**WEEK-2 MOCKITO SOLUTIONS**

**Exercise 1: Mocking and Stubbing**

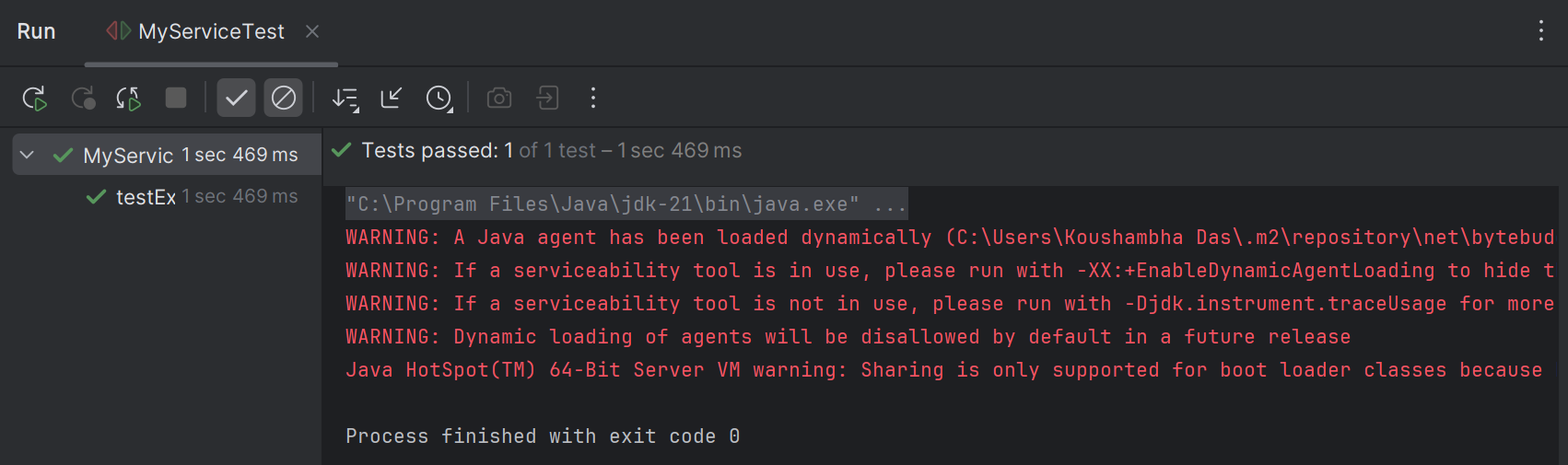
**CODE:**

package org.example.exc1;  
  
public interface ExternalApi {  
 String getData();  
}

package org.example.exc1;  
  
public class MyService {  
 private final ExternalApi externalApi;  
  
 public MyService(ExternalApi externalApi) {  
 this.externalApi = externalApi;  
 }  
  
 public String fetchData() {  
 return externalApi.getData();  
 }  
}

package org.example.exc1;  
  
import org.junit.Test;  
import org.mockito.Mockito;  
  
import static org.junit.Assert.*assertEquals*;  
import static org.mockito.Mockito.*when*;  
  
public class MyServiceTest {  
  
 @Test  
 public void testExternalApi() {  
 ExternalApi mockApi = Mockito.*mock*(ExternalApi.class);  
 *when*(mockApi.getData()).thenReturn("Mock Data");  
 MyService service = new MyService(mockApi);  
 String result = service.fetchData();  
 *assertEquals*("Mock Data", result);  
 }  
}

**OUTPUT**:



**Exercise 2: Verifying Interactions**

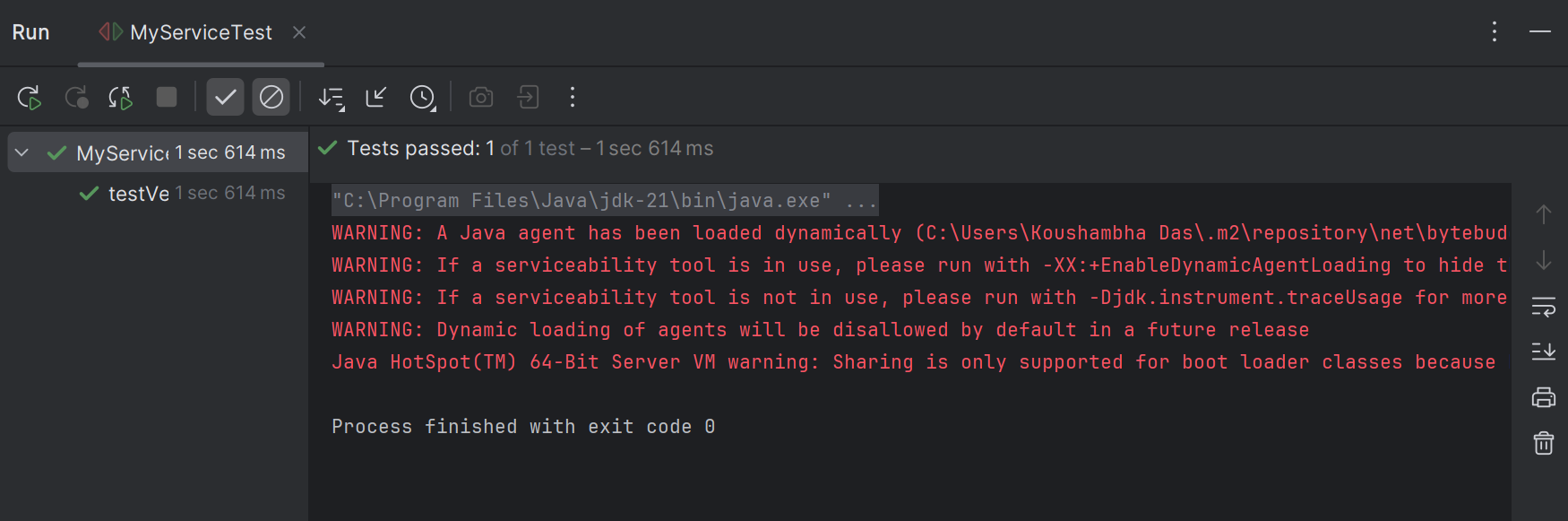
**Code:**

package org.example.exc1;  
  
public interface ExternalApi {  
 String getData();  
}

package org.example.exc1;  
  
public class MyService {  
 private final ExternalApi externalApi;  
  
 public MyService(ExternalApi externalApi) {  
 this.externalApi = externalApi;  
 }  
  
 public String fetchData() {  
 return externalApi.getData();  
 }  
}

package org.example.exc2;  
  
import static org.mockito.Mockito.\*;  
  
import org.example.exc1.ExternalApi;  
import org.example.exc1.MyService;  
import org.junit.jupiter.api.Test;  
import org.mockito.Mockito;  
  
public class MyServiceTest {  
 @Test  
 public void testVerifyInteraction() {  
 ExternalApi mockApi = Mockito.*mock*(ExternalApi.class);  
 MyService service = new MyService(mockApi);  
 service.fetchData();  
 *verify*(mockApi).getData();  
 }  
}

**Output:**

****