# 虚拟账户系统-用户使用说明书

目录

[一、引言 3](#_Toc516239872)

[1.1 编写目的 3](#_Toc516239873)

[1.2 项目背景 3](#_Toc516239874)

[1.3 定义 3](#_Toc516239875)

[1.4 参考资料 3](#_Toc516239876)

[二、项目概述 4](#_Toc516239877)

[2.1 目标 4](#_Toc516239878)

[2.2 功能 4](#_Toc516239879)

[2.3 性能 4](#_Toc516239880)

[2.3.1 精度 4](#_Toc516239881)

[2.3.2 时间特性 4](#_Toc516239882)

[三、运行环境 5](#_Toc516239883)

[3.1 硬件运行环境 5](#_Toc516239884)

[3.2 软件运行环境 5](#_Toc516239885)

[四、使用说明 5](#_Toc516239886)

[4.1 接口说明 5](#_Toc516239887)

[4.2 方法说明 5](#_Toc516239888)

[4.2.1 agencyLogin方法 5](#_Toc516239889)

[4.2.2 userLogin方法 6](#_Toc516239890)

[4.2.3 userRegister方法 7](#_Toc516239891)

[4.2.4 userPasswdChanging方法 7](#_Toc516239892)

[4.2.5 agencyInformation方法 8](#_Toc516239893)

[4.2.6 agencyAllUser方法 9](#_Toc516239894)

[4.2.7 userInformation方法 9](#_Toc516239895)

[4.2.8 freezeUnfreeze方法 10](#_Toc516239896)

[4.2.9 foundPasswd方法 11](#_Toc516239897)

[4.2.10 agencyTradeInformation方法 11](#_Toc516239898)

[4.2.11 userTradeInformation方法 12](#_Toc516239899)

[4.2.12 transferConsumer方法 13](#_Toc516239900)

[4.2.13 reCharge方法 13](#_Toc516239901)

[4.2.14 drawMoney方法 14](#_Toc516239902)

[4.3 求助查询 15](#_Toc516239903)

[五、附录 15](#_Toc516239904)

## 一、引言

## 1.1 编写目的

本说明书的编写目的是帮助前端开发团队成员了解“虚拟账户系统”的接口设计，便于对各种接口进行调用。

本说明书的读者对象为：“虚拟账户系统”对应前端开发团队成员。

## 1.2 项目背景

开发项目名称：虚拟账户系统

项目任务提出者：总项目负责人

项目开发者：林未团队

用户：“虚拟账户系统”对应前端开发团队成员

## 1.3 定义

MySQL：数据库

Tomcat：WEB服务器

Ubuntu 16.04 TLS 64位：运行环境

Mybatis：持久层框架

Druid：数据库连接池

Redis：缓存框架

Springframework：分层应用程序框架

Log4j：日志管理框架

Dubbo：分布式服务框架

## 1.4 参考资料

[1]“虚拟账户系统”的《软件开发计划书》

[2]“虚拟账户系统”的《需求规格说明书》

[3]“虚拟账户系统”的《软件设计说明书》

[4]“虚拟账户系统”的《测试分析报告》

## 二、项目概述

## 2.1 目标

近年来，互联网金融在我国迅速发展，金融服务平台炙手可热。开发此软件目的在于为金融服务系统提供商家运营平台，实现虚拟账户系统和清结算系统等的可视化，为商家提供便捷的账务管理和规范的信息管理。通过借助虚拟账户系统完成商家个人账户信息的管理。

