<Memory>

测试用例

server

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2016/9/8 | 1.0 | 对API的测试 | 许泽资 |
| 2016/9/10 | 1.1 | 简单修改 | 许泽资 |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. 简介 5

1.1 目的 5

1.2 范围 5

1.3 定义、首字母缩写词和缩略语 5

1.4 参考资料 5

1.5 概述 5

2. 工具类单元测试 5

2.1 statistics 5

2.2 Spring IoC 6

2.3 Restful Cookie Template 6

2.4 Log Service 7

2.5 MongoDB Utility 7

2.6 Edison Connection Utility 8

2.7 Get Age From Birthday 9

3. URI-API集成测试 UserBasic 10

3.1 /login POST 10

3.2 /logout POST 10

3.3 /identity/reg POST 10

3.4 /identity/detail/{username} GET 10

3.5 /identity/profile/{username} GET 10

3.6 /identity/profile POST 10

4. URI-API集成测试 Resource 10

4.1 /resources GET 10

4.2 /resources/latest GET 10

4.3 /resource/friends GET 10

4.4 /resources/latest/{timestamp} GET 10

4.5 /resources/friends/{timestamp} GET 10

4.6 /resources/self/{timestamp} GET 10

4.7 /resources POST 10

4.8 others 10

4.9 /resources/{resource\_id}/words GET/POST 10

4.10 /resources/{resource\_id}/image GET/POST 10

4.11 others 10

5. URI-API 集成测试 Friend 11

5.1 /friends GET 11

5.2 /friends/{username} DELETE 11

5.3 /friends/apply GET 11

5.4 /friends/apply/{username} POST 11

5.5 /friends/make GET 11

5.6 /friends/make/{username} POST/DELETE 11

6. URI-API 集成测试 Mark Music And Book 11

6.1 /resources/{resource\_id}/marks GET 11

6.2 /resources/{resource\_id}/marks/{timestamp} GET 11

6.3 /resources/resource\_id}/marks POST/GET/DELETE 11

6.4 /resources/{resource\_id}/music/{hashcode} POST 11

6.5 /resources/{resource\_id}/book/{ISBN} POST 11

7. URI-API 集成测试 Others 11

7.1 /statistics/visitors, ./resources/new, ./friends/new, ./resources GET 11

7.2 /recommend/user GET 11

8. 11

测试用例

# 简介

本测试用例文档，主要囊括了服务器端(back end), 的测试用例代码及其描述。通过对每一条URI-API的测试，来确定其可用性。内容包括但不限于：服务单元测试，集成测试，工具类单元测试。

## 目的

本文档旨在保证服务器端的服务个体、服务之间以及工具服务的可用性，并在测试结果的基础上给出建议。

## 范围

本文档主要包括后端的代码测试，包括单元测试以及集成测试。

## 定义、首字母缩写词和缩略语

|  |  |
| --- | --- |
| **文档用词** | **含义** |
| uri | 指代某一个服务API所对应的uri |
| 工具类 | 指代后端用于辅助完成系统而抽象的公共类型 |

## 参考资料

1. API文档： PART 1-UserBasic.json

PART 2-Resource.json

PART 3-Friend.json

PART 4-Mark.json

PART 5-MusicAndBook.json

PART 6-Tool.json

## 概述

本文档通过对每一个测试用例的描述或者输入代码，配合期待的输出值或者相应表现，来形成一个测试用例。测试人员通过对比来生成测试报告。测试用例的部分以产生所有输出信息为原则，完成边界值以及等价类覆盖。

# 工具类单元测试

给出测试的相关代码，相关的代码也可以在打包的UtilTest.java中直接执行测试。此部分的相关测试几乎没有前置条件与后置条件。

## statistics

独立功能测试

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 测试输入 | 1. 获取常数 2. 依次调用visit(), resource(), friend() 3. 获取常数 |
| 观察点 | 1. check()函数调用 |
| 预期结果 | 获取的常数比调用函数之前加一 |

交叉功能测试

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 测试输入 | 1. 获取常数 2. 交叉调用visit(), resource(), friend() 3. 获取常数 |
| 观察点 | 1. check()函数调用 |
| 预期结果 | 获取的常数比调用函数之前加一 |

时间跨度清零测试

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 测试输入 | 1. 在整点之前获取常数 2. 在整点之后调用visit(), resource(), friend() 3. 获取常数 |
| 观察点 | 1. check()函数调用 |
| 预期结果 | 获取的常数比调用函数之前不变 |

