**Exercise 2 : Verifying Interactions**

**🡪Components**

* **ExternalApi**  
  A simple class that simulates an external dependency with a **getData**() method.
* **MyService**  
  A service class that depends on ExternalApi. It has a **fetchData**() method that calls **getData**() from the API.
* **Ex2** (Test Class)  
  Contains a JUnit test method **testVerifyInteraction**() that:
  1. Creates a **mock** of ExternalApi using Mockito.
  2. Injects the mock into MyService.
  3. Calls **fetchData**() on the service.
  4. Uses **verify**() to check that **getData**() was called on the mock.

**🡪Purpose**

* **Mocking**: Isolates MyService from the real ExternalApi for testing.
* **Interaction Verification**: Ensures that MyService correctly calls the **getData**() method of its dependency.

**🡪Usage**

* Run the test method **testVerifyInteraction**(). The test will pass if **getData**() is called as expected.

**🡪Code Snippet:**

import org.junit.Test;

import static org.junit.Assert.\*;

import static org.mockito.Mockito.\*;

class ExternalApi {

    public String getData() {

        return "Real Data";

    }

}

class MyService {

    private ExternalApi api;

    public MyService(ExternalApi api) {

        this.api = api;

    }

    public String fetchData() {

        return api.getData();

    }

}

public class Ex2 {

    @Test

    public void testVerifyInteraction() {

        ExternalApi mockApi = mock(ExternalApi.class);

        MyService service = new MyService(mockApi);

        service.fetchData();

        verify(mockApi).getData();

    }

}