## 2.2 功能

“虚拟账户系统”是整个金融平台的账户模块，主要负责管理系统账户的，相应的内容有：

①适用于用户的功能：

 ◎注册、查询个人信息、修改个人信息及密码；

 ◎查询账户余额、以及账户的冻结和解冻；

 ◎账户间的转账、与其他平台的充值提现。

②系统的主要功能：

 ◎对用户信息的管理；

 ◎对余额信息的管理；

 ◎账户的交易功能。

③能够提供一定的安全机制，提供数据信息授权访问，防止随意删改，同时提供信息备份的服务。

## 2.3 性能

## 2.3.1 精度

交易过程中的各类金额数据，使用Decimal数据类型；交易过程中的各类日期时间数据，均精确至秒级别。

## 2.3.2 时间特性

①用户操作响应时间

注册、登录、账户结算的操作响应时间不超过5秒。

②后台管理者更新处理时间

所有数据的更新一般在5秒内响应。

## 三、运行环境

## 3.1 硬件运行环境

2.4 GHz Intel Xeon E5-2680 Broadwell（v4）双核

DDR4 2G

50G 机械硬盘

## 3.2 软件运行环境

Ubuntu 16.04 TLS 64位

MySQL 5.7

apache-tomcat 9.0.7

dubbo-monitor-simple 2.0.0

redis 4.0.9

zookeeper-3.4.10

jre 1.8.0\_162

## 四、使用说明

## 4.1 接口说明

虚拟账户系统服务接口信息如表4.1所示：

**表4.1 虚拟账户系统服务接口**

|  |  |
| --- | --- |
| **接口名称** | buaa.jj.accountservice.api.AccountService |
| **接口地址** | dubbo://140.143.4.171:20880/buaa.jj.accountservice.api.AccountService |
| **接口协议** | dubbo |
| **服务注册中心** | <dubbo:registry address="zookeeper://140.143.4.171:2181"/> |

## 4.2 方法说明

## 4.2.1 agencyLogin方法

①方法描述：机构登录方法。对登录机构名密码进行验证，成功登录返回机构ID，失败返回-1。

②请求参数：如表4.2所示。

**表4.2 请求参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数类型** | **参数名** | **参数描述** | **备注** |
| String | agency\_name | 机构名 | 无 |
| String | agency\_passwd | 机构密码 | SHA256后传入 |

③响应参数：如表4.3所示。

**表4.3 响应参数**

|  |  |
| --- | --- |
| **响应类型** | **响应描述** |
| int | 成功登录返回对应机构ID>=0，否则返回-1 |

④参考示例：假设数据库中已有机构abc。

|  |
| --- |
| accountService.agencyLogin("abc","0bfe935e70c321c7ca3afc75ce0d0ca2f98b5422e008bb31c00c6d7f1f1c0ad8"); |

⑤错误代码参考：

|  |
| --- |
| accountService.agencyLogin("saf","0bfe93ad6",12); //参数个数不一致 |

## 4.2.2 userLogin方法

①方法描述：用户登录方法。对登录用户名密码进行验证，成功登录返回用户ID，失败返回-1。（注：这里用户泛指用（商）户，下同。）

②请求参数：如表4.4所示。

**表4.4 请求参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数类型** | **参数名** | **参数描述** | **备注** |
| String | user\_name | 用户名 | 无 |
| String | user\_passwd | 用户密码 | SHA256后传入 |

③响应参数：如表4.5所示。

**表4.5 响应参数**

|  |  |
| --- | --- |
| **响应类型** | **响应描述** |
| int | 成功登录返回对应用户ID>=0，否则返回-1 |

④异常描述：如表4.6所示。

**表4.6 异常描述**

|  |  |
| --- | --- |
| **异常类型** | **异常描述** |
| UserNotExistException | 用户不存在，无法登录 |
| UserFrozenException | 用户被冻结，无法登录 |

⑤参考示例：假设数据库中已有用（商）户xyz。

|  |
| --- |
| accountService.userLogin("xyz","0bfe935e70c321c7ca3afc75ce0d0ca2f98b5422e008bb31c00c6d7f1f1c0ad6"); |

⑥错误代码参考：

|  |
| --- |
| accountService.userLogin("xyz","0bfe9c0d8",12); //参数个数不一致 |

## 4.2.3 userRegister方法

①方法描述：用户注册方法，成功注册返回对应用户ID。

②请求参数：如表4.7所示。

**表4.7 请求参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数类型** | **参数名** | **参数描述** | **备注** |
| String | user\_name | 用户（商户）名 | 无 |
| String | user\_passwd | 用户（商户）密码 | SHA256后传入 |
| String | user\_realname | 用户（商户）真实姓名（商户名称） | 无 |
| String | user\_tel | 用户（法人代表）电话 | 无 |
| String | user\_email | 用户（商户）Email | 无 |
| String | user\_identity | 用户身份认证号（商户许可证号） | 无 |
| int | under\_agency\_id | 用户（商户）所属机构 | 存在String类型的重载方法，under\_agency\_name，即用户（商户）所属机构名 |

