**Exercise 1: Mocking and Stubbing**

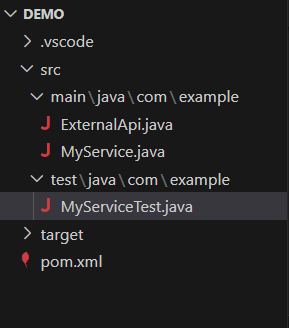
**Scenario:**

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.

**CODE:**

****

**Interface: ExternalApi.java**

package com.example;

public interface ExternalApi {

String getData();

}

**Service Class: MyService.java**

package com.example;

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

**Test Class: MyServiceTest.java**

package com.example;

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

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest {

@Test

public void testExternalApi() {

// Mocking: creating a mock object of ExternalApi

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

// Stubbing: defining behavior of mock object's method

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

MyService service = new MyService(mockApi);

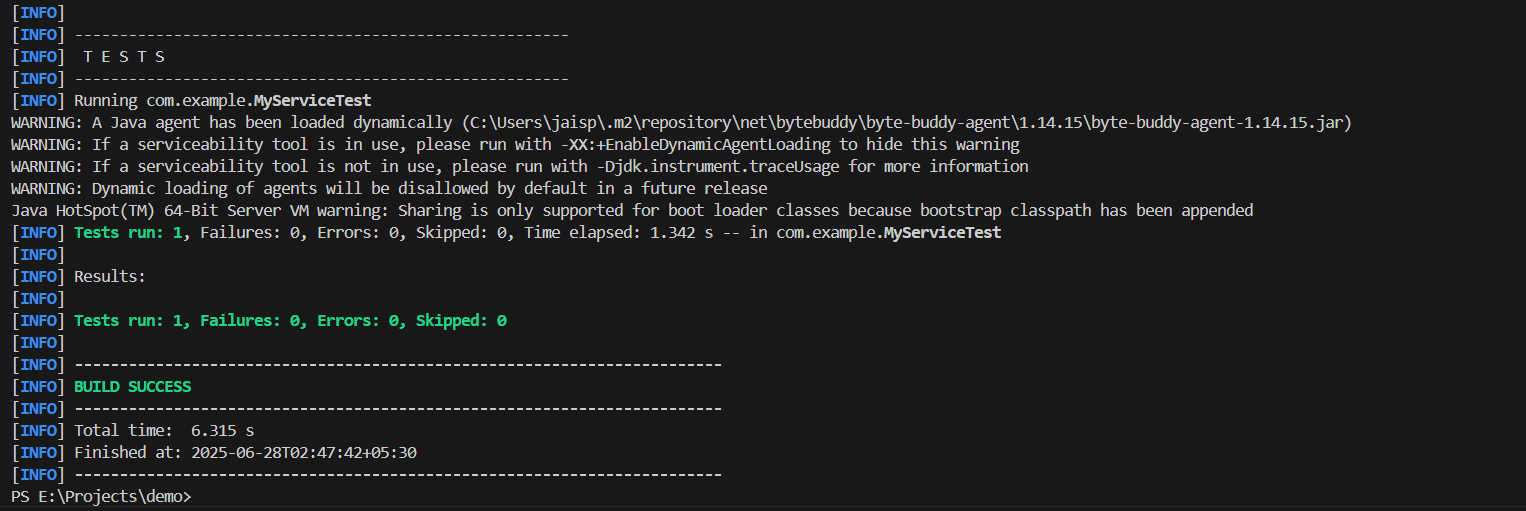
String result = service.fetchData();

assertEquals("Mock Data", result);

}

}

**Output:**

**

**Conclusion:**

* Created a mock of an external dependency using Mockito.
* Stubbed the method to return a predefined value.
* Verified the interaction using JUnit 5 assertions in combination with Mockito.