**Exercise 2: Verifying Interactions**

**Scenario:**

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

**Steps:**

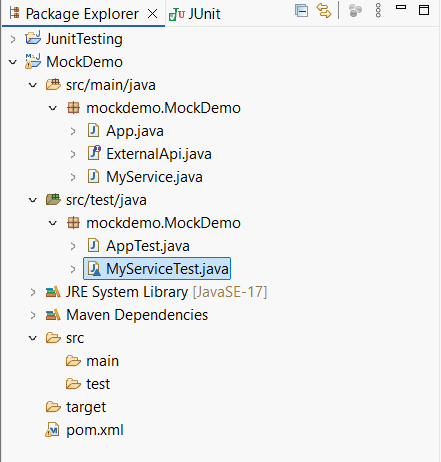
1. Create a mock object.

2. Call the method with specific arguments.

3. Verify the interaction.

**STEP 1 – CREATE A MAVEN PROJECT**

* Create a maven project using “maven-archetype-quickstart”.
* Below is the project structure



**STEP 2 – ADDING DEPENDENCIES TO pom.xml**

* Add Junit and Mokito dependencies to the project

|  |
| --- |
| <dependency>  <groupId>org.junit.jupiter</groupId>  <artifactId>junit-jupiter</artifactId>  <version>5.10.0</version>  <scope>test</scope>  </dependency>  <dependency>  <groupId>org.mockito</groupId>  <artifactId>mockito-core</artifactId>  <version>5.11.0</version>  <scope>test</scope>  </dependency> |

**STEP 3 – CREATE THE JAVA CLASSES**

* Create the external API Interface to create a mock object.

**Code:**

|  |
| --- |
| **package** mockdemo.MockDemo;  **public** **interface** ExternalApi {  String getData();  **void** sendData(String data);  } |

* Define a class to use the object

**Code:**

|  |
| --- |
| **package** mockdemo.MockDemo;  **public** **class** MyService {  **private** ExternalApi api;  **public** MyService(ExternalApi api) {  **this**.api = api;  }  **public** String fetchData() {  **return** api.getData();  }  **public** **void** saveData(String data) {  api.sendData(data);  }  } |

**STEP 4 – TESTING THE PROJECT**

* Create a class to use the mock object

Code:

|  |
| --- |
| @Test  void testVerifyFetchDataCalled() {  // Arrange  ExternalApi mockApi = mock(ExternalApi.class);  when(mockApi.getData()).thenReturn("Mock Result");  MyService service = new MyService(mockApi);  // Act  String result = service.fetchData();  // Assert  assertEquals("Mock Result", result);  // Verify interaction  verify(mockApi).getData();  }  @Test  void testVerifySendDataCalledWithCorrectArgument() {  // Arrange  ExternalApi mockApi = mock(ExternalApi.class);  MyService service = new MyService(mockApi);  String sampleData = "Hello World";  // Act  service.saveData(sampleData);  // Verify interaction with specific argument  verify(mockApi).sendData("Hello World");  }  @Test  void testVerifySendDataNotCalled() {  // Arrange  ExternalApi mockApi = mock(ExternalApi.class);  MyService service = new MyService(mockApi);  // Act  // (no call to saveData)  // Verify that sendData was never called  verify(mockApi, never()).sendData(anyString());  } |

**Output:**

