**集团统一登录对接文档**

[1. 统一登录整体说明 5](#_Toc475957242)

[1.1对接文档修改履历 5](#_Toc475957243)

[1.2 对接功能概述 6](#_Toc475957244)

[1.2 关于接口调用 7](#_Toc475957245)

[1.2.1 接口调用方式 7](#_Toc475957246)

[1.2.2接口调用注意事项 7](#_Toc475957247)

[1.2.3对接流程图(有数据迁移场景) 7](#_Toc475957248)

[2用户信息功能配置pom文件引入jar包 8](#_Toc475957249)

[3各环境dubbo地址 8](#_Toc475957250)

[4各公司数据同步到国美控股统一登录系统 8](#_Toc475957251)

[4.1各公司需要提供的信息 8](#_Toc475957252)

[4.2统一登录系统同步数据 9](#_Toc475957253)

[5各功能接口调用详细说明 9](#_Toc475957254)

[5.1注册前校验用户名，手机号，邮箱是否存在 9](#_Toc475957255)

[5.1.1接口名 9](#_Toc475957256)

[5.1.2Dubbo接口客户端配置 9](#_Toc475957257)

[5.1.3需要统一登录系统分配的参数 9](#_Toc475957258)

[5.1.4接口说明及调用Demo 9](#_Toc475957259)

[5.2注册新增用户 10](#_Toc475957260)

[5.2.1接口名 10](#_Toc475957261)

[5.2.2Dubbo接口客户端配置 10](#_Toc475957262)

[5.2.3需要统一登录系统分配的参数 10](#_Toc475957263)

[5.2.4接口说明及调用Demo 10](#_Toc475957264)

[5.3普通登录前的相关校验 12](#_Toc475957265)

[5.3.1普通登录前校验是否显示验证码 12](#_Toc475957266)

[5.3.2普通登录前校验用户名是否可以登录 13](#_Toc475957267)

[5.4普通登录 14](#_Toc475957268)

[5.4.1接口名 14](#_Toc475957269)

[5.4.2Dubbo接口客户端配置 14](#_Toc475957270)

[5.4.3需要统一登录系统分配的参数 14](#_Toc475957271)

[5.4.4接口说明及调用Demo 15](#_Toc475957272)

[5.5第三方登录 16](#_Toc475957273)

[5.5.1接口名 16](#_Toc475957274)

[5.5.2Dubbo接口客户端配置 16](#_Toc475957275)

[5.5.3需要统一登录系统分配的参数 16](#_Toc475957276)

[5.5.4接口说明及调用Demo 16](#_Toc475957277)

[5.6登录状态相关功能 17](#_Toc475957278)

[5.6.1需要导入的jar 17](#_Toc475957279)

[5.6.2dubbo接口和redis配置 18](#_Toc475957280)

[5.6.3登录或者注册后保存登录状态 18](#_Toc475957281)

[5.6.4验证登录状态 18](#_Toc475957282)

[5.6.5延长登录状态 19](#_Toc475957283)

[5.6.6不需要登录或者注册即可保存登录状态 20](#_Toc475957284)

[5.7根据userId获取用户信息 21](#_Toc475957285)

[5.7.1接口名 21](#_Toc475957286)

[5.7.2Dubbo接口客户端配置 21](#_Toc475957287)

[5.7.3需要统一登录系统分配的参数 21](#_Toc475957288)

[5.7.4接口说明及调用Demo 21](#_Toc475957289)

[5.8修改用户信息-校验手机号或邮箱是否已被使用 22](#_Toc475957290)

[5.8.1接口名 22](#_Toc475957291)

[5.8.2 Dubbo接口客户端配置 22](#_Toc475957292)

[5.8.3需要统一登录系统分配的参数 23](#_Toc475957293)

[5.8.4接口说明及调用Demo 23](#_Toc475957294)

[5.9 重置密码 23](#_Toc475957295)

[5.9.1接口名 23](#_Toc475957296)

[5.9.2Dubbo接口客户端配置 23](#_Toc475957297)

[5.9.3需要统一登录系统分配的参数 24](#_Toc475957298)

[5.9.4接口说明及调用Demo 24](#_Toc475957299)

[5.10修改及激活手机号 25](#_Toc475957300)

[5.10.1接口名 25](#_Toc475957301)

[5.10.2Dubbo接口客户端配置 25](#_Toc475957302)

[5.10.3需要统一登录系统分配的参数 25](#_Toc475957303)

[5.10.4接口说明及调用Demo 25](#_Toc475957304)

[5.11修改及激活邮箱 26](#_Toc475957305)

[5.11.1接口名 26](#_Toc475957306)

[5.11.2Dubbo接口客户端配置 26](#_Toc475957307)

[5.11.3需要统一登录系统分配的参数 26](#_Toc475957308)

[5.11.4接口说明及调用Demo 26](#_Toc475957309)

[5.12修改手机号激活状态 27](#_Toc475957310)

[5.12.1接口名 27](#_Toc475957311)

[5.12.2Dubbo接口客户端配置 27](#_Toc475957312)

[5.12.3需要统一登录系统分配的参数 27](#_Toc475957313)

[5.12.4接口说明及调用Demo 27](#_Toc475957314)

[5.13修改邮箱激活状态 28](#_Toc475957315)

[5.13.1接口名 28](#_Toc475957316)

[5.13.2Dubbo接口客户端配置 28](#_Toc475957317)

[5.13.3需要统一登录系统分配的参数 28](#_Toc475957318)

[5.13.4说明及调用Demo 28](#_Toc475957319)

[5.14第三方用户完善个人信息 29](#_Toc475957320)

[5.14.1接口名 29](#_Toc475957321)

[5.14.2Dubbo接口客户端配置 29](#_Toc475957322)

[5.14.3需要统一登录系统分配的参数 29](#_Toc475957323)

[5.14.4接口说明及调用Demo 29](#_Toc475957324)

[5.15校验用户是否可以找回密码 30](#_Toc475957325)

[5.15.1接口名 30](#_Toc475957326)

[5.15.2Dubbo接口客户端配置 30](#_Toc475957327)

[5.15.3需要统一登录系统分配的参数 31](#_Toc475957328)

[5.15.4接口说明及调用Demo 31](#_Toc475957329)

[5.16校验用户是否可以修改密码 32](#_Toc475957330)

[5.16.1接口名 32](#_Toc475957331)

[5.16.2Dubbo接口客户端配置 33](#_Toc475957332)

[5.16.3需要统一登录系统分配的参数 33](#_Toc475957333)

[5.16.4接口说明及调用Demo 33](#_Toc475957334)

[5.17修改密码 34](#_Toc475957335)

[5.17.1接口名 34](#_Toc475957336)

[5.17.2Dubbo接口客户端配置 34](#_Toc475957337)

[5.17.3需要统一登录系统分配的参数 34](#_Toc475957338)

[5.17.4接口说明及调用Demo 34](#_Toc475957339)

[5.18 查询第三方用户完善资料状态 35](#_Toc475957340)

[5.18.1接口名 35](#_Toc475957341)

[5.18.2Dubbo接口客户端配置 35](#_Toc475957342)

[5.18.3需要统一登录系统分配的参数 36](#_Toc475957343)

[5.18.4接口说明及调用Demo 36](#_Toc475957344)

[5.19校验用户是否可以找回密码，根据优先级返回用户(推荐) 36](#_Toc475957345)

[5.19.1接口名 36](#_Toc475957346)

[5.19.2Dubbo接口客户端配置 37](#_Toc475957347)

[5.19.3需要统一登录系统分配的参数 37](#_Toc475957348)

[5.19.4接口说明及调用Demo 37](#_Toc475957349)

[5.20用户授权 38](#_Toc475957350)

[5.20.1接口名 38](#_Toc475957351)

[5.20.2Dubbo接口客户端配置 38](#_Toc475957352)

[5.20.3需要统一登录系统分配的参数 39](#_Toc475957353)

[5.20.4接口说明及调用Demo 39](#_Toc475957354)

[5.21 查询用户是否可以升级一账通 39](#_Toc475957355)

[5.21.1接口名 39](#_Toc475957356)

[5.21.2Dubbo接口客户端配置 39](#_Toc475957357)

[5.21.3需要统一登录系统分配的参数 40](#_Toc475957358)

[5.21.4接口说明及调用Demo 40](#_Toc475957359)

[5.22手机号是否为被其它用户绑定 40](#_Toc475957360)

[5.22.1接口名 40](#_Toc475957361)

[5.22.2Dubbo接口客户端配置 41](#_Toc475957362)

[5.22.3需要统一登录系统分配的参数 41](#_Toc475957363)

[5.22.4接口说明及调用Demo 41](#_Toc475957364)

[5.23第三方首次登录后绑定到现有手机号的账号上并登陆 42](#_Toc475957365)

[5.23.1接口名 42](#_Toc475957366)

[5.23.2Dubbo接口客户端配置 42](#_Toc475957367)

[5.23.3需要统一登录系统分配的参数 42](#_Toc475957368)

[5.23.4接口说明及调用Demo 42](#_Toc475957369)

[5.24根据用户id查询用户授权状态以及授权时间 43](#_Toc475957370)

[5.24.1接口名 43](#_Toc475957371)

[5.24.2Dubbo接口客户端配置 43](#_Toc475957372)

[5.24.3需要统一登录系统分配的参数 43](#_Toc475957373)

[5.24.4接口说明及调用Demo 43](#_Toc475957374)

[6各功能所用到的接口及调用顺序 44](#_Toc475957375)

[6.1注册功能相关接口调用说明 44](#_Toc475957376)

[6.2普通登录功能相关接口调用说明 44](#_Toc475957377)

[6.3修改手机号功能相关接口调用说明 45](#_Toc475957378)

[6.4修改邮箱功能相关接口调用说明 45](#_Toc475957379)

[6.5找回密码功能相关接口调用说明 46](#_Toc475957380)

[6.6修改密码功能相关接口调用说明 46](#_Toc475957381)

[6.7第三方用户完善资料功能相关接口调用说明 46](#_Toc475957382)

[6.8第三方首次登录后，绑定到现有手机号的账号上并登陆说明 47](#_Toc475957383)

[7参考说明 47](#_Toc475957384)

[7.1第三方登录WhereFrom参考值 47](#_Toc475957385)

[7.2用户登录名login的生成规则 48](#_Toc475957386)

|  |  |
| --- | --- |
| 文档编写日期 | 2016-9-20 |
| 当前版本： | 1.2.2 |
| 国美控股统一登录开发小组 | 国美控股统一登录研发组 |
| 接口对接人 | 王桦，杨军鹏，杜瑞阳 |

# 统一登录整体说明

## 1.1对接文档修改履历

|  |  |  |  |
| --- | --- | --- | --- |
| **对接文档修改履历** | | | |
| **版本号** | **修改时间** | **修改人** | **修改内容** |
| 1.1.0 | 2016/9/20 | 王桦 | 新增第三方用户完善个人资料接口调用步骤（5.14） |
| 1.1.1 | 2016/10/15 | 王桦  栾嘉友 | 1.新增"不需要登录或者注册即可保存登录状态接口“（5.6.6）;  2.更新获取用户信息接口（5.7），找回密码接口（6.5）和修改密码接口（6.6）  3.修改dubbo接口调用的描述（1.2.2） |
| 1.1.2 | 2016/10/17 | 王桦  杜瑞阳 | 1. 修改“注册新增用户”接口的调用方法及初始化数据（5.2） 2. 修改“重置密码”接口(5.9)，改动方法名及接口入参 3. Pom.xml中需要配置的jar具体到版本（2和5.6.1） |
| 1.1.3 | 2016/10/27 | 杜瑞阳  王倩 | 1.每个接口添加分组unifyLogin  2.登录状态相关工具类和Filter修改  3.为金库添加新的接口，修改jar版本号  4.接口5.15返回类型修改为  MapResult<Map<String,Object>> |
| 1.1.4 | 2016/12/2 | 杨军鹏  杜瑞洋 | 1.修改 5.3.2普通登录前校验用户名是否可以登录 demo  2. 登录接口增加返回值(5.4)  3. 检查能否找回密码接口，增加不同渠道重复用户的场景(5.15) |
| 1.1.5 | 2016/12/13 | 刘芷维  杜瑞洋 | 1.增加5.2注册接口枚举返回类型  2. 增加接口5.19 检查用户能否找回密码（冲突用户会进行筛选），与5.15区分 |
| 1.1.6 | 2016/12/16 | 杨军鹏 刘芷维  康宁 | **本次主要改动：修改升级一账通功能相关接口**  1. 5.4普通登录返回字段新增“是否授权字段”。  2. 添加5.20用户授权接口 3. 修改验证用户是否存在接口 4. 注册对象增加授权字段，在extMap中增加扩展字段  5. 5.3.2普通登录前校验用户名是否可以登录添加扩展Map  6. 找回密码相关接口、修改用户信息相关接口添加扩展参数 |
| 1.1.7 | 2016/12/20 | 刘芷维 | 修改5.1注册前校验用户名，手机号，邮箱是否存在 |
| 1.1.8 | 2016/12/28 | 杨军鹏 | 新增5.21 查询用户是否可以升级一账通 |
| 1.1.9 | 2017/01/10 | 康宁 | 找回密码类、修改用户信息类接口返回枚举完善 |
| 1.2.0 | 2017/02/17 | 刘芷维 | 1.新增5.22手机号是否为被其它用户绑定  2.新增5.23第三方首次登录绑定手机后登陆 |
| 1.2.1 | 2017/02/20 | 康宁 | 1.修改5. 5.3普通登录前校验用户名是否可以登录  2.新增5.24 根据用户id查询用户授权状态以及授权时间 |
| 1.2.2 | 2017/02/23 | 杨军鹏 | 新增6.8第三方用户绑定手机号流程说明 |

## 1.2 对接功能概述

|  |  |
| --- | --- |
| **基础功能** | **功能描述** |
| 各子公司数据同步到集团统一登录系统 | 同步数据使用，不需要接口调用，需要开通从统一登录系统直接访问数据库权限；需要各公司提供表和字段的说明 |
| 注册 | 会为每个子公司分配注册来源；注册前要校验用户名，手机号，邮箱是否已经存在 |
| 第三方登录 | Oauth2.0相关各公司自己处理，Oauth2.0之后校验用户信息和保存用户信息调用接口 |
| 普通登录 |  |
| 注册校验用户名是否存在 | 注册时校验用户名使用 |
| 注册校验手机号或邮箱是否已被使用 | 注册时填写手机号或者邮箱校验使用 |
| 登陆后修改密码 |  |
| 登录前找回密码 | 可以根据登录名，手机号，邮箱找回 |
| 根据用户Id获取用户信息 | 只能根据统一登录系统的userId获取用户信息，如果是老数据，会包含各公司之前生成userId |
| 修改及激活手机号 |  |
| 修改及激活邮箱 |  |
| 修改手机号激活状态 |  |
| 修改邮箱激活状态 |  |
| 保存登录状态 | 登录成功后，会返回SCN，需要各公司把SCN保存在cookie中，同时集团统一登录系统里也会保存用户的登录状态 |
| 验证登录状态 | 在验证登录状态的时候，需要调用相应的方法和接口 |
| 第三方登录用户完善个人信息 | 如qq，微信等第三方登录用户，完善统一登录系统的密码，手机号等信息 |
| 根据授权信息送优惠券 | 获取用户授权相关信息后，可以根据授权信息向用户赠予优惠卷，以此激励用户授权 |

