**Name:** Paarthivi D

**Superset ID:** 6410961

**DN 4.0 - Java FSE Mandatory Hands-on**

**WEEK – 2**

**MOCKITO EXERCISES**

**EXERCISE 1: MOCKING AND STUBBING**

package com.example;

public interface ExternalApi {

    String getData();

}

**Service class**

package com.example;

public class MyService {

    private ExternalApi api;

    public MyService(ExternalApi api) {

        this.api = api;

    }

    public String fetchData() {

        return api.getData();

    }

}

**Test class**

package com.example.exercise1;

import org.junit.jupiter.api.Test;

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

import static org.mockito.Mockito.\*;

public class Exercise1Test {

    @Test

    public void testMockingAndStubbing() {

        ExternalApi mockApi = mock(ExternalApi.class);

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

        MyService service = new MyService(mockApi);

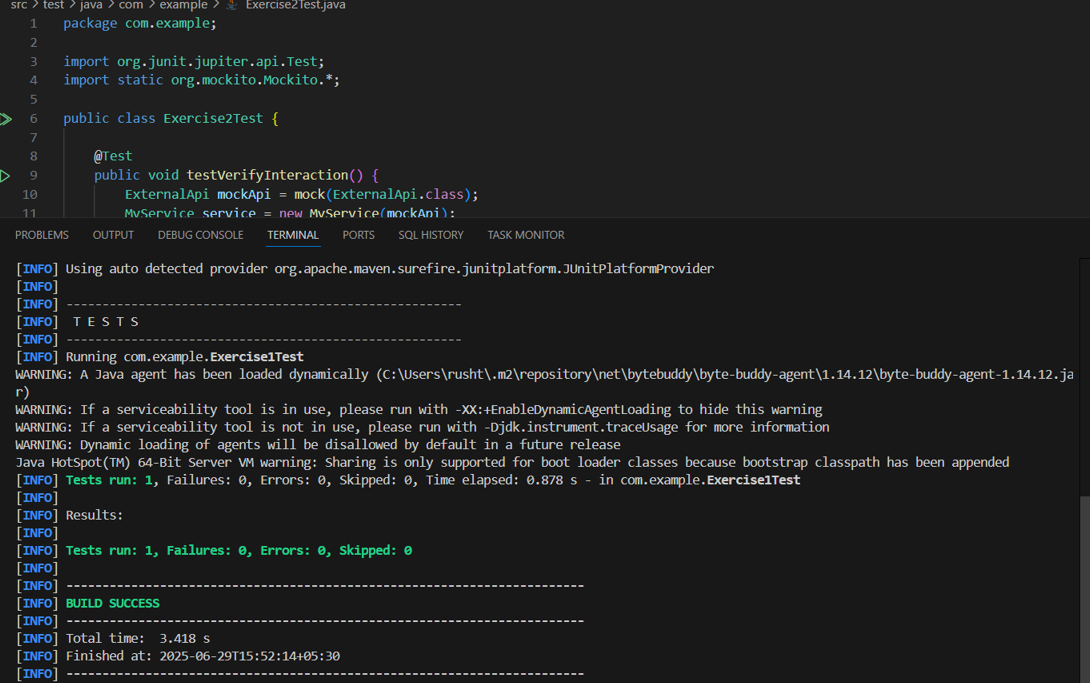
        String result = service.fetchData();

        assertEquals("Mock Data", result);

    }

}

**OUTPUT:**



**EXERCISE 2: VERIFYING INTERACTIONS**

package com.example;

public interface ExternalApi {

    String getData();

}

**Service class**

package com.example;

public class MyService {

    private ExternalApi api;

    public MyService(ExternalApi api) {

        this.api = api;

    }

    public String fetchData() {

        return api.getData();

    }

}

**Test class**

package com.example;

import org.junit.jupiter.api.Test;

import static org.mockito.Mockito.\*;

public class Exercise2Test {

    @Test

    public void testVerifyInteraction() {

        ExternalApi mockApi = mock(ExternalApi.class);

        MyService service = new MyService(mockApi);

        service.fetchData();

        verify(mockApi).getData();

    }

}

**OUTPUT:**

