https://www.linkedin.com/in/anshulagarwal30

Cucumber BDD Framework for REST Assured API Automation

1️⃣ Project Structure

```
api-automation-framework
|── src
|    ├── main
|    |    ├── java
|    |    |    ├── utils (Utility classes like ConfigReader, RequestBuilder)
|    ├── test
|    |    ├── java
|    |    |    ├── stepdefinitions (Step Definitions for Cucumber)
|    |    |    ├── runners (Test Runner classes)
|    |    |    ├── hooks (Setup & Teardown Hooks)
|    |    |    ├── testdata (Test data classes)
|    |    ├── resources
|    |    |    ├── features (Cucumber feature files)
|    |    |    ├── config.properties (Configuration file)
|── pom.xml (Maven dependencies)
|── Jenkinsfile (CI/CD setup)
```

2️⃣ Dependencies (pom.xml)

```xml
<dependencies>
   <!-- Cucumber Dependencies -->
   <dependency>
      <groupId>io.cucumber</groupId>
      <artifactId>cucumber-java</artifactId>
      <version>7.0.0</version>
   </dependency>
   <dependency>
      <groupId>io.cucumber</groupId>
      <artifactId>cucumber-junit</artifactId>
      <version>7.0.0</version>
      <scope>test</scope>
   </dependency>

   <!-- REST Assured -->
   <dependency>
      <groupId>io.rest-assured</groupId>
      <artifactId>rest-assured</artifactId>
      <version>5.0.1</version>
   </dependency>

   <!-- Jackson for JSON Parsing -->
   <dependency>
      <groupId>com.fasterxml.jackson.core</groupId>
      <artifactId>jackson-databind</artifactId>
      <version>2.14.0</version>
   </dependency>

   <!-- TestNG -->
   <dependency>
      <groupId>org.testng</groupId>
      <artifactId>testng</artifactId>
      <version>7.4.0</version>
   </dependency>
</dependencies>
```

3️⃣ Cucumber Feature File (CRUD API Operations)

📍 **Location**: src/test/resources/features/ApiTests.feature

```
Feature: API Automation using REST Assured

  Scenario: Verify GET API request
   Given I perform GET request on "/users/1"
   Then I should get response code 200

  Scenario: Verify POST API request
   Given I perform POST request on "/users"
   And I send request body
    """
    {
      "name": "John Doe",
      "job": "Software Engineer"
    }
    """
   Then I should get response code 201
   And response should contain "name" as "John Doe"

  Scenario: Verify PUT API request
   Given I perform PUT request on "/users/1"
   And I send request body
    """
    {
      "name": "Updated User",
      "job": "Senior Engineer"
    }
    """
   Then I should get response code 200

 Scenario: Verify DELETE API request
   Given I perform DELETE request on "/users/1"
   Then I should get response code 204
```

4️⃣ Step Definitions

📍 **Location**: src/test/java/stepdefinitions/ApiStepDefinitions.java

```java
import io.cucumber.java.en.*;
import io.restassured.response.Response;
import io.restassured.specification.RequestSpecification;

import static io.restassured.RestAssured.*;
import static org.junit.Assert.*;

public class ApiStepDefinitions {
    private RequestSpecification request;
    private Response response;
    private final String BASE_URL = "https://reqres.in/api";

    @Given("I perform GET request on {string}")
    public void i_perform_get_request_on(String endpoint) {
        response = given().when().get(BASE_URL + endpoint);
    }

    @Given("I perform POST request on {string}")
    public void i_perform_post_request_on(String endpoint) {
        request = given().header("Content-Type", "application/json");
    }

    @Given("I send request body")
    public void i_send_request_body(String body) {
        response = request.body(body).when().post(BASE_URL + "/users");
    }

    @Given("I perform PUT request on {string}")
    public void i_perform_put_request_on(String endpoint) {
        request = given().header("Content-Type", "application/json");
    }

    @Given("I perform DELETE request on {string}")
    public void i_perform_delete_request_on(String endpoint) {
        response = given().when().delete(BASE_URL + endpoint);
    }

    @Then("I should get response code {int}")
    public void i_should_get_response_code(int statusCode) {
        assertEquals(statusCode, response.getStatusCode());
    }

    @Then("response should contain {string} as {string}")
    public void response_should_contain_as(String key, String value) {
        assertEquals(value, response.jsonPath().getString(key));
    }
}
```

5️⃣ Test Runner Class

📍 **Location**: **src/test/java/runners/TestRunner.java**

```
import io.cucumber.junit.Cucumber;
import io.cucumber.junit.CucumberOptions;
import org.junit.runner.RunWith;

@RunWith(Cucumber.class)
@CucumberOptions(
        features = "src/test/resources/features",
        glue = "stepdefinitions",
        plugin = {"pretty", "html:target/cucumber-report.html"},
        monochrome = true
)
public class TestRunner {
}
```

6️⃣ TestNG Configuration (testng.xml)

📍 **Location:** testng.xml

```
<suite name="API Test Suite">
   <test name="API Tests">
      <classes>
         <class name="runners.TestRunner"/>
      </classes>
   </test>
</suite>
```

7️⃣ CI/CD Integration (Jenkinsfile)

📍 **Location:** Jenkinsfile

```
pipeline {
   agent any
   stages {
      stage('Checkout') {
         steps {
            git 'https://github.com/your-repo/api-automation.git'
         }
      }
      stage('Build') {
         steps {
            sh 'mvn clean install'
         }
      }
      stage('Run Tests') {
         steps {
            sh 'mvn test'
         }
      }
      stage('Generate Reports') {
         steps {
          sh 'mvn verify'
        }
      }
    }
}
```

8️⃣ Reports (Cucumber Reports)

📍 **Generate reports using Maven Cucumber Plugin**

```
<build>
   <plugins>
      <plugin>
         <groupId>net.masterthought</groupId>
         <artifactId>maven-cucumber-reporting</artifactId>
         <version>5.0.0</version>
         <executions>
            <execution>
               <id>report</id>
               <phase>verify</phase>
               <goals>
                  <goal>generate</goal>
               </goals>
            </execution>
         </executions>
      </plugin>
   </plugins>
</build>
```

Conclusion

🔷 This framework follows **Cucumber BDD** to automate **REST APIs using REST Assured**.
🔷 It supports **GET, POST, PUT, DELETE** requests.
🔷 Uses **TestNG for execution** and **Jenkins for CI/CD integration**.
🔷 Generates **Cucumber Reports** for test execution analysis.

🔥 **Next Steps**

- Implement **OAuth 2.0 authentication** for secured APIs.
- Add **custom logging & reporting** in test results.
- Integrate with **Allure Reports** for better visualization.

Happy Learning! 🌸