Mockito Exercise

Exercise 1: Mockito and Stubbing

Updated pom.xml:

<dependencies>

<!-- JUnit 4 -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

<!-- Mockito -->

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>3.12.4</version>

<scope>test</scope>

</dependency>

</dependencies>

ExternalApi.java:

package com.example;

public interface ExternalApi {

String getData();

}

MyService.java:

package com.example;

public class MyService {

private ExternalApi externalApi;

public MyService(ExternalApi externalApi) {

this.externalApi = externalApi;

}

public String fetchData() {

return externalApi.getData();

}

}

**MyServiceTest.java:**

package com.example;

import org.junit.Test;

import static org.junit.Assert.\*;

import static org.mockito.Mockito.\*;

public class MyServiceTest {

*@Test*

public void testExternalApi() {

// Step 1: Create mock object

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

// Step 2: Stub method to return predefined value

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

// Step 3: Use mock in the service

MyService service = new MyService(mockApi);

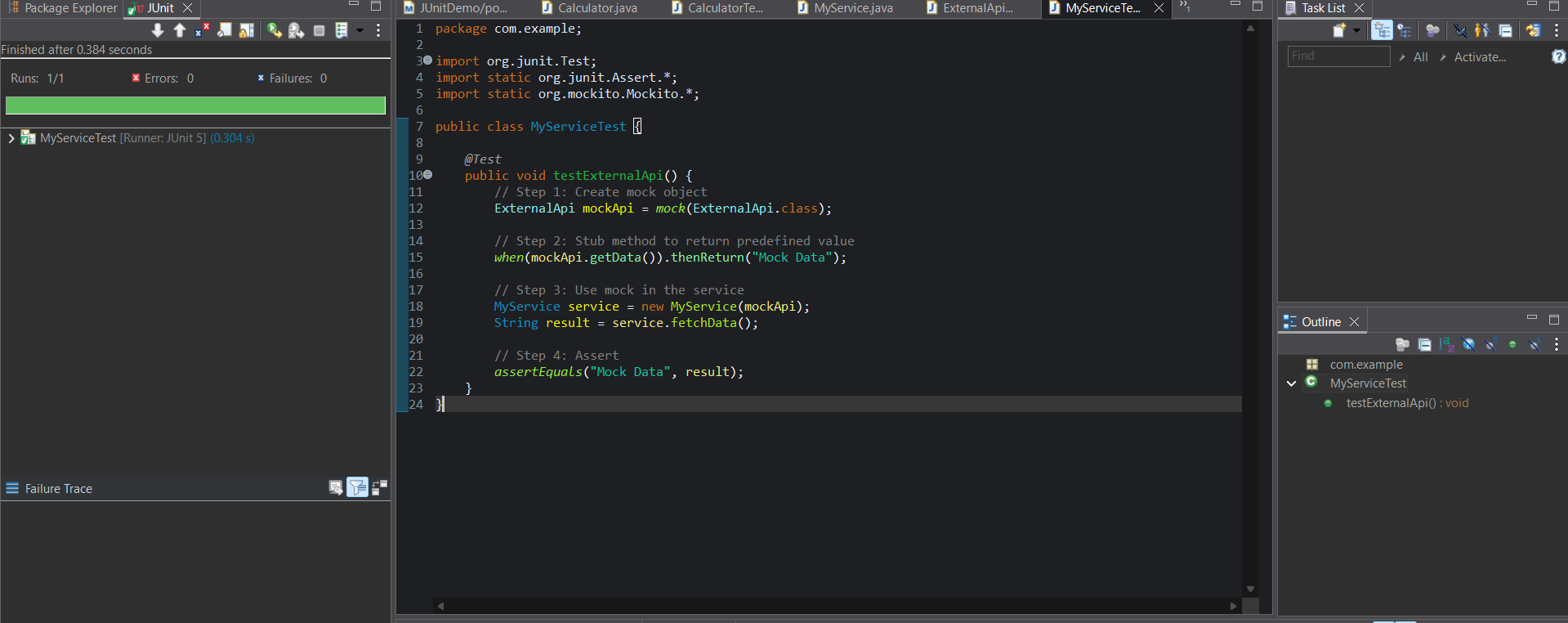
String result = service.fetchData();

// Step 4: Assert

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

}

}



Exercise2: Verifying Interactions:

MyServiceTest2.java:

package com.example;

import org.junit.Test;

import static org.mockito.Mockito.\*;

public class MyServiceTest2 {

@Test

public void testVerifyInteraction() {

// Step 1: Create mock

ExternalApi mockApi = mock(ExternalApi.class);

// Step 2: Inject mock into service

MyService service = new MyService(mockApi);

// Step 3: Call the method

service.fetchData();

// Step 4: Verify interaction

verify(mockApi).getData();

}

}