## Spring IoC

IoC.xml包含的目标，并且类符合

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 测试输入 | 1. 利用idGetter获取对应的服务实例 2. 利用classGetter获取对应的服务实例 |
| 预期结果 | 获取实例成功，对应类正确，无报错，方法可以正常调用 |

IoC.xml包含的目标，但类型不符合

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 测试输入 | 1. 利用idGetter获取对应的服务实例，但给一个错误的类 |
| 预期结果 | 获取实例失败，报错原因实例不符合。服务器不会宕机。 |

IoC.xml不包含的目标

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 测试输入 | 1. 利用idGetter获取随机名字的服务实例 2. 利用classGetter获取随机名字的服务实例 |
| 预期结果 | 获取实例失败，报错原因不存在对应bean。 |

## Restful Cookie Template

未登录状态下获取相应服务

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 未登录 |
| 测试输入 | {  “emotion”:1,  “type”:0  }  POST /resources |
| 观察点 | 1.setHeaders地点 |
| 预期结果 | 没有set headers，同时返回报错，未登录 |

未登录状态下登录

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 未登录 |
| 测试输入 | {  “name”:”Sturmfy”,  “password”:”nihao”  }  POST /identity |
| 观察点 | 1.setHeaders地点 |
| 预期结果 | 完成set headers，同时返回正确的实体信息，查看cookie有相应内容id |

登录状态下登录

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 未登录 |
| 测试输入 | {  “name”:”Sturmfy”,  “password”:”nihao”  }  POST /identity |
| 观察点 | 1. setHeaders地点 2. 返回信息 |
| 预期结果 | 完成set headers，同时返回正确的实体信息，查看cookie有相应内容id。返回信息中返回了上一个登录用户的用户名。 |

## Log Service

直接执行输出

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 测试输入 | 调用log()方法（无参数） |
| 预期结果 | 只打印时间，并占一行 |

执行加载内容，不执行输出

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 测试输入 | 不断调用log(String mes)方法（有参数） |
| 预期结果 | 无输出 |

加载内容，并在之后执行输出

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 测试输入 | 不断调用log(String mes)方法（有参数），最后调用log()方法（无参数） |
| 预期结果 | 输出日期 + 加载的内容，并占一行 |

## MongoDB Utility

存储图片，给出正确的信息

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 完成对mongoDB的连接 |
| 测试输入 | database = ‘BWPhoto’  bucket = ‘Profile’  filename = ‘a’  stream = …// 某个文件 |
| 观察点 | 无 |
| 预期结果 | 返回值为True，无报错信息；数据库检查有相关文件 |

存储图片，但输入流为空；或者文件名为 “”

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 完成对mongoDB的连接 |
| 测试输入 | filename = “”  or  stream = null |
| 观察点 | gfs.createFile() |
| 预期结果 | 返回值为False，服务器打印报错信息；数据库检查没有改变。 |

读取图片，图片存在

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 完成对mongoDB的连接 |
| 测试输入 | database = ‘BWPhoto’  bucket = ‘Profile’  filename = ‘a’ |
| 观察点 | 无 |
| 预期结果 | 返回值不为null，无报错信息；数据库检查有相关文件。文件读出展示后没有损坏。 |

读取图片，输入错误的文件信息

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 完成对mongoDB的连接 |
| 测试输入 | database = ‘wrong’  or  bucket = ‘wrong’  or  filename = ‘wrong’ |
| 观察点 | 实例is |
| 预期结果 | 返回值为null，服务器打印报错信息；数据库检查没有相关文件。 |

## Edison Connection Utility

输入正确的端口号，连接成功

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 未连接 |
| 测试输入 | port = ‘105’ |
| 观察点 | 构造函数 |
| 预期结果 | 打印连接成功相应信息，没有异常抛出。 |

连接失败

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 未连接 |
| 测试输入 | port = ‘0’ |
| 观察点 | 构造函数 |
| 预期结果 | 异常抛出，被接收后打印相关信息。 |

传输对象，并接受

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 连接成功 |
| 测试输入 | [{  “name”:”a”  },  {  “name”:”b”  }] |
| 观察点 | 实例json |
| 预期结果 | 发送成功，接收成功；屏幕打印相关信息；接受对象与发送的对象无差别。 |

传输对象，中间人为断开连接

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 连接成功 |
| 测试输入 | [{  “name”:”a”  },  {  “name”:”b”  }] |
| 观察点 | 实例json |
| 预期结果 | 发送失败，屏幕打印connection closed等相关信息。 |

## Get Age From Birthday

给定生日，目前日期与生日相同，年份相同或更大（正常边界值）

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | Calendar, util.Date类 |
| 测试输入 | 生日1996.6.21  日期2006.6.21 |
| 观察点 | 日期值，年龄差 |
| 预期结果 | 打印10岁 |

给定生日，年份更小（出生之前）

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | Calendar, util.Date类 |
| 测试输入 | 生日1996.6.21  日期1990.6.21 |
| 观察点 | 日期值，年龄差 |
| 预期结果 | 打印0岁 |

给定生日，月份相同，日期更晚（比生日晚，不到生日）

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | Calendar, util.Date类 |
| 测试输入 | 生日1996.6.21  日期2006.6.22 |
| 观察点 | 日期值，年龄差 |
| 预期结果 | 打印9岁 |

# URI-API集成测试 UserBasic

本部分的内容，以及之后的URI-API集成测试部分，均以一个URI+http方法为一个模块进行测试用例的书写

## /login POST

登录信息正确或者错误

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 未登录或者已经登录 |
| 测试输入 | {  **"name"**:**"Sturmfy"**,  **"password"**:**"1234"** } |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"success"**:**"Login Success with old user name1"** }  or  {  **"fail"**:**"User Not Exist Or Password Not Match"** } |
| 后置条件 | 成功 |

## /logout POST

已经登录情况下，登出

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"success"**:**"Log out success"** } |

未登录情况下，登出

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 未登录 |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"fail"**:**"No User Logged"** } |

## /identity/reg POST

注册（无论登录与否）

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 未登录或者已经登录 |
| 测试输入 | {  **"name"**:**"Sturmfy"**,  **"password"**:**"1234"**,  **"role"**: 1,  **"email"**:**"123@email.com"**,  **"age"**:**"20"**,  **"content"**:**"A cute boy ready for love"** } |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"fail"**:**"user exists"** }  or {  **"success"**:**"Register Success!"** }  如果登录的情况下注册成功，自动切换为该用户 |

## /identity/detail/{username} GET

登录情况下查询自己的信息

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | 自己的username |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"name"**:**"Sturmfy"**,  **"role"**: 1,  **"email"**:**"123@email.com"**,  **"age"**:**"20"**,  **"content"**:**"A cute boy ready for love"** }  不包括密码的所有详细信息 |

登录情况查询他人信息，或者未登录查询他人信息

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录或没有登录 |
| 测试输入 | 他人的username |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"name"**:**"Sturmfy"**,  **"age"**:**"20"**,  **"content"**:**"A cute boy ready for love"** }  简单信息 |

## /identity/profile/{username} GET

获取用户的头像

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录或没有登录 |
| 测试输入 | 某个人的username |
| 观察点 | 服务器输出 |
| 预期结果 | 图片文件流，可以直接用html代码或者其他文件流管道对接。显示的图片没有损坏错误 |

用户不存在

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录或没有登录 |
| 测试输入 | 某个不存在的人的username |
| 观察点 | 服务器输出 |
| 预期结果 | 默认的图片文件流，可以直接用html代码或者其他文件流管道对接。显示的图片没有损坏错误 |

## /identity/profile POST

上传

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | 图片文件流 |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"success"**:**"image upload succeeded"** } |

上传，未登录

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 未登录 |
| 测试输入 | 图片文件流 |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"fail"**:**"Something's wrong"** } |

上传，文件大小超过限制

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | 图片文件流 |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"fail"**:**"Something's wrong"** } |

# URI-API集成测试 Resource

## /resources GET

登录，获取

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | 无 |
| 观察点 | 服务器输出 |
| 预期结果 | [  {  **"id"**:1,  **"name"**:**"Sturmfy"**,  **"status"**:**"1"** }  {  **"id"**:72,  **"name"**:**"Sturmfy"**,  **"memory\_id"**:1,  **"image\_id"**:2,  **"status"**:**'1'** } ] |

未登录，获取

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 未登录 |
| 测试输入 | 无 |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"fail"**:**"User Not Logged"** } |

## /resources/latest GET

获取最新可以可浏览的资源

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 无 |
| 测试输入 | 无 |
| 观察点 | 服务器输出 |
| 预期结果 | [  {  **"id"**:1,  **"name"**:**"name1"**,  **"timestamp"**:1468909652402  } ] |

## /resource/friends GET

获取到朋友的最新的可以浏览的信息

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | 无 |
| 观察点 | 服务器输出 |
| 预期结果 | [  {  **"id"**:1,  **"name"**:**"name1"**,  **"timestamp"**:1468909652402  } ] |

未登录

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 未登录 |
| 测试输入 | 无 |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"fail"**:**"User Not Logged"** } |

