**Week – 2**

**HANDS ON**

**Mockito exercises**

**Exercise 2: Verifying Interactions**

**Code**

1. **Pom.xml**

<?xml version="1.0" encoding="UTF-8"?>

<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>com.example</groupId>

<artifactId>mockito-demo</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<maven.compiler.source>22</maven.compiler.source>

<maven.compiler.target>22</maven.compiler.target>

<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>5.12.0</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

1. **ExternalApi.java**

package com.example;

public class ExternalApi {

public String getData() {

return "Real Data";

}

}

1. **MyService.java**

package com.example;

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

1. **MyServiceTest.java**

package com.example;

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.assertEquals;

import static org.mockito.Mockito.\*;

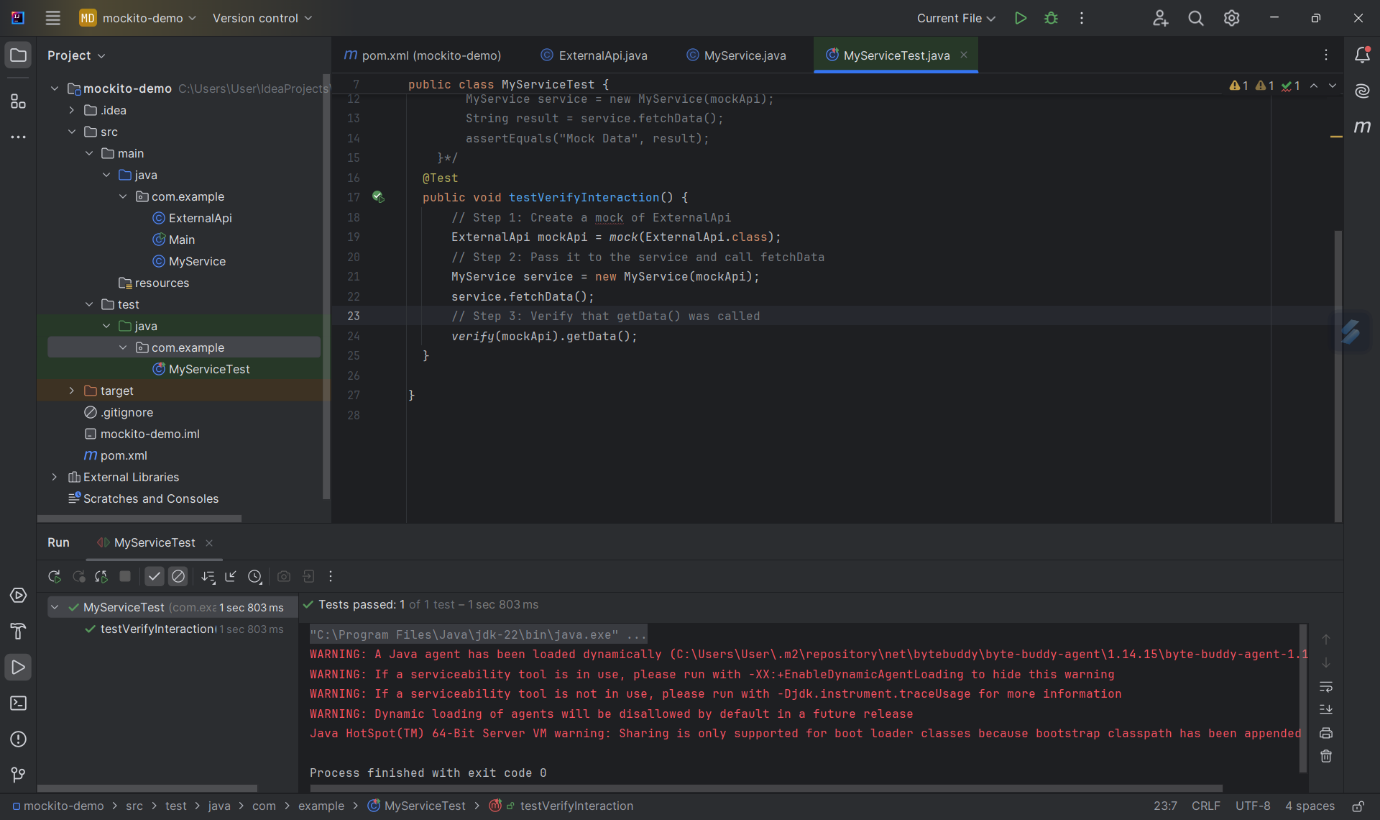
public class MyServiceTest {

@Test

public void testVerifyInteraction() {  
 // Step 1: Create a mock of ExternalApi  
 ExternalApi mockApi = *mock*(ExternalApi.class);  
 // Step 2: Pass it to the service and call fetchData  
 MyService service = new MyService(mockApi);  
 service.fetchData();  
 // Step 3: Verify that getData() was called  
 *verify*(mockApi).getData();  
}

}

**OUTPUT**

****