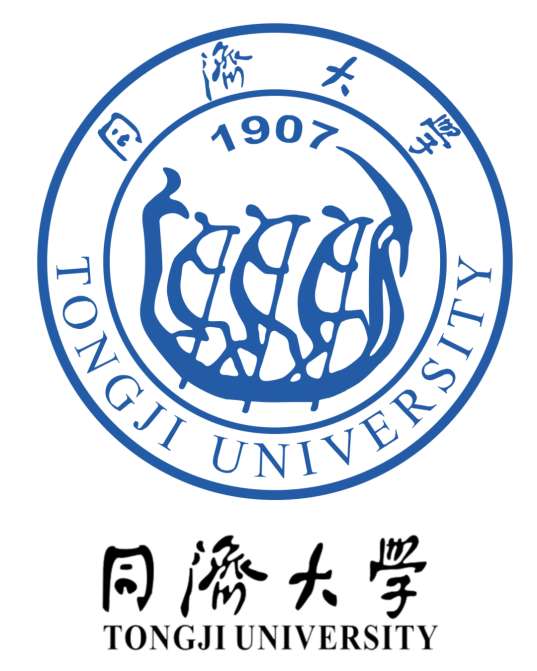
同济微博系统

软件测试项目 单元测试文档



1751324 贺思奇

1752079 尤 皓

1551531 吴国栋

1652657 刘 寅

目录

[1. 引言 3](#_Toc44452572)

[1.1 文档目的 3](#_Toc44452573)

[1.2 预期读者 3](#_Toc44452574)

[2. 项目概述 4](#_Toc44452575)

[2.1 系统结构 4](#_Toc44452576)

[2.2 主体功能 4](#_Toc44452577)

[2.3 其他功能 5](#_Toc44452578)

[3. 参考资料 5](#_Toc44452579)

[4. 测试背景 5](#_Toc44452580)

[4.1 测试说明 5](#_Toc44452581)

[4.2 测试方法 6](#_Toc44452582)

[4.3 测试对象 6](#_Toc44452583)

[4.4 测试对象分析 6](#_Toc44452584)

[4.4.1 user controller模块 7](#_Toc44452585)

[4.4.2 relation controller 模块 7](#_Toc44452586)

[4.4.3 blog-at controller模块 7](#_Toc44452587)

[4.4.4 blog home/profile/square controller 类模块 8](#_Toc44452588)

[5. 用例分析与实现 8](#_Toc44452589)

[5.1 user controller 模块用例分析与实现 8](#_Toc44452590)

[5.1.1 register用例分析与实现 8](#_Toc44452591)

[5.1.2 login 用例分析与实现 10](#_Toc44452592)

[5.1.3 changeInfo 用例分析与实现 11](#_Toc44452593)

[5.1.4 changePassword 用例分析与实现 12](#_Toc44452594)

[5.1.5 deleteUser 用例分析与实现 14](#_Toc44452595)

[5.1.6 logout 用例分析与实现 16](#_Toc44452596)

[5.2 relation controller 模块用例分析与实现 17](#_Toc44452597)

[5.2.1 follow 用例分析与实现 17](#_Toc44452598)

[5.2.2 unfollow 用例分析与实现 18](#_Toc44452599)

[5.2.3 getFans 用例分析与实现 19](#_Toc44452600)

[5.2.4 getFollowers 用例分析与实现 20](#_Toc44452601)

[5.3 blog-at controller模块用例分析与实现 21](#_Toc44452602)

[5.3.1 getAtMeCount用例分析与实现 21](#_Toc44452603)

[5.3.2 getAtMeBlogList用例分析与实现 23](#_Toc44452604)

[5.3.3 markAsRead用例分析与实现 24](#_Toc44452605)

[5.4 blog home/profile/square controller模块用例分析与实现 26](#_Toc44452606)

[5.4.1 Atorage\_BLOG\_ID用例分析与实现 26](#_Toc44452607)

[5.4.2 Load\_Blog\_ID用例分析与实现 28](#_Toc44452608)

[5.4.3 Load\_homepage用例分析与实现 29](#_Toc44452609)

[6. 回归测试 30](#_Toc44452610)

# 引言

## 文档目的

本文档为“慕课-微博系统”单元测试活动划定测试的范围，提供方法、工具以及进度上的指导。主要阐述单元测试活动的过程、如何控制单元测试、单元测试的规划以及记录测试的结果。保证项目的单元测试过程能够准确面向需求文档中的功能需求，并覆盖其中绝大部分的要求。记录测试过程中的发现的缺陷和错误，跟踪其处理过程，给后续的集成测试、系统测试以及系统的运维提供支持和帮助。

