**TDD using JUnit5 and Mockito**

**Mockito 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.

***// create a External API interface- ExternalApi.java***

package mockito;

public interface ExternalApi {

String getData();

}

***//create a class- MyService.java***

package mockito;

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

**//create a class - MyServiceTest.java**

package mockito;

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() {

*//Step 1: Create a mock object for the external API*

ExternalApi mockApi = mock(ExternalApi.class);

*//Step 2: Stub the method to return a predefined value*

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

*//Step 3: Use the mock object in the service*

MyService service = new MyService(mockApi);

String result = service.fetchData();

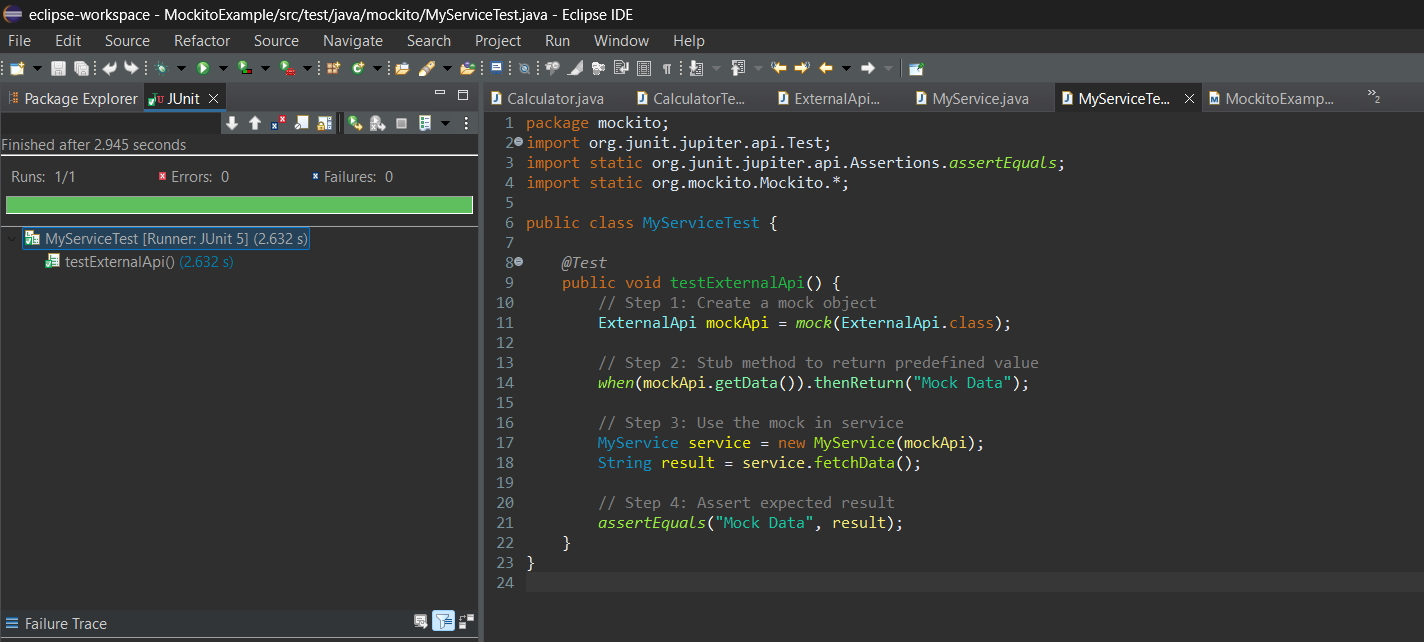
*//Step 4: Assert that the result matches the stubbed value*

assertEquals("Mock Data", result);

}

}

**Output:**



**Exercise 2: Verifying Interactions**

Scenario:

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

***// create a External API interface- ExternalApi.java***

package mockito;

public interface ExternalApi {

String getData();

}

***//create a class- MyService.java***

package mockito;

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

***//create a class - MyServiceTest.java***

package mockito;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

import static org.mockito.Mockito.\*;

public class MyServiceTest {

@Test

public void testVerifyInteraction() {

*// Step 1: Create a mock object*

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

*// Step 2: Call the method with specific arguments*

MyService service = new MyService(mockApi);

service.fetchData(); *// internally calls mockApi.getData()*

*// Step 3: Verify the interaction*

verify(mockApi).getData();

}

}

**Output:**

