**Exercise 1: Mocking and Stubbing**

Scenario:  
You need to test a service (MyService) that depends on an external API (ExternalApi). The goal is to use Mockito to mock the API and stub its method to return a predefined value.

ExternalApi.java

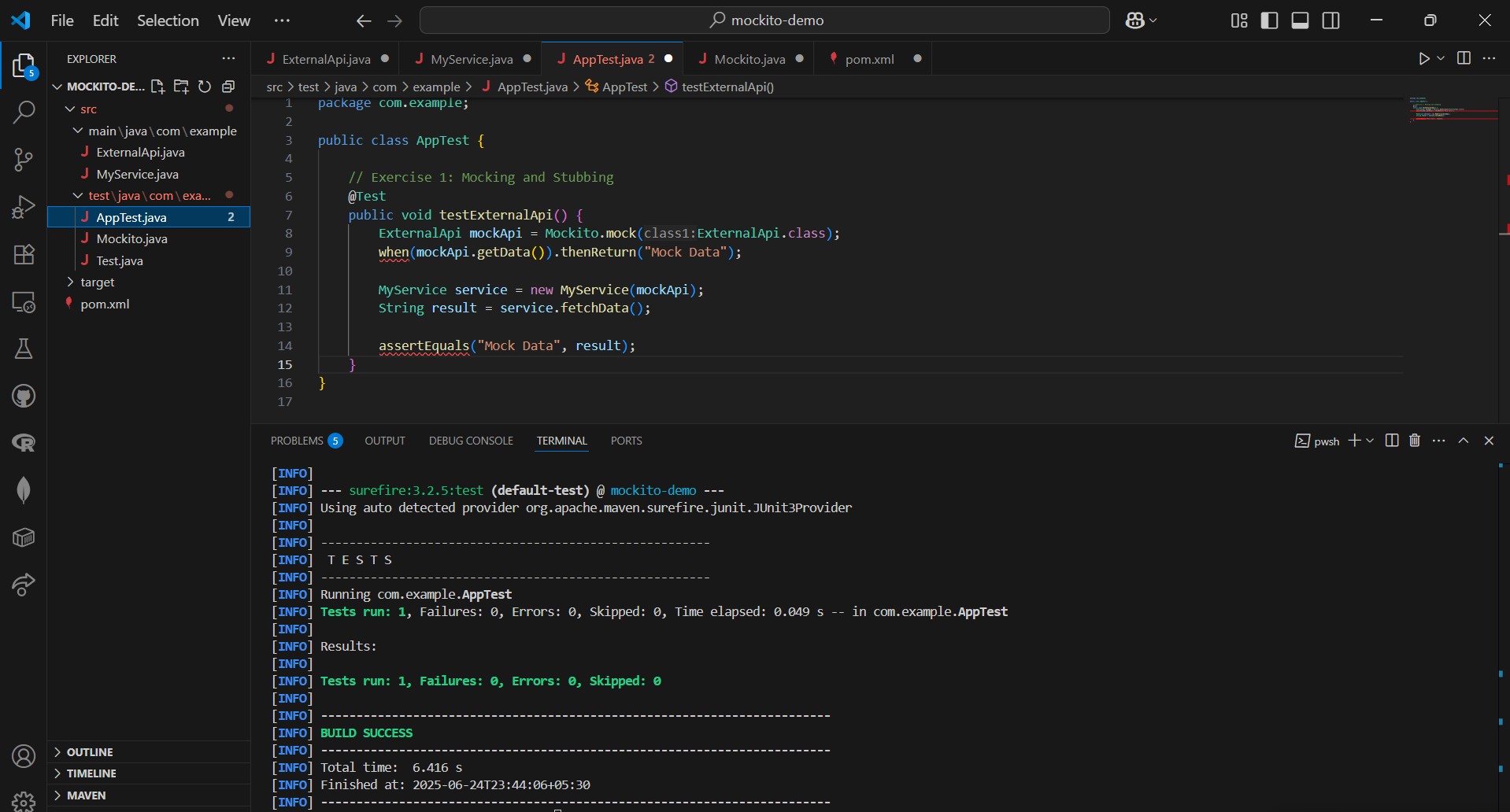
package com.example;  
  
public interface ExternalApi {  
 String getData();  
}

MyService.java

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

AppTest.java

package com.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 AppTest {  
  
 @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);  
 }  
}



**Exercise 2: Verifying Interactions**

Scenario:  
You need to ensure that the method getData() on the external API is actually called by the service.

ExternalApi.java

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

MyService.java

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

AppTest.java

package com.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 AppTest {  
  
 @Test  
 public void testVerifyInteraction() {  
 ExternalApi mockApi = Mockito.mock(ExternalApi.class);  
 MyService service = new MyService(mockApi);  
  
 service.fetchData();  
  
 verify(mockApi).getData(); // Verifies interaction  
 }  
}  
  