## 预期读者

本文档的预期读者包括但是不限于：

1. 测试人员：指导和核查单元测试人员的工作，对后续的测试人员提供测试资料，以帮助其更快地明确测试重点，设计测试方案。
2. 开发人员：指导项目的开发者修正项目缺陷和错误，补全需求文档中列举的功能点，指导后续的开发过程。
3. 项目经理：对整个项目的流程进行监管，方便经理对测试工作进行系统地安排。

我们强烈地希望读者能够在一定程度上掌握如下的相关知识，这能够帮助读者更好地理解本文档。

（1）熟悉计算机的基本网络操作。

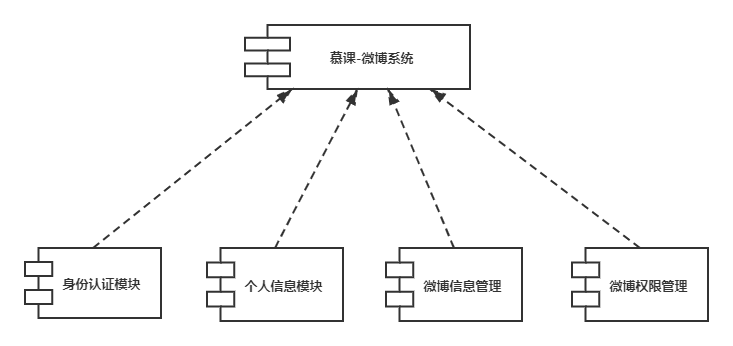
（2）熟悉当前常用的测试工具。

# 项目概述

## 系统结构

系统由以下四部分组成：

1. 身份认证系统
2. 个人信息系统
3. 微博信息系统
4. 微博权限系统



## 主体功能

* 修改自己的展示信息（头像、昵称等）
* 发布微博内容
* 对微博进行留言回复，被回复的用户会收到通知
* 关注/取关用户，以及相关的内容展示

## 其他功能

* 注册登录功能
* 修改密码功能

# 参考资料

1. 《高级软件测试技术》
2. 《Software Testing Foundations》
3. 《慕课-微博系统概要设计规约》
4. 《慕课-微博系统详细设计规约》

# 4. 测试背景

## 4.1 测试说明

本测试主要是针对本系统的单元测试，不包括硬件、系统测试以及之后的集成测试。本测试能够进行的主要前提是代码编写的完成。

测试需要完成的主要内容包括以下几个方面：

1. 对项目后端每个模块中的函数进行测试，确保其能正确响应，返回结果。
2. 验证结果的正确性，包括在输入非法数据时，函数能正确的判断，并返回预设的结果。
3. 对单元测试过程中发现的缺陷和错误进行记录跟踪，在处理后进行回归测试。

## 测试方法

本次测试综合考虑了软件测试课程中涉及到的各种测试方法。

（1）保证所有的语句、分支被覆盖；

（2）参考等价类划分方法；

（3）参考边界值分析方法；

（4）参考使用错误猜测方法；

（5）测试脚本使用 Javascript 语言实现，在node.js的SuperTest开发单元测试模块下进行测试代码编写，使用Jest作为测试框架，和驱动 （Drive）代码、桩（Stub）代码构成一个可执行系统；

（6）覆盖率使用Jest自带的覆盖率测试工具完成；

（7）某函数的缺陷被修正后必须回归与该函数相关的所有单元测试用例。

## 4.3 测试对象

（1） user controller类

（2） relation controller类

（3） blog-at controller 类

（4） blog home/profile/square controller 类

## 4.4 测试对象分析

我们面向“慕课-微博系统”的后端业务逻辑层，针对其中的函数进行测试。Node.js后端开发的架构为FaaS(Function as a Service)，因此对其中每个函数的测试，即可转化为对整个项目业务逻辑的测试。只要覆盖了controller层的所有代码，即实现了100%覆盖率的单元测试。在测试对象的安排时，将blog-home, blog-profile和blog-square三个功能相对应的模块组合在一起，进行测试。

### 4.4.1 user controller模块

|  |  |  |
| --- | --- | --- |
| 标志符 | 名称 | 代码行（LOC） |
| UN\_USR\_001 | register(username, password, gender) | 41 |
| UN\_USR\_002 | login(ctx, username,password) | 36 |
| UN\_USR\_003 | changeInfo(ctx, { nickname, city, picture }) | 58 |
| UN\_USR\_004 | changePassword( username, password, newPassword) | 48 |
| UN\_USR\_005 | deleteCurUser(username) | 16 |
| UN\_USR\_006 | Logout(ctx) | 5 |

