**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:**

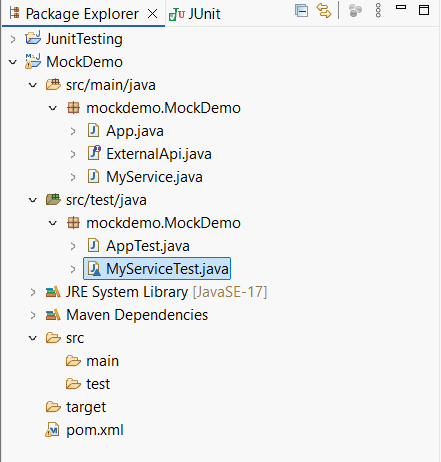
1. Create a mock object for the external API.

2. Stub the methods to return predefined values.

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

**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;  public interface ExternalApi {  String getData();  } |

* 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() {  System.out.print("The API has been fetched.");  return api.getData();  }  } |

**STEP 4 – TESTING THE PROJECT**

* Create a class to use the mock object

**Code:**

|  |
| --- |
| package mockdemo;  import org.junit.jupiter.api.Test;  import static org.junit.jupiter.api.Assertions.assertEquals;  import static org.mockito.Mockito.\*;  class MyServiceTest {  @Test  void testFetchDataReturnsMockedValue() {  ExternalApi mockApi = mock(ExternalApi.class);  when(mockApi.getData()).thenReturn("Mock Data");  MyService service = new MyService(mockApi);  String result = service.fetchData();  assertEquals("Mock Data", result);  }  @Test  void testFetchDataCalledOnce() {  ExternalApi mockApi = mock(ExternalApi.class);  when(mockApi.getData()).thenReturn("Another Mock");  MyService service = new MyService(mockApi);  service.fetchData();  verify(mockApi, times(1)).getData();  }  } |

**Output:**

