100 REST ASSURED INTERVIEW QUESTIONS AND ANSWERS (PART 01)

1. What is Rest Assured?

Rest Assured is a Java library designed for testing RESTful web services, simplifying the process of sending HTTP requests and validating responses.

2. How do you set up Rest Assured in a Maven project?

You need to include the Rest Assured dependency in the pom.xml file of your Maven project.

3. What are the key components of Rest Assured?

The key components include given(), when(), and then(), which help in structuring API tests.

4. How do you send a GET request using Rest Assured?

You can send a GET request by using the get() method with the desired URL.

5. How do you validate the status code of a response?

You can validate the status code using response.getStatusCode() or using the then() method for assertions.

6. What is the purpose of given, when, and then in Rest Assured?

Given is used to set up the request, when specifies the action to perform, and then validates the response.

7. How do you extract the response body as a string?

You can extract the response body using response.getBody().asString().

8. What is the purpose of contentType in Rest Assured?

The contentType method is used to set the media type of the request, such as application/json.

9. How do you send a POST request with a JSON body?

You can send a POST request with a JSON body by specifying the body using the body() method along with contentType().

10. How do you handle query parameters in a request?

You can handle query parameters using the queryParam() method while building your request.

11. How do you perform assertions on response data?

You can perform assertions using methods like body(), statusCode(), and header() in the response validation.

12. What is the use of authentication in Rest Assured?

Authentication can be handled using basic authentication or bearer tokens to access protected resources.

13. How do you specify request headers?

You can specify request headers using the header() method while building your request.

14. What are response specifications in Rest Assured?

Response specifications define common validations that can be reused across multiple tests.

15. How do you create request specifications?

Request specifications allow you to define common request parameters and can be reused in multiple tests.

16. What is the use of filters in Rest Assured?

Filters are used to manipulate requests and responses, such as logging requests and responses for debugging purposes.

17. How do you test different HTTP methods using Rest Assured?

You can test various HTTP methods such as GET, POST, PUT, and DELETE by using their respective methods.

18. How do you handle JSON path and XML path in Rest Assured?

JSON path is used to extract values from JSON responses, while XML path is used for extracting values from XML responses.

19. What is the purpose of the statusCode() method?

The statusCode() method is used to assert the expected HTTP status code in the response.

20. How do you validate the response time of an API?

You can validate response time using the time() method to ensure it meets performance requirements.

21. What are the different ways to extract values from the response?

You can extract values using jsonPath() for JSON responses and xmlPath() for XML responses.

22. How do you handle complex JSON responses?

You can navigate complex JSON structures using jsonPath() to retrieve nested values.

23. What are some common HTTP status codes, and what do they represent?

Common HTTP status codes include
- 200 **(OK)**
- 404 **(Not Found)**
- 500 **(Internal Server Error)**
- 401 **(Unauthorized)**

24. How do you integrate Rest Assured with TestNG?

You can integrate Rest Assured with TestNG by creating test methods annotated with @Test and using Rest Assured within those methods.

25. What is the use of the log() method in Rest Assured?

The log() method is used to log request and response details for debugging purposes.

26. How do you handle data-driven testing with Rest Assured?

Data-driven testing can be handled using TestNG's @DataProvider or JUnit's parameterized tests to run multiple test cases with different inputs.

27. How do you use JSON Schema validation in Rest Assured?

You can use the jsonSchema() method to validate the response against a JSON schema.

28. What is the difference between assert and verify in Rest Assured?

Assert will stop the execution if the condition fails, while verify will continue executing the remaining tests regardless of the failure.

29. How do you send form parameters in a POST request using Rest Assured?

You can send form parameters using the formParam() method when building the request.

30. How do you test a PUT request using Rest Assured?

You can test a PUT request by using the put() method with the required URL and body.

31. How do you handle file uploads in Rest Assured?

You can handle file uploads using the multiPart() method along with the POST or PUT request.

32. What are some common challenges while testing APIs?

Common challenges include handling authentication, managing different environments, and validating response data formats.

33. How do you set up logging in Rest Assured?

You can set up logging by using the log() method with parameters like log().all() to log all request and response details.

34. What is the purpose of the Response class in Rest Assured?

The Response class represents the response received from an HTTP request, allowing you to access status codes, headers, and body content.

35. How can you validate response headers in Rest Assured?

You can validate response headers using the header() method to check for specific header values.

36. What is a mock server in the context of API testing?

A mock server simulates an API to test client-side logic without depending on the actual API.

37. How do you integrate Rest Assured with Cucumber?

You can integrate Rest Assured with Cucumber by creating step definitions that utilize Rest Assured methods for API testing.

38. How do you handle timeouts in Rest Assured?

You can handle timeouts using the config() method to set connection and read timeouts.

39. What is the significance of the base URI in Rest Assured?

The base URI simplifies the request URL by allowing you to define a common endpoint that all requests can use.

40. How do you validate JSON data types in a response?

You can validate JSON data types using jsonPath() and assertions to check the type of specific fields in the response.

41. What is the use of the queryParam() method in Rest Assured?

The queryParam() method is used to add query parameters to the URL of the request.

42. How do you handle different content types in Rest Assured?

You can handle different content types by setting the appropriate content type in the request using contentType().

43. How do you make a request with a custom timeout using Rest Assured?

You can set custom timeouts using the config() method with a specified timeout duration.

44. What are the different assertion methods available in Rest Assured?

Different assertion methods include body(), statusCode(), header(), and cookie() for validating various aspects of the response.

45. How do you test for SSL certificates using Rest Assured?

You can test for SSL certificates by disabling SSL verification or by providing a valid certificate when making requests.

46. How do you handle pagination in API responses using Rest Assured?

You can handle pagination by extracting the pagination details from the response and iterating over the pages to fetch data.

47. How do you perform negative testing with Rest Assured?

Negative testing can be performed by sending invalid data or requests to ensure the API responds with appropriate error messages.

48. What is the use of the `extract()` method in Rest Assured?

The `extract()` method is used to retrieve data from the response for further assertions or processing.

49. How do you test webhooks with Rest Assured?

You can test webhooks by simulating the webhook request to the defined endpoint and validating the response.

50. How do you handle multiple environment configurations in Rest Assured?

You can handle multiple environment configurations by externalizing configuration values using properties files or environment variables.