### 4.4.2 relation controller 模块

|  |  |  |
| --- | --- | --- |
| 标志符 | 名称 | 代码行（LOC） |
| UN\_REL\_001 | Follow(myUserId, curUserId) | 13 |
| UN\_REL\_002 | unfollow(myUserId, curUserId) | 22 |
| UN\_REL\_003 | getFans(userId) | 37 |
| UN\_REL\_004 | getFollowers(userId) | 40 |

### 4.4.3 blog-at controller模块

|  |  |  |
| --- | --- | --- |
| 标志符 | 名称 | 代码行（LOC） |
| UN\_GMC\_001 | getAtMeCount(userId) | 29 |
| UN\_GMC\_002 | getAtMeBlogList(userId, pageIndex = 0) | 28 |
| UN\_MAR\_001 | markAsRead(userId) | 42 |

### 4.4.4 blog home/profile/square controller 类模块

|  |  |  |
| --- | --- | --- |
| 标志符 | 名称 | 代码行（LOC） |
| UN\_ABI\_001 | Atorage\_Blog\_ID  (blog\_ID,user\_id) | 25 |
| UN\_LBD\_001 | Load\_Blog\_ID  (blog\_ID,user\_id) | 24 |
| UN\_LBD\_002 | Load\_homepage(userId) | 38 |
| UN\_LBD\_003 | Load\_firset\_pagedata(useID,data,  picture) | 72 |
| UN\_LBD\_004 | Load\_squarepage(userID,data,  picture) | 51 |

# 用例分析与实现

## 5.1 user controller 模块用例分析与实现

### 5.1.1 register用例分析与实现

a.标识符定义

UN\_USR\_001

b.被测特性

（1）检测register函数的字符检查模块

（2）检测register函数对重复注册的识别

（3）检测register函数对正确输入的响应

c.测试方法

（1）传入错误数据的注册信息，检查能否返回error!=0

（2）传入正确的注册信息，检查能否返回error=0

（3）传入重复的注册信息，检查能否返回error!=0

d. 测试项标示

|  |  |  |
| --- | --- | --- |
| 测试项标志符 | 测试项描述 | 优先级 |
| UN\_USR\_001\_001 | 传入错误数据的注册信息，检查能否返回error!=0 | 高 |
| UN\_USR\_001\_002 | 传入正确的注册信息，检查能否返回error=0 | 高 |
| UN\_USR\_001\_003 | 传入重复的注册信息，检查能否返回error!=0 | 高 |

e.测试通过/失败标准

所有的用例都必须被执行，且没有发现错误。

f .对应用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | UN\_USR\_001 | | | | | |
| 优先级 | 高 | | | | | |
| 测试项描述 | 调用register函数，传入注册信息，检查返回值中的error | | | | | |
| 前置条件 | 无 | | | | | |
| 用例编号 | 测试方法 | 输入参数 | 预期结果 | 真实结果 | 是否有BUG | 是否解决 |
| 001 | 等价类 | 错误注册数据 | Error=10002 | Error=10002 | 无 | / |
| 002 | 等价类 | 正确注册数据 | Error=0 | Error=0 | 无 | / |
| 003 | 等价类 | 重复注册数据 | Error=10001 | Error=10001 | 无 | / |

### 5.1.2 login 用例分析与实现

a.标识符定义

UN\_USR\_002

b.被测特性

（1）检测用户名输入错误的情况

（2）检测用户密码输入错误的情况

（3）检测输入正确数据的情况

（4）检测重复登录的情况

c.测试方法

白盒测试，以登录函数的每一个分支作为一个等价类，进行测试

d. 测试项标示

|  |  |  |
| --- | --- | --- |
| 测试项标志符 | 测试项描述 | 优先级 |
| UN\_USR\_002\_001 | 传入不存在的用户名 | 中 |
| UN\_USR\_002\_002 | 传入错误的密码 | 高 |
| UN\_USR\_002\_003 | 传入正确的用户名和密码，且用户未登录 | 高 |
| UN\_USR\_002\_004 | 传入已登录的用户的用户名和密码 | 中 |

