**Mockito Hands-On Exercises**

**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.

Solution Code:

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest {

@Test

public void testVerifyInteraction() {

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

MyService service = new MyService(mockApi);

service.fetchData();

verify(mockApi).getData();

} }

**CODE:**

ExternalApi.java

package org.example;  
  
public interface ExternalApi {  
 String getData();  
}

MyService.java

package org.example;  
  
public class MyService {  
 private ExternalApi api;  
 public MyService(ExternalApi api) {  
 this.api = api;  
 }  
 public String fetchData() {  
 return api.getData();  
 }  
}

MyServiceTest.java

package org.example;  
import org.junit.jupiter.api.Test;  
import org.mockito.Mockito;  
import static org.mockito.Mockito.\*;  
public class MyServiceTest {  
  
 @Test  
 public void testVerifyInteraction() {  
 ExternalApi mockApi = Mockito.*mock*(ExternalApi.class);  
 MyService service = new MyService(mockApi);  
 service.fetchData();  
 *verify*(mockApi).getData();  
 System.*out*.println("Verified: mockApi.getData() was called.");  
 }  
}

**OUTPUT:**

