# Week-2

**TDD using JUnit5 and Mockito**

#### Exercise 1: Mocking and Stubbing

#### ****ExternalApi.java****

public interface ExternalApi {

String getData();

}

#### ****MyService.java****

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

#### ****MyServiceTest.java****

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

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

public class MyServiceTest {

@Test

public void testExternalApi() {

// Create mock object

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

// Stub the method to return predefined value

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

// Use the mock in service

MyService service = new MyService(mockApi);

String result = service.fetchData();

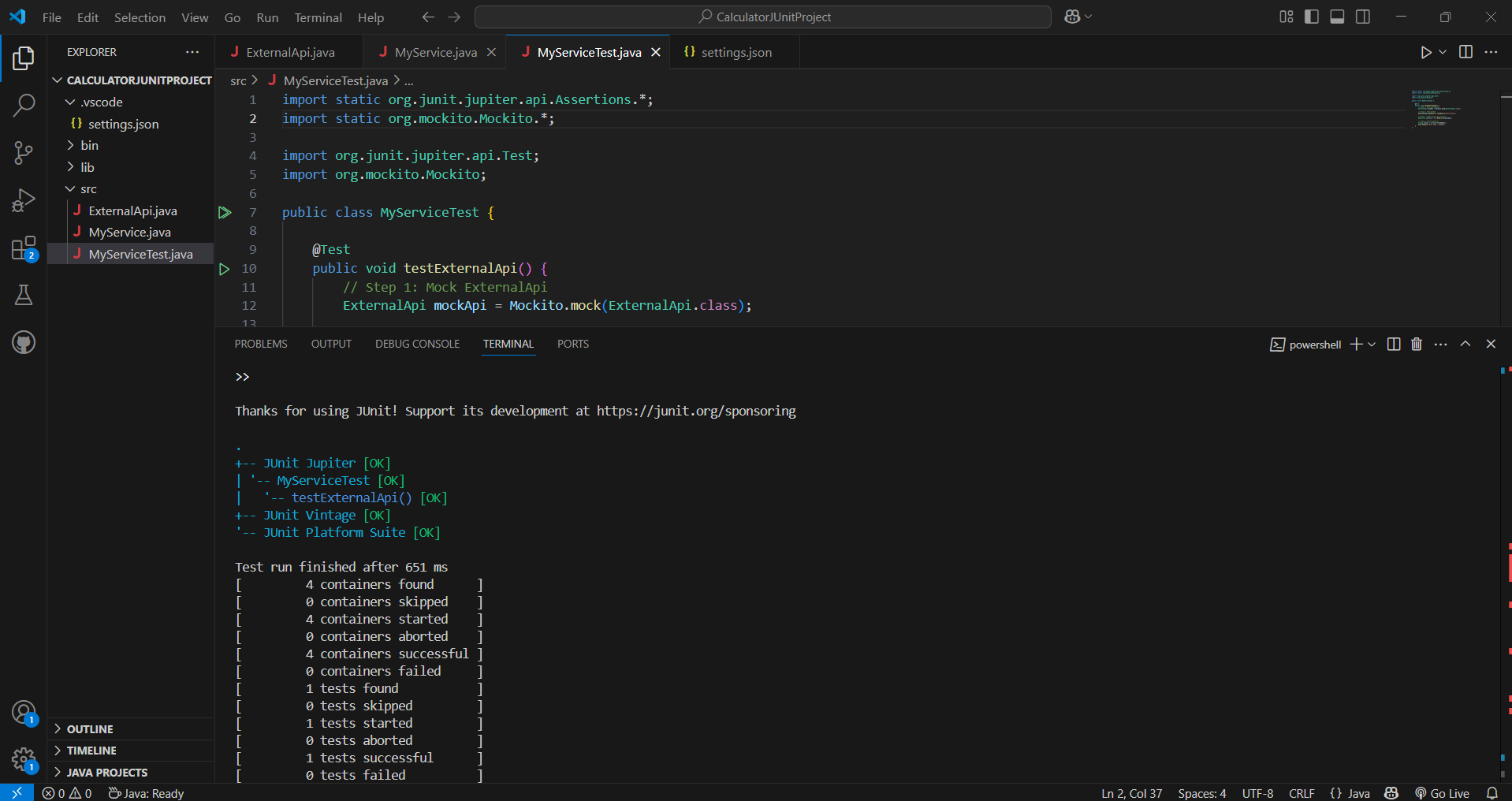
// Verify the result

assertEquals("Mock Data", result);

