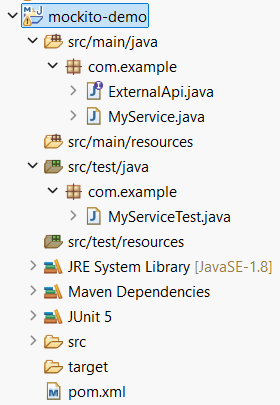
**Exercise 1: Mocking and Stubbing**



**ExternalApi.java**

package com.example;

public interface ExternalApi {

String getData();

}

**MyService.java**

**package** com.example;

**public** **class** MyService {

**private** ExternalApi api;

**public** MyService(ExternalApi api) {

**this**.api = api;

}

**public** String fetchData() {

**return** api.getData();

}

}

**MyServiceTest.java**

package com.example;

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() {

// Step 1: Create mock

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

// Step 2: Stub method

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

// Step 3: Inject into service

MyService service = new MyService(mockApi);

// Step 4: Call and assert

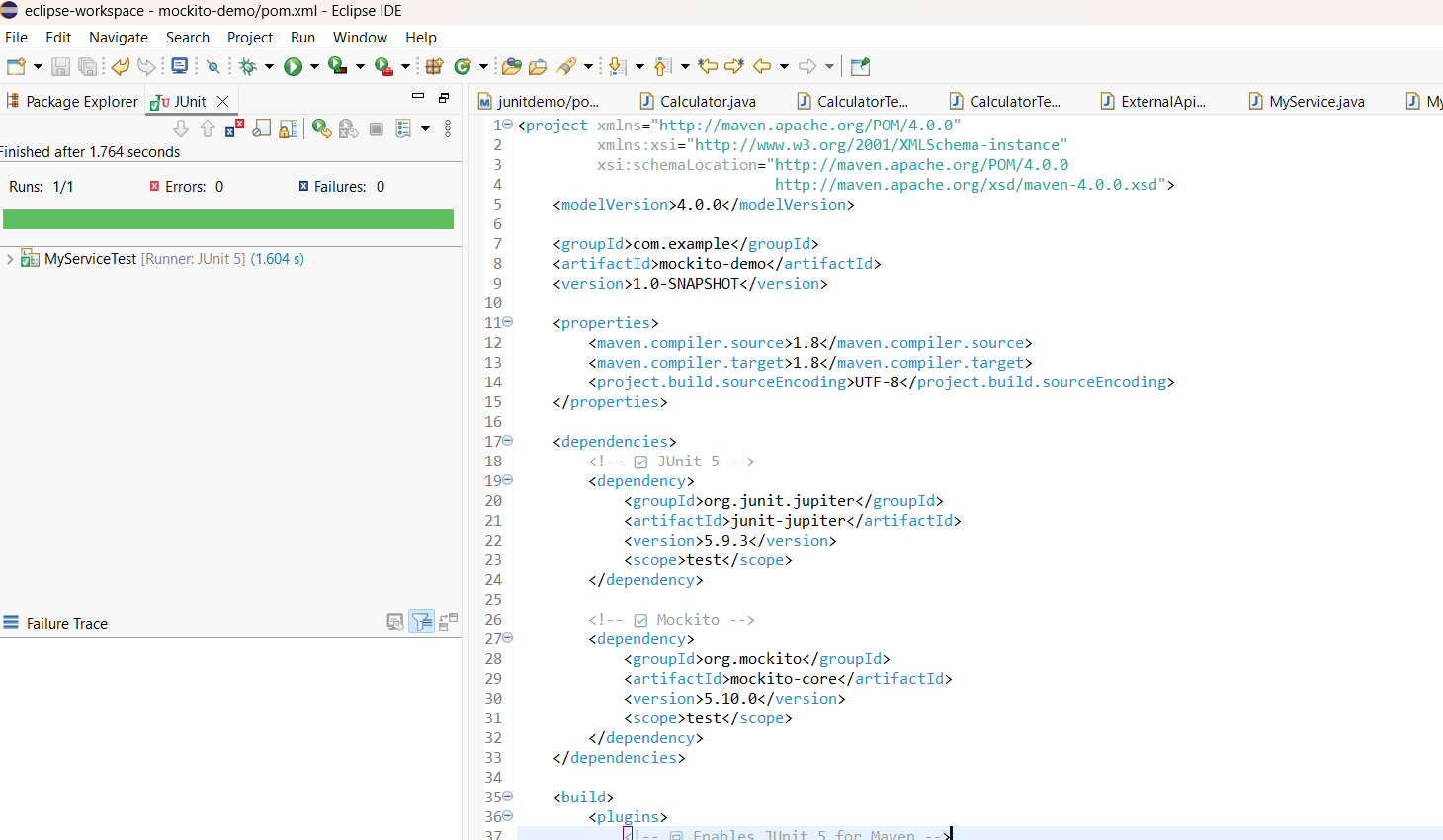
String result = service.fetchData();

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

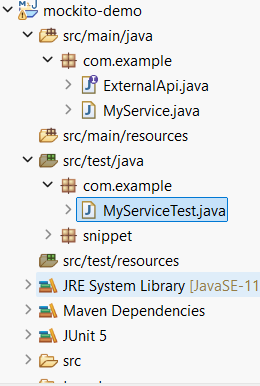
}

}

OUTPUT:



**Exercise 2: Verifying Interactions**

****

**ExternalApi.java**

public interface ExternalApi {

void getData();

}

**MyService.java**

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public void fetchData() {

api.getData(); // This call is what we verify in the test

}

}

**MyServiceTest.java**

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest {

@Test

public void testVerifyInteraction() {

// Step 1: Create a mock object

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

// Step 2: Inject mock into service and call the method

MyService service = new MyService(mockApi);

service.fetchData();

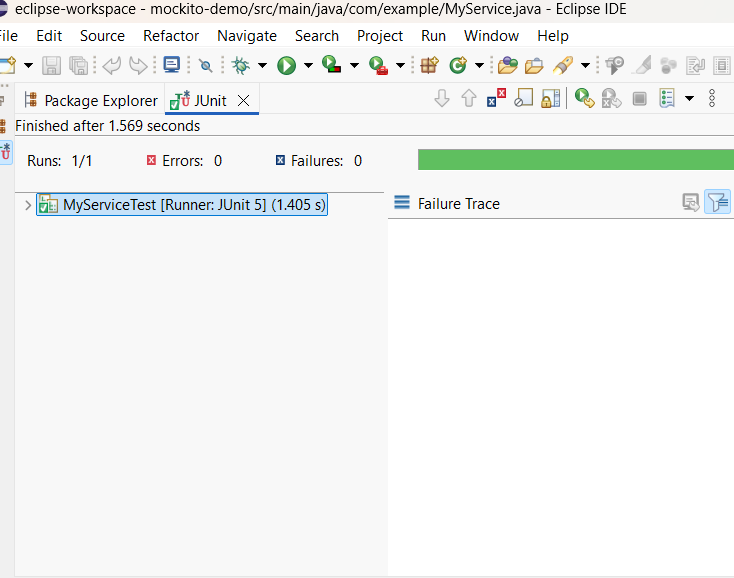
// Step 3: Verify that mockApi.getData() was called

verify(mockApi).getData(); // Test will fail if getData() was not called

}

}

**OUTPUT:**

****