e.测试通过/失败标准

所有的用例都必须被执行，且没有发现错误。

f .对应用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | UN\_USR\_002 | | | | | |
| 优先级 | 高 | | | | | |
| 测试项描述 | 调用login函数，传入用户信息，检查返回值中的error | | | | | |
| 前置条件 | 数据库中存在用户信息 | | | | | |
| 用例编号 | 测试方法 | 输入参数 | 预期结果 | 真实结果 | 是否有BUG | 是否解决 |
| 001 | 等价类 | UserName=100，  Password=“123” | Error=10004 | Error=10004 | 无 | / |
| 002 | 等价类 | Username=“abc”,  Password=”1234” | Error=10004 | Error=10004 | 无 | / |
| 003 | 等价类 | UserName=”abc”  Password=”123” | Error=0 | Error=0 | 无 | / |
| 003 | 等价类 | UserName=”abc”  Password=”123” | Error=0 | Error=0 | 无 | / |

### 5.1.3 changeInfo 用例分析与实现

a.标识符定义

UN\_USR\_003

b.被测特性

（1）检测输入完整信息的结果

（2）检测输入不完整信息的结果

（2）检测传入图片过大的结果

c.测试方法

白盒测试，以changeinfo函数的每一个分支作为一个等价类，进行测试

d. 测试项标示

|  |  |  |
| --- | --- | --- |
| 测试项标志符 | 测试项描述 | 优先级 |
| UN\_USR\_003\_001 | 输入完成的info信息 | 高 |
| UN\_USR\_003\_002 | 输入不完成的info信息 | 中 |
| UN\_USR\_003\_003 | 传入的图片大于1M | 中 |

e.测试通过/失败标准

所有的用例都必须被执行，且没有发现错误。

f .对应用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | UN\_USR\_003 | | | | | |
| 优先级 | 高 | | | | | |
| 测试项描述 | 调用changeinfo函数，传入信息，检查返回值中的error | | | | | |
| 前置条件 | 数据库中存在用户信息 | | | | | |
| 用例编号 | 测试方法 | 输入参数 | 预期结果 | 真实结果 | 是否有BUG | 是否解决 |
| 001 | 等价类 | 不含city的info信息 | Error=10009 | Error=10009 | 无 | / |
| 002 | 等价类 | 包含所有的info信息 | Error=0 | Error=0 | 无 | / |
| 002 | 等价类 | Img信息大于1M | Error=10007 | Error=0 | 有 | 回归测试 |

### 5.1.4 changePassword 用例分析与实现

a.标识符定义

UN\_USR\_004

b.被测特性

（1）检测输入错误原密码的结果

（2）检测输入正确原密码，新密码的结果

（3）检测输入新密码长度小于3的结果

（4）检测输入新密码长度等于3的结果

c.测试方法

白盒测试，以密码的schema 检测代码的条件为边界值，进行测试

d. 测试项标示

|  |  |  |
| --- | --- | --- |
| 测试项标志符 | 测试项描述 | 优先级 |
| UN\_USR\_004\_001 | 输入错误的原密码 | 高 |
| UN\_USR\_004\_002 | 输入正确的原密码，新密码长度大于3 | 高 |
| UN\_USR\_004\_003 | 输入的新密码长度小于3 | 中 |
| UN\_USR\_004\_004 | 输入的新密码长度等于3 | 中 |

e.测试通过/失败标准

所有的用例都必须被执行，且没有发现错误。

f .对应用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | UN\_USR\_004 | | | | | |
| 优先级 | 高 | | | | | |
| 测试项描述 | 调用chagePassword函数，传入信息，检查返回值中的error | | | | | |
| 前置条件 | 数据库中存在用户信息 | | | | | |
| 用例编号 | 测试方法 | 输入参数 | 预期结果 | 真实结果 | 是否有BUG | 是否解决 |
| 001 | 等价类 | OldPassword=”1234” | Error=10006 | Error=10006 | 无 | / |
| 002 | 边界值 | OldPassword=”123”  NewPassword=”1234” | Error=0 | Error=0 | 无 | / |
| 003 | 边界值 | OldPassword=”1234”  NewPassword=”12” | Error=10009 | Error=10009 | 无 | / |
| 004 | 边界值 | OldPassword=”1234”  NewPassword=”123” | Error=0 | Error=0 | 无 | / |