③响应参数：如表4.8所示。

**表4.8 响应参数**

|  |  |
| --- | --- |
| **响应类型** | **响应描述** |
| int | 成功返回用户ID，失败抛出异常 |

④异常描述：如表4.9所示。

**表4.9 异常描述**

|  |  |
| --- | --- |
| **异常类型** | **异常描述** |
| NameDuplicateException | 注册用户的用户名重复，无法注册 |
| AgencyNotExistException | 注册用户的所属机构不存在，无法注册 |
| UserAgencyDuplicateException | 注册用户的身份证号在该机构下已注册过帐号，无法注册 |

⑤参考示例：假设数据库中无用（商）户af。

|  |
| --- |
| accountService.userRegister("af","0bfe935e70c321c7ca3afc75ce0d0ca2f98b5422e008bb31c00c6d7f1f1c0ad8","lfy","1381462","f6s@126.com","3422347",1); |

⑥错误代码参考：

|  |
| --- |
| accountService.userRegister("af","lfy","1381462","f6s@126.com","1234567890",1); //缺少必填参数 |

## 4.2.4 userPasswdChanging方法

①方法描述：用户（商户）修改密码，成功返回true。

②请求参数：如表4.10所示。

**表4.10 请求参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数类型** | **参数名** | **参数描述** | **备注** |
| int | user\_id | 用户（商户）ID | 无 |
| String | old\_passwd | 用户（商户）原密码 | SHA256后传入 |
| String | new\_passwd | 用户（商户）新密码 | SHA256后传入 |

③响应参数：如表4.11所示。

**表4.11 响应参数**

|  |  |
| --- | --- |
| **响应类型** | **响应描述** |
| Boolean | 成功返回true，失败抛出异常 |

④异常描述：如表4.12所示。

**表4.12 异常描述**

|  |  |
| --- | --- |
| **异常类型** | **异常描述** |
| UserNotExistException | 用户不存在，无法更改密码 |
| UserFrozenException | 用户被冻结，无法更改密码 |

⑤参考示例：假设数据库中存在ID为3的用（商）户。

|  |
| --- |
| accountService.userPasswdChanging(3,"0bfe935e70c321c7ca3afc75ce0d0ca2f98b5422e008bb31c00c6d7f1f1c0ad6","0bfe935e70c321c7ca3afc75ce0d0ca2f98b5422e008bb31c00c6d7f1f1c0ad8"); |

⑥错误代码参考：

|  |
| --- |
| accountService.userPasswdChanging(3,"sdfg"); //缺少必填参数 |

## 4.2.5 agencyInformation方法

①方法描述：查询机构信息。

②请求参数：如表4.13所示。

**表4.13 请求参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数类型** | **参数名** | **参数描述** | **备注** |
| int | agency\_id | 机构ID | 无 |

③响应参数：如表4.14所示。

**表4.14 响应参数**

|  |  |
| --- | --- |
| **响应类型** | **响应描述** |
| Map<String,Object> | 成功返回机构信息，失败返回空Map |

④Map字段描述：如表4.15所示，注意键大小写敏感。

**表4.15 Map字段描述**

|  |  |
| --- | --- |
| **键** | **值描述** |
| agencyID | 机构ID |
| agencyName | 机构名 |
| agentName | 机构代理姓名 |
| agentTel | 机构代理电话 |
| agentEmail | 机构代理邮箱 |

⑤参考示例：假设数据库中存在ID为1的机构。

|  |
| --- |
| accountService.agencyInformation(1); |

⑥错误代码参考：

|  |
| --- |
| accountService.agencyInformation(); //缺少必填参数 |

## 4.2.6 agencyAllUser方法

①方法描述：获取机构下所有用户ID。

②请求参数：如表4.16所示。

**表4.16 请求参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数类型** | **参数名** | **参数描述** | **备注** |
| int | agency\_id | 机构ID | 无 |

③响应参数：如表4.17所示。

**表4.17 响应参数**

|  |  |
| --- | --- |
| **响应类型** | **响应描述** |
| List<Integer> | 成功返回机构用户ID列表，失败返回空表 |

④参考示例：假设数据库中存在ID为1的机构。

|  |
| --- |
| accountService.agencyAllUser(1); |

⑤错误代码参考：

|  |
| --- |
| accountService.agencyAllUser(); //缺少必填参数 |

## 4.2.7 userInformation方法

①方法描述：查询用户信息。

②请求参数：如表4.18所示。

**表4.18 请求参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数类型** | **参数名** | **参数描述** | **备注** |
| int | user\_id | 用户ID | 无 |

③响应参数：如表4.19所示。

**表4.19 响应参数**

|  |  |
| --- | --- |
| **响应类型** | **响应描述** |
| Map<String,Object> | 成功返回用户信息，失败返回空Map |

④Map字段描述：如表4.20所示，注意键大小写敏感。

**表4.20 Map字段描述**

|  |  |
| --- | --- |
| **键** | **值描述** |
| userID | 用户ID |
| userName | 用户名 |
| userTel | 用户电话 |
| userEmail | 用户邮箱 |
| userIdentity | 用户身份证明 |
| agency | 所属机构 |
| availableBalance | 可用余额 |
| frozenBalance | 冻结余额 |
| isFrozen | 是否被冻结（冻结为true） |

⑤参考示例：假设数据库中存在ID为3的用（商）户。

|  |
| --- |
| accountService.userInformation(3); |

⑥错误代码参考：

|  |
| --- |
| accountService.userInformation(); //缺少必填参数 |

## 4.2.8 freezeUnfreeze方法

①方法描述：冻结、解冻账户。

②请求参数：如表4.21所示。

**表4.21 请求参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数类型** | **参数名** | **参数描述** | **备注** |
| int | user\_id | 用户ID | 无 |
| boolean | is\_frozen | 操作类型 | true代表冻结，false代表解冻 |

③响应参数：如表4.22所示。注意该方法不判断操作前用户的账户状态，例：若账户原本已经冻结，is\_frozen传入true，账户状态不变，但仍返回1，即认为冻结成功。

**表4.22 响应参数**

|  |  |
| --- | --- |
| **响应类型** | **响应描述** |
| int | 0操作无效，1冻结，2解冻 |

④异常描述：如表4.23所示。

**表4.23 异常描述**

|  |  |
| --- | --- |
| **异常类型** | **异常描述** |
| UserNotExistException | 不存在该ID的用户 |

⑤参考示例：假设数据库中存在ID为3的用（商）户。

|  |
| --- |
| accountService.freezeUnfreeze(3,true); |

⑥错误代码参考：

|  |
| --- |
| accountService.freezeUnfreeze(3,2); //参数类型不一致 |

## 4.2.9 foundPasswd方法

①方法描述：用身份认证直接替换现有密码。

②请求参数：如表4.24所示。

**表4.24 请求参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数类型** | **参数名** | **参数描述** | **备注** |
| String | user\_name | 用户名 | 无 |
| String | user\_identity | 身份证号/商户许可证号 | 根据用户身份增加相应前缀，用户为Y，商户为S |
| String | user\_passwd | 用户新密码 | SHA256 |

③响应参数：如表4.25所示。

**表4.25 响应参数**

|  |  |
| --- | --- |
| **响应类型** | **响应描述** |
| boolean | true 成功，false失败 |

④异常描述：如表4.26所示。

**表4.26 异常描述**

|  |  |
| --- | --- |
| **异常类型** | **异常描述** |
| UserNotExistException | 用户不存在，无法找回密码 |
| UserFrozenException | 用户被冻结，无法找回密码 |

⑤参考示例：假设数据库中存在用（商）户xyz。

|  |
| --- |
| accountService.foundPasswd("xyz","1234567890","0bfe935e70c321c7ca3afc75ce0d0ca2f98b5422e008bb31c00c6d7f1f1c0ad6"); |

⑥错误代码参考：

|  |
| --- |
| accountService.foundPasswd("xyz","1234567890","0bfe935e70c321c7ca3afc75ce0d0ca2f98b5422e008bb31c00c6d7f1f1c0ad6", false); //参数个数不一致 |

## 4.2.10 agencyTradeInformation方法

①方法描述：获取机构交易记录。

②请求参数：如表4.27所示。

**表4.27 请求参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数类型** | **参数名** | **参数描述** | **备注** |
| int | agency\_id | 机构ID | 无 |
| String | start\_date | 记录开始日期 | 无 |
| String | end\_date | 记录结束日期 | 无 |
| int | trade\_type | 交易类型 | 0充值，1提现，2转账与消费 |

③响应参数：如表4.28所示。其中map的key为交易ID，map的value为交易信息。

**表4.28 响应参数**

|  |  |
| --- | --- |
| **响应类型** | **响应描述** |
| List<Map<Integer,Object>> | 由交易编号、交易明细组成的键值对的列表 |

④参考示例：假设数据库中存在ID为6的机构。

|  |
| --- |
| accountService.agencyTradeInformation(6,"2018-04-01 13:50:31","2018-05-01 13:50:31",1); |

⑤错误代码参考：

|  |
| --- |
| accountService.agencyTradeInformation(6,"2018-06-01 13:50:31","2018-05-01 13:50:31",1); //记录开始日期晚于结束日期 |

## 4.2.11 userTradeInformation方法

①方法描述：获取用户交易记录。

②请求参数：如表4.29所示。

**表4.29 请求参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数类型** | **参数名** | **参数描述** | **备注** |
| int | user\_id | 用户ID | 无 |
| String | start\_date | 记录开始日期 | 无 |
| String | end\_date | 记录结束日期 | 无 |
| int | trade\_type | 交易类型 | 0充值，1提现，2转账与消费 |

③响应参数：如表4.30所示。其中map的key为交易ID，map的value为交易信息。

**表4.30 响应参数**

|  |  |
| --- | --- |
| **响应类型** | **响应描述** |
| List<Map<Integer,Object>> | 由交易编号、交易明细组成的键值对的列表 |

④参考示例：假设数据库中存在ID为6的用（商）户。

|  |
| --- |
| accountService.userTradeInformation(6,"2018-04-01 13:50:31","2018-05-01 13:50:31",1); |

⑤错误代码参考：

|  |
| --- |
| accountService.userTradeInformation(6,"2018-06-01 13:50:31","2018-05-01 13:50:31",1); //记录开始日期晚于结束日期 |

## 4.2.12 transferConsumer方法

①方法描述：转账与消费。

②请求参数：如表4.31所示。

**表4.31 请求参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数类型** | **参数名** | **参数描述** | **备注** |
| int | pay\_user\_id | 付款方用户ID | 无 |
| int | get\_user\_id | 收款方用户ID | 无 |
| double | amount | 转账额 | 无 |
| boolean | trade\_type | 交易类型 | false转账，true消费 |

③响应参数：如表4.32所示。

**表4.32 响应参数**

|  |  |
| --- | --- |
| **响应类型** | **响应描述** |
| boolean | 成功返回true，失败返回false |

④异常描述：如表4.33所示。

**表4.33 异常描述**

|  |  |
| --- | --- |
| **异常类型** | **异常描述** |
| UserNotExistException | 用户不存在，无法转帐或消费 |
| UserFrozenException | 用户被冻结，无法转账或消费 |

⑤参考示例：假设数据库中存在ID为3和5的用（商）户。

|  |
| --- |
| accountService.transferConsume(3,5,100.00,false); |

⑥错误代码参考：

|  |
| --- |
| accountService.transferConsume(3,5,-100.00,false); //转账额为负数 |

## 4.2.13 reCharge方法

①方法描述：充值。

②请求参数：如表4.34所示。

**表4.34 请求参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数类型** | **参数名** | **参数描述** | **备注** |
| int | user\_id | 用户ID | 无 |
| double | amount | 充值金额 | 无 |
| boolean | recharge\_platform | 使用平台 | false微信，true支付宝 |

③响应参数：如表4.35所示。

**表4.35 响应参数**

|  |  |
| --- | --- |
| **响应类型** | **响应描述** |
| boolean | 成功返回true，失败返回false |

④异常描述：如表4.36所示。

**表4.36 异常描述**

|  |  |
| --- | --- |
| **异常类型** | **异常描述** |
| UserNotExistException | 用户不存在，无法充值 |

