

API testing has been considered the future of software testing and it has advantages in the ability to test for Backend functionality, time effectiveness, language independence, and GUI integration. It becomes a must for software testing projects to ensure product quality and at the same time important part of Testing/SDET interview questions and answers. In the post below we will check API Testing Interview Questions and Answers along with real time examples.

Content will be focused on below :

- REST API testing interview Questions
- Advanced API testing Interview Questions
- Rest Assured API Testing interview Questions for experienced

WHAT IS AN API?

Imagine we are sitting at a table in a restaurant with a menu of choices to order from. The kitchen is the part of the “system” that will prepare your order. What is missing is a critical link to communicate your order to the kitchen and deliver your food back to your table. That’s where the waiter or API comes in. The waiter is the messenger – or API – that takes your request or order and tells the kitchen – the system – what to do. Then the waiter delivers the response back to you; in this case, it is the food.

Here is a real-life API example. We may be familiar with the process of searching for flights online. Just like the restaurant, you have a variety of options to choose from, including different cities, departure and return dates, and more. Let us imagine that you’re booking you are flight on an airline website.

We choose departure city and date, return city and date, cabin class, as well as other variables. In order to book your flight, you interact with the airline’s website to access their database and see if any seats are available on those dates and what the costs might be.

However, what if we are not using the airline’s website? What if we are using an online travel service, such as Kayak or Expedia, which aggregates information from a number of airline databases?

The travel service, in this case, interacts with the airline's API. The API is the interface that, like your helpful waiter, can be asked by that online travel service to get information from the airline's database to book seats, baggage options, etc. The API then takes the airline's response to your request and delivers it right back to the online travel service, which then shows you the most updated, relevant information.

VERY IMP: [BEGINNERS GUIDE FOR API TESTING](#)

Explain the major components of an API?

https://www.linkedin.com/posts/sidharth-shukla-77b53145_sidpost-api-software-activity-7089077112594182146-LpDw?utm_source=share&utm_medium=member_desktop

21. How to use Basic authentication in automation?

```
Response resp = given()
    .auth()
    .basic("sid", "sid").when().get("https://reqres.in/api/users/2");
```

22. How to use Pre-emptive authentication in automation?

```
Response resp1 = given()
    .auth()
    .preemptive().basic("sid",
"sid").when().get("https://reqres.in/api/users/2");
```

23. How to use digest authentication in automation?

```
Response resp2 = given()
    .auth()
    .digest("sid", "sid").when().get("https://reqres.in/api/users/2");
```

24. How to use OAuth2 authentication in automation?

```
Response resp3 = given()
```

```
.auth()  
.oauth2("").when().get("https://reqres.in/api/users/2");
```

25. How to use OAuth authentication in automation?

```
Response resp4 = given()  
    .auth()  
    .oauth("consumerKey", "consumerSecret", "accessToken",  
"secretToken").when().get("https://reqres.in/api/users/2");
```

26. How to use header for authorization(oauth2) in automation?

```
Response resp5 = given().header("Authorization","accessToken")  
    .when().get("https://reqres.in/api/users/2");  
  
}
```

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

- All Web services are APIs but not all APIs are Web services.
- Web service uses three styles of use: SOAP, REST and XML-RPC for communication whereas API may be exposed in multiple ways.
- Web service needs a network to operate but APIs don't need a network to operate.

Explain API Chaining with examples ?

https://www.linkedin.com/posts/sidharth-shukla-77b53145_sidpost-apitesting-testautomation-activity-7118075315326746624-tJQh?utm_source=share&utm_medium=member_desktop

30. 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, REST Server provides access to resources and client accesses and makes these resources available.

- Each resource is identified by URIs or global IDs, and REST uses multiple ways to represent a resource, such as text, JSON, and XML.

What all 4xx status code you have used in your project, can you explain some of the error codes?

https://www.linkedin.com/posts/sidharth-shukla-77b53145_testing-testautomation-automation-activity-7091624769585905664-eUKK?utm_source=share&utm_medium=member_desktop

Explain what statelessness means in REST?

