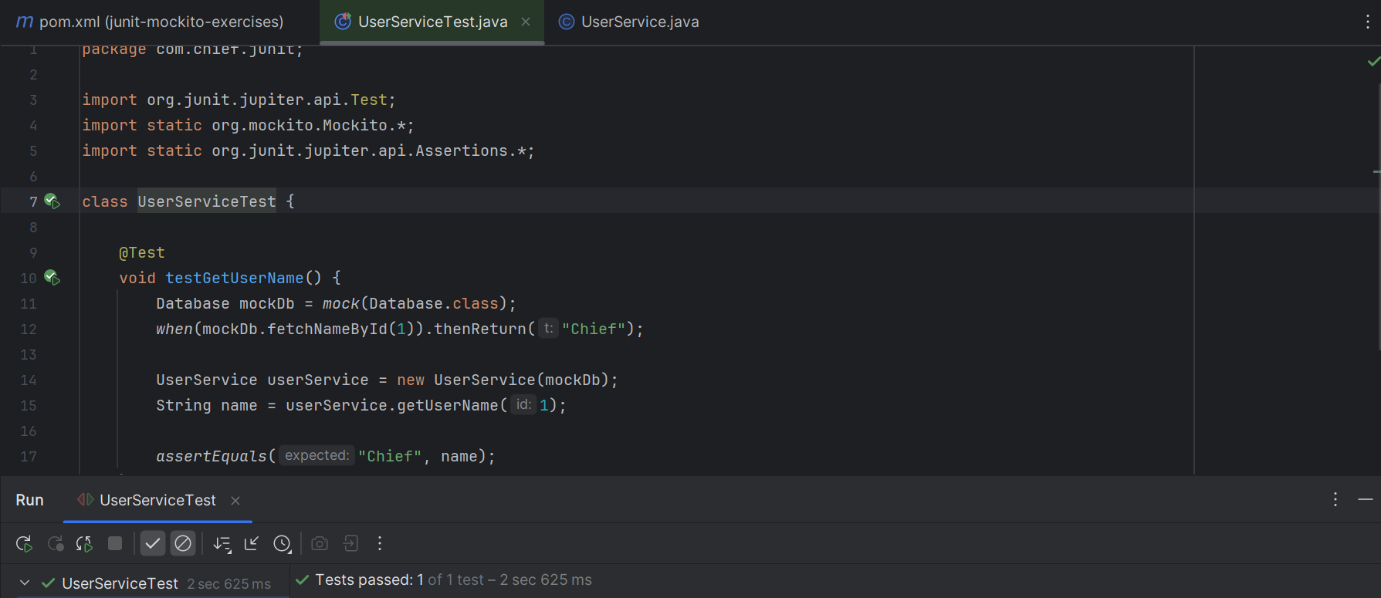
MOCKITO & SFL4J

Exercise 1:

package com.chief.junit;  
  
import org.junit.jupiter.api.Test;  
import static org.mockito.Mockito.\*;  
import static org.junit.jupiter.api.Assertions.\*;  
  
class UserServiceTest {  
  
 @Test  
 void testGetUserName() {  
 Database mockDb = *mock*(Database.class);  
 *when*(mockDb.fetchNameById(1)).thenReturn("Chief");  
  
 UserService userService = new UserService(mockDb);  
 String name = userService.getUserName(1);  
  
 *assertEquals*("Chief", name);  
 }  
}



Exercise 2:

package com.chief.junit;  
  
import org.junit.jupiter.api.Test;  
import static org.mockito.Mockito.\*;  
  
public class InteractionTest {  
  
 @Test  
 void testInteractionWithMock() {  
 Database mockDb = *mock*(Database.class);  
 mockDb.fetchNameById(100);  
 *verify*(mockDb, *times*(1)).fetchNameById(100);  
 }  
}



Exercise S LF4J:

package com.chief.junit;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
public class LoggingExample {  
  
 private static final Logger *logger* = LoggerFactory.*getLogger*(LoggingExample.class);  
  
 public static void main(String[] args) {  
 *logger*.error("❌ This is an error message");  
 *logger*.warn("⚠️ This is a warning message");  
 }  
}

