**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.

Steps: 1. Create a mock object for the external API.

2. Stub the methods to return predefined values.

3. Write a test case that uses the mock object.

**Code**

**DemoApi.java**

package org.example;  
  
public interface DemoApi {  
 String getData();  
}

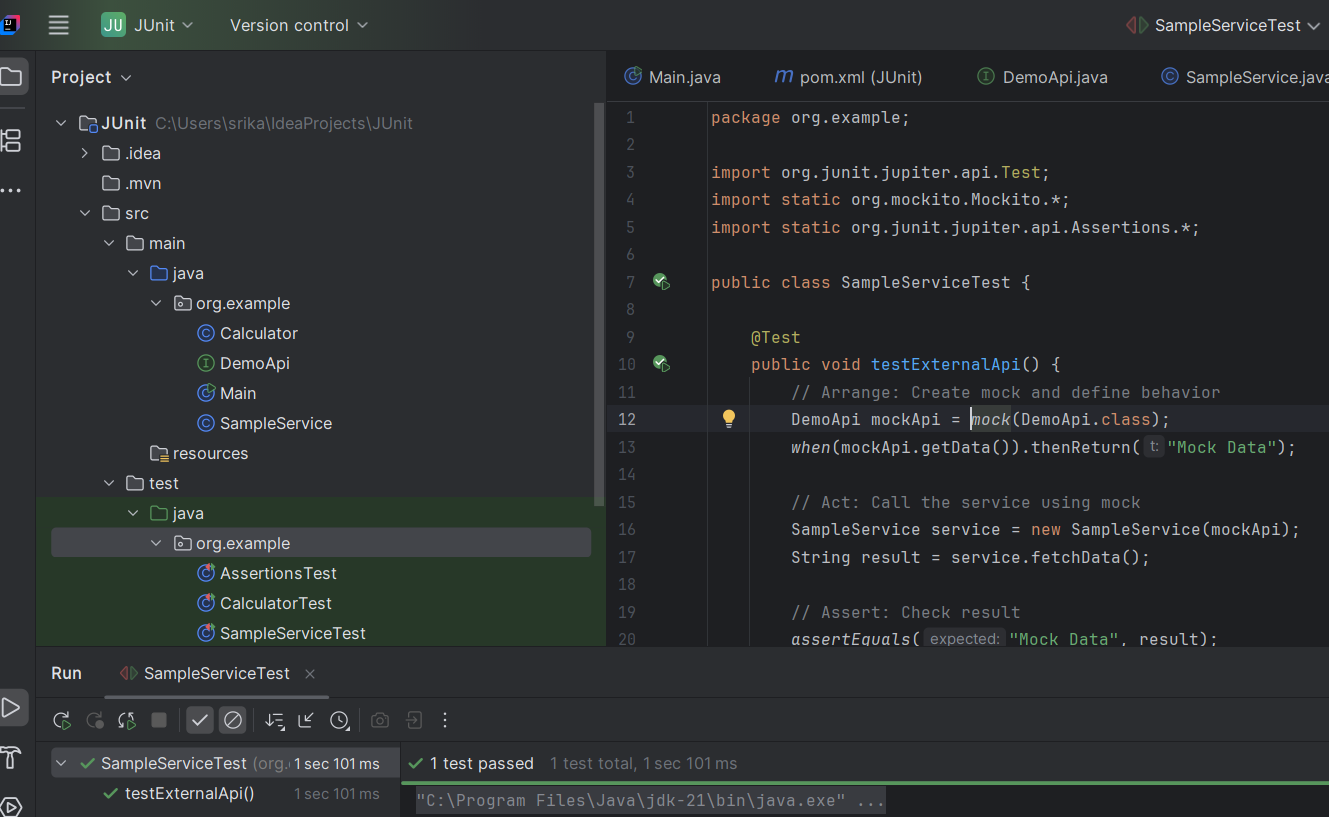
**SampleService.java**

package org.example;  
  
public class SampleService {  
 private final DemoApi api;  
  
 public SampleService(DemoApi api) {  
 this.api = api;  
 }  
  
 public String fetchData() {  
 return api.getData();  
 }  
}

**SampleServiceTest.java**

package org.example;  
import org.junit.jupiter.api.Test;  
import static org.mockito.Mockito.\*;  
import static org.junit.jupiter.api.Assertions.\*;  
  
public class SampleServiceTest {  
 @Test  
 public void testExternalApi() {  
   
 DemoApi mockApi = *mock*(DemoApi.class);  
 *when*(mockApi.getData()).thenReturn("Mock Data");  
 SampleService service = new SampleService(mockApi);  
 String result = service.fetchData();  
 *assertEquals*("Mock Data", result);  
 }  
}

