**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.

2. Stub the methods to return predefined values.

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

Create a maven folder as MockitoDemo –

In that, create a ExternalApi in src/main/java as-

**Program:**

**public** **interface** ExternalApi {

String getData();

}

Then after, create a MyService.java in src/main/java as-

**Program:**

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

**private** ExternalApi api;

**public** MyService(ExternalApi api) {

**this**.api = api;

}

**public** String fetchData() {

**return** api.getData();

}

}

Later, create a MyServiceTest.java in src/test/java as-

**Program:**

import static org.mockito.Mockito.\*;

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

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest {

@Test

public void testFetchDataWithMockApi() {

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

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

MyService service = new MyService(mockApi);

String result = service.fetchData();

assertEquals("Mock Response", result, "Expected mocked data to be returned");

verify(mockApi).getData();

}

}

It is all about the Mocking and Stubbing in Mockito

**Exercise 2: Verifying Interactions**

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

**Steps:**

1. Create a mock object.

2. Call the method with specific arguments.

3. Verify the interaction.

I have created a VerifyingInteraction named maven project.

In that, I have created ExternalApi.java in src/main/java as-

**Program-**

**public** **interface** ExternalApi {

String getData();

}

Then after, I have created MyService.java as-

**Program-**

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

**private** ExternalApi api;

**public** MyService(ExternalApi api) {

**this**.api = api;

}

**public** String fetchData() {

**return** api.getData();

}

}

Later, I have created MyServicTest.java in src/test/java as-

**Program:**

import org.junit.jupiter.api.Test;

import static org.mockito.Mockito.\*;

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

import org.mockito.Mockito;

@SuppressWarnings("unused")

public class MyServiceTest {

@Test

public void testVerifyInteraction() {

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

MyService service = new MyService(mockApi);

service.fetchData();

verify(mockApi).getData();

}

}

It is all about Verifying Interactions in Mockito