REST Assured Extracting Response Data

In **Rest Assured**, extracting response data is essential when you need to verify response values or pass them between API calls. Here's a step-by-step guide with examples on how to extract different types of response data.

✅ Step 1: Basic Setup

Add the required Maven dependency:

```xml
<dependency>
    <groupId>io.rest-assured</groupId>
    <artifactId>rest-assured</artifactId>
    <version>5.4.0</version>
    <scope>test</scope>
</dependency>
```

✅ Step 2: Extracting Response as a Whole

```java
import io.restassured.response.Response;
import static io.restassured.RestAssured.*;

public class ExtractExample {
    public static void main(String[] args) {
        Response response =
            given()
                .baseUri("https://jsonplaceholder.typicode.com")
            .when()
                .get("/posts/1");

        System.out.println("Response Body:");
        System.out.println(response.asPrettyString());
    }
}
```

✅ Step 3: Extracting Specific Values

You can extract values like **`id`**, **`title`**, **`userId`** etc.

🔹 Using `.path()` method:

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

```
System.out.println("Title: " + title);
```

- **Using `.jsonPath()` method:**

```java
import io.restassured.path.json.JsonPath;

Response response =
    given()
        .baseUri("https://jsonplaceholder.typicode.com")
    .when()
        .get("/posts/1");

JsonPath json = response.jsonPath();
String body = json.getString("body");
System.out.println("Body: " + body);
```

✅ Step 4: Extracting List of Values

```java
List<Integer> ids =
    given()
        .baseUri("https://jsonplaceholder.typicode.com")
    .when()
        .get("/posts")
    .then()
        .extract()
        .path("id");

System.out.println("All IDs: " + ids);
```

✅ Step 5: Extracting Data from Headers

```java
String contentType =
    given()
        .baseUri("https://jsonplaceholder.typicode.com")
    .when()
        .get("/posts/1")
    .then()
        .extract()
        .header("Content-Type");

System.out.println("Content-Type: " + contentType);
```

✅ Step 6: Extracting Status Code

```java
int statusCode =
    given()
        .baseUri("https://jsonplaceholder.typicode.com")
    .when()
        .get("/posts/1")
    .then()
        .extract()
        .statusCode();

System.out.println("Status Code: " + statusCode);
```

✅ Step 7: Store and Use Extracted Data

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

// Use in another request or assertion
System.out.println("Extracted Title: " + postTitle);
```

🔚 Summary

Task	Method Used
Whole Response	**`Response response = given...`**
Single Field	**`.extract().path("field")`**
JSON Path	**`.jsonPath().get("field")`**
Header	**`.extract().header("name")`**
Status Code	**`.extract().statusCode()`**
List of Fields	**`.extract().path("list")`**

If you'd like, I can give you a **mini-assignment** or a **mock API lab** on extracting response data. Would you like that?

Storing response values from one API call and using them in a **subsequent API request** is a common and powerful practice in API testing and automation with **Rest Assured**.

✅ Use Case Example

1. **POST** request to create a user → Get the `id` from the response.
2. **GET** request to retrieve the user → Use the stored `id`.

✅ Step-by-Step: Store and Use Response Data

🟢 Step 1: Send POST Request and Store ID

```java
import io.restassured.response.Response;
import static io.restassured.RestAssured.*;

public class StoreAndUse {
    public static void main(String[] args) {

        // Step 1: Create a new user (POST)
        Response postResponse =
            given()
                .baseUri("https://reqres.in")
                .header("Content-Type", "application/json")
                .body("{ \"name\": \"Koti\", \"job\": \"QA\" }")
            .when()
                .post("/api/users");

        // Extract user ID from response
        String userId = postResponse.jsonPath().getString("id");
        System.out.println("User ID extracted: " + userId);

        // Step 2: Use the user ID in GET request (this API is dummy; modify accordingly)
        Response getResponse =
            given()
                .baseUri("https://reqres.in")
            .when()
                .get("/api/users/" + userId);

        // Print GET response
        System.out.println("GET Response:");
        System.out.println(getResponse.asPrettyString());
    }
}
```

🔁 Reusable Version with Variables

```java
String userId =
    given()
        .baseUri("https://reqres.in")
        .header("Content-Type", "application/json")
        .body("{ \"name\": \"Koti\", \"job\": \"Engineer\" }")
    .when()
        .post("/api/users")
    .then()
        .extract()
        .path("id");
```

```
// Use stored userId in next API call
given()
    .baseUri("https://reqres.in")
.when()
    .get("/api/users/" + userId)
.then()
    .statusCode(200);
```

✅ Things You Can Store:

Type	Example Extraction
Path Value	`.path("id")`
Header	`.header("Content-Type")`
Cookie	`.cookie("JSESSIONID")`
List/Array	`.path("data.id[0]")`
Status Code	`.statusCode()`

📌 Tip:

If you're doing multiple steps (e.g., login → get token → use token), store values in `String`, `Map`, or even static variables across methods/classes.

Would you like a **token-based login + protected API** flow example as well?

Great question!

In **Rest Assured**, both `extract().path()` and `extract().response()` are used to extract data **from API responses**, but they serve slightly different purposes depending on what you need:

✅ `extract().path()` – Direct Value Extraction

- Use it when you want to **directly get a value** (like an ID, name, token, etc.) from the response.
- Returns the **specific value** (not the whole response).

🔹 Example:

```
java

String userId =
    given()
        .baseUri("https://reqres.in")
        .header("Content-Type", "application/json")
        .body("{ \"name\": \"Koti\", \"job\": \"QA\" }")
    .when()
        .post("/api/users")
    .then()
        .statusCode(201)
        .extract()
        .path("id");

System.out.println("Extracted User ID: " + userId);
```

✅ `extract().response()` – Full Response Extraction

- Use this when you want to store the **entire response object** for further inspection or multiple extractions.
- You can later use `.path()`, `.jsonPath()`, `.getStatusCode()`, etc., on this `Response` object.

🔹 Example:

```
java

import io.restassured.response.Response;

Response response =
    given()
        .baseUri("https://reqres.in")
        .header("Content-Type", "application/json")
        .body("{ \"name\": \"Koti\", \"job\": \"QA\" }")
    .when()
        .post("/api/users")
    .then()
        .statusCode(201)
        .extract()
        .response();

String id = response.path("id");
String createdAt = response.path("createdAt");

System.out.println("ID: " + id);
System.out.println("Created At: " + createdAt);
```

⚖️ Summary: When to Use What?

Use Case	Method	Return Type
Extract a **single field**	`.extract().path("key")`	Value (String, int, etc.)
Extract **multiple fields later**	`.extract().response()`	**`Response`** object

✅ Best Practice

If you only need **one value**, use:

```java
String token = extract().path("token");
```

If you need **many values or headers/status**, use:

```java
Response response = extract().response();
```

Would you like a **real-world scenario** showing `extract().response()` for token-based auth?