## 1.2 关于接口调用

### 1.2.1 接口调用方式

统一使用Dubbo,Dubbo相关文档都在这里：<http://dubbo.io/Home-zh.htm>

### 1.2.2接口调用注意事项

* + - 必须和集团统一登录系统应用在同一个内网，如果内网不通的，需要打通内网
    - 老数据在导入到国美集团统一登录数据库时会新生成一个userId，获取用户信息只能用统一登录系统的userId获取。

### 1.2.3对接流程图(有数据迁移场景)



# 2用户信息功能配置pom文件引入jar包

<dependency>

<groupId>com.gome</groupId>

<artifactId>userCenter-model</artifactId>

<version>STABLE-1.0.11</version>

</dependency>

<dependency>

<groupId>com.gome</groupId>

<artifactId>userCenter-facade</artifactId>

<version>STABLE-1.0.11</version>

</dependency>

# 3各环境dubbo地址

***<!-- UAT环境测试联调环境-->***

<dubbo:registry

protocol=*"zookeeper"*address=*"对接的时候提供"* />

***<!—4344环境和生产共用一个数据库，不对用户开放 -->***

<dubbo:registry

protocol="zookeeper"

address="*对接的时候提供*" />

***<!--生产 -->***

<dubbo:registry

protocol="zookeeper"

address="对接的时候提供" />

# 4各公司数据同步到国美控股统一登录系统

## 4.1各公司需要提供的信息

* 需要各公司开通统一登录系统直连各公司测试和生产数据库的权限

具体信息：ip，端口号，用户名，密码

* 需要各公司提供登录和注册相关表和字段的详细描述

具体信息：用户id，用户名，密码，手机号，邮箱，第三方登录id，第三方登录用户名所在的数据库名和所属表的建表sql

## 4.2统一登录系统同步数据

统一登录系统会通过程序直连各公司的数据库，分别把各公司生产和测试环境的数据迁移到统一登录系统中。

# 5各功能接口调用详细说明

## 5.1注册前校验用户名，手机号，邮箱是否存在

### 5.1.1接口名

com.gome.userCenter.facade.register.IUserRegisterFacade.existUser(String login,String invokeFrom,Map extMap)

说明：

* 包括对login,mobile,email是否存在的校验
* 注册，修改手机号，修改邮箱，完善第三方用户信息时，都会用到此接口

### 5.1.2Dubbo接口客户端配置

<dubbo:reference interface="com.gome.userCenter.facade.register.IUserRegisterFacade" id="userRegisterFacade" group="unifyLogin"/>

### 5.1.3需要统一登录系统分配的参数

* 调用方标识：invokeFrom
* 扩展字段：extMap

### 5.1.4接口说明及调用Demo

**Publicvoid**existUser(){

Map<String,Object> extMap=new HashMap<String,Object>();

extMap.put("companyName", "gomeOnline");//统一登录系统会为每个公司分配自己

**MapResult<Void>**result = userRegisterFacade.existUser("test",”invoke”, extMap);//参数要查询的用户名

**if**(result.isSuccess){

//用户名存在; 用户所在渠道编号，格式为Set<String>，即该手机号可能存在于多个渠道

Set<String> keys=result.getExtParams.get("subCompanyCode");

}**else**{

//用户名不存在

}

}

错误码枚举：以result.isSuccess 为准判断手机号是否被占用即可，无错误码枚举

## 5.2注册新增用户

### 5.2.1接口名

com.gome.userCenter.facade.register.IUserRegisterFacade. registerUnifyUser(GomeUnifyRegisterUsergomeUser,Map<String,Object>extMap)

### 5.2.2Dubbo接口客户端配置

<dubbo:reference interface="com.gome.userCenter.facade.register.IUserRegisterFacade" id="userRegisterFacade" group="unifyLogin"/>

### 5.2.3需要统一登录系统分配的参数

* 注册来源:GomeUnifyRegisterUser的registerSource
* 站点: GomeUnifyRegisterUser的registerSite
* 调用渠道:extMap的invokeChannel
* 公司名称:extMap的companyName
* 说明：registerSource，registerSite和invokeChannel可以一样

### 5.2.4接口说明及调用Demo

说明：

* 如果对接方公司没有login，统一登录系统会自动生成login且全集团唯一，生成login规则如7.2描述
* 普通注册分为手机注册和邮箱注册，统一登录系统会根据registerType值分取mobile或者email并对mobile，email的格式做校验；如果registerType和对应取出的mobile或者email的值不匹配，会返回注册失败。

**Publicvoid**registerUser() {

IUserRegisterFacadeuserRegisterFacade = (IUserRegisterFacade) context.getBean("userRegisterFacade");

GomeUnifyRegisterUsergomeUser = newGomeUnifyRegisterUser();

gomeUser.setRegisterIp("127.0.0.1");//用户ip--必填

gomeUser.setHostName("gome\_test\_225");//接口调用端服务器名称--必填

gomeUser.setHostPort(80);//接口调用应用端口号

gomeUser.setLogin("13461337564");//登录名—非必填

gomeUser.setPassword("1234567abc");//明文密码--必填

gomeUser.setMobile("13461337564");//手机号

// gomeUser.setEmail("wanghua306@126.com");////手机和邮箱二选一

gomeUser.setRegisterType(“mobileReg”);//手机注册

// gomeUser.setRegisterType(“emailReg”);//手机注册和邮箱注册二选一

gomeUser.setRegisterSource(“gomeOnline”);//统一登录系统会为每个公司分配自己的RegisterSource--必填

gomeUser.setRegisterSite("gome");//统一登录系统会为每个公司分配各自的站点--必填

Map<String, Object>extMap = **new**HashMap<String, Object>();

extMap.put("userRegisterType", "Person");//个人注册--必填

extMap.put("invokeChannel", "");//统一登录系统会为每个公司分配调用渠道--必填

extMap.put("isAuthorization", "Y");//授权码 Y是授权 N 不授权

extMap.put("companyName", "gome");//统一登录系统会为每个公司分配名字--必填

extMap.put("isBindSnsUser", "true");//当注册绑定第三方用户时设置

GomeSNSUser gomeSNSUser = **new** GomeSNSUser(**null**,"baidu","999999999","若非群玉山头见");//构造第三方用户

extMap.put("gomeSnsUser", "gome");//当注册绑定第三方用户时，设置

UserRegisterResult<GomeUnifyRegisterUser>result=userRegisterFacade.registerUnifyUser(gomeUser, extMap);

**if**(result.isSuccess()){//result一定会返回，不用判断null，注册成功

System.*out*.println("注册成功"+JSONObject.*toJSONString*(result));

System.*out*.println(“SCN需要调用方保存到cooki*e"+ result.extMap().get("SCN"));*

}**else**{//注册失败

System.*out*.println("注册失败页面提示信息"+result.getMessage());

System.*out*.println("注册失败异常堆栈信息"+result.getExceptionMessage());