## /resources/latest/{timestamp} GET

获取时间点之前的resource（之前没有）

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | 久远的时间 |
| 观察点 | 服务器输出 |
| 预期结果 | [] |

正常的时间

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | 正常的时间 |
| 观察点 | 服务器输出 |
| 预期结果 | [  {  **"id"**:1,  **"name"**:**"name1"**,  **"timestamp"**:1468909652402  } ] |

未登录：见前面 /resources/latest GET

## /resources/friends/{timestamp} GET

获取时间点之前的resource（之前没有）

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | 久远的时间 |
| 观察点 | 服务器输出 |
| 预期结果 | [] |

正常的时间

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | 正常的时间 |
| 观察点 | 服务器输出 |
| 预期结果 | [  {  **"id"**:1,  **"name"**:**"name1"**,  **"timestamp"**:1468909652402  } ] |

未登录：见前面 /resources/friends/latest GET

## /resources/self/{timestamp} GET

获取时间点之前的resource（之前没有）

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | 久远的时间 |
| 观察点 | 服务器输出 |
| 预期结果 | [] |

正常的时间

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | 正常的时间 |
| 观察点 | 服务器输出 |
| 预期结果 | [  {  **"id"**:1,  **"name"**:**"name1"**,  **"timestamp"**:1468909652402  } ] |

未登录：见前面 /resources/self/latest GET

## /resources POST

创建一个resource（正常）

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | {  “emotion”:1,  “type”:2  } |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"id"**:1,  **"name"**:**"name1"**,  **"timestamp"**:1468909652402 } |

创建一个resource 未登录

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 未登录 |
| 测试输入 | 无 |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"fail"**:**"User Not Logged"** } |

创建，内容类型错误

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | {  “emotion”:1,  “type”:2  } |
| 观察点 | 服务器输出 |
| 预期结果 | 服务器报错 |

## others

改变resource的公开属性

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | /resource/{resource\_id}/public  /resource/{resource\_id}/private |
| 观察点 | 服务器输出 |
| 预期结果 | 返回成功信息  检查数据库，内容更改成功 |

改变resource公开属性，但是resource不存在或者未登录

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录或者未登录 |
| 测试输入 | /resource/{resource\_id}/public  /resource/{resource\_id}/private  resource\_id不存在 |
| 观察点 | 服务器输出 |
| 预期结果 | 返回失败信息  检查数据库，内容没有变化 |

## /resources/{resource\_id}/words GET/POST

获取resource的word

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | resource\_id |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"id"**:2,  **"content"**:**"life is so different here. when I feel the life before** , I indeed get my life.**",  "timestamp"**:1248794565 } |

创建或者更改resource的word

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | resource\_id  {  **"content"**:**"I wanna change my life"** } |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"fail"**:**"failed for some reason"** }  or {  **"success"**:**"create succeeded"** } |

创建或获取resource的word，但没有登录或者resource不存在

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | resource\_id  以及对应的输入数据（见上） |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"fail"**:**"failed for some reason"** } |

## /resources/{resource\_id}/image GET/POST

获取resource的image

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | resource\_id |
| 观察点 | 服务器输出 |
| 预期结果 | 图片文件流 |

创建或者更改resource的image

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | resource\_id  图片文件流 |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"fail"**:**"Something's wrong"** }  or {  **"success"**:**"image upload succeeded"** } |

创建resource的image，但没有登录或者resource不存在

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录或未登录 |
| 测试输入 | resource\_id  以及对应的输入数据（见上） |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"fail"**:**"failed for some reason"** } |

获取resource的image，但是没有登录或resource不存在

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录或未登录 |
| 测试输入 | resource\_id  以及对应的输入数据（见上） |
| 观察点 | 服务器输出 |
| 预期结果 | 获得默认的图片的图片流 |

## others

创建tag

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | resource\_id  tag\_name  /resources/resource\_id/tags/{tag\_name} |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"success"**:**"succeeded"** } |

获取tags

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | resource\_id  tag\_name  /resources/resource\_id/tags GET |
| 观察点 | 服务器输出 |
| 预期结果 | [  {  **"name"**:**"heart"** }  {  **"name"**:**"love"** }  {  **"name"**:**"gay"** } ] |

未登录或者发生错误

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录或未登录 |
| 测试输入 | resource\_id  tag\_name  /resources/resource\_id/tags GET  or  /resources/resource\_id/tags/{tag\_name} POST |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"fail"**:**"failed for some reason"** } |

# URI-API 集成测试 Friend

## /friends GET

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | 无 |
| 观察点 | 服务器输出 |
| 预期结果 | [  {  **"myName"**:**"Sturmfy"**,  **"frName"**:**"name1"** } ] |