### 5.1.5 deleteUser 用例分析与实现

a.标识符定义

UN\_USR\_005

b.被测特性

（1）传入一个用户信息，根据id删除该用户

c.测试方法

白盒测试，以deleteUser函数的每一个分支作为一个等价类，进行测试

d. 测试项标示

|  |  |  |
| --- | --- | --- |
| 测试项标志符 | 测试项描述 | 优先级 |
| UN\_USR\_005\_001 | 传入已登录的用户信息。 | 低 |

e.测试通过/失败标准

所有的用例都必须被执行，且没有发现错误。

f .对应用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | UN\_USR\_005 | | | | | |
| 优先级 | 低 | | | | | |
| 测试项描述 | 调用deleteUser函数，传入用户信息，检查返回值中的error | | | | | |
| 前置条件 | 数据库中存在用户信息,且用户已经登录 | | | | | |
| 用例编号 | 测试方法 | 输入参数 | 预期结果 | 真实结果 | 是否有BUG | 是否解决 |
| 001 | 等价类 | UserName=“abc”，  Password=“123”， | Error=0 | Error=10010 | 有 | 回归测试 |

### 5.1.6 logout 用例分析与实现

a.标识符定义

UN\_USR\_006

b.被测特性

（1）传入一个用户信息，根据id登出该用户（在redis中注销记录）

c.测试方法

白盒测试，以logout函数的每一个分支作为一个等价类，进行测试

d. 测试项标示

|  |  |  |
| --- | --- | --- |
| 测试项标志符 | 测试项描述 | 优先级 |
| UN\_USR\_005\_001 | 传入已登录的用户信息。 | 中 |

e.测试通过/失败标准

所有的用例都必须被执行，且没有发现错误。

f .对应用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | UN\_USR\_006 | | | | | |
| 优先级 | 中 | | | | | |
| 测试项描述 | 调用logout函数，传入用户信息，检查返回值中的error | | | | | |
| 前置条件 | 数据库中存在用户信息,且用户已经登录 | | | | | |
| 用例编号 | 测试方法 | 输入参数 | 预期结果 | 真实结果 | 是否有BUG | 是否解决 |
| 001 | 等价类 | UserName=“zx”，  Password=“321”， | Error=0 | Error=0 | 无 | / |

## 5.2 relation controller 模块用例分析与实现

### follow 用例分析与实现

a.标识符定义

UN\_REL\_001

b.被测特性

（1）传入用户A和B的id，令用户A关注B

（2）传入用户A和B的id，A已经关注B，令用户A再次关注B

c.测试方法

黑盒测试，测试follow函数对重复关注的结果。

d. 测试项标示

|  |  |  |
| --- | --- | --- |
| 测试项标志符 | 测试项描述 | 优先级 |
| UN\_REL\_001\_001 | 传入用户A和B。 | 中 |
| UN\_REL\_001\_002 | 传入用户A和B | 中 |

e.测试通过/失败标准

所有的用例都必须被执行，且没有发现错误。

f .对应用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | UN\_REL\_001 | | | | | |
| 优先级 | 中 | | | | | |
| 测试项描述 | 调用follow函数，传入用户信息，检查返回值中的error | | | | | |
| 前置条件 | 数据库中存在多个用户信息 | | | | | |
| 用例编号 | 测试方法 | 输入参数 | 预期结果 | 真实结果 | 是否有BUG | 是否解决 |
| 001 | 错误猜测 | myUser=“zx”，  OpUser=“abc”， | Error=0 | Error=0 | 无 | / |
| 002 | 错误猜测 | myUser=“zx”，  OpUser=“abc”， | Error=0 | Error=0 | 无 | / |

### unfollow 用例分析与实现

a.标识符定义

UN\_REL\_002

b.被测特性

（1）传入用户A和B的id，A已经关注B, 令用户A取消关注B

c.测试方法

传入用户A和B的id，A已经关注B, 令用户A取消关注B

d. 测试项标示

|  |  |  |
| --- | --- | --- |
| 测试项标志符 | 测试项描述 | 优先级 |
| UN\_REL\_002\_001 | 传入用户A和B | 中 |

e.测试通过/失败标准

所有的用例都必须被执行，且没有发现错误。

f .对应用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | UN\_REL\_002 | | | | | |
| 优先级 | 中 | | | | | |
| 测试项描述 | 调用unfollow函数，传入用户信息，检查返回值中的error | | | | | |
| 前置条件 | 数据库中存在用户A和B的信息，且A已经关注B。 | | | | | |
| 用例编号 | 测试方法 | 输入参数 | 预期结果 | 真实结果 | 是否有BUG | 是否解决 |
| 001 | 等价类 | myUser=“zx”，  OpUser=“abc” | Error=0 | Error=0 | 无 | / |

