**Exercise 1: Mocking and Stubbing**

**Code:**

ExternalApi.java

package Cognizant\_training\_week\_2\_modules.Cognizant\_training\_week\_2\_modules;

public interface ExternalApi {

String getData();

}

MyService.java

package Cognizant\_training\_week\_2\_modules.Cognizant\_training\_week\_2\_modules;

public class MyService {

private ExternalApi externalApi;

public MyService(ExternalApi externalApi) {

this.externalApi = externalApi;

}

public String fetchData() {

return externalApi.getData();

}

}

MyServiceTest.java

package Cognizant\_training\_week\_2\_modules.Cognizant\_training\_week\_2\_modules;

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

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

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();

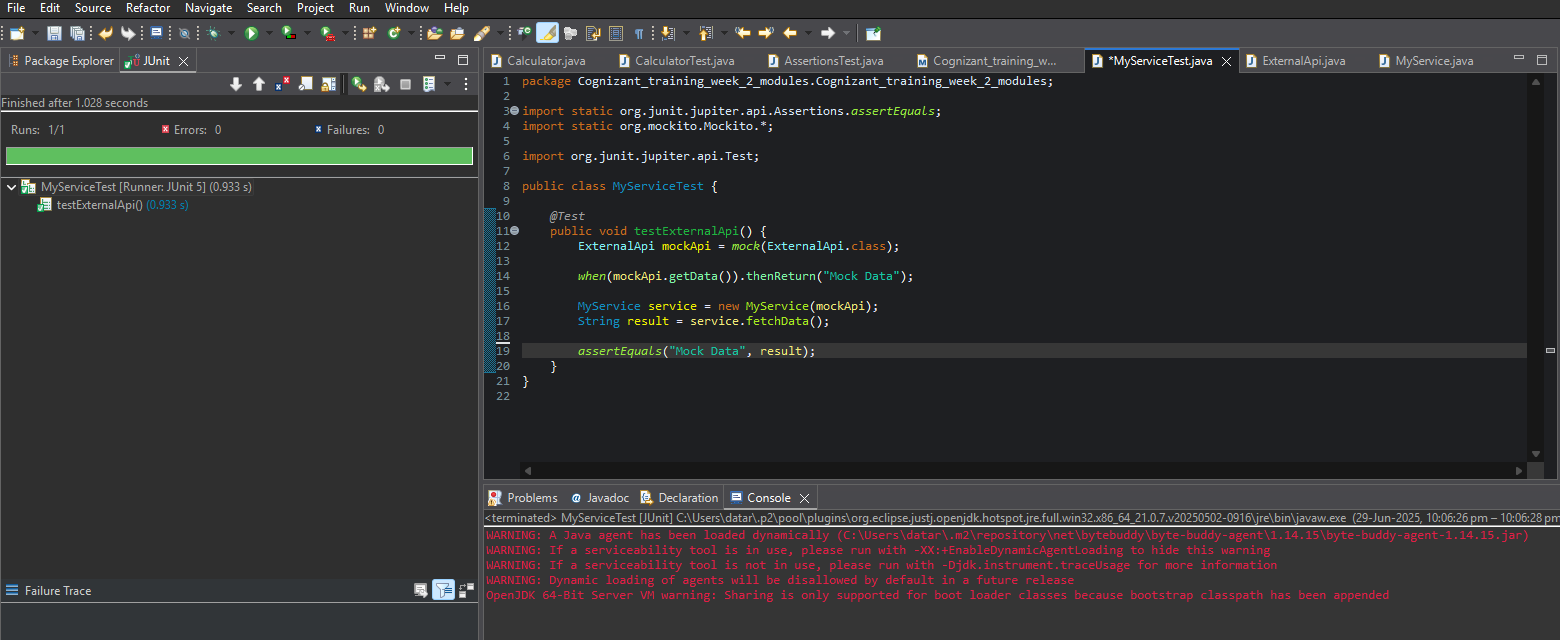
// Assert the result

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

}

}

**Output:**



**Exercise 2: Verifying Interactions**

**Code:**

ExternalApi.java

package Cognizant\_training\_week\_2\_modules.Cognizant\_training\_week\_2\_modules;

public interface ExternalApi {

String getData();

}

MyService.java

package Cognizant\_training\_week\_2\_modules.Cognizant\_training\_week\_2\_modules;

public class MyService {

private ExternalApi externalApi;

public MyService(ExternalApi externalApi) {

this.externalApi = externalApi;

}

public String fetchData() {

return externalApi.getData();

}

}

MyServiceTest.java

package Cognizant\_training\_week\_2\_modules.Cognizant\_training\_week\_2\_modules;

import org.junit.jupiter.api.Test;

import static org.mockito.Mockito.\*;

public class MyServiceTest {

*@Test*

public void testVerifyInteraction() {

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

MyService service = new MyService(mockApi);

service.fetchData();

*verify*(mockApi).getData();

}

}

**Output:**

