**Exercise 2: Verifying Interactions**

**The Interface – ExternalApi**

// ExternalApi.java

public interface ExternalApi {

String getData();

}

**The Class That Uses It – MyService**

// MyService.java

public class MyService {

private ExternalApi externalApi;

public MyService(ExternalApi externalApi) {

this.externalApi = externalApi;

}

public String fetchData() {

return externalApi.getData(); // We want to verify that this gets called

}

}

**Add Maven Dependencies**

**pom.xml:**

<dependencies>

<!-- JUnit 5 -->

<dependency>

<groupId>org.junit.jupiter</groupId>

<artifactId>junit-jupiter</artifactId>

<version>5.9.3</version>

<scope>test</scope>

</dependency>

<!-- Mockito -->

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>5.10.0</version>

<scope>test</scope>

</dependency>

</dependencies>

**The Test Case – MyServiceTest**

// MyServiceTest.java

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

public class MyServiceTest {

@Test

public void testVerifyInteraction() {

// Step 1: Create a mock object

ExternalApi mockApi = mock(ExternalApi.class);

// Step 2: Inject mock into the service

MyService service = new MyService(mockApi);

// Step 3: Call the method

service.fetchData();

// Step 4: Verify that mockApi.getData() was called

verify(mockApi).getData(); // Passes if called at least once

}

}

**Verifying with Arguments:** verify(mockApi).getData("abc123");

**Output:**

