**TDD using JUnit5 and Mockito**

**Mockito 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.

**Steps:**

1. Create a mock object for the external API.

ExternalApi.java

package example.mockito\_demo;

public interface ExternalApi {

String getData();}

2. Stub the methods to return predefined values.

MyService.java

package example.mockito\_demo;

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

3. Write a test case that uses the mock object.

MyServiceTest.java

package example.mockito\_demo;

import org.junit.jupiter.api.Test;

import static org.mockito.Mockito.\*;

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

public class MyServiceTest {

*@Test*

public void testExternalApi() {

// Arrange - Create mock

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

// Act - Stub method

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

// Inject and test

MyService service = new MyService(mockApi);

String result = service.fetchData();

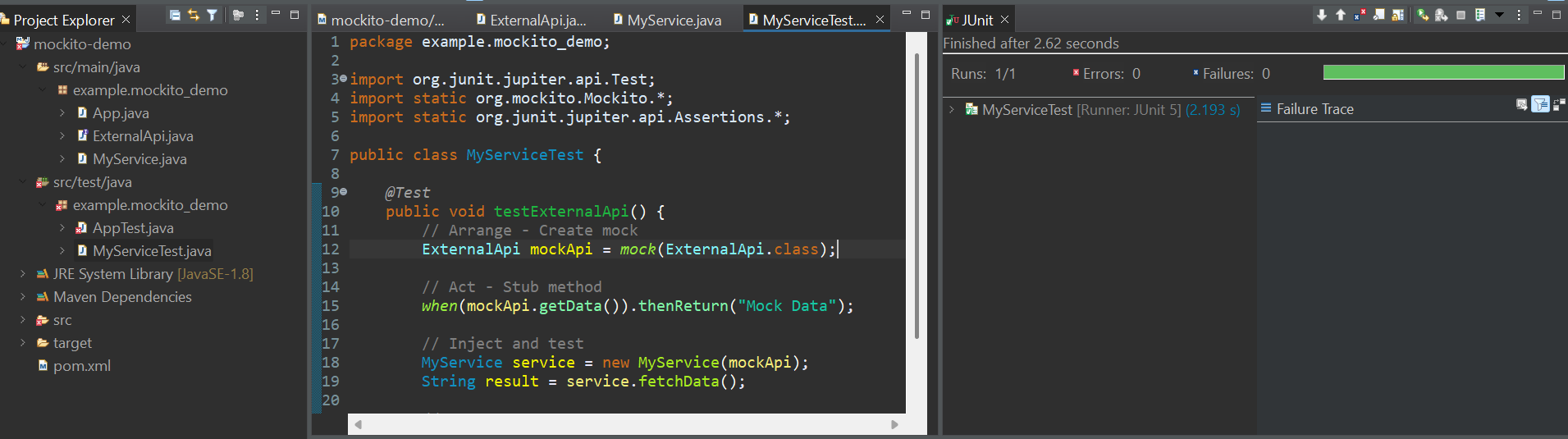
// Assert

*assertEquals*("Mock Data", result);

}

}

**Output:**

****

**Exercise 2: Verifying Interactions**

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

**Steps:**

1. Create a mock object.

ExternalApi.java

package example.mockito\_demo;

public interface ExternalApi {

String getData();}

2. Call the method with specific arguments.

MyService.java

package example.mockito\_demo;

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

3. Verify the interaction.

MyServiceTest.java

package example.mockito\_demo;

import org.junit.jupiter.api.Test;

import static org.mockito.Mockito.\*;

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

public class MyServiceTest {

@Test

public void testVerifyGetDataCalled() {

// Step 1: Create a mock of ExternalApi

ExternalApi mockApi = mock(ExternalApi.class);

// Step 2: Stub getData() to return "Mock Response"

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

// Step 3: Inject mock into service

MyService service = new MyService(mockApi);

// Step 4: Call fetchData()

String result = service.fetchData();

// Step 5: Verify that getData() was called

verify(mockApi).getData();

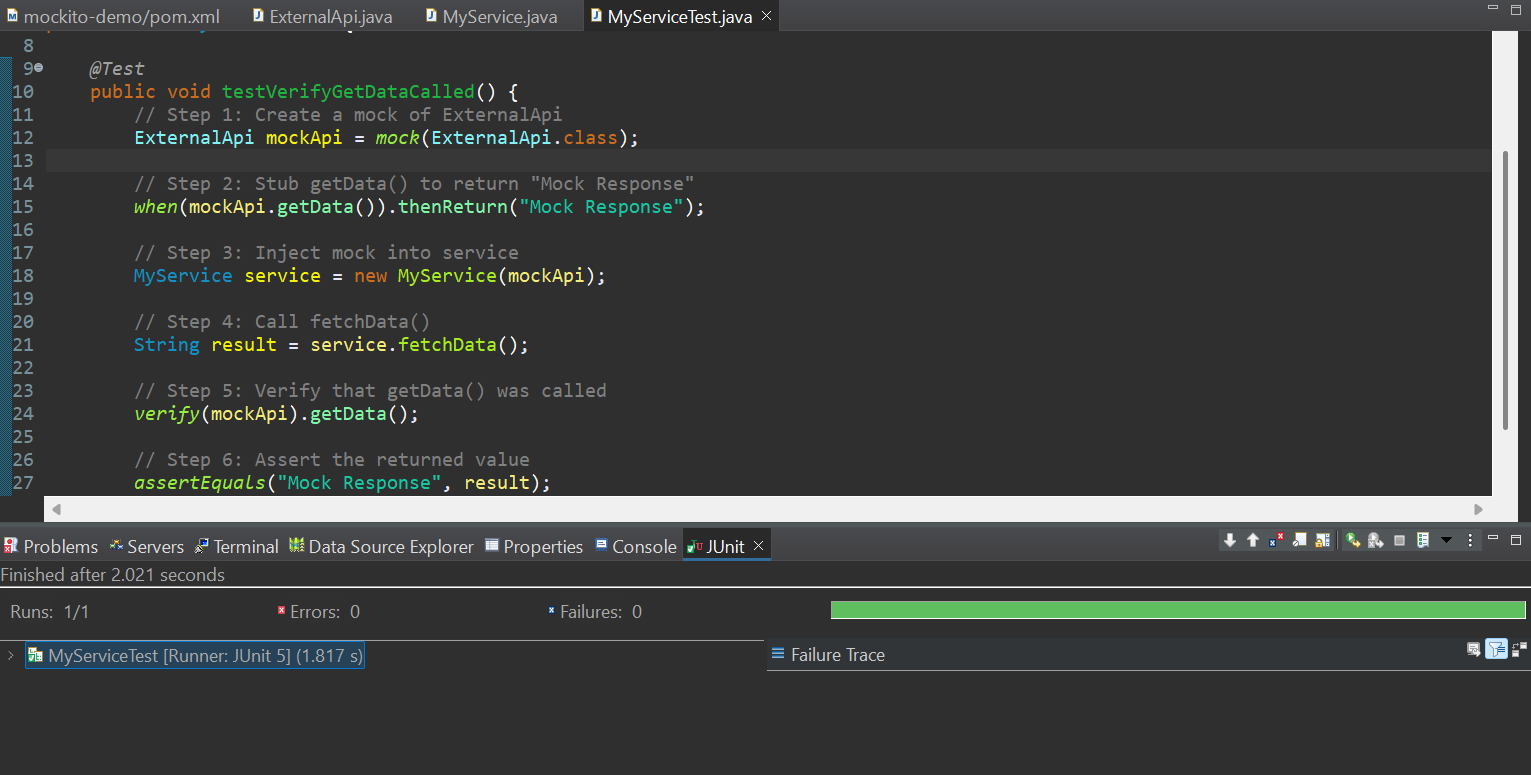
// Step 6: Assert the returned value

assertEquals("Mock Response", result);

}

}

**Output:**

****