### getFans 用例分析与实现

a.标识符定义

UN\_REL\_003

b.被测特性

（1）传入用户A的id，获取A的所有fans信息

c.测试方法

传入用户A的id，获取A的所有follower信息

d. 测试项标示

|  |  |  |
| --- | --- | --- |
| 测试项标志符 | 测试项描述 | 优先级 |
| UN\_REL\_003\_001 | 传入用户A | 中 |

e.测试通过/失败标准

所有的用例都必须被执行，且没有发现错误。

f .对应用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | UN\_REL\_003 | | | | | |
| 优先级 | 中 | | | | | |
| 测试项描述 | 调用follow函数，传入用户信息，检查返回值中的error | | | | | |
| 前置条件 | 数据库中存在用户A，且用户A存在粉丝 | | | | | |
| 用例编号 | 测试方法 | 输入参数 | 预期结果 | 真实结果 | 是否有BUG | 是否解决 |
| 001 | 等价类 | myUser=“zx”， | Error=0 | Error=0 | 无 | / |

### 5.2.4 getFollowers 用例分析与实现

a.标识符定义

UN\_REL\_004

b.被测特性

（1）传入用户A的id，获取A的所有followers信息

c.测试方法

传入用户A的id，获取A的所有follower信息

d. 测试项标示

|  |  |  |
| --- | --- | --- |
| 测试项标志符 | 测试项描述 | 优先级 |
| UN\_REL\_004\_001 | 传入用户A | 中 |

e.测试通过/失败标准

所有的用例都必须被执行，且没有发现错误。

f .对应用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | UN\_REL\_004 | | | | | |
| 优先级 | 中 | | | | | |
| 测试项描述 | 调用follow函数，传入用户信息，检查返回值中的error | | | | | |
| 前置条件 | 数据库中存在用户A，且用户A有Follow其他人 | | | | | |
| 用例编号 | 测试方法 | 输入参数 | 预期结果 | 真实结果 | 是否有BUG | 是否解决 |
| 001 | 等价类 | myUser=“abc” | Error=0 | Error=0 | 无 | / |

## 5.3 blog-at controller模块用例分析与实现

### 5.3.1 getAtMeCount用例分析与实现

a.标识符定义

UN\_GMC\_001\_001

UN\_GMC\_001\_002

b.被测特性

（1）当 blog 包含“/”时，会对 blog 做处理，把 blog 里的字符串进行包装，并添加到 blog\_list 里

发送出去；

（2）当 blog 不包含“/”时，不发送信息。

c.测试方法

（1）传入带“/”的 blog，检查是否有信息被发送；

（2）传入不带“/”的 blog，检查是否有信息被发送。

d. 测试项标示

|  |  |  |
| --- | --- | --- |
| 测试项标志符 | 测试项描述 | 优先级 |
| UN\_GMC\_001\_001 | 传入带“/”的blog，检查是否有信息被发送 | 低 |
| UN\_GMC\_001\_002 | 传入不带“/”的blog检查是否有信息被发送 | 高 |

e.测试通过/失败标准

所有的用例都必须被执行，且没有发现错误。

f .对应用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | UN\_GMC\_001\_001 | | | | | |
| 优先级 | 低 | | | | | |
| 测试项描述 | 传入带“/”的blog，检查是否有信息被发送 | | | | | |
| 前置条件 | 无 | | | | | |
| 用例编号 | 测试方法 | 输出参数 | 预期结果 | 真实结果 | 是否有BUG | 是否解决 |
| 001 | 等价类 | 带‘/’的blog参数 | 检测到微博正确发送 | 检测到微博正确发送 | 无 | / |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | UN\_GMC\_001\_002 | | | | | |
| 优先级 | 高 | | | | | |
| 测试项描述 | 传入不带“/”的blog检查是否有信息被发送 | | | | | |
| 前置条件 | 无 | | | | | |
| 用例编号 | 测试方法 | 输出参数 | 预期结果 | 真实结果 | 是否有BUG | 是否解决 |
| 001 | 等价类 | 不带“/”的blog信息 | 检测到微博发送 | 检测到微博发送 | 无 | / |

