**Exercise 2: Verifying Interactions**

**Scenario:**

You need to verify that a method is called **with specific arguments** or **a specific number of times**.

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

import org.junit.jupiter.api.Test;

public class MyServiceTest {

@Test

public void testVerifyInteraction() {

// Step 1: Create a mock object

ExternalApi mockApi = mock(ExternalApi.class);

// Stub the method (optional, if you care about return value)

when(mockApi.getData()).thenReturn("Mocked");

// Step 2: Call the method through the service

MyService service = new MyService(mockApi);

String result = service.fetchData();

// Step 3: Verify the interaction

verify(mockApi).getData(); // Verifies that getData() was called once

// Optional assertion to confirm the return value

assertEquals("Mocked", result);

}

}