**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**

**1.ExternalApi.java**

public interface ExternalApi {

String getData();

}

**2.MyService.java**

public class MyService {

private ExternalApi externalApi;

public MyService(ExternalApi externalApi) {

this.externalApi = externalApi;

}

public String fetchData() {

// Calls external API method

return externalApi.getData();

}

}

**3.MyServiceTest.java**

import org.junit.jupiter.api.Test;

import static org.mockito.Mockito.\*;

public class MyServiceTest {

@Test

public void testVerifyInteraction() {

// 1. Create mock

ExternalApi mockApi = mock(ExternalApi.class);

// 2. Inject into service

MyService service = new MyService(mockApi);

// 3. Call the method

service.fetchData();

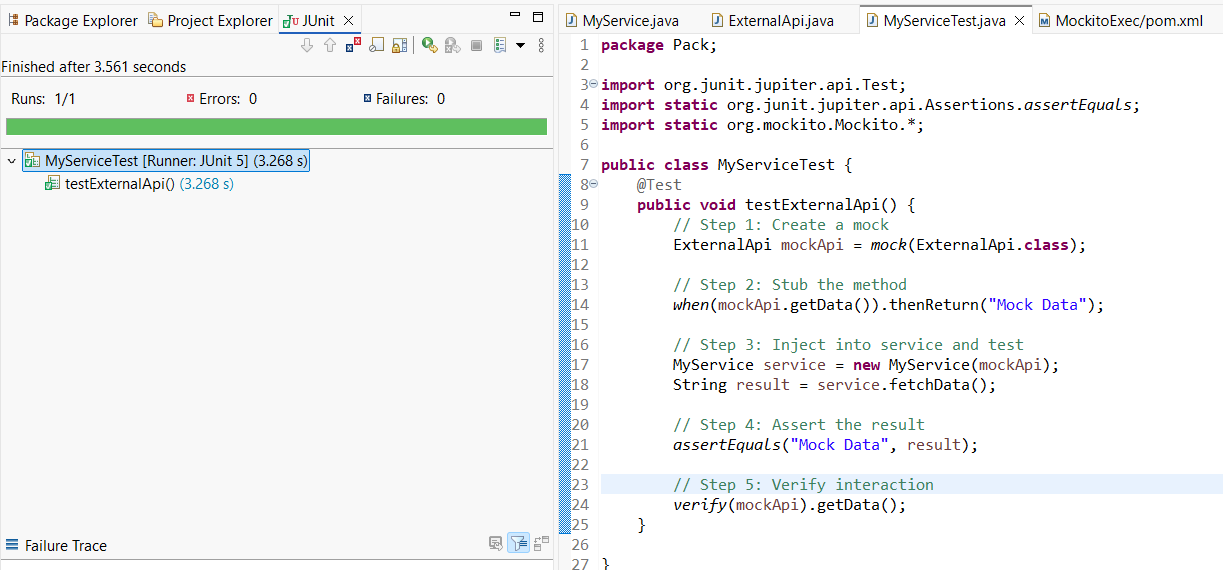
// 4. Verify interaction

verify(mockApi).getData();

}

}

**Output:**



**Exercise 2: Verifying Interactions**

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

**1.ExternalApi.java**

package Pack;

public interface ExternalApi {

String getData(String userId);

}

**2.Service.java**

package Pack;

public class MyService {

private ExternalApi externalApi;

public MyService(ExternalApi externalApi) {

this.externalApi = externalApi;

}

public String fetchData() {

return externalApi.getData("admin"); // Depends on ExternalApi

}

}

**3.ServiceTest.java**

package Pack;

import org.junit.jupiter.api.Test;

import static org.mockito.Mockito.\*;

public class MyServiceTest {

@Test

public void testExternalApi() {

// Step 1: Create a mock

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

// Step 2: Stub the method

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

// Step 3: Inject into service and test

MyService service = new MyService(mockApi);

String result = service.fetchData();

*verify*(mockApi).getData("admin");

System.*out*.println("Result: " + result);

}

}

**Output:**

