**Mockito Exercices**

**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.**

**ExternalApi.java:**

public interface ExternalApi {

String getData();

}

**ExternalApi.java:**

public class MyService {

private ExternalApi externalApi;

public MyService(ExternalApi externalApi) {

this.externalApi = externalApi;

}

public String fetchData() {

return externalApi.getData();

}

}

**Exercise 2: Verifying Interactions**

**Scenario:**

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

**MyService.java:**

import static org.mockito.Mockito.\*;

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

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest {

@Test

public void testExternalApi() {

// Step 1: Create a mock for the external API

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

// Step 2: Stub the method to return a predefined value

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

// Step 3: Write a test using the mock object

MyService service = new MyService(mockApi);

String result = service.fetchData();

// Assertion

assertEquals("Mock Data", result);

}

}

A screenshot of a computer program

AI-generated content may be incorrect.