**Exercise 1: Mocking and Stubbing**

**Question:** You need to test a service that depends on an external API. Use Mockito to mock the external API and stub its methods.

**CODE:**

**ExternalApi.java:**

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

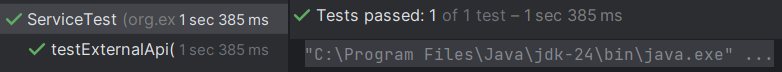
**MyService.java:**

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

**ServiceTest.java:**

package org.example;  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.*assertEquals*;  
import static org.mockito.Mockito.\*;  
  
public class ServiceTest {  
 @Test  
 public void testExternalApi() {  
  
 ExternalApi mockApi = *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**

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

**CODE:**

**ExternalApi.java:**

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

**MyService.java**:

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

**MyServiceTest.java**:

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

**OUTPUT:**

