**Exercise 1:**

**Scenario**: Testing a service class (MyService) that depends on an external API (ExternalApi).

1. Created file named ExternalApi.java, inside the **main** folder.

*package com.example;*

*public class MyService {*

*private final ExternalApi api;*

*public MyService(ExternalApi api) {*

*this.api = api;*

*}*

*public String fetchData() {*

*return api.getData();*

*}*

*}*

1. Created the test file, called MyServicesTest.java, inside **test** folder.
2. Added the Mockito dependencies to the pom.xml file.

*Dependency:*

*<!-- Mockito -->*

*<dependency>*

*<groupId>org.mockito</groupId>*

*<artifactId>mockito-core</artifactId>*

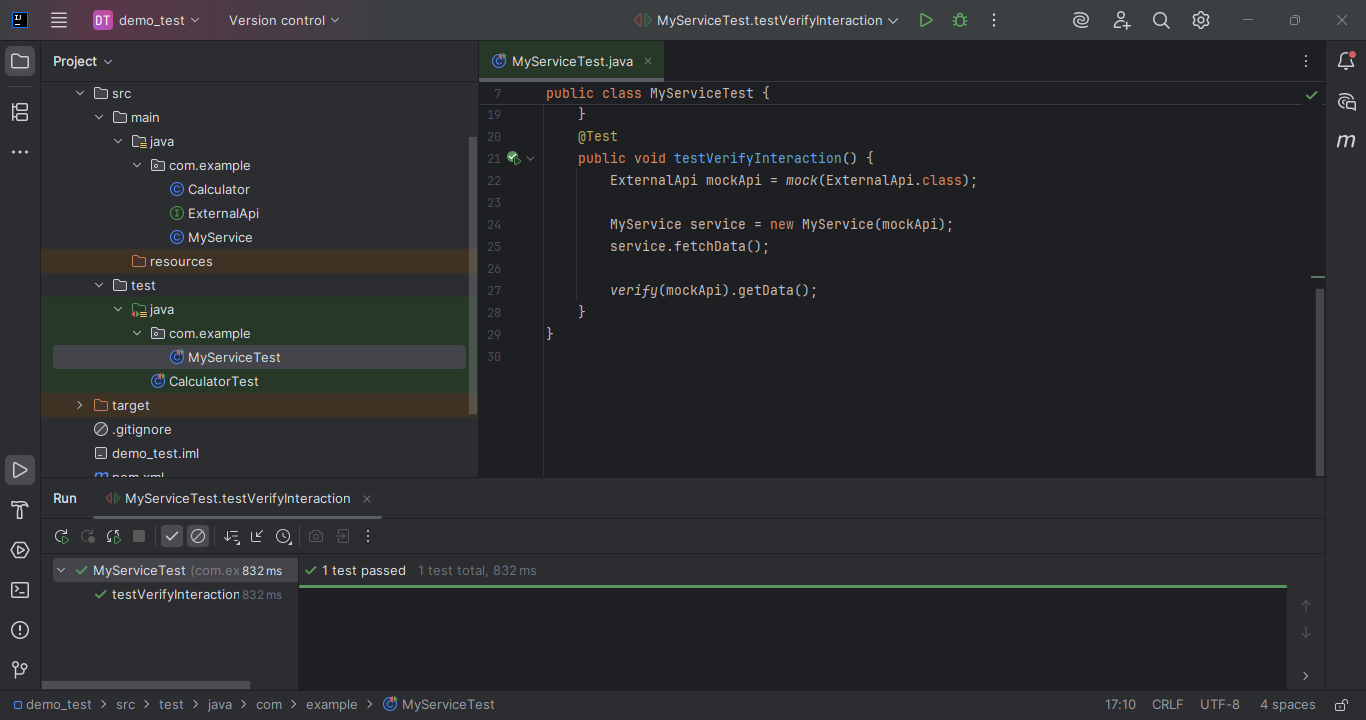
*<version>5.11.0</version>*

*<scope>test</scope>*

*</dependency>*

1. Run the test.

**OUTPUT:**



**Exercise 2: Verifying Interactions**

**Scenario:** Make sure a method on our mock object was actually called — and with the expected arguments.

1. Using the same ExternalApi created above, and MyService. Added a new method to MyServicesTest.java

CODE:

*@Test*

*public void testVerifyInteraction() {*

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

*MyService service = new MyService(mockApi);*

*service.fetchData(); // This should trigger mockApi.getData()*

*verify(mockApi).getData();*

*}*

Last line checks whether the method was called or not.

1. Run the test

**OUTPUT:**

