**Cognizant Digital Nurture 4.0**

**Name: Siri Chandana Chittipolu**

**Email:** [**sirichittipolu11@gmail.com**](mailto:sirichittipolu11@gmail.com)

**Superset ID: 6386277**

**Mandatory 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.**

**Solution:**

ExternalApi.java :

package com.example.demo;

public interface ExternalApi {

String getData();

}

MyService.java:

package com.example.junit\_demo2;

import com.example.demo.ExternalApi;

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

MyServiceTest.java:

package com.example.junit\_demo2;

import org.junit.jupiter.api.Test;

import com.example.demo.ExternalApi;

import static org.mockito.Mockito.\*;

import static org.junit.jupiter.api.Assertions.\*;

public class MyServiceTest {

*@Test*

public void testExternalApi() {

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

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

MyService service = new MyService(mockApi);

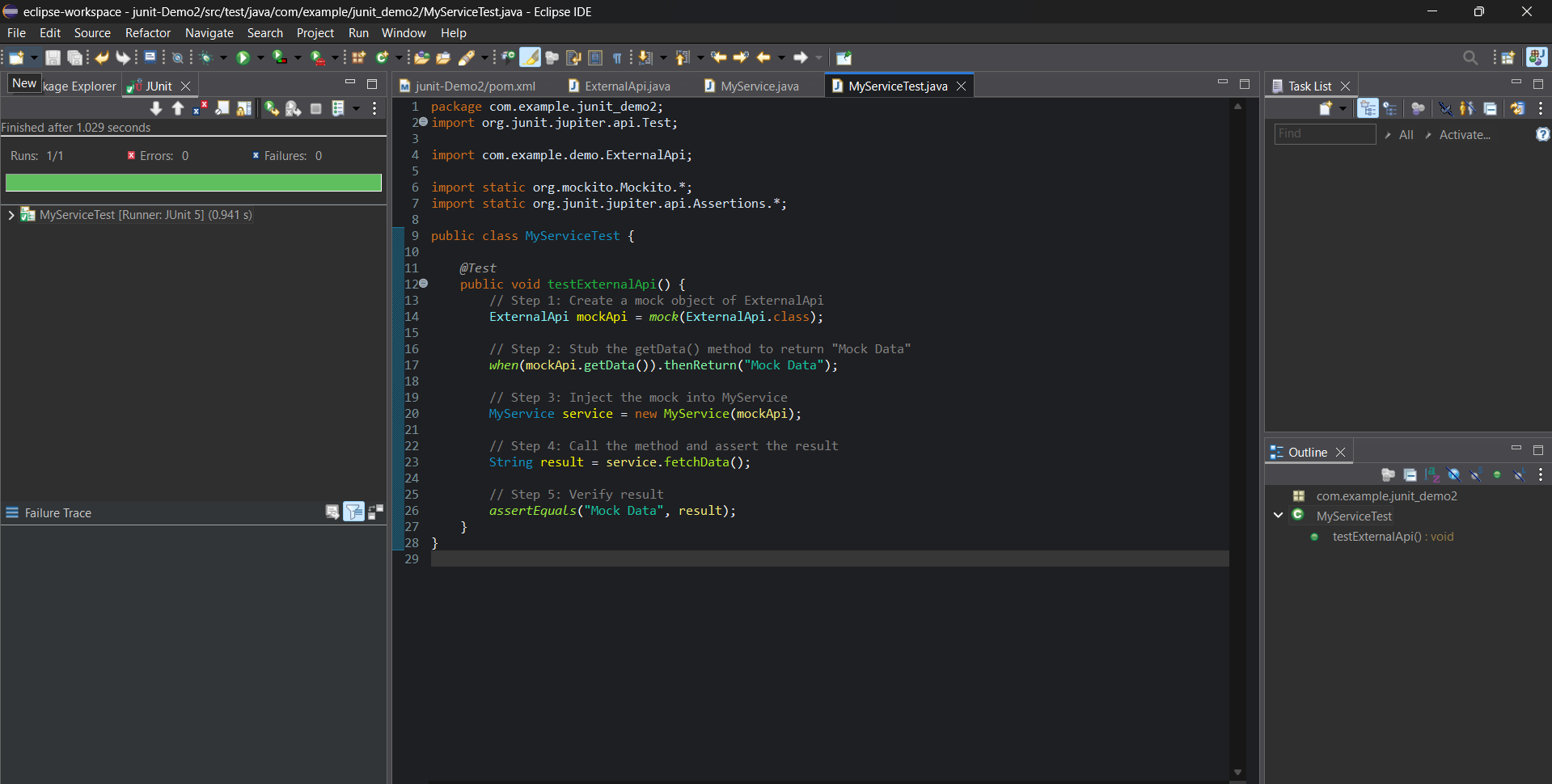
String result = service.fetchData();