Ans: Statelessness means that the client and server don't store information about each other's state. Since the server stores no information, it treats each client request as a new request.

As a consequence, the following conditions would apply:

- The client request contains all information required for the server to process the request
- Client application is responsible for storing session state

Explain different types of parameters in API, like path param, query param ?

https://www.linkedin.com/posts/sidharth-shukla-77b53145_java-automation-javavirtualmachine-activity-7080762585028956161-qS-V?utm_source=share&utm_medium=member_desktop

How to send a Nested JSON object as payload?

Ans:

```
Ex: {"space": {  
  "name": "myspace",  
  "id": "sidharth"  
}}
```

```
HashMap<String,Object> mainobj= new HashMap<String,Object>();
HashMap<String,String> subobj= new HashMap<String,String>();
subobj.put("name","QA");
subobj.put("id","sidharth");
mainobj.put("space",subobj);
```

What is JSON Schema and how to perform Schema Testing with Rest Assured?

Ans: [Click Here For Answer With Code](#)

What are the disadvantages of REST API?

Ans:

- Doesn't enforce security practices
- HTTP method limits you to synchronous requests
- Due to statelessness, you might be unable to maintain state (e.g. in sessions)

Explain Idempotency in API with examples:

https://www.linkedin.com/posts/sidharth-shukla-77b53145_testing-sidpost-apitesting-activity-7178593745385238528-ysS4?utm_source=share&utm_medium=member_desktop

What is Input injection and what are different ways to simulate user input?

Ans::

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

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

Which purpose does the OPTIONS method serve for the RESTful Web services?

Ans: The OPTIONS Method lists down all the operations of a web service supports. It creates read-only requests to the server.

1. What are the types of methods most used in RestAPI Testing?

GET/POST/PUT/PATCH/DELETE/HEAD/OPTIONS

P1 : GET POST DELETE

P2 : PUT

P3 : PATCH

2. What are the types of Status codes?

1xx informational response – the request was received, continuing process

2xx successful – the request was successfully received, understood, and accepted

3xx redirection – further action needs to be taken in order to complete the request

4xx client error – the request contains bad syntax or cannot be fulfilled

5xx server error – the server failed to fulfil an apparently valid request

3. What are the status codes you have come across in your API testing project?

<https://automationreinvented.blogspot.com/2019/03/what-are-most-used-api-status-codes.html>

4. What is GET Method?

GET Retrieve information about the REST API resource

5. What is POST Method?

POST Create a REST API resource

6. What is PUT Method?

PUT Update a REST API resource

7. What is DELETE Method?

DELETE Delete a REST API resource or related component

8. What is HEAD method?

The HEAD method asks for a response identical to that of a GET request, but without the response body.

This is useful for retrieving meta-information written in response headers, without having to transport the entire content

9. What is OPTIONS method?

The OPTIONS method returns the HTTP methods that the server supports for the specified URL.

This can be used to check the functionality of a web server by requesting '*' instead of a specific resource.

10. PUT VS POST

PUT vs POST : An Example

Let's list down when to use POST and when to use PUT :

GET /user-management/users: Get all users

POST /user-management/users: Create a new user

GET /user-management/users/{id} : Get the user information identified by "id"

PUT /user-management/users/{id} : Update the user information identified by "id"

It is good practice to use:

- PUT for UPDATE operations.
- POST for CREATE operations.
- PUT is Idempotent
- POST is not Idempotent.

