

100 REST ASSURED INTERVIEW QUESTIONS AND ANSWERS (PART 02)

51. How do you mock a REST API using Rest Assured?

You can mock a REST API using libraries like WireMock or Mockito to simulate API responses.

52. What is the purpose of the `requestSpecification` in Rest Assured?

The `requestSpecification` is used to define common request parameters, headers, and authentication for multiple requests.

53. How do you perform a DELETE request using Rest Assured?

You can perform a DELETE request by using the `delete()` method with the appropriate URL.

54. What is the use of `responseSpecification` in Rest Assured?

The `responseSpecification` is used to define common validations for responses that can be reused in multiple tests.

55. How do you handle special characters in query parameters?

You can encode special characters in query parameters using URI encoding methods to ensure valid requests.

56. How can you verify the presence of a specific value in the response body?

You can verify the presence of a specific value using the `body()` method with assertions in the response validation.

57. How do you extract a specific field value from a JSON response?

You can extract a specific field value using `jsonPath()` method followed by the field name.

58. What is the difference between `get()` and `post()` methods in Rest Assured?

The `get()` method is used for retrieving data, while the `post()` method is used for sending data to create a resource.

59. How do you handle versioning in APIs using Rest Assured?

You can handle versioning by including version numbers in the URL or request headers and validating responses accordingly.

60. What are common best practices for API testing with Rest Assured?

Common best practices include organizing tests logically, using request and response specifications, and handling authentication properly.

100 REST ASSURED INTERVIEW QUESTIONS AND ANSWERS (PART 02)

61. How do you validate the structure of a JSON response?

You can validate the structure of a JSON response using JSON Schema validation or by asserting the presence of expected fields.

62. How do you chain multiple requests in Rest Assured?

You can chain multiple requests by using the response from one request as input for another request.

63. What is the significance of the `given()` method in Rest Assured?

The `given()` method is used to set up the initial conditions for the request, such as parameters and headers.

64. How do you handle API response time validation?

You can handle API response time validation by using the `time()` method to assert that the response time is within acceptable limits.

65. How can you perform a health check on an API using Rest Assured?

You can perform a health check by sending a GET request to the health endpoint and validating the response status.

66. How do you use Rest Assured with BDD style syntax?

You can use BDD style syntax by chaining the methods in a more natural language format, such as `given()`, `when()`, and `then()`.

67. How do you handle asynchronous requests in Rest Assured?

You can handle asynchronous requests by sending the request and then polling the response until the expected result is achieved.

68. How do you set up a base path in Rest Assured?

You can set up a base path using the `basePath()` method to define a common path for all requests.

69. What are the advantages of using Rest Assured over other API testing tools?

Rest Assured provides a simple syntax, built-in support for BDD style, and integration with Java testing frameworks.

70. How do you handle API rate limiting in your tests?

You can handle API rate limiting by implementing waits or retries in your tests to comply with the API's usage limits.

100 REST ASSURED INTERVIEW QUESTIONS AND ANSWERS (PART 02)

71. How do you assert JSON response values against expected values?

You can assert JSON response values against expected values using methods like `body()` and `jsonPath()` for validation.

72. How do you handle API error responses in your tests?

You can handle API error responses by validating the status code and error message in the response body.

73. How do you send custom HTTP headers in a request?

You can send custom HTTP headers using the `header()` method while building your request.

74. What is the purpose of the ``baseURI`` method in Rest Assured?

The ``baseURI`` method is used to define a common base URL for all requests, simplifying request construction.

75. How do you log the request and response details for debugging purposes?

You can log the request and response details using the `log()` method, specifying options like `log().all()` to log everything.

76. How do you validate XML response data in Rest Assured?

You can validate XML responses using `xmlPath()` to navigate and assert values in the XML structure.

77. How do you handle OAuth 2.0 authentication in Rest Assured?

You can handle OAuth 2.0 by including the access token in the Authorization header using the `header()` method.

78. What is the difference between response time and latency in API testing?

Response time is the time taken to get a response, while latency is the delay before the response starts after a request is sent.

79. How do you set a default port in Rest Assured?

You can set a default port using `RestAssured.port` to specify a port number that all requests will use.

80. How can you capture cookies in Rest Assured?

You can capture cookies from the response using the `getCookies()` method on the response object.

100 REST ASSURED INTERVIEW QUESTIONS AND ANSWERS (PART 02)

81. How do you check if a JSON array contains a specific value?

You can use `jsonPath()` to access the JSON array and assert whether it contains a specific value.

82. How do you validate the status line of a response?

You can validate the status line by using the `statusLine()` method to compare it against an expected value.

83. How do you handle form authentication in Rest Assured?

You can handle form authentication using the `formAuth()` method, providing a username and password.

84. How do you verify that an element in a JSON array is of a certain type?

You can use `jsonPath()` to access the element and then assert its type using appropriate assertions.

85. How do you test API caching using Rest Assured?

You can test API caching by making multiple requests to the same endpoint and comparing response headers for cache control.

86. How do you set the Content-Encoding header in Rest Assured?

You can set the Content-Encoding header using the `header()` method and specifying the encoding type, like `gzip`.

87. How do you use a JSON file as a request body?

You can use a JSON file as a request body by reading the file contents and passing it to the `body()` method.

88. How do you perform authorization using API keys in Rest Assured?

You can include the API key as a header or query parameter, depending on how the API is designed to receive it.

89. How do you verify that a field in the response body is not null?

You can use `jsonPath()` with an assertion to verify that the field value is not null.

90. How do you handle redirects in Rest Assured?

You can handle redirects by setting redirection handling in the request specification or using `redirect()`.

100 REST ASSURED INTERVIEW QUESTIONS AND ANSWERS (PART 02)

91. How do you validate a JSON response against a specific schema in Rest Assured?

You can use JSON Schema validation by loading the schema and using it with `jsonSchema()` for comparison.

92. How do you handle response compression, such as gzip, in Rest Assured?

Rest Assured automatically decompresses gzip responses, so no extra configuration is typically required.

93. What is the purpose of the `patch()` method in Rest Assured?

The `patch()` method is used to update a resource partially, sending only the modified data in the request body.

94. How do you validate multiple response elements in a single assertion?

You can use the `and()` method in the `then()` block to chain multiple assertions in a single validation step.

95. How do you extract multiple values from the response using `jsonPath()`?

You can extract multiple values by using `jsonPath().getList()` to retrieve a list of values from the response.

96. How do you handle non-standard HTTP methods in Rest Assured?

You can use the `request()` method with the custom HTTP method name to handle non-standard methods.

97. How do you validate nested JSON elements in Rest Assured?

You can navigate nested JSON elements using `jsonPath()` and specify the full path to the element.

98. How do you set default parameters for multiple requests?

You can set default parameters by configuring the `requestSpecification` with the default parameters.

99. How do you verify that a JSON response field meets a regular expression pattern?

You can use `jsonPath()` to extract the field value and then apply a regular expression to validate it.

100. How do you ensure idempotency in API requests?

Idempotency in APIs can be ensured by using HTTP methods like GET, PUT, and DELETE, which do not alter data on multiple requests.