**Mockito Hands-On Exercises**

**Exercise 1: Mocking and Stubbing**

**To test a service that depends on an external API. Use Mockito to mock the external API and stub its methods.**

**Add Mockito and JUnit Dependencies**

**pom.xml:**

<dependencies>

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>5.11.0</version>

<scope>test</scope>

</dependency>

<dependency>

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

<artifactId>junit-jupiter</artifactId>

<version>5.10.2</version>

<scope>test</scope>

</dependency>

</dependencies>

**Inside src/main/java/com.example.mockito\_exercise**

1. **Select Interface**

**ExternalApi.java**

package com.example.mockito\_exercise;

public interface ExternalApi {

String getData();

}

1. **Select Class**

**MyService.java**

package com.example.mockito\_exercise;

public class MyService {

private ExternalApi externalApi;

public MyService(ExternalApi externalApi) {

this.externalApi = externalApi;

}

public String fetchData() {

return externalApi.getData();

}

}

**Inside src/test/java/com.example.mockito\_exercise**

**Select JUnit Test Case**

**MyServiceTest.java**

package com.example.mockito\_exercise;

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

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest {

@Test

public void testExternalApi() {

// Create mock for ExternalApi

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

// Stub the method to return predefined value

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

// Inject mock into service

MyService service = new MyService(mockApi);

// Call method under test

String result = service.fetchData();

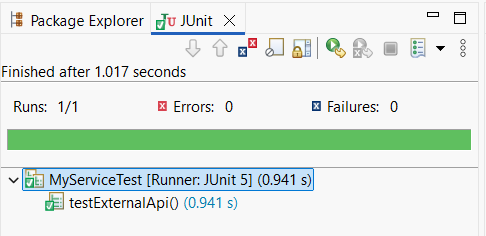
// Assert expected result

assertEquals("Mock Data", result);

}

}

**OUTPUT:**

****