Mockito Hands-On Exercises

## 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.

**Api.java :**

**package org.example;**

public interface Api {

String get();

}

**MainTest.java**

package org.example;

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

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;public class MainTest

@Test

public void testApiCall() {

Api api = mock(Api.class);

when(api.get()).thenReturn("Mock");

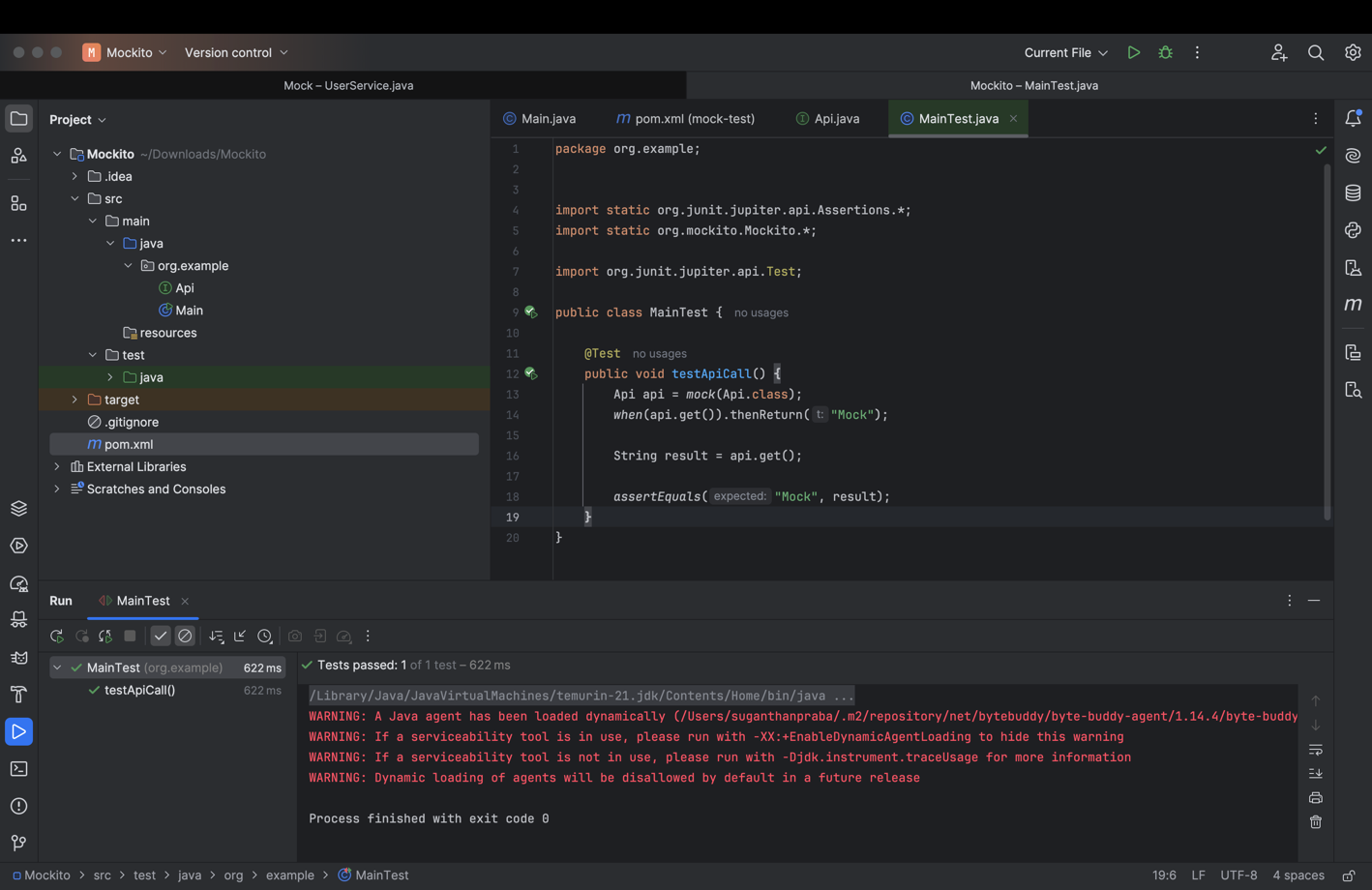
String result = api.get();

## assertEquals("Mock", result);

## }

## }

## Output:



## Exercise 2: Verifying Interactions:

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

**Api.java:**

package org.example;

public interface Api {

String get();

}

**UseApi.java:**

package org.example;

public class UseApi {

private Api api;

public UseApi(Api api) {

this.api = api;

}

public String call() {

return api.get();

}

}

**CheckCall.java :**

package org.example;

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

public class CheckCall {

@Test

public void testGetCalledOnce() {

Api mockApi = mock(Api.class);

UseApi useApi = new UseApi(mockApi);

useApi.call();

verify(mockApi).get();

}

## Output:

