# Testing api:

import io.restassured.RestAssured;

import org.junit.BeforeClass;

import org.junit.Test;

import java.net.URI;

import java.net.http.HttpClient;

import java.net.http.HttpRequest;

import java.net.http.HttpResponse;

import static io.restassured.RestAssured.given;

import static org.hamcrest.Matchers.\*;

public class WebApiTest {

private static final String BASE\_URL\_REST\_ASSURED = "http://your-api-base-url"; // Replace with your actual API base URL

private static final String BASE\_URL\_HTTP\_CLIENT = "http://manbas2527-api-base-url"; // Replace with your actual API base URL

// Helper method to send HTTP GET requests using HttpClient

private static HttpResponse<String> sendGetRequest(String baseUrl, String endpoint) throws Exception {

HttpClient client = HttpClient.newHttpClient();

HttpRequest request = HttpRequest.newBuilder()

.uri(URI.create(baseUrl + endpoint))

.GET()

.build();

return client.send(request, HttpResponse.BodyHandlers.ofString());

}

@BeforeClass

public static void setup() {

// Set the base URI for the API using RestAssured

RestAssured.baseURI = BASE\_URL\_REST\_ASSURED;

}

@Test

public void verifyApiEndpointAvailability\_RestAssured() {

given()

.when()

.get("/api/home")

.then()

.statusCode(200);

}

@Test

public void verifyApiEndpointAvailability\_HttpClient() throws Exception {

HttpResponse<String> response = sendGetRequest(BASE\_URL\_HTTP\_CLIENT, "/api/home");

int statusCode = response.statusCode();

System.out.println("Verify API Endpoint Availability (HttpClient) - Response Status Code: " + statusCode);

assert statusCode == 200 : "Expected status code 200, but got " + statusCode;

}

@Test

public void verifyResponseFormat\_RestAssured() {

given()

.when()

.get("/sample-endpoint")

.then()

.contentType("application/json");

}

@Test

public void verifyResponseFormat\_HttpClient() throws Exception {

HttpResponse<String> response = sendGetRequest(BASE\_URL\_HTTP\_CLIENT, "/sample-endpoint");

String contentType = response.headers().firstValue("Content-Type").orElse("");

System.out.println("Verify Response Format (HttpClient) - Content-Type: " + contentType);

assert contentType.equalsIgnoreCase("application/json") : "Expected JSON response";

}

@Test

public void testUnauthorizedAccess\_RestAssured() {

given()

.when()

.get("/protected-endpoint")

.then()

.statusCode(401);

}

@Test

public void testUnauthorizedAccess\_HttpClient() throws Exception {

HttpResponse<String> response = sendGetRequest(BASE\_URL\_HTTP\_CLIENT, "/protected-endpoint");

int statusCode = response.statusCode();

System.out.println("Test Unauthorized Access (HttpClient) - Response Status Code: " + statusCode);

assert statusCode == 401 : "Expected status code 401, but got " + statusCode;

}

@Test

public void testPositiveScenario\_RestAssured() {

given()

.when()

.get("/positive-scenario-endpoint")

.then()

.statusCode(200)

.body("someKey", equalTo("expectedValue"))

.body("anotherKey", containsString("partialExpectedValue"));

}

@Test

public void testPositiveScenario\_HttpClient() throws Exception {

HttpResponse<String> response = sendGetRequest(BASE\_URL\_HTTP\_CLIENT, "/positive-scenario-endpoint");

int statusCode = response.statusCode();

String responseBody = response.body();

System.out.println("Test Positive Scenario (HttpClient) - Response Status Code: " + statusCode);

System.out.println("Response Body: " + responseBody);

assert statusCode == 200 : "Expected status code 200, but got " + statusCode;

assert responseBody.contains("expectedValue") : "Response body does not contain expected value";

// Add more assertions based on the expected response body

}

@Test

public void testAnotherScenario\_RestAssured() {

given()

.when()

.get("/another-scenario-endpoint")

.then()

.statusCode(200)

.body("someKey", equalTo("anotherExpectedValue"));

}

@Test

public void testAnotherScenario\_HttpClient() throws Exception {

HttpResponse<String> response = sendGetRequest(BASE\_URL\_HTTP\_CLIENT, "/another-scenario-endpoint");

int statusCode = response.statusCode();

String responseBody = response.body();

System.out.println("Test Another Scenario (HttpClient) - Response Status Code: " + statusCode);

System.out.println("Response Body: " + responseBody);

assert statusCode == 200 : "Expected status code 200, but got " + statusCode;

assert responseBody.contains("anotherExpectedValue") : "Response body does not contain expected value";

}

}