transactional reliability is crucial.
3. **GraphQL** – GraphQL is a query language and runtime for APIs, allowing clients to request exactly the data they need. It offers greater efficiency by reducing over-fetching or under-fetching of data and is favored in applications requiring high flexibility in data fetching.
4. **gRPC (gRPC Remote Procedure Call)** – gRPC is an open-source protocol developed by Google. It uses HTTP/2 for communication, supports multiple programming languages, and is highly performant. It is particularly suited for microservices architectures.
5. **WebSocket APIs** – WebSocket is a protocol for two-way communication between a client and server over a single, long-lived connection. WebSocket APIs are commonly used in real-time applications like chat apps, live updates, and multiplayer gaming.

Each protocol has its strengths and ideal scenarios for implementation, giving developers the flexibility to choose the one that best fits their project’s needs.

1. **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](https://www.softwaretestingmaterial.com/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.

1. **What is the difference between manual API testing and automated API testing?**
2. **What are the common API Testing Types?**
3. **What are the most important best practices for API testing?**

**Top 25 API Testing Interview Questions – Rahul Shetty**

1. **What all challenges are included under API testing?**

API documentation

Access to DB

Authorization overhead

1. **What is difference between PUT and POST method?**

POST request: create a new object on the server

PUT request: Update the object in server with new value.

1. **What are commonly used HTTP methods?**

**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 resources in a server.

**DELETE**: It lets you delete data from a server.

1. **List out few Authentications technique used in API’s?**

Session/Cookie based authentication

Basic authentication

Digest authentication

OAuth

1. **Why is API testing determined as the most suitable form of automation testing?**
2. **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.

1. **What exactly needs to verify in API testing?**
2. We will verify the accuracy of the data.
3. Will see the HTTP status code.
4. We will see the response time.
5. Error code in case API returns any errors.
6. Authorization would be check.
7. Non-functional testing such as performance and security testing.
8. **What are path parameters and query parameters for below API request URL?**

http:/rahulshettyacademy.com/orders/112234?location=IND

1. **What are the core components of an HTTP request?**
2. HTTP request methods like: GET, POST, PUT, DELETE
3. Base uniform resource identifier (URI)
4. Resources and Parameters
5. Request Headers, which caries metadata (as key-value pairs) for the HTTP request message.
6. Request Body, which indicates the message content or resource representation.
7. **What could be the HTTP method for below API scenario? Answer it is GET or POST**

**Scenario**: An API which has endpoint, parameters, headers, cookies, Payload

1. **What is the difference between API and UI testing?**

**UI**: (User Interface) testing means the testing 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 how users can interact with app elements such as images, fonts, layout etc. are checked.

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

1. **What protocol is used by the RESTFUL Web Services?**

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

1. **What is Soap Web Servies?**

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

1. **How do we represent a resource in REST?**

Using HTTP methods.

1. **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 should be used to create a resource.

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.

1. **Can you use POST request instead of PUT to create a resource?**

Yes, we can. Because POSY is super set of other HTTP method expect GET.

1. **What do you understand by payload in Restful Web Services?**

Payload/body is the secured input data which is sent to APIs to process the request. Payload is generally represented in json format in REST API’s.

1. **How do we differentiate path and query parameters in API request endpoint?**
2. **What is Rest Assured?**

It is Java library which can automate Rest API’s.

1. **How would we define API details in REST Assured Automation?**

We shall define all the request details and send it to server in GIVEN, WHEN THEN methods.

1. **What is JSON serialization and deserialization in REST assured?**

**Serialization** in REST assured context is a process of converting a Java object into request body (payload).

Rest assured also support deserialization by converting response body to Java object.

1. **List out few common Json parsing techniques used in Rest assured automation?**

Json Path

Deserialization of json using POJO classes.

1. **How would you send attachments to API using Rest assured test?**

Using MultiPart method

1. **Different status codes and their descriptions?**

|  |  |  |
| --- | --- | --- |
| **CODE** | **Status** | **Description** |
| 200 | OK | The request was successfully completed. |
| 201 | Created | A new resource was successfully created. |
| 400 | Bad Request | The request was invalid. |
| 401 | Unauthorized | The request did not include an authentication token or the authentication token was expired. |
| 403 | Forbidden | The client did not have permission to access the requested resource. |
| 404 | Not Found | The requested resource was not found. |
| 405 | Method Not Allowed | The HTTP method in the request was not supported by the resource. For example, the DELETE method cannot be used |
| 409 | Conflicts | The request could not be completed due to a conflict. |
| 500 | Internal Server Error | The request was not completed due to an error on the server side. |
| 501 | Service Unavailable | The server was unavailable. |

1. Get the Json Path of ‘site’ from below Json.