**Mockito Hands-On Exercises**

**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.

**Notifier.java**

package com.example.junit\_demo;

public interface Notifier {

String getData();

}

**NotificationServiceTest.java**

package com.example.junit\_demo;

import com.example.junit\_demo.Notifier;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

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

import static org.mockito.Mockito.*when*;

public class NotificationServiceTest {

*@Test*

public void testExternalApi() {

Notifier mockApi = Mockito.*mock*(Notifier.class);

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

NotificationService service = new NotificationService(mockApi);

String result = service.fetchData();

*assertEquals*("Mock Data", result);

}

}

**NotificationService.java**

package com.example.junit\_demo;

import com.example.junit\_demo.Notifier;

public class NotificationService {

private Notifier notifier;

public NotificationService(Notifier notifier) {

this.notifier = notifier;

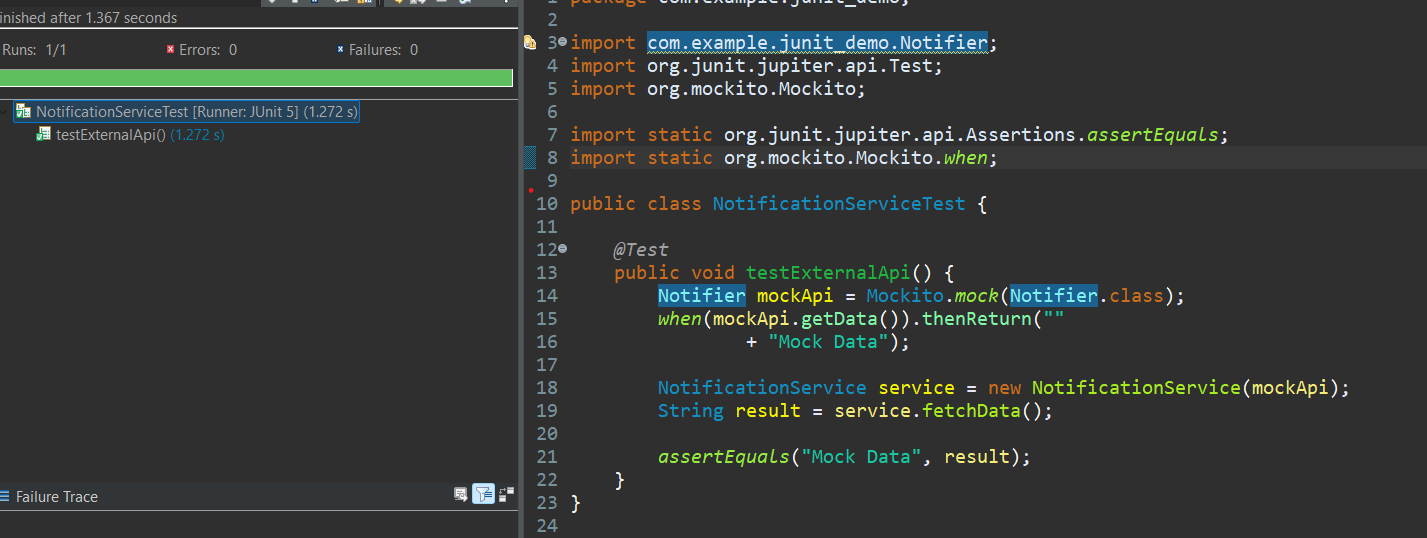
}

public String fetchData() {

return notifier.getData();

}

}



**Exercise 2: Verifying Interactions**

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

**ExternalApi.java**

package com.example.junit\_demo;

public interface ExternalApi {

String getData();

}

**MyServiceTest.java**

package com.example.junit\_demo;

import org.junit.jupiter.api.Test;

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

import static org.mockito.Mockito.\*;

public class MyServiceTest {

*@Test*

public void testExternalApi() {

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

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

MyService service = new MyService(mockApi);

String result = service.fetchData();

*assertEquals*("Mock Data", result);

}

*@Test*

public void testVerifyInteraction() {

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

MyService service = new MyService(mockApi);

service.fetchData(); // Should call getData()

// ✅ Verifies that getData() was called once

*verify*(mockApi).getData();

}

}

**MyService.java**

package com.example.junit\_demo;

public class MyService {

private ExternalApi externalApi;

public MyService(ExternalApi externalApi) {

this.externalApi = externalApi;

}

public String fetchData() {

return externalApi.getData();

}

}

