Rest Assured is a Java library for testing and validating RESTful web services. It simplifies the process of making HTTP requests and validating the responses. Here are some common ways to validate responses using Rest Assured.

Basic Setup

Before we start with the examples, make sure you have Rest Assured and other dependencies set up in your project. If you are using Maven, include the following dependencies in your pom.xml:

xml

Copy code

<dependency>

<groupId>io.rest-assured</groupId>

<artifactId>rest-assured</artifactId>

<version>5.0.0</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.hamcrest</groupId>

<artifactId>hamcrest</artifactId>

<version>2.2</version>

<scope>test</scope>

</dependency>

Validating Status Code

You can validate the HTTP status code returned by the server:

java

Copy code

import io.restassured.RestAssured;

import static io.restassured.RestAssured.\*;

import static org.hamcrest.Matchers.\*;

public class ApiTest {

public static void main(String[] args) {

RestAssured.baseURI = "https://jsonplaceholder.typicode.com";

given()

.when()

.get("/posts/1")

.then()

.statusCode(200);

}

}

Validating Response Headers

You can also validate specific response headers:

java

Copy code

import io.restassured.RestAssured;

import static io.restassured.RestAssured.\*;

import static org.hamcrest.Matchers.\*;

public class ApiTest {

public static void main(String[] args) {

RestAssured.baseURI = "https://jsonplaceholder.typicode.com";

given()

.when()

.get("/posts/1")

.then()

.statusCode(200)

.header("Content-Type", equalTo("application/json; charset=utf-8"));

}

}

Validating Response Body

You can validate the contents of the response body:

Validating JSON Fields

java

Copy code

import io.restassured.RestAssured;

import static io.restassured.RestAssured.\*;

import static org.hamcrest.Matchers.\*;

public class ApiTest {

public static void main(String[] args) {

RestAssured.baseURI = "https://jsonplaceholder.typicode.com";

given()

.when()

.get("/posts/1")

.then()

.statusCode(200)

.body("userId", equalTo(1))

.body("id", equalTo(1))

.body("title", equalTo("sunt aut facere repellat provident occaecati excepturi optio reprehenderit"))

.body("body", startsWith("quia et suscipit"));

}

}

Validating Nested JSON Fields

For nested JSON fields, you can use dot notation or square brackets:

java

Copy code

import io.restassured.RestAssured;

import static io.restassured.RestAssured.\*;

import static org.hamcrest.Matchers.\*;

public class ApiTest {

public static void main(String[] args) {

RestAssured.baseURI = "https://jsonplaceholder.typicode.com";

given()

.when()

.get("/posts/1/comments")

.then()

.statusCode(200)

.body("[0].email", equalTo("Eliseo@gardner.biz"))

.body("[1].name", equalTo("et fugit eligendi deleniti quidem qui sint nihil autem"))

.body("[2].postId", equalTo(1));

}

}

Validating with JSONPath

You can also use JSONPath expressions to validate responses:

java

Copy code

import io.restassured.RestAssured;

import static io.restassured.RestAssured.\*;

import static org.hamcrest.Matchers.\*;

public class ApiTest {

public static void main(String[] args) {

RestAssured.baseURI = "https://jsonplaceholder.typicode.com";

given()

.when()

.get("/posts")

.then()

.statusCode(200)

.body("size()", equalTo(100)) // Validate size of the array

.body("find { it.id == 1 }.title", equalTo("sunt aut facere repellat provident occaecati excepturi optio reprehenderit"))

.body("findAll { it.userId == 1 }.size()", equalTo(10)); // Validate number of posts by userId 1

}

}

Validating XML Response

If your API returns XML, you can validate it similarly using XMLPath:

java

Copy code

import io.restassured.RestAssured;

import static io.restassured.RestAssured.\*;

import static org.hamcrest.Matchers.\*;

public class ApiTest {

public static void main(String[] args) {

RestAssured.baseURI = "https://www.w3schools.com";

given()

.when()

.get("/xml/note.xml")

.then()

.statusCode(200)

.body("note.to", equalTo("Tove"))

.body("note.from", equalTo("Jani"))

.body("note.heading", equalTo("Reminder"))

.body("note.body", equalTo("Don't forget me this weekend!"));

}

}

Complete Example

Combining various validation techniques, here is a complete example:

java

Copy code

import io.restassured.RestAssured;

import static io.restassured.RestAssured.\*;

import static org.hamcrest.Matchers.\*;

public class ApiTest {

public static void main(String[] args) {

RestAssured.baseURI = "https://jsonplaceholder.typicode.com";

given()

.when()

.get("/posts/1")

.then()

.statusCode(200)

.header("Content-Type", equalTo("application/json; charset=utf-8"))

.body("userId", equalTo(1))

.body("id", equalTo(1))

.body("title", equalTo("sunt aut facere repellat provident occaecati excepturi optio reprehenderit"))

.body("body", startsWith("quia et suscipit"));

}

}

Conclusion

Rest Assured provides a comprehensive way to validate API responses using various matchers and assertions. You can validate status codes, headers, and response bodies (both JSON and XML). By leveraging these capabilities, you can create robust API tests that ensure your endpoints are f1unctioning as expected.