

Rest Assured JSON Validation

Validating JSON responses using **Rest Assured** in Java is a core task for API testing. Here's a complete guide with **code examples** and **different types of validation**.

✓ Setup

Add these dependencies in `pom.xml` for Maven:

```
xml

<dependencies>
  <dependency>
    <groupId>io.rest-assured</groupId>
    <artifactId>rest-assured</artifactId>
    <version>5.3.0</version>
    <scope>test</scope>
  </dependency>
</dependencies>
```

✓ Sample JSON Response

Assume the API returns:

```
json

{
  "id": 101,
  "name": "John",
  "email": "john@example.com",
  "roles": ["admin", "user"]
}
```

✓ Basic JSON Validation

```
java

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

public class JsonValidationTest {
  public static void main(String[] args) {
    baseUrl = "https://api.example.com";

    given()
      .when()
      .get("/users/101")
      .then()
      .statusCode(200)
      .body("id", equalTo(101))
      .body("name", equalTo("John"))
      .body("email", containsString("@example.com"))
      .body("roles", hasItems("admin", "user"));
  }
}
```

✓ Validate Nested JSON

Assume response:

```
json
{
  "data": {
    "user": {
      "id": 101,
      "details": {
        "name": "John",
        "active": true
      }
    }
  }
}
```

Validation:

```
java

.body("data.user.id", equalTo(101))
.body("data.user.details.name", equalTo("John"))
.body("data.user.details.active", equalTo(true));
```

✓ Extract Values & Validate Programmatically

```
java

Response response = given()
    .when()
    .get("/users/101");

int id = response.jsonPath().getInt("id");
String name = response.jsonPath().getString("name");

assert id == 101;
assert name.equals("John");
```

✓ Validate List/Array

Response:

```
json
{
  "users": [
    {"id": 1, "name": "A"},
    {"id": 2, "name": "B"}
  ]
}
```

Validation:

```
java

.body("users.size()", equalTo(2))
.body("users.name", hasItems("A", "B"))
.body("users[0].id", equalTo(1));
```

✅ Validate with JSON Schema

Add dependency:

```
xml

<dependency>
  <groupId>io.rest-assured</groupId>
  <artifactId>json-schema-validator</artifactId>
  <version>5.3.0</version>
</dependency>
```

Validation:

```
java

import static io.restassured.module.jsv.JsonSchemaValidator.*;

.body(matchesJsonSchemaInClasspath("user_schema.json"));
```

If you'd like, I can also give you a **mini-project or assignment** for JSON validation in Rest Assured. Would you like that?