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

}

}

**Output:**



**Exercise 2: Verifying Interactions**

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

**Solution:**

ExternalApi.java :

package com.example.demo;

public interface ExternalApi {

String getData();

}

MyService.java:

package com.example.junit\_demo2;

import com.example.demo.ExternalApi;

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

MyServiceTest.java:

package com.example.junit\_demo2;

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import com.example.demo.ExternalApi;

public class MyServiceTest {

*@Test*

public void testVerifyInteraction() {

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

MyService service = new MyService(mockApi);

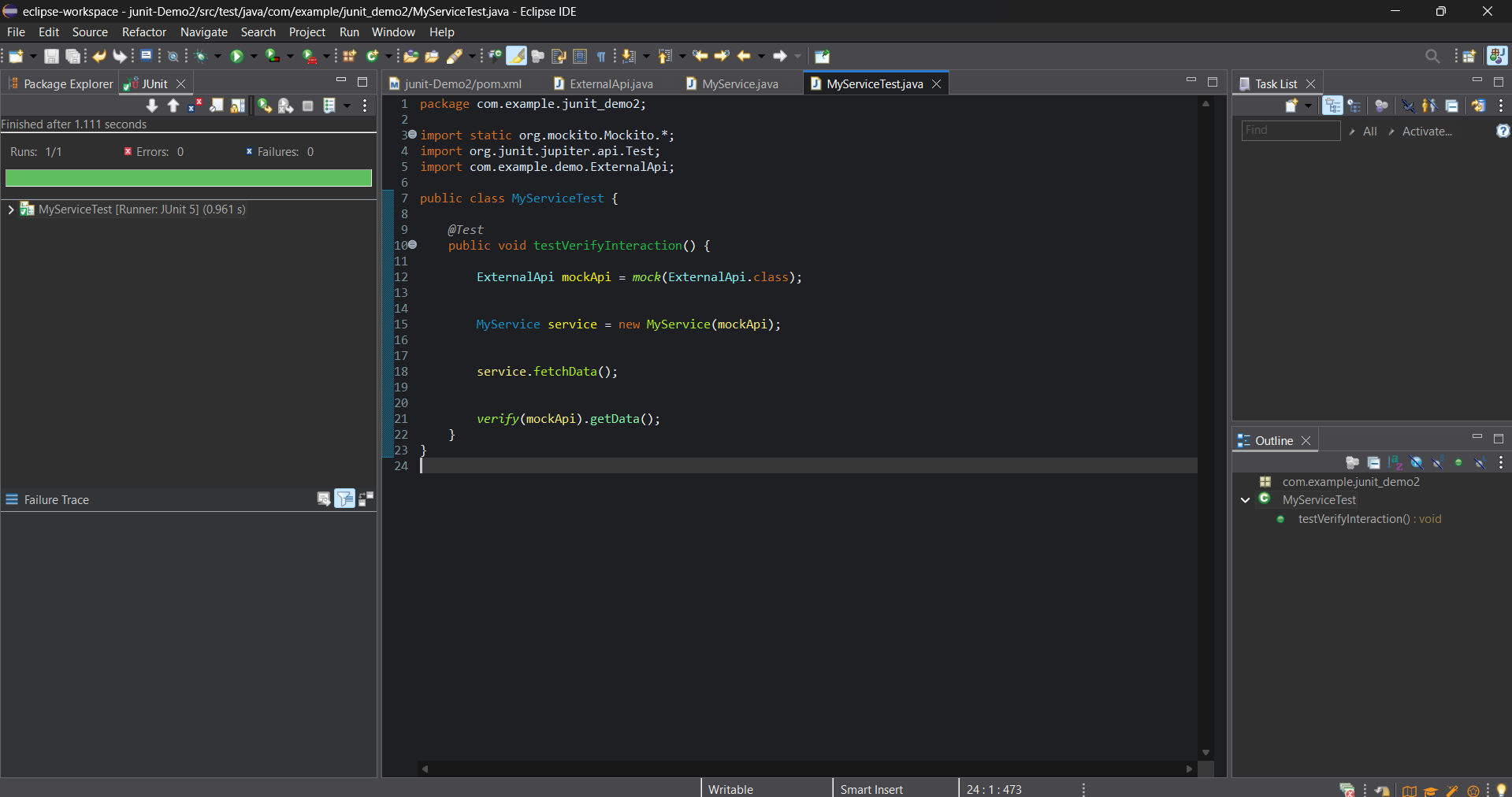
service.fetchData();

*verify*(mockApi).getData();

}

}

* We first create a mock object of the ExternalApi interface using Mockito.
* Then, we pass this mock into the MyService class and call its method.
* Finally, we use verify() to check if the expected method getData() was called on the mock.

****