10. Http vs HTTPS

In HTTP, URL begins with "http://" whereas URL starts with "https://"

1. HTTP uses port number 80 for communication and HTTPS uses 443
2. HTTP is considered to be unsecured and HTTPS is secured

3. HTTP Works at Application Layer and HTTPS works at Transport Layer
4. In HTTP, Encryption is absent and Encryption is present in HTTPS
5. HTTP does not require any certificates and HTTPS needs SSL Certificates

11. Automate GET method and validate the status code?

```
@Test(description="Verify status code for GET method-users/2 as 200")
public static void verifyStatusCodeGET() {

    Response resp = given()
        .when()
        .get("https://reqres.in/api/users/2");

    assertEquals(resp.getStatusCode(),200);

}
```

12. Automate GET method and fetch response body?

```
@Test(description="Verify status code for GET method-users/2 as 200")
```

```
public static void verifyStatusCodeGET() {
```

```
    Response resp=given().when().get("https://reqres.in/api/users/2");
```

```
    assertEquals(resp.getBody().asString(),200);
```

}

13. Automate GET method and verify value from response

body?(validate that total number pages =12)

```
@Test(description="Verify status code for GET method-users/2 as 200")
public static void verifyStatusCodeGET() {

    Response
    resp=given().when().get("https://reqres.in/api/users");

    System.out.println(resp.path("total").toString());

    assertEquals(resp.getStatusCode(),200);
    assertEquals(resp.path("total").toString(),"12");
}
```

14. How to pass query param with GET method in Rest Assured?

API Query parameters can be defined as **the optional key-value pairs that appear after the question mark in the URL**. Basically, they are extensions of the URL that are utilized to help determine specific content or action based on the data being delivered. Query parameters are appended to the end of the URL, using a '?'

```
@Test
    public void validateQueryParamInGiven() {

        Response resp = given().queryParams("page",
"2").

when().get("https://reqres.in/api/users");

        assertEquals(resp.getStatusCode(),200);

        System.out.println(resp.getBody().asString());
    }
```

15. How to pass header for GET method in Rest Assured?

```
@Test
    public void validateGivenHeader() {

        Response resp = given().header("Content-Type",
"application/json").

when().get("https://gorest.co.in/public-api/users");

        assertEquals(resp.getStatusCode(),200);
        System.out.println(resp.getBody().asString());
    }
```

16. How to automate PATCH method in rest Assured?

The HTTP PATCH method can be used when a resource needs to be updated.

This method is especially useful if a resource is large and the changes being made are small.

```
@Test(description="validate with jsonpath and json object and pass post body as json file")
    public void MethodValidationPUT() throws IOException, ParseException {

        FileInputStream file = new FileInputStream(new File
(System.getProperty("user.dir")+"\\TestData\\put.json"));

        Response resp =
            given().header("Content-Type" ,
"application/json").body(IOUtils.toString(file, "UTF-8")).
            when().patch("https://reqres.in/api/users/2");

        assertEquals(resp.getStatusCode(),200);
        assertEquals(resp.path("job"),"tester");

    }
```

17. How to automate PUT method in Rest Assured?

A PUT method puts or places a file or resource precisely at a specific URI. In case a file or a resource already exists at that URI, the PUT method replaces that file or resource. If there is no file or resource, PUT creates a new one.

```

@Test(description="validate with jsonpath and json object and pass post body as json
file")
    public void MethodValidationPUT() throws IOException, ParseException {

        FileInputStream file = new FileInputStream(new File
(System.getProperty("user.dir")+"\\TestData\\put.json"));

        Response resp =
            given().header("Content-Type" ,
"application/json").body(IOUtils.toString(file,"UTF-8")).
            when().put("https://reqres.in/api/users/2");

        assertEquals(resp.getStatusCode(),200);
        assertEquals(resp.path("job"),"tester");

    }

```

18. How to automate POST method in Rest Assured?

POST requests are **used to send data to the API server to create or update a resource**. The data sent to the server is stored in the request body of the HTTP request

```

@Test(description="validate with jsonpath and json object and pass post body as json
file")
    public void MethodValidationPOST() throws IOException, ParseException {

        FileInputStream file = new FileInputStream(new File
(System.getProperty("user.dir")+"\\TestData\\put.json"));

        Response resp =
            given().header("Content-Type" ,
"application/json").body(IOUtils.toString(file,"UTF-8")).
            when().post("https://reqres.in/api/users");

        assertEquals(resp.getStatusCode(),201);
        assertEquals(resp.path("job"),"tester");

    }

```

What is the major drawback of using SOAP?