## /friends/{username} DELETE

存在用户

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | username |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"success"**:**"success"** } |

用户不存在

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | username |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"fail"**:**"fail"** } |

## /friends/apply GET

获取所有的申请

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | 无 |
| 观察点 | 服务器输出 |
| 预期结果 | [  {  **"applyer"**:**"Sturmfy"**,  **"applyee"**:**"name1"**,  **"status"**:**"-"** }  {  **"applyer"**:**"Sturmfy"**,  **"applyee"**:**"name2"**,  **"status"**:**"y"** }  {  **"applyer"**:**"Sturmfy"**,  **"applyee"**:**"name3"**,  **"status"**:**"n"** } ] |

## /friends/apply/{username} POST

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | username |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"success"**:**"success"** } |

## /friends/make GET

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | 无 |
| 观察点 | 服务器输出 |
| 预期结果 | [  {  **"applyer"**:**"Sturmfy"**,  **"applyee"**:**"name1"**,  **"status"**:**"-"** } ] |

## /friends/make/{username} POST/DELETE

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | username |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"success"**:**"success"** } |

申请人不在申请列表

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | username |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"fail"**:**"failed"** } |

## 未登录错误

当未登录的时候一律返回

{

“fail”:”User Not Logged”

}

# URI-API 集成测试 Mark Music And Book

## /resources/{resource\_id}/marks GET

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | resource\_id |
| 观察点 | 服务器输出 |
| 预期结果 | [  {  **"id"**:12,  **"item\_id"**:235,  **"this\_name"**:**"name1"**,  **"content"**:**"how do you feel about this?"**,  **"time"**:1228242156224,  **"mark\_on\_id"**:**null** }  {  ...  } ] |

resource\_id 不存在

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | 错误的resource\_id |
| 观察点 | 服务器输出 |
| 预期结果 | []为空 |

## /resources/{resource\_id}/marks/{timestamp} GET

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | resource\_id， timestamp |
| 观察点 | 服务器输出 |
| 预期结果 | [  {  **"id"**:12,  **"item\_id"**:235,  **"this\_name"**:**"name1"**,  **"content"**:**"how do you feel about this?"**,  **"time"**:1228242156224,  **"mark\_on\_id"**:**null** }  {  ...  } ] |

久远的timestamp

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | resource\_id, timestamp |
| 观察点 | 服务器输出 |
| 预期结果 | []为空 |

## /resources/{resource\_id}/music/{hashcode} POST

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | resource\_id, hashcode |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"success"**:**"7aed5f321596a3d"** } |

## /resources/{resource\_id}/book/{ISBN} POST

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | resource\_id, ISBN |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"success"**:**"978-123-7-1234-1234"** } {  **"fail"**:**"Invalid ISBN Format"** } |

# URI-API 集成测试 Others

## /statistics/visitors, ./resources/new, ./friends/new, ./resources GET

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 测试输入 | 直接获取 |
| 观察点 | 服务器输出 |
| 预期结果 | {  **"amount"**:**"1068"** } |

## /recommend/user GET

要求已经登录

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 测试输入 | 直接获取 |
| 观察点 | 服务器输出 |
| 预期结果 | [  “a”,  “b”  ] |

# 权限控制测试

## 资源获取

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录，并且用不同身份登录 |
| 测试输入 | 1. 分别创建各自的private资源 2. 分别获取 |
| 观察点 | 返回输出 |
| 预期结果 | 不会有对方的private资源 |

## 推荐获取

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录，并且用不同身份登录 |
| 测试输入 | 1. 分别创建各自的public资源 2. 分别获取推荐 |
| 观察点 | 返回输出 |
| 预期结果 | 获取的推荐用户不完全一致 |

## 跨域修改

|  |  |
| --- | --- |
| 项目 | 描述或伪代码 |
| 前置条件 | 已经登录 |
| 测试输入 | 1. 通过API对不属于自己的resource进行修改 2. 对不属于自己的private资源进行查看 |
| 观察点 | 服务器输出 |
| 预期结果 | 无修改权限，被拒绝，给出拒绝信息 |

# 说明

文档当中把应该测试的部分悉数加入。但依旧有一些需要注意的点。

1. 有些API极为简单，已经略过。
2. 实现方法类似的API酌情可以在实际测试的时候略过。但注意服务背后其他细节导致的不同。
3. 与安卓端的集成测试以及系统测试将在安卓端文档中描述。
4. 性能测试以及压力测试直接在测试报告中给出。
5. 数据库完整性测试直接在测试报告中给出。