Mockito Exercices

**Exercise 1: Mocking and Stubbing**

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.

**ExternalApi.java**

**public** **interface** ExternalApi {

String getData();

}

**MyService.java**

**public** **class** MyService {

**private** ExternalApi externalApi;

**public** MyService(ExternalApi externalApi) {

**this**.externalApi = externalApi;

}

**public** String fetchData() {

**return** externalApi.getData();

}

}

**MyServiceTest.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** MyServiceTest {

@Test

**public** **void** testExternalApiWithMocking() {

// Create a mock of ExternalApi

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

// Stub the getData() method

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

// Inject the mock and test the service

MyService service = **new** MyService(mockApi);

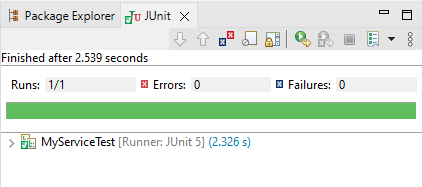
String result = service.fetchData();

// Validate the stubbed behavior

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

}

}



**Exercise 2: Verifying Interactions**

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.

**MyServiceVerificationTest.java**

**import** **static** org.mockito.Mockito.\*;

**import** org.junit.jupiter.api.Test;

**import** org.mockito.Mockito;

**public** **class** MyServiceVerificationTest {

@Test

**public** **void** testVerifyExternalApiInteraction() {

// Create a mock object

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

// Inject the mock into the service and call the method

MyService service = **new** MyService(mockApi);

service.fetchData(); // This should trigger mockApi.getData()

// Verify that getData() was called on the mock

*verify*(mockApi).getData();

}

}