### 5.3.2 getAtMeBlogList用例分析与实现

a.标识符定义

UN\_GMC\_002\_001

b.被测特性

（1) 参数如果被 / 分割为 2 部分，将 blog 处理后与当前时间 起插入数据库；

（2）blog 参数如果被 / 分割后的数组长度不为 2，不做处理，不插入数据。

c.测试方法

（1）传入带一个 / 字符的 blog 参数，检查数据库是否插入数据；

（2）传入不带或带多个 / 字符的 blog 参数，检查数据库是否插入数据。

d.测试项标示

|  |  |  |
| --- | --- | --- |
| 测试项标志符 | 测试项描述 | 优先级 |
| UN\_GMC\_002\_001 | 传入带一个 / 字符的 blog参数 | 高 |
| UN\_GMC\_002\_002 | 传入不带或者带多个 / 字符的 blog 参数 | 高 |

e.测试通过/失败标准

所有的用例都必须被执行，且没有发现错误。

f .对应用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | UN\_GMC\_002\_001 | | | | | |
| 优先级 | 高 | | | | | |
| 测试项描述 | 传入带一个 / 字符的 blog 参数 | | | | | |
| 前置条件 | 无 | | | | | |
| 用例编号 | 测试方法 | 输出参数 | 预期结果 | 真实结果 | 是否有BUG | 是否解决 |
| 001 | 等价类 | 带 一 个 /  字符的 blog  参数 | 数 据 库 插 入 新 数 据 ， 这 个 新 数 据 由 blog与当前  时间组成 | 数 据 库 插 入 新 数 据 ， 这 个 新 数 据 由 blog与当前 时间组成 | 无 | / |

### 5.3.3 markAsRead用例分析与实现

a.标识符定义

UN\_MAR\_001

b.被测特性

（1）group 中缺失参数会触发插入错误，执行失败；

（2）group中参数完整，插入成功，执行成功。

c.测试方法

（1）group 参数使用数据不完整的 user 对象，预期插入失败；

（2）group 参数使用数据完整的 user 对象，预期插入成功。

d.测试项标示

|  |  |  |
| --- | --- | --- |
| 测试项标志符 | 测试项描述 | 优先级 |
| UN\_MAR\_001\_001 | addNewGroup 插入数据失败 | 高 |
| UN\_MAR\_001\_002 | addNewGroup 插入数据失败 | 高 |

e.测试通过/失败标准

所有的用例都必须被执行，且没有发现错误。

f .对应用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | UN\_MAR\_001\_001 | | | | | |
| 优先级 | 高 | | | | | |
| 测试项描述 | addNewGroup 插入数据成功，返回 List*<*user*>* | | | | | |
| 前置条件 | 无 | | | | | |
| 用例编号 | 测试方法 | 输出参数 | 预期结果 | 真实结果 | 是否有BUG | 是否解决 |
| 001 | 等价类 | 数据完整的 group | 插入成功，  返回所有数  据 | 插入成功，  返回所有数  据 | 无 | / |

## 5.4 blog home/profile/square controller模块用例分析与实现

### 5.4.1 Atorage\_BLOG\_ID用例分析与实现

a.标识符定义

**UN\_ABI\_001**

b.被测特性

（1）blog\_Id 在数据库中不存在时返回空哈希表；

（2）blog\_Id 在数据库中存在返回一个 Map*<*String,List*<*blog*>>*，执行成功。

c.测试方法

采用边界值方法，考虑 blog\_Id 在数据库中是否存在，即是否返回空哈希表，分为返回空哈希表和

返回 Map*<*String,List*<*blogr*>>*。

d.测试项标示

|  |  |  |
| --- | --- | --- |
| 测试项标志符 | 测试项描述 | 优先级 |
| **UN\_ABI\_001**\_001 | getblog 返回空哈希表 | 高 |
| **UN\_ABI\_001**\_002 | getblog 返 回 类 型 为  Map*<*String,List*<*blog\_Id*>>* | 高 |

e.测试通过/失败标准

所有的用例都必须被执行，且没有发现错误。

f .对应用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | **UN\_ABI\_001**\_001 | | | | | |
| 优先级 | 高 | | | | | |
| 测试项描述 | getblog 返回空哈希表 | | | | | |
| 前置条件 | 无 | | | | | |
| 用例编号 | 测试方法 | 输出参数 | 预期结果 | 真实结果 | 是否有BUG | 是否解决 |
| 001 | 边界值 | blog\_Id=100 | NULL | NULL | 无 | / |
| 002 | 边界值 | blog\_Id=50 | NULL | NULL | 无 | / |
| 003 | 边界值 | blog\_Id=0 | NULL | NULL | 无 | / |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | **UN\_ABI\_001**\_002 | | | | | |
| 优先级 | 高 | | | | | |
| 测试项描述 | getblog返 回 类 型 为  Map*<*String,List*<*blog*>>* | | | | | |
| 前置条件 | 无 | | | | | |
| 用例编号 | 测试方法 | 输出参数 | 预期结果 | 真实结果 | 是否有BUG | 是否解决 |
| 001 | 边界值 | blog\_Id=100 |  |  | 无 | / |
| 002 | 边界值 | blog\_Id=50 |  |  | 无 | / |
| 003 | 边界值 | blog\_Id=0 | NULL | NULL | 无 | / |

