MOCKITO HANDS-ON EXERCISE:

EXERCISE 1: MOCKING AND STUBBING &

EXERCISE 2: VERFYING INTERACTIONS

//MAIN.JAVA

package org.example;

//TIP To <b>Run</b> code, press <shortcut actionId="Run"/> or

// click the <icon src="AllIcons.Actions.Execute"/> icon in the gutter.

public class Main {

public static void main(String[] args) {

//TIP Press <shortcut actionId="ShowIntentionActions"/> with your caret at the highlighted text

// to see how IntelliJ IDEA suggests fixing it.

System.out.printf("Hello and welcome!");

for (int i = 1; i <= 5; i++) {

//TIP Press <shortcut actionId="Debug"/> to start debugging your code. We have set one <icon src="AllIcons.Debugger.Db\_set\_breakpoint"/> breakpoint

// for you, but you can always add more by pressing <shortcut actionId="ToggleLineBreakpoint"/>.

System.out.println("i = " + i);

}

}

}

// EXTERNAL API

public interface ExternalApi {

String getData();

}

//MY SERVICE

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

//MY SERVICE TEST1

import static org.mockito.Mockito.\*;

import static org.junit.jupiter.api.Assertions.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest1 {

@Test

public void testExternalApi() {

ExternalApi mockApi = Mockito.mock(ExternalApi.class);

when(mockApi.getData()).thenReturn("Mock Data");

MyService service = new MyService(mockApi);

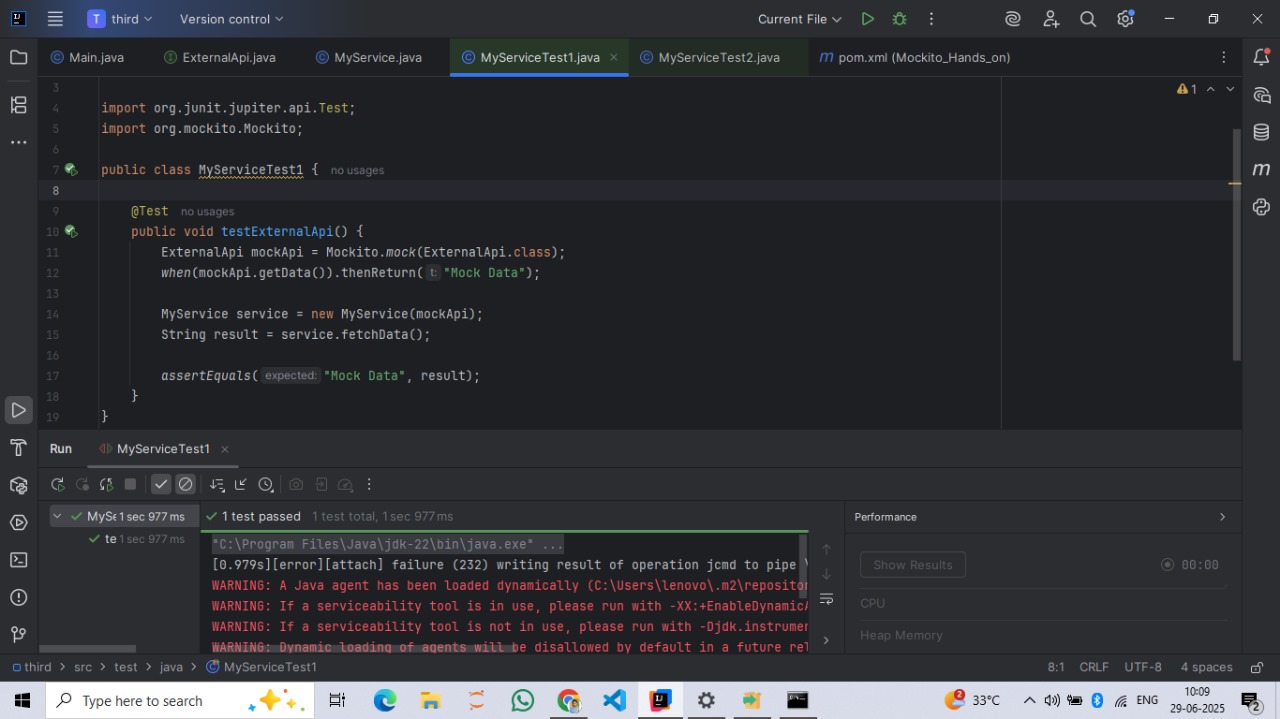
String result = service.fetchData();

assertEquals("Mock Data", result);

}

}

OUTPUT:



//MY SERVICE TEST 2

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest2 {

@Test

public void testVerifyInteraction() {

ExternalApi mockApi = Mockito.mock(ExternalApi.class);

MyService service = new MyService(mockApi);

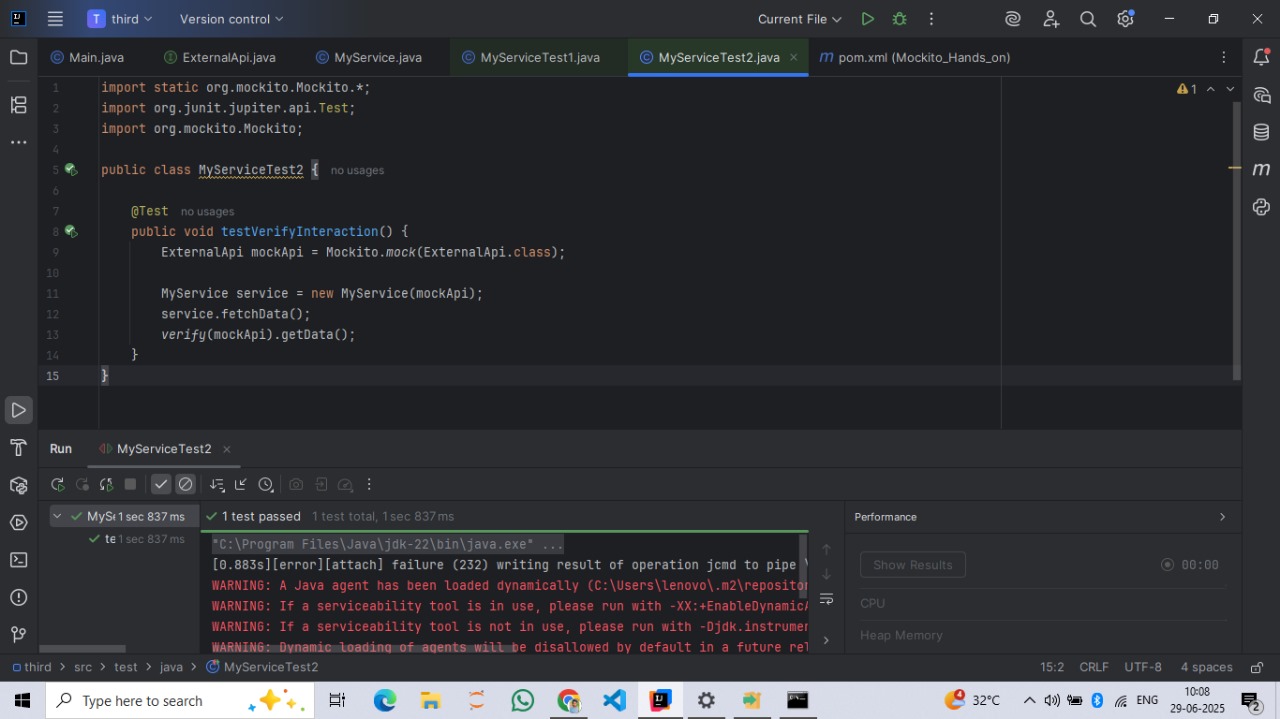
service.fetchData();

verify(mockApi).getData();

}

}

OUTPUT:



//POM FILE

<?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>org.example</groupId>

<artifactId>Mockito\_Hands\_on</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<maven.compiler.source>21</maven.compiler.source>

<maven.compiler.target>21</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>5.4.1</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>

</project>