}

}

**Output:**

****

**Exercise 2: Verifying Interactions**

### ExternalApi.java (Interface)

public interface ExternalApi {

String getData();

}

### MyService.java

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public void fetchData() {

api.getData(); // we're just verifying this is called

}

}

### VerifyInteractionTest.java

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class VerifyInteractionTest {

@Test

public void testVerifyInteraction() {

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

MyService service = new MyService(mockApi);

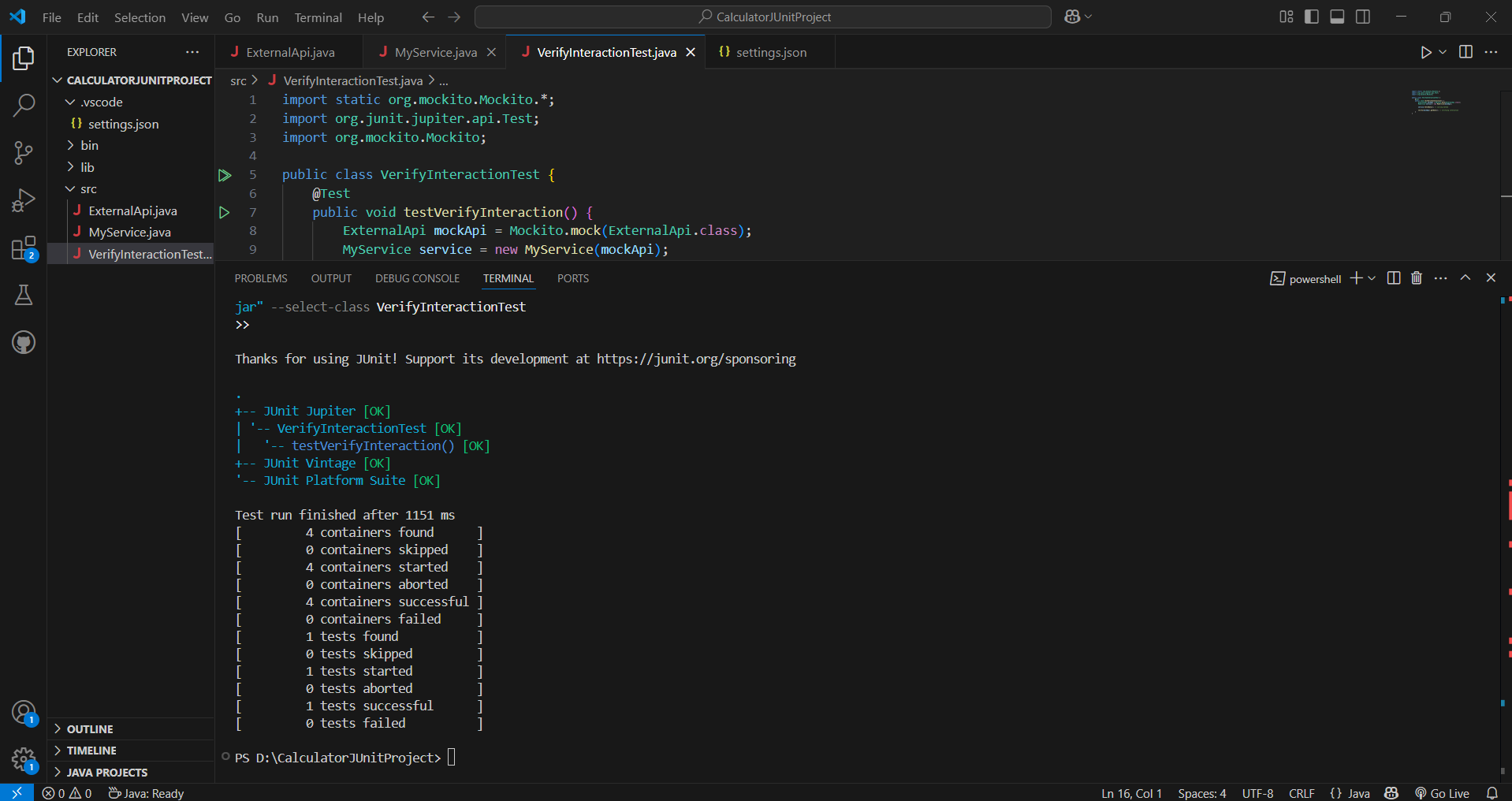
service.fetchData(); // calling method

verify(mockApi).getData(); // verifying interaction

}

}

**Output:**

****