### 5.4.2 Load\_Blog\_ID用例分析与实现

a.标识符定义

**UN\_LBD\_001**

b.被测特性

（1）数据库中无数据，返回空值；

（2）数据库中有数据，返回 blogGroup 中的所有数据，执行成功。

c.测试方法

（1）数据库无对应数据时查询，返回空值；

（2）数据库中有对应数据时查询，返回 List*<*blogGroup*>*，执行成功。

d.测试项标示

|  |  |  |
| --- | --- | --- |
| 测试项标志符 | 测试项描述 | 优先级 |
| **UN\_LBD\_001**\_001 | 对空数据库查询，返回空值 | 高 |
| **UN\_LBD\_001**\_002 | 数据库中存在数据，返回 List*<*blogGroup*>* | 高 |

e.测试通过/失败标准

所有的用例都必须被执行，且没有发现错误。

f .对应用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | **UN\_LBD\_001**\_001 | | | | | |
| 优先级 | 高 | | | | | |
| 测试项描述 | 对空数据库查询，返回空值 | | | | | |
| 前置条件 | 无 | | | | | |
| 用例编号 | 测试方法 | 输出参数 | 预期结果 | 真实结果 | 是否有BUG | 是否解决 |
| 001 | 等价类 | 0 | NULL | NULL | 无 | / |

### 5.4.3 Load\_homepage用例分析与实现

a.标识符定义

**UN\_LBD\_002**

b.被测特性

传入用户的cookies信息，得到用户发送的bloglist。

c.测试方法

传入用户的cookies信息，得到用户发送的bloglist。

d.测试项标示

|  |  |  |
| --- | --- | --- |
| 测试项标志符 | 测试项描述 | 优先级 |
| **UN\_LBD\_002**\_001 | 传入用户的cookies信息，得到用户发送的bloglist。 | 高 |

e.测试通过/失败标准

所有的用例都必须被执行，且没有发现错误。

f .对应用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | **UN\_LBD\_002**\_001 | | | | | |
| 优先级 | 高 | | | | | |
| 测试项描述 | 传入用户的cookies信息，得到用户发送的bloglist。 | | | | | |
| 前置条件 | 无 | | | | | |
| 用例编号 | 测试方法 | 输出参数 | 预期结果 | 真实结果 | 是否有BUG | 是否解决 |
| 001 | 等价类 | bloglist | datacount不等于0 | datacount等于0 | 有 | 回归测试 |

# 回归测试

在前面的测试环节中，我们发现了有 3 个测试的函数存在着 bug，因此我们针对这 3 个函数进行了修改，在完成修改之后，我们针对这三个函数进行了回归测试，测试结果如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | UN\_USR\_005 | | | | | |
| 优先级 | 低 | | | | | |
| 测试项描述 | 调用deleteUser函数，传入用户信息，检查返回值中的error | | | | | |
| 前置条件 | 数据库中存在用户信息,且用户已经登录 | | | | | |
| 用例编号 | 测试方法 | 输入参数 | 预期结果 | 真实结果 | 是否有BUG | 是否解决 |
| 001 | 等价类 | UserName=“abc”，  Password=“123”， | Error=0 | Error=0 | 无 | 是 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | UN\_USR\_003 | | | | | |
| 优先级 | 高 | | | | | |
| 测试项描述 | 调用changeinfo函数，传入信息，检查返回值中的error | | | | | |
| 前置条件 | 数据库中存在用户信息 | | | | | |
| 用例编号 | 测试方法 | 输入参数 | 预期结果 | 真实结果 | 是否有BUG | 是否解决 |
| 002 | 等价类 | Img信息大于1M | Error=10007 | Error=10007 | 无 | 是 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试项编号 | **UN\_LBD\_002\_001** | | | | | |
| 优先级 | 高 | | | | | |
| 测试项描述 | 传入用户的cookies信息，得到用户发送的bloglist。 | | | | | |
| 前置条件 | 无 | | | | | |
| 用例编号 | 测试方法 | 输出参数 | 预期结果 | 真实结果 | 是否有BUG | 是否解决 |
| 001 | 等价类 | bloglist | datacount不等于0 | datacount不等于0 | 无 | / |