**Ques 1 : Mocking and Stubbing**

**Code :**

**--ExternalApi.java**

public interface ExternalApi {  
 String getData();  
}

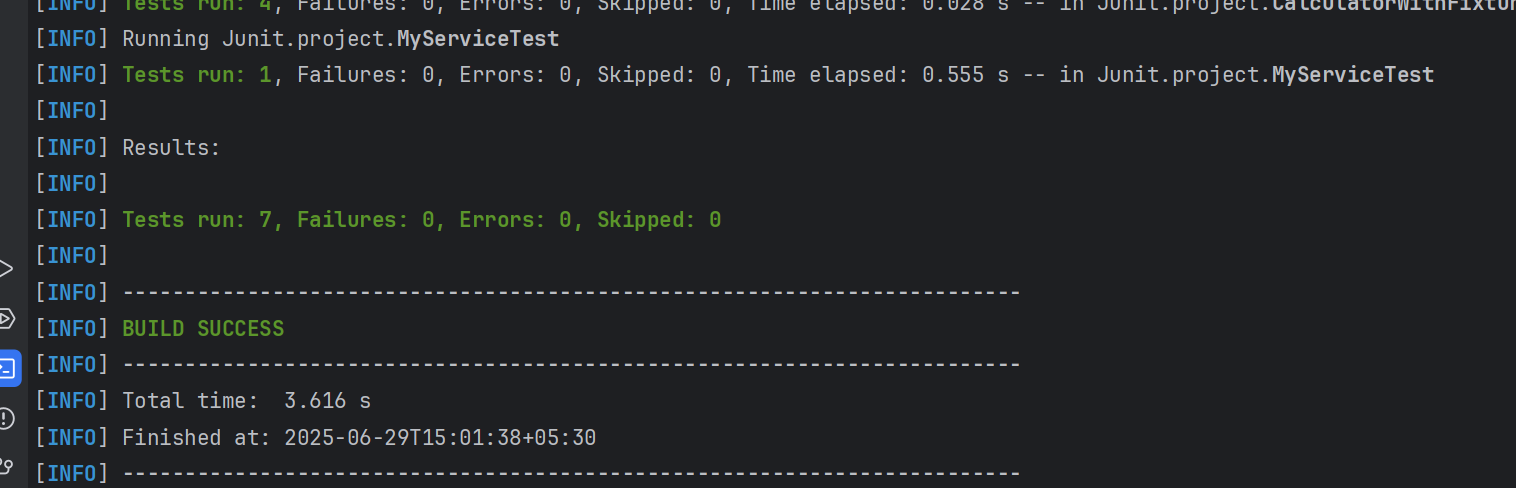
**--MyService.java**

public class MyService {  
 private ExternalApi api;  
  
 public MyService(ExternalApi api) {  
 this.api = api;  
 }  
  
 public String fetchData() {  
 return api.getData();  
 }  
}

**--MyServiceTest.java**

import static org.mockito.Mockito.\*;  
import static org.junit.Assert.assertEquals;  
  
import org.junit.Test;  
import org.mockito.Mockito;  
  
public class MyServiceTest {  
 @Test  
 public void testExternalApi() {  
 // Step 1: Create a mock object  
 ExternalApi mockApi = Mockito.mock(ExternalApi.class);  
  
 // Step 2: Stub the method  
 when(mockApi.getData()).thenReturn("Mock Data");  
  
 // Step 3: Inject mock and test  
 MyService service = new MyService(mockApi);  
 String result = service.fetchData();  
  
 // Assert  
 assertEquals("Mock Data", result);  
 }  
}

Output



**Ques 2 : Verifying Interaction**

**Code :**

**--MyServiceTest.java**

package Junit.project;  
  
import static org.mockito.Mockito.\*;  
import org.junit.jupiter.api.Test;  
import org.mockito.Mockito;  
import static org.junit.jupiter.api.Assertions.\*;  
  
// Mocked External API  
interface ExternalApi {  
 String getData();  
}  
  
// Service class that uses ExternalApi  
class MyService {  
 private ExternalApi api;  
  
 public MyService(ExternalApi api) {  
 this.api = api;  
 }  
  
 public String fetchData() {  
 return api.getData();  
 }  
}  
  
// Test class using Mockito to verify interaction  
public class MyServiceTest {  
  
 @Test  
 public void testVerifyInteraction() {  
 // Step 1: Create a mock object  
 ExternalApi mockApi = Mockito.*mock*(ExternalApi.class);  
  
 // Step 2: Call the method with specific arguments (mocked call)  
 MyService service = new MyService(mockApi);  
 service.fetchData();  
  
 // Step 3: Verify the interaction  
 *verify*(mockApi).getData(); // this checks if getData() was called on mockApi  
 }  
}

Output

