Q. What is RESTful API.?

A RESTful API (Representational State Transfer API) is a web service that uses HTTP requests to interact with resources on a server. It follows six key principles:

1. **Client-Server:** Separates client and server, allowing independent evolution.
2. **Stateless:** Each request from the client is self-contained, with no stored session on the server.
3. **Cacheable:** Responses can be cached to improve performance.
4. **Uniform Interface:** Uses consistent HTTP methods like GET, POST, PUT, and DELETE for CRUD operations.
5. **Layered System:** Can have multiple layers (like authentication) for scalability.
6. **Code on Demand** (optional): Server can send executable code to the client if needed.

RESTful APIs are scalable, flexible, and widely supported, making them ideal for building cross-platform applications.

Q. How RESTful APIs Work?

RESTful APIs use HTTP requests to perform operations on "resources," which are any piece of data.

Resources are typically represented as URIs (Uniform Resource Identifiers), like /users or /products.

When we interact with a RESTful API, we use HTTP methods to perform actions on these resources:

- GET /products/123: Retrieves the product with ID 123.
- POST /products: Creates a new product with data in the request body.
- PUT /products/123: Updates the product with ID 123.
- DELETE /products/123: Deletes the product with ID 123.

Q. What is RestAssured and why is it used?

RestAssured is a Java library that simplifies the process of testing RESTful APIs. It provides a fluent API that makes writing and reading API tests easy.

It's used because it:

- Provides a concise and readable syntax for writing API tests.
- Supports various HTTP methods (GET, POST, PUT, DELETE, etc.).
- Handles JSON and XML responses.
- Allows for easy validation of responses.
- Integrates well with other testing frameworks like JUnit and TestNG.

given()

content type, set cookies, add auth, add param, set heads info etc...

when()

use to get, post, put, delete

then()

validate status code, extract response, extract headers cookies & response body.

Q. How do you set up a RestAssured test?

We start by creating a base URL using RestAssured.baseURI.

Then, we can send requests using methods like given(), when(), and then().

Q. How do you send different HTTP requests (GET, POST, PUT, DELETE) using RestAssured?

Use the methods: get(), post(), put(), and delete().

Q. How do you handle request parameters and headers?

Use the params() and headers() methods to add parameters and headers to our requests.

Q. How do you validate the response status code and body content?

We can use the statusCode() and body() methods to validate the response status code and body content.

Q. How do you handle dynamic response values?

Use regular expressions or JSON path expressions to extract dynamic values and validate them.

Q. How do you validate the status code in RestAssured?

Using the statusCode() method in the then() block:

Q. How can you validate the JSON response body in RestAssured?

Using the **body()** method:

```
        .then()
         .statusCode(200)
         .body("userId", equalTo(1))
         .body("id", equalTo(1))
         .body("title", equalTo("String"));
```

the body() method checks that the userId, id, and title fields in the response JSON match the expected values.

Q. How do you send a POST request with a JSON body using RestAssured?

Use the body() method to include the request payload:

```
given()  .header("Content-Type", "application/json")
                .body("{ \"name\": \"John\", \"job\": \"Developer\" }")
```

Q. How do you handle authentication in RestAssured?

It supports various authentication mechanisms, like Basic, Digest, OAuth, etc.

For Basic authentication, we can use:

```
        given()
         .auth()
         .basic("username", "password")
      .when()
         .get("https://api.example.com/protected/resource")
         .then()
         .statusCode(200);
```

Q. How do you extract a response value using RestAssured?

The extract() method allows us to extract specific information from the response, such as response body, headers, cookies, status code, etc.

For example, to extract a JSON response field:

```
String title = given()
  .when()
  .get("https://jsonplaceholder.typicode.com/posts/1")
  .then()
  .extract()
  .path("title");
```

Q. What is the difference between param(), queryParam(), and formParam()?

- **param():** A generic method for adding parameters to a request. It automatically decides whether the parameter should go into the query string or body based on the HTTP method.
- **queryParam():** Specifically adds the parameter as a query string in the URL.
- **formParam():** Adds the parameter to the body of a POST request as form data.

