Mocking and Stubbing in JUnit using Mockito

# ****Exercise 1: Mocking and Stubbing****

# Scenario:

In this exercise, we aim to test a service that depends on an external API. Instead of calling the actual API during testing, we use Mockito to create a mock object, stub its methods to return predefined values, and validate our service logic based on those values.

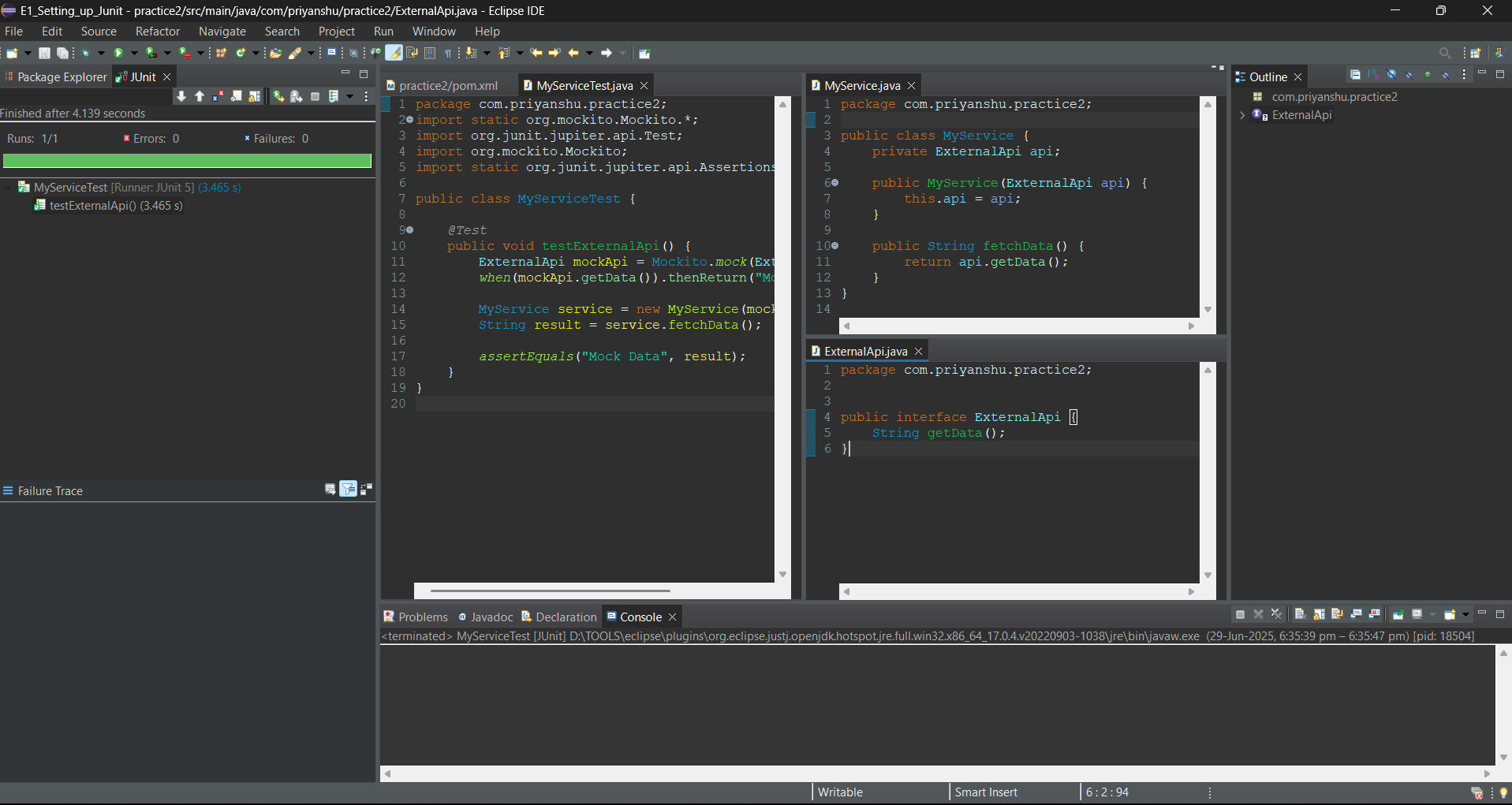
# Code Summary:

• `ExternalApi`: Interface that defines the external dependency with a `getData()` method.

• `MyService`: Uses `ExternalApi` to fetch data via its `fetchData()` method.

• `MyServiceTest`: JUnit test class using Mockito to mock the API and verify behavior.

# Output:



# ****Exercise 2: Verifying Interactions****

## ****Objective:****

To ensure that a method in a mocked dependency is invoked with the correct arguments. This helps validate that your code interacts with its collaborators properly.

## ****Code:****

@Test

public void testVerifyInteraction() {

ExternalApi mockApi = mock(ExternalApi.class);

MyService service = new MyService(mockApi);

service.fetchData();

verify(mockApi).getData();

}

## ****Output:****