**WEEK 2 – ASSIGNMENT**

**Superset ID:** 6390124

**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.

**Steps:**

1. Create a mock object for the external API.

2. Stub the methods to return predefined values.

3. Write a test case that uses the mock object.

**Solution Code:**

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest {

@Test

public void testExternalApi() {

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

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

MyService service = new MyService(mockApi);

String result = service.fetchData();

assertEquals("Mock Data", result);

}

}

**MyService.java**

package com.example;

public class MyService {

    private final ExternalApi externalApi;

    public MyService(ExternalApi externalApi) {

        this.externalApi = externalApi;

    }

    public String fetchData() {

        return externalApi.getData();

    }

}

**ExternalApi.java**

package com.example;

public interface ExternalApi {

    String getData();

}

**MyServiceTest.java**

package com.example;

import org.junit.jupiter.api.Test;

import org.junit.jupiter.api.extension.ExtendWith;

import org.mockito.Mock;

import org.mockito.junit.jupiter.MockitoExtension;

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

import static org.mockito.Mockito.when;

@ExtendWith(MockitoExtension.class)

public class MyServiceTest {

    @Mock

    ExternalApi mockApi;

    @Test

    public void testExternalApi() {

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

        MyService service = new MyService(mockApi);

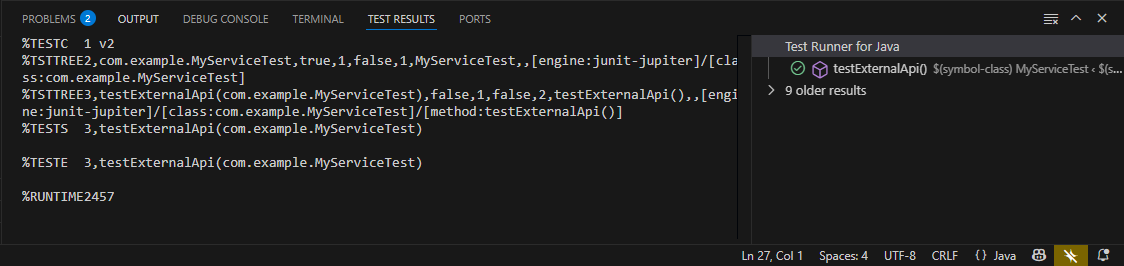
        String result = service.fetchData();

        assertEquals("Mock Data", result);

    }

}

**Output:**



**Exercise 2:** Verifying Interactions

**Scenario:**

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

**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();

}

}

**MyServiceTest1.java**

package com.example;

import org.junit.jupiter.api.Test;

import org.junit.jupiter.api.extension.ExtendWith;

import org.mockito.Mock;

import org.mockito.junit.jupiter.MockitoExtension;

import static org.mockito.Mockito.verify;

@ExtendWith(MockitoExtension.class)

public class MyServiceTest1 {

    @Mock

    ExternalApi mockApi;

    @Test

    public void testVerifyInteraction() {

        // Arrange

        MyService service = new MyService(mockApi);

        // Act

        service.fetchData();

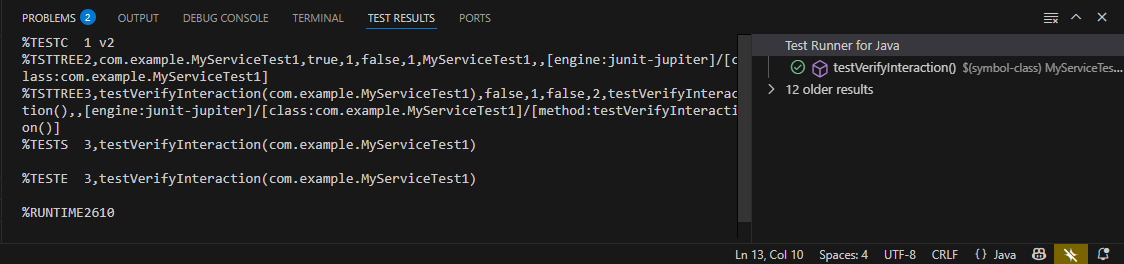
        // Assert / Verify that getData() was called once

        verify(mockApi).getData(); // Verifies that mockApi.getData() was called exactly once

    }

}

**Output:**

****