⑤参考示例：假设数据库中存在ID为4的用（商）户。

|  |
| --- |
| accountService.reCharge(4,6.00,true); |

⑥错误代码参考：

|  |
| --- |
| accountService.reCharge(4,-6.00,true); //充值金额为负数 |

## 4.2.14 drawMoney方法

①方法描述：提现。

②请求参数：如表4.37所示。

**表4.37 请求参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数类型** | **参数名** | **参数描述** | **备注** |
| int | user\_id | 用户ID | 无 |
| double | amount | 取款金额 | 无 |
| boolean | recharge\_platform | 使用平台 | false微信，true支付宝 |

③响应参数：如表4.38所示。

**表4.38 响应参数**

|  |  |
| --- | --- |
| **响应类型** | **响应描述** |
| boolean | 成功返回true，失败返回false |

④异常描述：如表4.39所示。

**表4.39 异常描述**

|  |  |
| --- | --- |
| **异常类型** | **异常描述** |
| UserNotExistException | 用户不存在，无法提现 |

⑤参考示例：假设数据库中存在ID为4的用（商）户。

|  |
| --- |
| accountService.drawMoney(6,4.00,false); |

⑥错误代码参考：

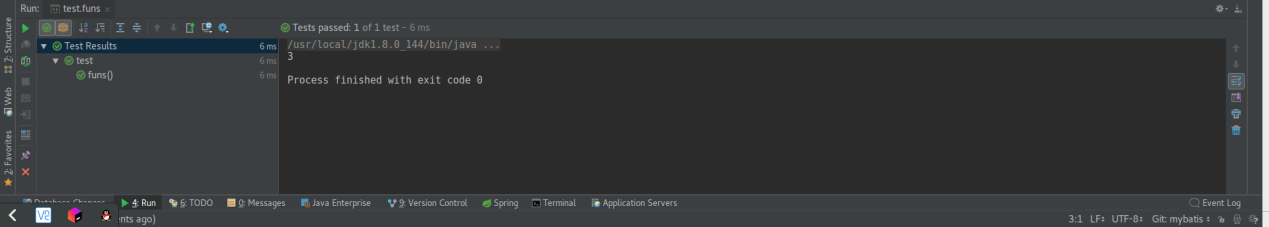
|  |
| --- |
| accountService.drawMoney(6,-4.00,false); //取款金额为负数 |

## 4.3 求助查询

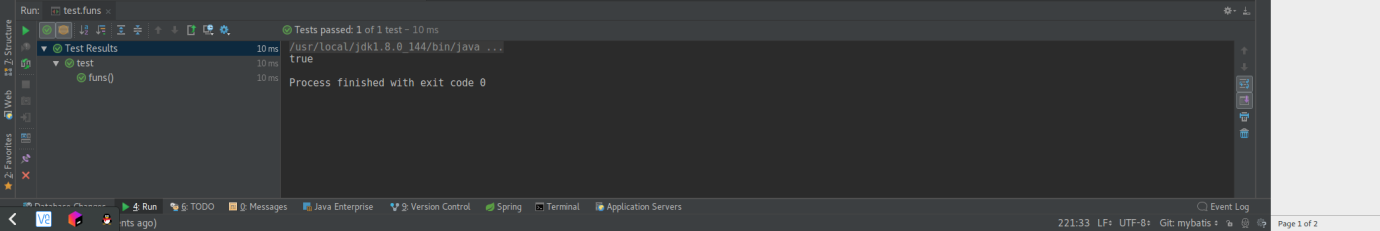
联系本项目负责人林未，QQ：369200411。

## 五、附录

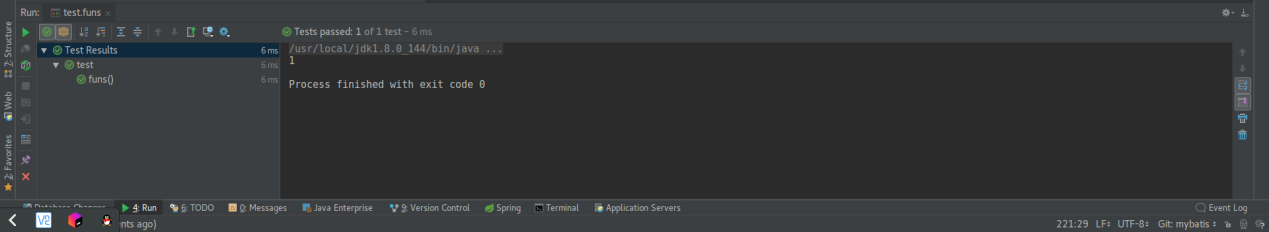
以下为接口内各方法调用实例结果图：



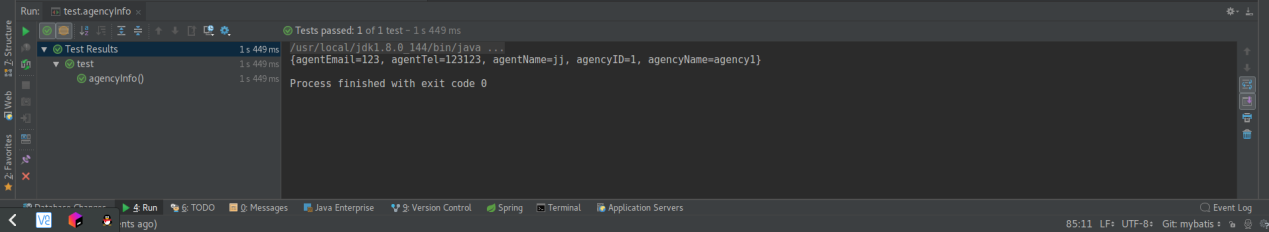
**图5.1 登录/注册结果截图**



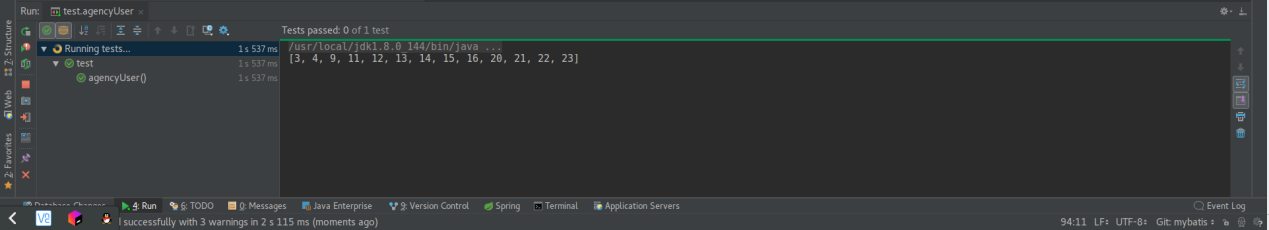
**图5.2 修改密码/找回密码/所有交易结果截图**



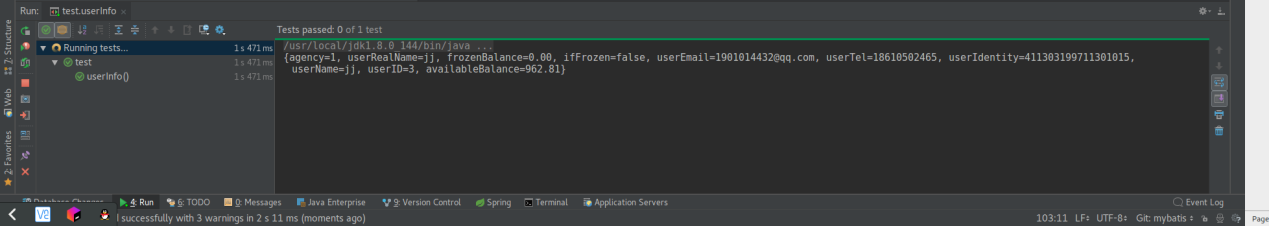
**图5.3 冻结解冻结果截图**



**图5.4 查询机构信息结果截图**



**图5.5 查询机构用户列表结果截图**



**图5.6 查询用户信息结果截图**