Q. How do you handle dynamic URLs or endpoints in RestAssured?

We can handle dynamic URLs by using path parameters:

```
String endpoint = "https://jsonplaceholder.typicode.com/posts/{id}";
given()
      .pathParam("id", 1)
 .when()
      .get(endpoint)
 .then()
      .statusCode(200);
```

Q. How do you pass path parameters in RestAssured?

We can pass path parameters using the pathParam() method:

```
given().pathParam("userId", 1)
```

Q. How do you handle query parameters in RestAssured?

We can add query parameters using the queryParam() method:

```
given()
        .queryParam("id", "123")
 .when()
        .get("https://api.example.com/users")
 .then().statusCode(200);
```

Q. How do you send form data with a POST request in RestAssured?

We can send form data using the formParam() method:

```
given()
        .formParam("username", "rajat")
        .formParam("password", "password")
 .when()
        .post("https://api.example.com/login")
 .then().statusCode(200);
```

Q. How do you validate headers in RestAssured responses?

We can validate headers in the response using the header() method in the then() block:

```
given()
  .when()
        .get("https://api.example.com/resource")
  .then()
        .statusCode(200)
        .header("Content-Type", "application/json");
```

Q. How do you chain multiple validations in RestAssured?

By using multiple assertions in the then() block:

```
.then()
 .statusCode(200)
 .body("userId", equalTo(1))
 .body("id", equalTo(1))
 .body("title", notNullValue());
```

Q. How do you handle cookies in RestAssured?

We can extract cookies from the response using the extract().cookie() method:

```
String sessionId = given()
 .when()
 .get("https://api.example.com/login")
 .then()
 .statusCode(200)
 .extract()
 .cookie("Stirng");
```

We can then pass this cookie in subsequent requests using the cookie() method:

```
given()
 .cookie("JSESSIONID", sessionId)
 .when()
 .get("https://api.example.com/secure-page")
 .then().statusCode(200);
```

Q. How do you handle SSL certificates in RestAssured?

For SSL certificate validation, we can use **relaxedHTTPSValidation**():

```
given()
```

```
.relaxedHTTPSValidation()
.when()
.get("https://self-signed.badssl.com/")
.then()
.statusCode(200);
```

Q. How do you read response data using JsonPath in RestAssured?

We can use JsonPath to read specific data from the JSON response:

```
Response response = given()
 .when()
 .get("https://jsonplaceholder.typicode.com/posts/1");
 JsonPath jsonPath = response.jsonPath();
      String title = jsonPath.get("title");
      System.out.println("Title: " + title);
```

Q. How do you send multipart form data (file upload) in RestAssured?

We can send multipart form data (e.g., for file upload) using multiPart():

```
File file = new File("path/to/file.txt");
given()
 .multiPart("file", file)
 .when()
 .post("https://api.example.com/upload")
 .then()
 .statusCode(200);
```

Q. How do you handle XML responses in RestAssured?

RestAssured supports XML validation using xmlPath():

```
Response response = given()
 .when()
```

```
.get("https://api.example.com/xml-endpoint");
    String name = response.xmlPath().getString("person.name");
    System.out.println("Name: " + name);
```

Q. What is the difference between response.asString() and response.getBody().asString() in RestAssured?

- **response.asString():** Converts the entire response to a String format.
- **response.getBody().asString():** Specifically converts only the body part of the response to a String format.

Q. What is the role of Hamcrest in REST-assured?

Hamcrest is a matcher library to define conditions for assertions.

For example, equalTo() is a Hamcrest matcher used to compare values, while hasItems() is used to check for specific items in arrays.

Q. How do you validate the elements of an array in REST-assured?

We can validate an array using the hasItems() matcher:

```
get("/events?id=390").then().assertThat()
.body("odds.price", hasItems("1.30", "5.25"));
```

This validates that the odds array in the JSON response contains prices 1.30 and 5.25.

Q. How do you handle anonymous JSON roots in REST-assured?

Anonymous JSON roots are arrays without a key.

We can validate them using $ or an empty string as the path:

```
when().get("/json").then().body("$", hasItems(1, 2, 3));
```

Q. How do you measure response time in REST-assured?

We can use the time() or timeIn() methods to measure response time:

```
Response response = RestAssured.get("/users/eugenp");
long timeInMS = response.time();  // In milliseconds
long timeInS = response.timeIn(TimeUnit.SECONDS);  // In seconds
```