**MOCKITO EXERCISES**

**EXERCISE 1: MOCKING AND STUBBING**

Create a java maven project in eclipse.

Update the pom.xml file and update the Maven project.

**ExternalApi.java** in src/main/java

package com.example;

public interface ExternalApi {

String getData();

}

**Service.java**

package com.example;

public class MyService {

private ExternalApi externalApi;

public MyService(ExternalApi externalApi) {

this.externalApi = externalApi;

}

public String fetchData() {

return externalApi.getData();

}

}

**ServiceTest.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() {

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

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

MyService service = new MyService(mockApi);

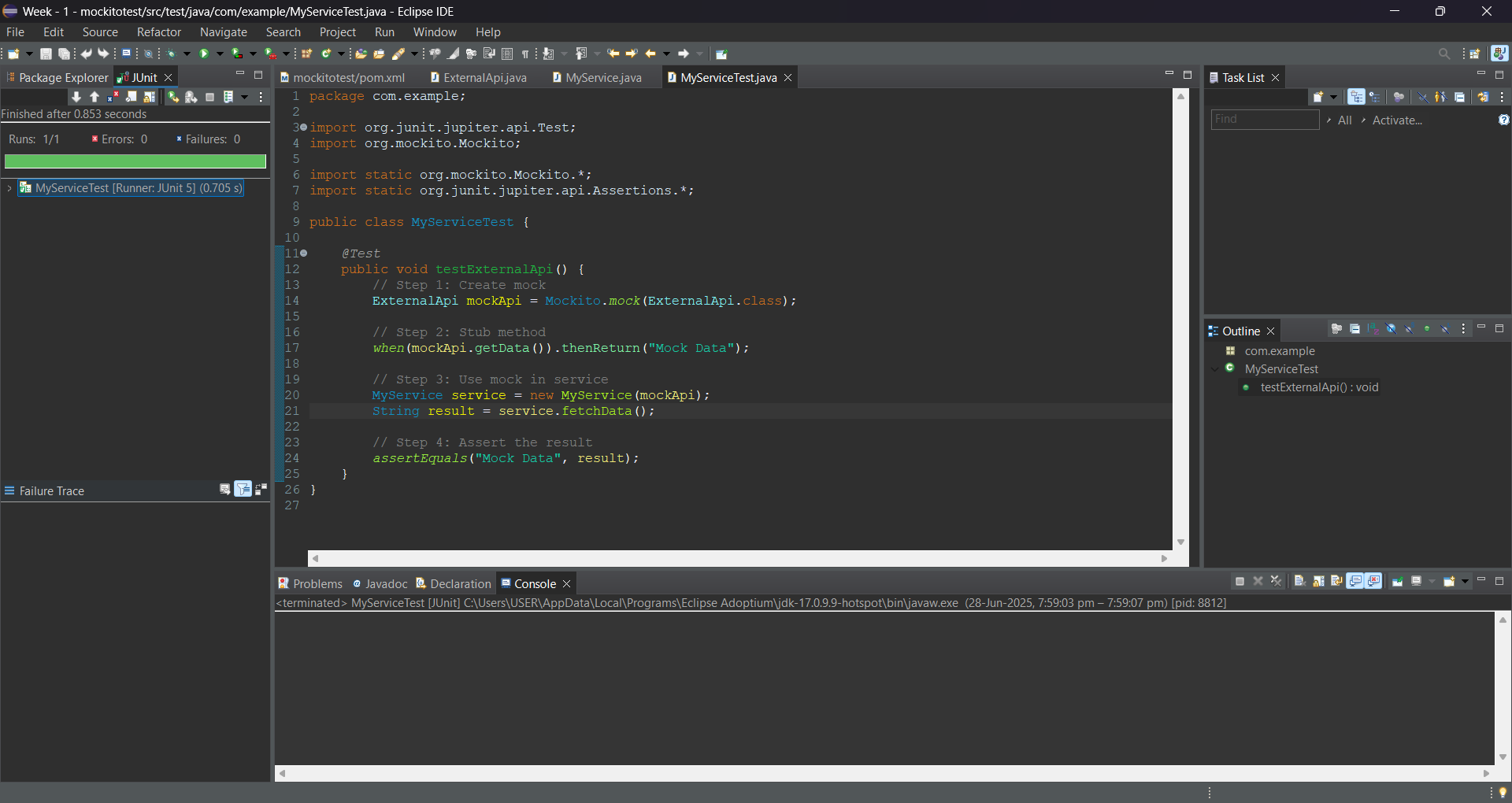
String result = service.fetchData();

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

}

}

**OUTPUT**



**EXERCISE 2: VERIFYING INTERACTIONS**

Create a java maven project in eclipse

Update the pom.xml file and update the Maven project

**ExternalApi.java**

package com.example;

public interface ExternalApi {

String getData();

}

**Service.java**

package com.example;

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

**ServiceTest.java**

package com.example;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

import static org.mockito.Mockito.\*;

public class MyServiceTest {

*@Test*

public void testVerifyInteraction() {

// Step 1: Mock the ExternalApi

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

// Step 2: Pass mock to service

MyService service = new MyService(mockApi);

service.fetchData();

// Step 3: Verify that getData() was called

*verify*(mockApi).getData();

}

}

**OUTPUT**

