Exercise 4:

package com.example.demo;

public interface ExternalApi {

String getData();

String getData(String userId);

void send(String userId, String message);

}

package com.example.demo;

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

public String process\_msg(String userId) {

api.send(userId, "Hello " + userId);

return api.getData(userId);

}

}

package com.example.demo;

import static org.mockito.Mockito.\*;

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

import org.junit.jupiter.api.Test;

public class MyServiceTest {

*@Test*

public void testFetchData() {

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

// when(mockApi.getData()).thenReturn("mockdata");

// when(mockApi.getData("user1234")).thenReturn("mock-data with parameter");

// doNothing().when(mockApi).send(anyString(), anyString());

// MyService service = new MyService(mockApi);

// String result1 = service.process("user1234");

// verify(mockApi).send("user123", "Hello user123");

// System.out.println( " user123 verified.");

// verify(mockApi).getData(anyString());

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

*when*(mockApi.getData("user123")).thenReturn("mock-data with parameter");

*doNothing*().when(mockApi).send(*anyString*(), *anyString*());

MyService service = new MyService(mockApi);

String result = service.process\_msg("user123");

*verify*(mockApi).send("user123", "Hello user123");

*verify*(mockApi).getData("user123");

*assertEquals*("mock-data with parameter", result);

System.***out***.println("Test passed. Result: " + result);

}

}

A screenshot of a computer program

AI-generated content may be incorrect.