测试对象：UserServiceImpl类

测试人员：刘畅

测试日期：2020/6/22

# 一、测试对象描述

该类作为service层的一个业务类，其主要功能为调用Dao层中的UserDao接口使用JAVA持久层API对数据库进行增删改查操作，涉及到的数据库表有User表。

为了方便测试，我们使用mock对UserDao进行接口的模拟，作为测试对象的桩来进行测试。

该类需要测试的方法如下：

|  |  |  |
| --- | --- | --- |
| 方法标识符 | 方法名 | 测试关注点 |
| Unit\_001\_Function\_001 | User findByUserID(String userID) | 返回的结果与UserDao中findByUserID函数返回结果是否符合。 |
| Unit\_001\_Fuction\_002 | User findById(int id) | 返回的结果与UserDao中findById函数返回结果是否符合。 |
| Unit\_001\_Fuction\_003 | Page<User> findByUserID(Pageable pageable) | 是否能正确调用UserDao中的findByUserID（pageable）函数 |
| Unit\_001\_Function\_004 | User checkLogin(String userID, String password) | 返回的结果与UserDao中findByUserIDAndPassword函数返回结果是否符合。 |
| Unit\_001\_Fuction\_005 | int create(User user) | 是否能正确调用userDao中的save函数和findAll函数 |
| Unit\_001\_Fuction\_006 | void delByID(int id) | 是否能正确调用userDao中的deleteById函数 |
| Unit\_001\_Fuction\_007 | void updateUser(User user) | 是否能正确调用userDao中的save函数 |
| Unit\_001\_Fuction\_008 | int countUserID(String userID) | 是否能正确调用userDao中countByUserID并返回正确的结果 |

# 二、测试用例设计

## 2.1、Unit\_001\_Fuction\_001

待测试的方法为UserServiceImpl类中的findByUserID(String userID)方法

该方法即为调用UserDao层findByUserID函数返回User类型的函数，为确认该方法是否正确调用函数和返回正确参数值，我们只需设计一个测试用例

测试用例表如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 驱动函数 | 输入变量 | 预期输出 | 执行结果 |
| 1 | return\_user\_by\_UserID() | String userID="user"; | res.getUserID() res.getPassword() res.getEmail() res.getPhone() isadmin,res.getIsadmin() res.getUserName() res.getPicture()匹配  *verify*(userDao).findByUserID(userID);通过 |  |

## 2.2、Unit\_001\_Fuction\_002

待测试的方法为UserServiceImpl类中的findById(int id)方法

该方法即为调用UserDao层findByID函数返回User类型的函数，为确认该方法是否正确调用函数和返回正确参数值，我们只需设计一个测试用例

测试用例如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 驱动函数 | 输入变量 | 预期输出 | 执行结果 |
| 1 | return\_User\_by\_id() | int id=2; | res.getUserID() res.getPassword() res.getEmail() res.getPhone() isadmin,res.getIsadmin() res.getUserName() res.getPicture()匹配  *verify*(userDao).findById(id);通过 |  |

## 2.3、Unit\_001\_Fuction\_003

待测试的方法为UserServiceImpl类中的findByUserID(Pageable pageable)方法

该方法即为调用UserDao层findAllByIsAdmin函数，返回pageable类型的User信息的函数，我们只需判断该方法是否调用findAllByIsAdmin函数

测试用例如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 驱动函数 | 输入变量 | 预期输出 | 执行结果 |
| 1 | return\_user\_list\_paged() | Pageable pageable= PageRequest.*of*(0,10); | *verify*(userDao).findAllByIsadmin(0,pageable);  通过 |  |

## 2.4、Unit\_001\_Function\_004

待测试的方法为UserServiceImpl类中的checkLogin(String userID, String password)方法

该方法即为调用UserDao层findByUserIDAndPassword函数，返回User类型的函数，为测试该方法是否调用正确的函数和返回正确的参数，我们设计如下测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 驱动函数 | 输入变量 | 预期输出 | 执行结果 |
| 1 | check\_userID\_password\_matched() | String userID="user"; String password="password"; | res.getUserID() res.getPassword() res.getEmail() res.getPhone() isadmin,res.getIsadmin() res.getUserName() res.getPicture()匹配  *verify*(userDao).findByUserIDAndPassword(userID,password);通过 |  |

