**Mockito Exercises**

**Exercise 1: Mocking and Stubbing**

ExternalApi.java

public interface ExternalApi {

String getData();

}

MyService.java

public class MyService {

private final ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.assertEquals;

import static org.mockito.Mockito.\*;

public class MyServiceTest {

@Test

public void testFetchData() {

ExternalApi mockApi = mock(ExternalApi.class);

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

MyService service = new MyService(mockApi);

String result = service.fetchData();

assertEquals("Mocked Response", result);

}

}

MyServiceTest.java

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.assertEquals;

import static org.mockito.Mockito.\*;

public class MyServiceTest {

@Test

public void testFetchData() {

ExternalApi mockApi = mock(ExternalApi.class);

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

MyService service = new MyService(mockApi);

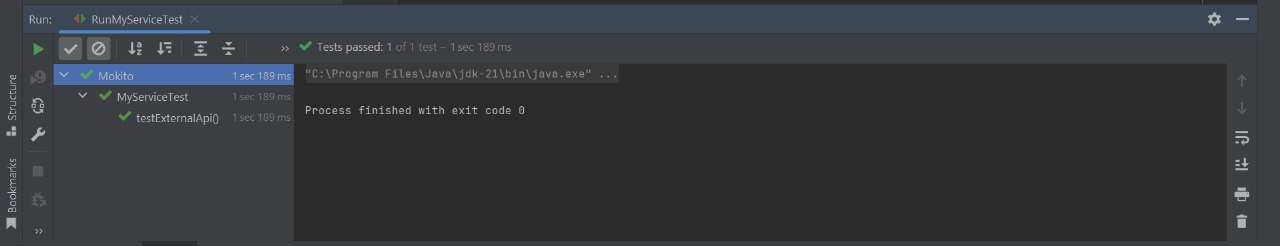
String result = service.fetchData();

assertEquals("Mocked Response", result);

}

}

**Output:-**

****

**Exercise 2: Verifying Interactions**

ExternalApi.java

public interface ExternalApi {

String getData();

}

MyService.java

public class MyService {

private final ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public void fetchData() {

api.getData(); // This should be verified in the test

}

}

MyServiceTest.java

import org.junit.jupiter.api.Test;

import static org.mockito.Mockito.\*;

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

public class MyServiceTest {

@Test

public void testVerifyInteraction() {

ExternalApi mockApi = mock(ExternalApi.class);

MyService service = new MyService(mockApi);

service.fetchData();

verify(mockApi).getData(); // Passes if getData() was called exactly once

}

}

Output:-

