2. Automate a POST API — User Login

POST: https://api.ecommerce.com/v1/auth/login
Request Body:

json

CopyEdit

```json
{
  "email": "user@example.com",
  "password": "securePassword123"
}
```

Expected Response:

json

CopyEdit

```json
{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6..."
}
```

Step 1: Add Dependencies (pom.xml)

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

  <!-- JSON Parsing (Optional) -->
  <dependency>
    <groupId>org.json</groupId>
    <artifactId>json</artifactId>
    <version>20230227</version>
```

```xml
            </dependency>


            <!-- JUnit -->
            <dependency>
                <groupId>org.junit.jupiter</groupId>
                <artifactId>junit-jupiter</artifactId>
                <version>5.10.0</version>
                <scope>test</scope>
            </dependency>
        </dependencies>
```

Step 2: Create the Test Class

```java
import io.restassured.RestAssured;
import io.restassured.response.Response;
import org.json.JSONObject;
import org.junit.jupiter.api.Test;


import static io.restassured.RestAssured.given;
import static org.junit.jupiter.api.Assertions.assertEquals;
import static org.junit.jupiter.api.Assertions.assertTrue;


public class LoginApiTest {

    @Test
    public void testLoginSuccess() {
        // Set Base URI
        RestAssured.baseURI = "https://api.ecommerce.com/v1";


        // Create JSON payload
        JSONObject requestBody = new JSONObject();
        requestBody.put("email", "user@example.com");
        requestBody.put("password", "securePassword123");
```

```java
    // Make POST request
    Response response = given()
        .header("Content-Type", "application/json")
        .body(requestBody.toString())
        .when()
        .post("/auth/login");


    // Assertions
    assertEquals(200, response.getStatusCode(), "Expected 200 OK");
    String token = response.jsonPath().getString("token");
    assertTrue(token != null && token.length() > 10, "Token should be returned");
  }
}
```