WEEK-2

**Mockito exercises**

**Exercise 1: Mocking and Stubbing**

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.jupiter.api.Test;

import org.mockito.Mockito;

import static org.mockito.Mockito.\*;

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

public class MyServiceTest {

    @Test

    public void testExternalApi() {

        // Step 1: Create a mock object

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

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

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

        // Step 3: Use the mock in the service

        MyService service = new MyService(mockApi);

        // Step 4: Call the method and assert

        String result = service.fetchData();

        assertEquals("Mock Data", result);

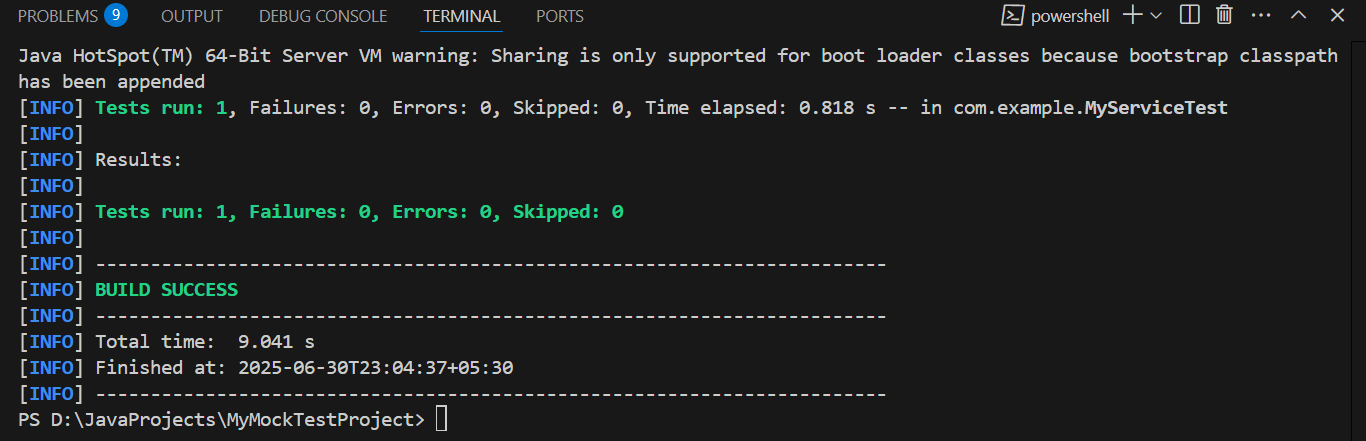
        // Step 5 (Optional): Verify interaction

        verify(mockApi).getData();

    }

}

**Output:**



**Exercise 2: Verifying Interactions**

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.jupiter.api.Test;

import org.mockito.Mockito;

import static org.mockito.Mockito.\*;

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

public class MyServiceTest {

    @Test

    public void testExternalApi() {

        // Step 1: Create a mock object

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

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

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

        // Step 3: Use the mock in the service

        MyService service = new MyService(mockApi);

        // Step 4: Call the method and assert

        //String result = service.fetchData();

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

        service.fetchData();

        // Step 5 (Optional): Verify interaction

        verify(mockApi).getData();

    }

}

Output:

