

REST Assured Assertions

In **REST Assured**, **assertions** are used to **validate** the response of an API against expected values. This is crucial in **API testing** to ensure that the API behaves as expected.

♦ Common Assertions in REST Assured

1. Status Code Assertion

```
java

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

2. Content-Type Assertion

```
java

.then()
.contentType(ContentType.JSON);
```

3. Response Body Assertion (using Hamcrest Matchers)

✔ Check if JSON key has a specific value

```
java

.then()
.body("name", equalTo("John"));
```

✔ Check nested fields

```
java

.then()
.body("data.id", equalTo(101));
```

✔ Check if list contains value

```
java

.then()
.body("items", hasItem("Apple"));
```

✔ Check list size

```
java

.then()
  .body("items.size()", equalTo(3));
```

✓ Check multiple assertions

```
java

.then()
  .body("status", equalTo("success"))
  .body("code", equalTo(200));
```

◆ Import Statements (for assertions)

```
java

import static io.restassured.RestAssured.*;
import static org.hamcrest.Matchers.*;
```

◆ Example: Complete Assertion

```
java

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://reqres.in";

        given()
        .when()
        .get("/api/users/2")
        .then()
        .statusCode(200)
        .body("data.id", equalTo(2))
        .body("data.email", equalTo("janet.weaver@reqres.in"));
    }
}
```

◆ Useful Hamcrest Matchers

Matcher	Description
<code>equalTo()</code>	Checks for equality
<code>hasItem()</code>	Checks if a list has a value
<code>hasItems()</code>	Checks multiple values in a list
<code>containsString()</code>	Checks if string contains text
<code>not()</code>	Asserts negative conditions

Let me know if you want assertion examples for XML responses, JSON schema validation, or response time validation.

You can **validate response time** in REST Assured using `.time()` or `.time(lessThan(...))` from Hamcrest matchers.

✓ Basic Response Time Validation

```
java

given()
.when()
.get("https://reqres.in/api/users/2")
.then()
.time(lessThan(2000L)); // Response time should be < 2000 milliseconds
```

- **2000L** = 2000 milliseconds
- **lessThan()** is from `org.hamcrest.Matchers`

🔍 Check and Print Actual Response Time

```
java

long time =
given()
.when()
.get("https://reqres.in/api/users/2")
.time(); // returns time in milliseconds

System.out.println("Response time: " + time + " ms");
```

🔧 Assert Response Time with JUnit (Optional)

```
java

import static org.junit.Assert.assertTrue;

long responseTime =
given()
.when()
.get("https://reqres.in/api/users/2")
.time();

assertTrue(responseTime < 2000); // JUnit assertion
```

📦 Required Import

Make sure to import:

```
java

import static org.hamcrest.Matchers.lessThan;
```

Would you like to integrate this into a test framework like **TestNG** or **JUnit**?

Chained assertions in **REST Assured** allow you to validate multiple parts of a response in a clean and readable way, improving **test maintainability** and **reporting clarity**.

✓ Example: Chained Assertions for Clean Reporting

```
java

given()
.when()
.get("https://reqres.in/api/users/2")
.then()
    .assertThat()
        .statusCode(200)
        .contentType("application/json; charset=utf-8")
        .body("data.id", equalTo(2))
        .body("data.email", equalTo("janet.weaver@reqres.in"))
        .body("support.text", containsString("support"));
```

🔍 Why Chaining Is Useful

- Groups all validations neatly under `.then()`
- Generates better **failure reports** in test runners like TestNG or JUnit
- Encourages **clean and readable** code

📦 Required Imports

```
java

import static io.restassured.RestAssured.*;
import static org.hamcrest.Matchers.*;
```

🔧 Full JUnit Example

```
java

import org.junit.Test;
import static io.restassured.RestAssured.*;
import static org.hamcrest.Matchers.*;

public class ChainedAssertionTest {
    @Test
    public void validateUserData() {
        given()
        .when()
        .get("https://reqres.in/api/users/2")
        .then()
            .assertThat()
                .statusCode(200)
                .contentType("application/json; charset=utf-8")
                .body("data.first_name", equalTo("Janet"))
                .body("data.last_name", equalTo("Weaver"))
                .body("support.url", equalTo("https://reqres.in/#support-heading"));
    }
}
```

💡 Tip for Better Reporting

If you're using **TestNG** or **ExtentReports**, each chained `body()` statement can be wrapped in a `try-catch` to capture detailed failure logs, or you can split assertions per test case method.

Let me know if you'd like this in **TestNG + ExtentReports**, or with a reusable utility structure for assertions.