Ans: When using SOAP, users often get 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.

When to use contract testing?

Ans: Contract testing is immediately applicable anywhere where you have two services that you need to communicate - such as an API client and a web front-end. Although a single client and a single service is a common use case, contract testing really shines in an environment with many services (as is common for a microservice architecture). Having well-formed contract tests makes it easy for developers to avoid version hell. Contract testing is a killer app for microservice development and deployment.

To learn more about the use of the pact in contract testing please refer:
[ContractTestingWithPact](#)

What are the status codes you have come across in your API testing project?

Ans: [Click Here For Status Codes](#)

How to fix the error import io.restassured.RestAssured cannot be resolved?

Do you have the scope set to test when you are adding the mentioned dependencies? This limits the code from accessing that dependency's classes within the source code. That is, we can access those classes within your **testsources** (ex:
`${project.dir}/src/test/java/<package>`,
`${project.dir}/test/<package>`).

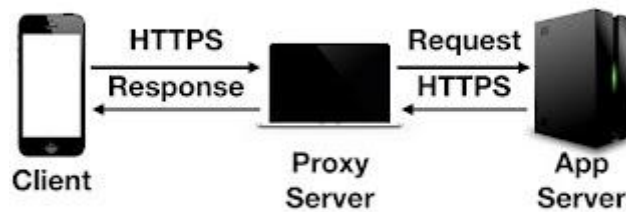
If that is not your intended use case, just remove the scope attribute.

```
<dependency>
  <groupId>io.rest-assured</groupId>
  <artifactId>rest-assured</artifactId>
  <version>4.1.1</version>
</dependency>
```

74. How to check API responses in mobile (Android/iOS) browser?

Ans: Using Fiddler or Charles Proxy. A proxy server is an intermediary for requests which travel from client to server and vice-versa.

A proxy can exist on the same machine as the client or server, it can also exist on a separate machine. This is the case for the setup we are going to use in the current context where we will have a client (mobile phone with an application we want to debug), a proxy server (our PC) and a Server (which communicates with the client).



How to validate Query Parameters in API Testing?

https://www.linkedin.com/posts/sidharth-shukla-77b53145_sidpost-apitesting-qualityassurance-activity-7105754145323679744-k0ai?utm_source=share&utm_medium=member_desktop

Scenario 1:

Question:

You are tasked with testing the authentication functionality of an API using Rest Assured. How would you approach this scenario?

Answer:

Firstly, I would ensure that I have clear documentation or understanding of the authentication mechanism used by the API, whether it's basic authentication, OAuth, or token-based authentication. Then, I would write test cases using Rest Assured to verify that the authentication process works as expected. This would involve sending requests with valid credentials and ensuring that the API responds with the expected status codes and authentication tokens. Additionally, I would simulate scenarios such as providing invalid credentials or missing authentication tokens to verify that the API handles these cases appropriately, returning the correct error responses.

Scenario 2:

Question:

You need to test an API endpoint that retrieves user data based on certain criteria using Rest Assured. How would you design your test cases for this scenario?

Answer:

To test the user data retrieval endpoint, I would first identify the criteria that can be used to filter or retrieve specific user data, such as user IDs, usernames, or other attributes. Then, I would design test cases to verify that the endpoint returns the

correct user data based on different combinations of criteria. This would involve sending requests with various parameters using Rest Assured and validating that the API responds with the expected user data. I would also include test cases to verify edge cases, such as requesting data for non-existent users or providing invalid criteria, to ensure that the API handles these scenarios gracefully and returns appropriate error responses. Additionally, I would consider testing performance aspects by sending requests with different load levels to assess the endpoint's scalability and response times.

API Testing External Resource with 100+ Q&A

<https://drive.google.com/file/d/1vlhMeb0jSB9fH6Z78nMG-iDg6pP8HADn/view?usp=sharing>