**Exercise 2: Mocking External Services (RESTful APIs)**

You need to test a service that calls an external RESTful API.

**ApiService.java :-**

package org.example1;  
  
public class ApiService {  
  
 private final RestClient restClient;  
  
 public ApiService(RestClient restClient) {  
 this.restClient = restClient;  
 }  
  
 public String fetchData() {  
 String response = restClient.getResponse();  
 return "Fetched " + response;  
 }  
}

**RestClient.java :-**

package org.example1;  
  
public interface RestClient {  
 String getResponse();  
}

**ApiServiceTest.java :-**

package org.example1;  
  
import org.junit.jupiter.api.Test;  
import static org.mockito.Mockito.\*;  
import static org.junit.jupiter.api.Assertions.\*;  
  
public class ApiServiceTest {  
  
 @Test  
 public void testServiceWithMockRestClient() {  
 // Step 1: Create mock REST client  
 RestClient mockRestClient = *mock*(RestClient.class);  
  
 // Step 2: Stub method to return mock response  
 *when*(mockRestClient.getResponse()).thenReturn("Mock Response");  
  
 // Step 3: Inject mock and verify service logic  
 ApiService apiService = new ApiService(mockRestClient);  
 String result = apiService.fetchData();  
  
 // Assert that the mocked result is used correctly  
 *assertEquals*("Fetched Mock Response", result);  
  
 // Optional: verify interaction  
 *verify*(mockRestClient, *times*(1)).getResponse();  
 }  
}

**pom.xml :-**

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>org.example1</groupId>  
 <artifactId>MOcking\_external\_service</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <packaging>jar</packaging>  
  
 <name>MOcking\_external\_service</name>  
 <url>http://maven.apache.org</url>  
  
 <properties>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
 </properties>  
  
 <dependencies>  
 <!-- JUnit 5 -->  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter</artifactId>  
 <version>5.10.0</version>  
 <scope>test</scope>  
 </dependency>  
  
 <!-- Mockito -->  
 <dependency>  
 <groupId>org.mockito</groupId>  
 <artifactId>mockito-core</artifactId>  
 <version>5.11.0</version>  
 <scope>test</scope>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-surefire-plugin</artifactId>  
 <version>3.1.2</version>  
 <configuration>  
 <argLine>-XX:+EnableDynamicAgentLoading</argLine>  
 </configuration>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>

**Output :-**