## 2.5、Unit\_001\_Function\_005

待测试的方法为UserServiceImpl类中的create（User user）方法

该方法即为调用UserDao层的save函数和findAll函数，返回findAll函数结果的函数，为测试该方法是否调用正确的函数，我们设计如下测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 驱动函数 | 输入变量 | 预期输出 | 执行结果 |
| 1 | user\_register() | int id=1; String userID="user"; String password="password"; String email="222@qq.com"; String phone="12345678901"; int isadmin=0; String user\_name="nickname"; String picture="picture"; User user=new User(id,userID,user\_name,password,email,phone,isadmin,picture); | *verify*(userDao).save(user); *verify*(userDao).findAll();通过 |  |

## 2.6、Unit\_001\_Fuction\_006

待测试的方法为UserServiceImpl类中的delByID(int id)方法

该方法即为调用UserDao层的deleteById函数，为测试该方法是否调用正确的函数，我们设计如下测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 驱动函数 | 输入变量 | 预期输出 | 执行结果 |
| 1 | del\_user\_by\_id() | id=1;  id=-1 | *verify*(userDao).deleteById(1); *verify*(userDao).deleteById(-1); *verify*(userDao,*times*(2)).deleteById(*anyInt*());通过 |  |

## 2.7、Unit\_001\_Function\_007

待测试的方法为UserServiceImpl类中的updateUser(User user)方法

该方法即为调用UserDao层的save函数，为测试该方法是否调用正确的函数，我们设计如下测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 驱动函数 | 输入变量 | 预期输出 | 执行结果 |
| 1 | update\_user\_info() | int id=1; String userID="user"; String password="password"; String email="222@qq.com"; String phone="12345678901"; int isadmin=0; String user\_name="nickname"; String picture="picture"; User user=new User(id,userID,user\_name,password,email,phone,isadmin,picture); | *verify*(userDao).save(user);通过 |  |

## 2.8、Unit\_001\_Function\_008

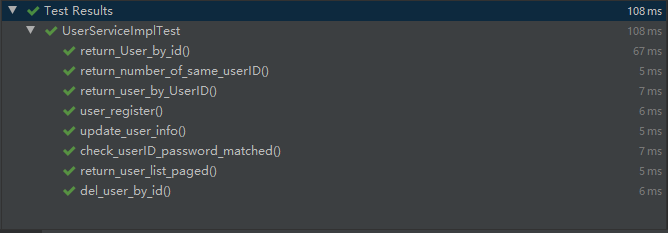
待测试的方法为UserServiceImpl类中的countUserID(String userID)方法

该方法即为调用UserDao层的countByUserID函数，为测试该方法是否调用正确的函数，我们设计如下测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 驱动函数 | 输入变量 | 预期输出 | 执行结果 |
| 1 | return\_number\_of\_same\_userID() | String userID="user"; | *verify*(userDao,*times*(2)).countByUserID(userID);通过 |  |

# 三、测试用例的执行与结果

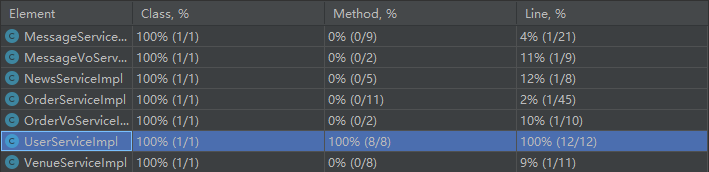
测试用例执行如下图所示：



所有测试用例均通过测试。

# 四、测试结果分析

测试用例对于UserServiceImpl类的覆盖度如下图所示：



从该图可得知，测试对于UserServiceImpl类的方法覆盖度（Method）达到100%，对其语句覆盖度（Line）也达到100%，经测试经理评审，该单元通过了单元测试。