**Mockito exercises**

**Exercise 1: Mocking and Stubbing**

class Main {

public static void main(String[] args) {

ExternalApi mockApi = new ExternalApi() {

@Override

public String getData() {

return "Mock Data";

}

};

MyService service = new MyService(mockApi);

String result = service.fetchData();

if ("Mock Data".equals(result)) {

System.out.println("Test Passed: " + result);

} else {

System.out.println("Test Failed: Got " + result);

}

}

}

interface ExternalApi {

String getData();

}

class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

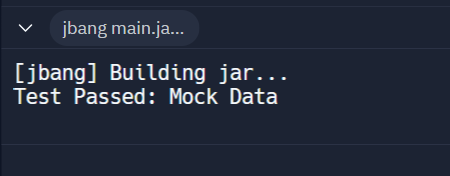
return api.getData();

} }





OUTPUT :



**Exercise 2: Verifying Interactions**

class Main {

public static void main(String[] args) {

ExternalApiMock mockApi = new ExternalApiMock();

MyService service = new MyService(mockApi);

service.fetchData();

if (mockApi.wasGetDataCalled()) {

System.out.println("Test Passed: getData() was called.");

} else {

System.out.println("Test Failed: getData() was NOT called.");

}

}

}

// Simulate external API interface

interface ExternalApi {

String getData();

}

// Mock class that records interaction

class ExternalApiMock implements ExternalApi {

private boolean getDataCalled = false;

@Override

public String getData() {

getDataCalled = true;

return "Mock Data";

}

public boolean wasGetDataCalled() {

return getDataCalled;

}

}

// Service that uses ExternalApi

class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

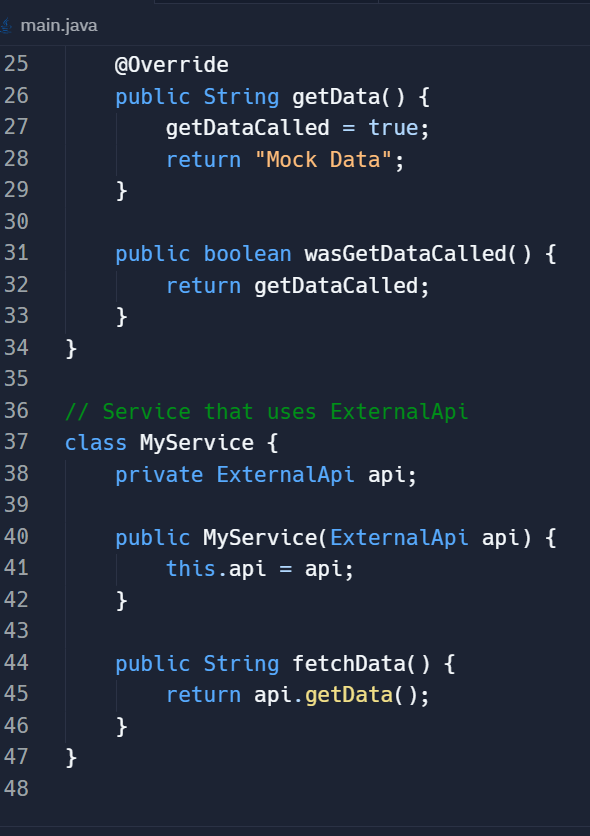
public String fetchData() {

return api.getData();

}

}





OUTPUT :

