**MOCKITO 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:**

**DEPENDENCIES FOR pom.xml**

<dependencies>  
 <dependency>  
 <groupId>org.mockito</groupId>  
 <artifactId>mockito-core</artifactId>  
 <version>4.8.1</version>  
 <scope>test</scope>  
</dependency>  
  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter-api</artifactId>  
 <version>5.9.0</version>  
 <scope>test</scope>  
 </dependency>  
  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter-engine</artifactId>  
 <version>5.9.0</version>  
 <scope>test</scope>  
 </dependency>  
  
</dependencies>

**1.ExternalApi.java**

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

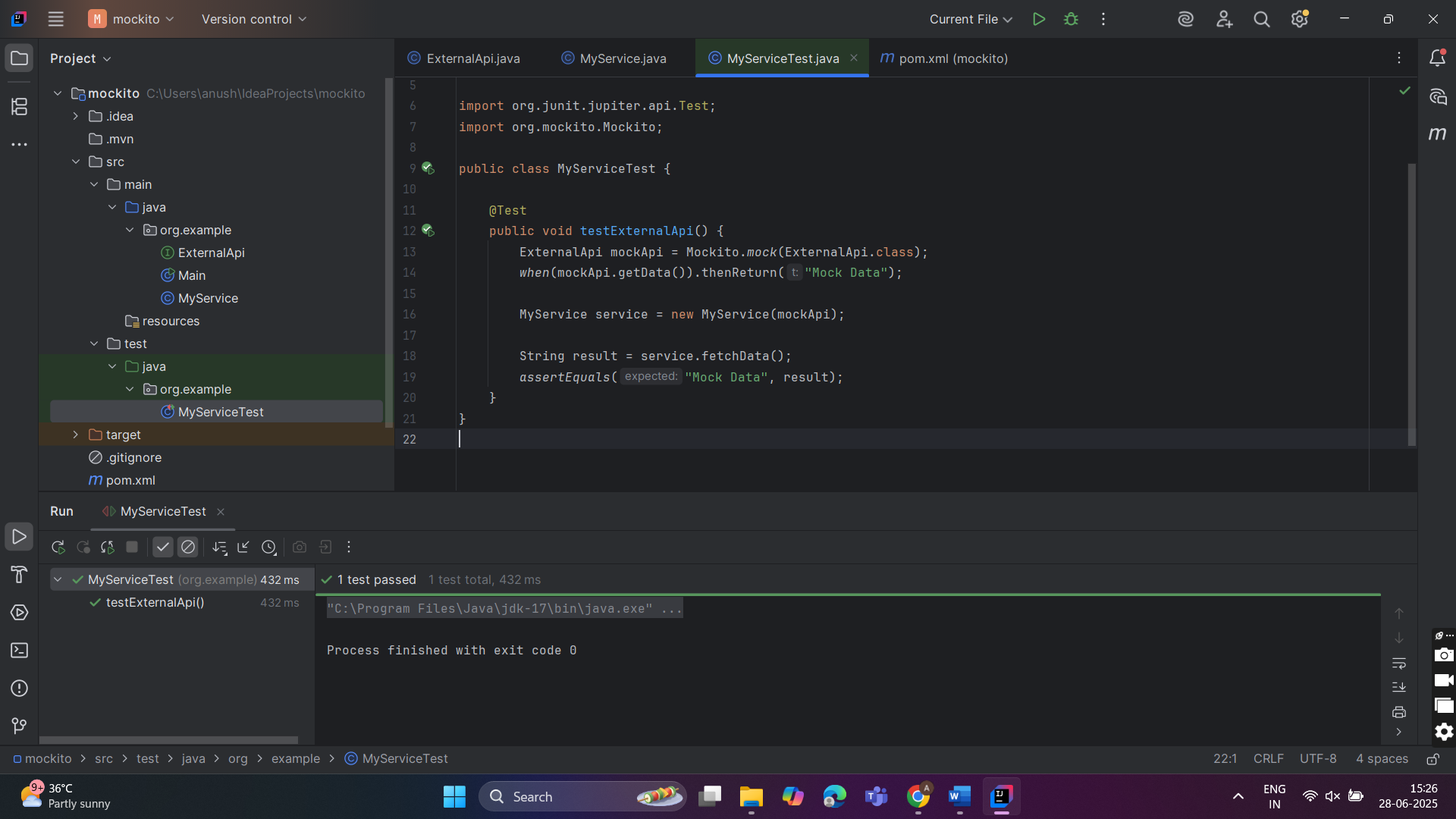
**2.MyService.java**

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

**3.MyServiceTest.java**

package org.example;  
  
import static org.mockito.Mockito.\*;  
import static org.junit.jupiter.api.Assertions.\*;  
  
import org.junit.jupiter.api.Test;  
import org.mockito.Mockito;  
  
public class MyServiceTest {  
  
 @Test  
 public void testExternalApi() {  
 ExternalApi mockApi = Mockito.*mock*(ExternalApi.class);  
 *when*(mockApi.getData()).thenReturn("Mock Data");  
  
 MyService service = new MyService(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.

**Steps:**

1. Create a mock object.
2. Call the method with specific arguments.
3. Verify the interaction.

**CODE:**

**1.ExternalApi.java**

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

**2.MyService.java**

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

**3.MyServiceTest.java**

package org.example;  
  
import static org.mockito.Mockito.\*;  
import org.junit.jupiter.api.Test;  
import org.mockito.Mockito;  
  
public class MyServiceTest {  
  
 @Test  
 public void testVerifyInteraction() {  
 ExternalApi mockApi = Mockito.*mock*(ExternalApi.class);  
  
 MyService service = new MyService(mockApi);  
 service.fetchData();  
  
 *verify*(mockApi).getData();  
 }  
}

**OUTPUT**:

