Rest Assured Authentication

In **Rest Assured**, handling **authentication** is straightforward and supports multiple authentication mechanisms such as:

✅ 1. Basic Authentication

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

✅ 2. Digest Authentication

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

✅ 3. Preemptive Basic Authentication

(Sends credentials **without** waiting for server challenge)

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

✅ 4. Form Authentication

(Used when login is through a form, like web apps)

```java
given()
    .auth()
    .form("username", "password", FormAuthConfig.formAuthConfig()
        .withAutoDetectionFrom("https://example.com/login"))
.when()
    .get("https://example.com/dashboard")
.then()
    .statusCode(200);
```

✅ 5. OAuth 2.0 Token Authentication

```java
given()
    .auth()
    .oauth2("your_access_token")
.when()
    .get("https://example.com/api")
.then()
    .statusCode(200);
```

> You usually get the token via a separate token request before using it in the API call.

✅ 6. Bearer Token in Header (Manual)

Sometimes used if `.oauth2()` isn't flexible enough:

```java
given()
    .header("Authorization", "Bearer your_access_token")
.when()
    .get("https://example.com/api")
.then()
    .statusCode(200);
```

🔁 Optional: Reusable Authentication in a `RequestSpecification`

```java
RequestSpecification reqSpec = new RequestSpecBuilder()
    .setBaseUri("https://example.com")
    .addHeader("Authorization", "Bearer your_access_token")
    .build();

given()
    .spec(reqSpec)
.when()
    .get("/api")
.then()
    .statusCode(200);
```

💡 Tip: If dealing with session-based auth (cookies), you may use:

```java
Response res = given()
    .formParam("username", "admin")
    .formParam("password", "admin123")
.when()
    .post("/login");

String sessionId = res.getCookie("JSESSIONID");

given()
    .cookie("JSESSIONID", sessionId)
.when()
    .get("/secure-endpoint")
.then()
    .statusCode(200);
```

Would you like me to show a real-world mini-project combining token-based authentication and test validation using Rest Assured?

Here's a **real-world example** of handling **token-based login in a banking application** using **Rest Assured**.

✅ Scenario: Token-Based Login in Banking App

Workflow:

1. Login with username/password → get a token.
2. Use the token to access secure endpoints like `/balance` or `/transfer`.

🔧 1. Login Request – Get the Token

Endpoint: `POST /api/login`

```
json

// Request Body:
{
  "username": "john.doe",
  "password": "securePass123"
}
```

Sample Response:

```
json

{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6Ikp..."
}
```

✅ Java Code to Get Token:

```
java

import io.restassured.response.Response;

String baseURI = "https://banking.example.com";

Response loginResponse = given()
     .baseUri(baseURI)
     .header("Content-Type", "application/json")
     .body("{ \"username\": \"john.doe\", \"password\": \"securePass123\" }")
.when()
     .post("/api/login");

String token = loginResponse.jsonPath().getString("token");
System.out.println("Token: " + token);
```

🔐 2. Use Token to Access Secure Endpoint

Endpoint: `GET /api/account/balance`

Sample Response:

```json
{
  "accountNumber": "987654321",
  "balance": 12500.75,
  "currency": "INR"
}
```

✅ Java Code:

```java
given()
    .baseUri(baseURI)
    .header("Authorization", "Bearer " + token)
.when()
    .get("/api/account/balance")
.then()
    .statusCode(200)
    .body("balance", greaterThan(0.0f))
    .body("currency", equalTo("INR"));
```

🧪 Full Example in One Test

```java
@Test
public void testTokenBasedLoginAndBalanceCheck() {
    String baseURI = "https://banking.example.com";

    // Step 1: Login and extract token
    String token = given()
        .baseUri(baseURI)
        .header("Content-Type", "application/json")
        .body("{ \"username\": \"john.doe\", \"password\": \"securePass123\" }")
    .when()
        .post("/api/login")
    .then()
        .statusCode(200)
        .extract()
        .jsonPath()
        .getString("token");

    // Step 2: Use token to check balance
    given()
        .baseUri(baseURI)
        .header("Authorization", "Bearer " + token)
    .when()
        .get("/api/account/balance")
    .then()
        .statusCode(200)
        .body("balance", greaterThan(0.0f))
        .body("currency", equalTo("INR"));
}
```

💡 Tips:

- Save tokens in a config file or environment variable if reused across tests.
- Use **`RequestSpecification`** for better reusability if multiple secure APIs share the same token.

Would you like a reusable **RestAssured framework** to structure login, token reuse, and tests for the banking app?