

2da. Parte.

Nombre Completo: CI: Fecha:...../...../.....

6. Dado el proyecto “logeapi” realizar el **test simulando se tiene la funcionalidad implementada de la clase ICuenta** a la cual accedemos través de su interface. (se debe lograr a una cobertura de 100%)

Proje... x FavoritesFilesServices

logeapi

Source Packages

com.com470.mockito1app

Mockito1appApplication.java

com.com470.mockito1app.controller

ExcepcionCuentaEnUso.java

ExcepcionUsuarioDesconocido.java

ICuenta.java

IRepositorioCuentas.java

com.com470.mockito1app.service

GestorLogin.java

Test Packages

com.com470.mockito1app

com.com470.mockito1app.service

GestorLoginTest.java

Other Sources

Dependencies

Runtime Dependencies

Test Dependencies

Java Dependencies

Project Files

pom.xml

nbactions.xml

rbc2ppApi

rbc2ppApp

rejpapp

spring-boot-data-jpa

usoJUnit

webrejp

wsrejp

Output - Test (GestorLogin) xCode Coverage Report for logeoapi xGestorLogin.java x

SourceHistory

1package com.com470.mockitolapp.controller;

2

3

4public interface ICuenta {

5

6boolean claveCorrecta(String candidata);

7void entrarCuenta();

8void bloquearCuenta();

9boolean estaBloqueada();

10boolean estaEnUso();

11}

12

13

Code Coverage:0,00 %

Test Results x

com.com470:logeapi:jar:0.0.1-SNAPSHOT x

Tests passed: 100,00 %

All 10 tests passed. (0,283 s)

com.com470.mockito1app.service.GestorLoginTest passed

testUsuarioIncorrecto passed (0,188 s)

testAccesoDenegadoCuentaBloqueada passed (0,005 s)

testAccederTrasDosIntentos passed (0,003 s)

testBloquearTrasTresIntentos passed (0,001 s)

testBloquearTrasCuatroIntentos passed (0,001 s)

testCuentaEnUso passed (0,001 s)

testAccesoConcedidoALaPrimera passed (0,0 s)

testAccederTrasBloqueoConOtroUsuario passed (0,002 s)

testAccesoDenegadoALaPrimera passed (0,0 s)

testAccederTrasUnIntento passed (0,0 s)

logeapi

Source Packages

com.com470.mockito1app

Mockito1appApplication.java

com.com470.mockito1app.controller

ExcepcionCuentaEnUso.java

ExcepcionUsuarioDesconocido.java

ICuenta.java

IRepositorioCuentas.java

com.com470.mockito1app.service

GestorLogin.java

Test Packages

com.com470.mockito1app

com.com470.mockito1app.service

GestorLoginTest.java

Other Sources

Dependencies

Runtime Dependencies

Test Dependencies

Java Dependencies

Project Files

pom.xml

nbactions.xml

rbc2ppApi

rbc2ppApp

rejpapp

spring-boot-data-jpa

usoJUnit

webrejp

wsrejp

Total Coverage:Output - Test (GestorLogin)

FilenameCoverage

com.com470.mockito1app.Mockito1appApplication0,00 %

com.com470.mockito1app.service.GestorLogin100,00 %

com.com470.mockito1app.controller.ExcepcionUsuarioDesconocido100,00 %

com.com470.mockito1app.controller.ExcepcionCuentaEnUso100,00 %

Total88,00 %

Clear ResultsRun All Tests

logeapi

Source Packages

com.com470.mockito1app

Mockito1appApplication.java

com.com470.mockito1app.controller

ExcepcionCuentaEnUso.java

ExcepcionUsuarioDesconocido.java

ICuenta.java

IRepositorioCuentas.java

com.com470.mockito1app.service

GestorLogin.java

Test Packages

com.com470.mockito1app

com.com470.mockito1app.service

GestorLoginTest.java

Other Sources

Dependencies

Runtime Dependencies

Test Dependencies

Java Dependencies

Project Files

pom.xml

nbactions.xml

rbc2ppApi

rbc2ppApp

rejpapp

spring-boot-data-jpa

usoJUnit

webrejp

wsrejp

SourceHistory

1package com.com470.mockitolapp.service;

2

3import com.com470.mockitolapp.controller.ExcepcionCuentaEnUso;

4import com.com470.mockitolapp.controller.ICuenta;

5import com.com470.mockitolapp.controller.IRepositorioCuentas;

6import com.com470.mockitolapp.controller.ExcepcionUsuarioDesconocido;

7

8public class GestorLogin {

9

10private IRepositorioCuentas repo;

11private int numFallos;

12private static final int MAX_FALLOS = 3;

13private String usuarioAnterior = "";

14

15public GestorLogin(IRepositorioCuentas repo) {

16this.repo = repo;

17numFallos = 0;

18}

19

20public void acceder(String usuario, String clave) {

21ICuenta cuenta = repo.buscar(usuario);

22

23if (cuenta == null)

24throw new ExcepcionUsuarioDesconocido();

25

26if (cuenta.estaEnUso())

27throw new ExcepcionCuentaEnUso();

28}

Code Coverage:100,00 %

Ejemplo de los test a realizar.

```
@Mock
ICuenta cuenta;

@Before
public void inicializarTest() {
    when(repo.buscar("pepe")).thenReturn(cuenta);
}

@Test
public void testAccesoConcedidoALaPrimera() {
    when(cuenta.claveCorrecta("1234")).thenReturn(true);
    /*completar */
}

@Test
public void testAccesoDenegadoALaPrimera() {
    /*completar */
}

@Test(expected = ExcepcionUsuarioDesconocido.class)
public void testUsuarioIncorrecto() {
    when(repo.buscar("bartolo")).thenReturn(null);
    /*completar */
}

@Test
public void testBloquearTrasTresIntentos() {
    /*completar */
}

@Test
public void testAccederTrasDosIntentos() {
    /*completar */
}

@Test
public void testAccederTrasUnIntento() {
    /*completar */
}

@Test
public void testBloquearTrasCuatroIntentos() {
    when(cuenta.claveCorrecta("1234")).thenReturn(false);
    GestorLogin login = new GestorLogin(repo);
    login.acceder("pepe", "1234");
    /*completar */
}

@Test
public void testAccesoDenegadoCuentaBloqueada() {
    /*completar */
}

@Test(expected = ExcepcionCuentaEnUso.class)
public void testCuentaEnUso() {
    /*completar */
}
```

Descargar el proyecto de:

<https://github.com/jcbergman/loggeoapi>

Buena suerte!!!

Nota. - Deben subir a la plataforma ecampus lo sgte:

- El archivo test.java del test realizado.
 - La captura del test de métodos y cobertura alcanzado, (jpg o pdf)
 - El enlace donde está depositado si código en GitHub (para comprobar su ejecución)
- Si omiten algo obviamente la nota es menor, se verificará si existe repetidos de código del test/