**Week-2 Mockito Exercises**

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.

**Code:**

**ExternalAPI.java**

public interface ExternalAPI

{

String getData();

}

**MyService.java**

public class MyService

{

private ExternalAPI api;

// Constructor injection

public MyService(ExternalAPI api)

{

this.api = api;

}

public String fetchData()

{

return api.getData(); // Uses the API

}

}

**MyServiceTestClass.java**

import static org.mockito.Mockito.\*;

import static org.junit.jupiter.api.Assertions.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTestClass

{

@Test

public void testExternalApi()

{

// Step 1: Mock the External API

ExternalAPI mockApi = Mockito.*mock*(ExternalAPI.class);

// Step 2: Stub the method getData() to return "Mock Data"

*when*(mockApi.getData()).thenReturn("Mock Data");

// Step 3: Pass mock to the service and call the method

MyService service = new MyService(mockApi);

String result = service.fetchData();

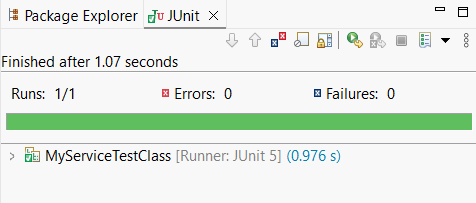
// Step 4: Assert the result

*assertEquals*("Mock Data", result);

}

}

**Output:**

****

1. **Verifying Interactions**

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

**Code:**

**ExternalAPI.java**

public interface ExternalAPI

{

String getData();

}

**MyService.java**

public class MyService

{

private ExternalAPI api;

// Constructor injection

public MyService(ExternalAPI api)

{

this.api = api;

}

public String fetchData()

{

return api.getData(); // Uses the API

}

}

**MyServiceTestClass.java**

import static org.mockito.Mockito.\*;

import static org.junit.jupiter.api.Assertions.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTestClass

{

@Test

public void testVerifyInteraction()

{

ExternalAPI mockApi = Mockito.mock(ExternalAPI.class);

MyService service = new MyService(mockApi);

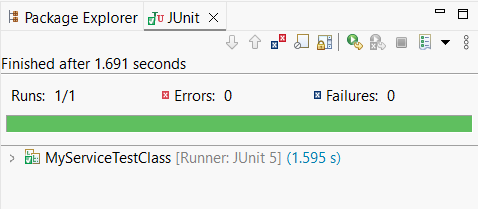
service.fetchData(); // calling method

verify(mockApi).getData(); // verifying the method was called

}

}

**Output:**

****