**Exercise 2: Verifying Interactions**

**Scenario:**

You need to ensure that a method is called with specific arguments.

**Steps:**

1. Create a mock object.
2. Call the method with specific arguments.
3. Verify the interaction.

**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.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest {

@Test

public void testVerifyInteraction() {

// Mocking: creating a mock object of ExternalApi

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

MyService service = new MyService(mockApi);

// Act: call method that should trigger the interaction

service.fetchData();

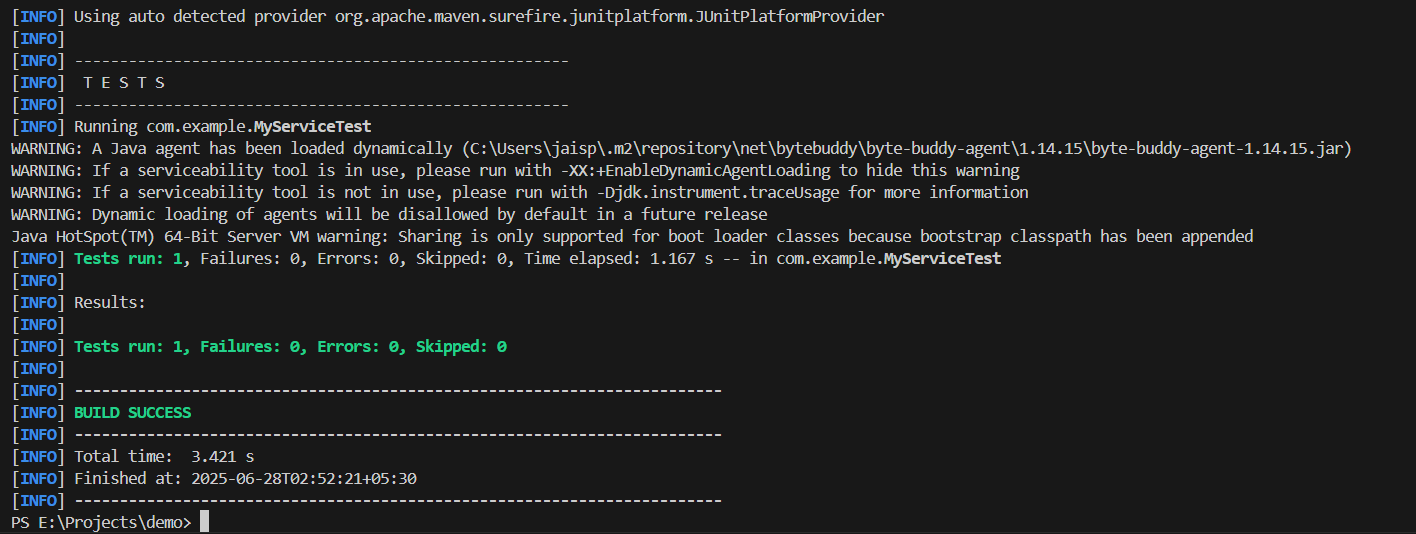
// Verify: check if getData() was called on mockApi

verify(mockApi).getData();

}

}

**Output:**

**

**Conclusion:**

* Verified that getData() was invoked.
* Demonstrated interaction-based testing using Mockito.
* Ensured expected behavior between service and dependency through verification.