**Output**

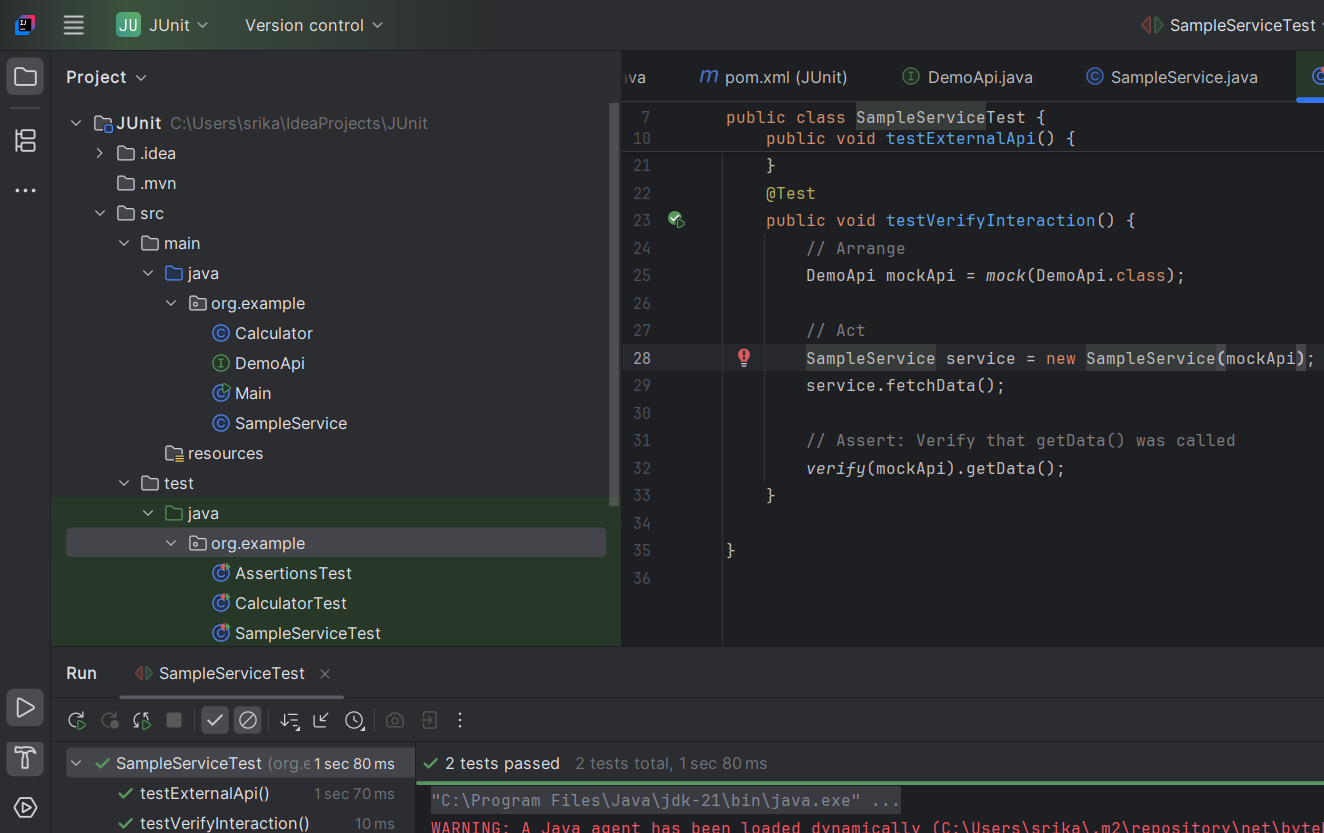


**Exercise 2: Verifying Interactions Scenario:** You need to ensure that a method is called with specific arguments.

@Test  
public void testVerifyInteraction() {  
 DemoApi mockApi = *mock*(DemoApi.class);  
 SampleService service = new SampleService(mockApi);

service.fetchData();  
 *verify*(mockApi).getData();  
}

**Output**

****