**JAVA FSE WEEK 2**

**Mockito**

**Mandatory Hands-On Exercises**

**Exercise 1: Mocking and Stubbing**

**Scenario:** You need to test a service that depends on an external API. Use Mockito to mock the external API and stub its methods.

MyServiceTest.java:

package example;

import org.junit.Test;

import static org.junit.Assert.\*;

import static org.mockito.Mockito.\*;

public class MyServiceTest {

@Test

public void testExternalApi() {

// Arrange: create a mock of the external API

ExternalApi mockApi = mock(ExternalApi.class);

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

// Act: pass the mock to MyService and call fetchData()

MyService service = new MyService(mockApi);

String result = service.fetchData();

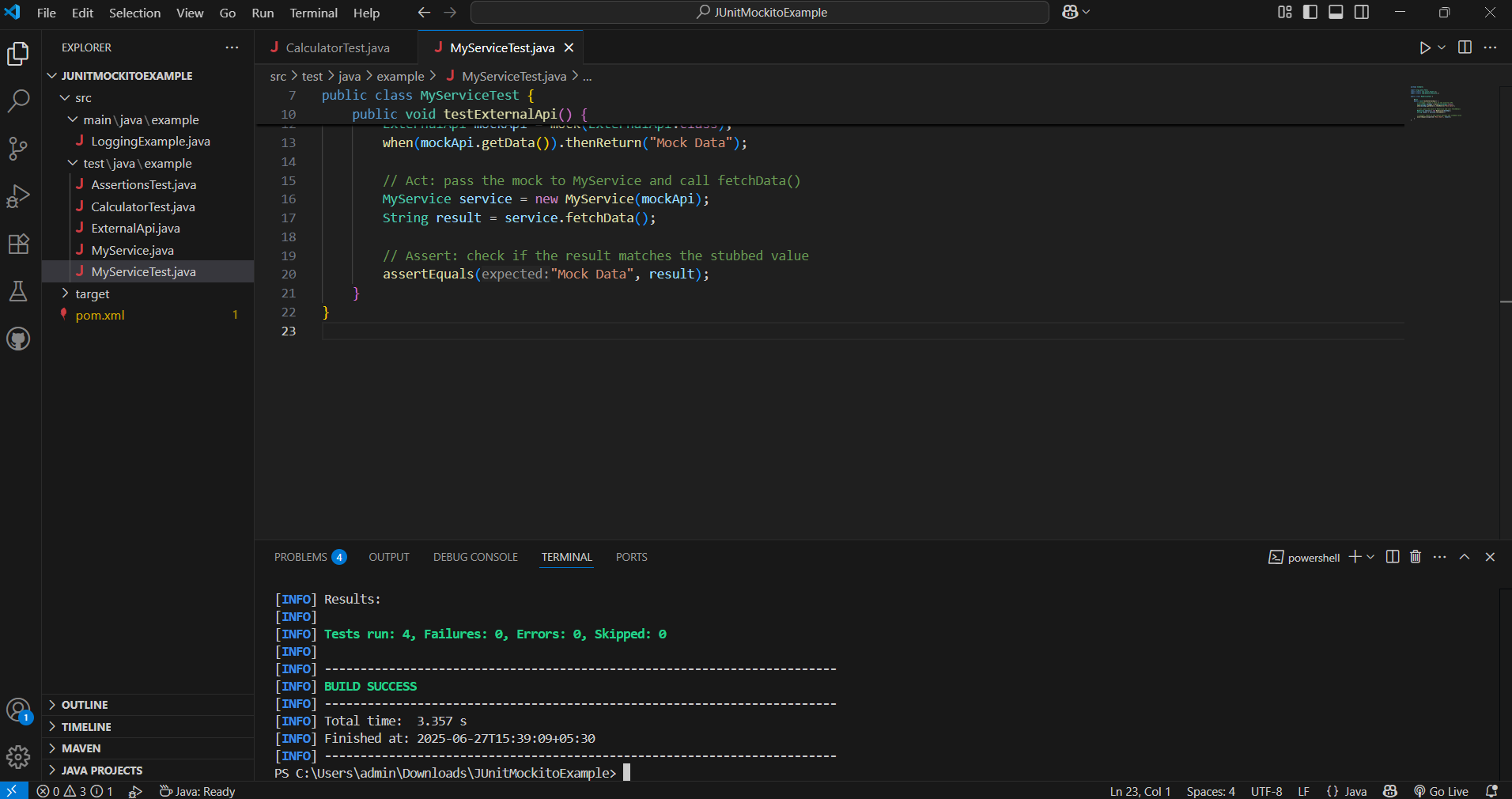
// Assert: check if the result matches the stubbed value

assertEquals("Mock Data", result);

}

}

**Output:**

****

**Exercise 2: Verifying Interactions**

**Scenario:** You need to ensure that a method is called with specific arguments.

MyServiceTest.java:

package example;

import org.junit.Test;

import static org.mockito.Mockito.\*;

public class MyServiceTest {

@Test

public void testVerifyInteraction() {

// Arrange: create a mock of the external API

ExternalApi mockApi = mock(ExternalApi.class);

// Act: call fetchData() using the mock

MyService service = new MyService(mockApi);

service.fetchData();

// Assert: verify that getData() was called exactly once

verify(mockApi).getData();

}

}

**OUTPUT:**