}

}

注册枚举返回类型：

reg\_verify\_suc(0000,"校验通过！"),

gomeRegisterUser\_null(0001,"注册用户为空"),

//1 system|开头系统异常

system\_reg\_error(1000,"system|注册失败，请稍后再试"),

system\_user\_exsit(1001,"system|用户已存在，注册失败"),

system\_reg\_ifFull(1002,"system|今日注册名额已用完，请明天再试"),

system\_ip\_ifFull(1003,"system|该IP48小时注册次数超过限制"),

system\_reg\_ifFull\_over50(1004,"system|该IP48小时内超过50个用户，且50%的密码都一致"),

//2 login|登录错误

login\_repeat\_error(2000,"login|请勿重复提交"),

login\_userName\_null(2001,"login|请输入用户名"),

login\_email\_len\_error(2002,"login|邮箱地址长度应在4-50个字符之间"),

login\_email\_error(2003,"login|邮箱地址不正确，请重新输入"),

login\_userName\_len\_null(2004,"login|用户名长度只能在4-20个字符之间"),

login\_userName\_error(2005,"login|用户名只能由中文、英文、数字及\"-\"、\"\_\"组成"),

login\_phone\_error(2006,"login|请输入正确的手机号"),

login\_userName\_dig(2007,"login|不能是纯数字"),

login\_phoneOrEmail\_null(2008,"login|请输入手机号或邮箱"),

//3 password|密码错误

pwd\_null(3001,"password|请输入密码"),

pwd\_len\_error(3002,"password|密码长度应为6-20个字符"),

pwd\_char\_same(3003,"password|密码不能为同一字符"),

pwd\_not\_dig(3004,"password|密码不能全为数字"),

pwd\_illegal(3005,"password|非法字符，请使用字母加数字或符号的组合，6-20个字符"),

pwd\_easy(3006,"password|该密码比较简单，有被盗风险"),

;

## 5.3普通登录前的相关校验

### 5.3.1普通登录前校验是否显示验证码

#### 5.3.1.1接口名

com.gome.userCenter.facade.login.IUserLoginFacade.isShowAuthenticCodeWithGrade(String login,String ipAddress,String invokeFrom,Map<String, String> captExtMap)

#### 5.3.1.2Dubbo接口客户端配置

<dubbo:reference

interface="com.gome.userCenter.facade.login.IUserLoginFacade" id="userLoginFacade" group="unifyLogin"/>

#### 5.3.1.3需要统一登录系统分配的参数

* 调用方标识：invokeFrom

#### 5.3.1.4接口说明及调用Demo

**Publicvoid**testShowAuthenticCode(){

//第一个参数用户名

//第二个参数用户登录ip

Result<ShowCaptchGrade> result = userLoginFacade.isShowAuthenticCodeWithGrade("whtest","21.0.8.9", invokeFrom, **null**);

**if**(result.isSuccess()){

//前台页面显示验证码

}

}

### 5.3.2普通登录前校验用户名是否可以登录

#### 5.3.2.1接口名

com.gome.userCenter.facade.login.IUserLoginFacade.isConflictOrSnsOrNikeName(String login,String invokeFrom, Map<String, Object> extMap)

#### 5.3.2.2接口客户端配置

<dubbo:referenceinterface="com.gome.userCenter.facade.login.IUserLoginFacade" id="userLoginFacade" group="unifyLogin"/>

#### 5.3.2.3需要统一登录系统分配的参数

* 调用方标识：invokeFrom

#### 5.3.2.4接口说明及调用Demo

public void testIsConflictOrSnsOrNikeName (){

String loginName = "test";

Map<String, Object> extMap = **new** HashMap<String, Object>();

extMap.put("companyName", "gomeOnline");

UserResult<Map<String, Object>> validateResult = loginFacade.isConflictOrSnsOrNikeName(loginName,"invokeFrom",extMap);

if (validateResult.isSuccess()) {

Map<String, Object> resultMap = validateResult.getBuessObj();

boolean isNikename=(Boolean) resultMap.get("isNikename");//是否是昵称

String whereFrom = (String)resultMap.get("whereFrom");//用户来源

if((!"gome".equalsIgnoreCase(whereFrom))

&& (!"hqm".equalsIgnoreCase(whereFrom))){

//第三方用户名不允许登录

}

if("hqm".equalsIgnoreCase(whereFrom) && loginName.startsWith("hqm-")){

//如果是门店用户 ，不能使用hqm-开头的login去登录，不允许登录

}

if(isNikename){

//昵称不允许登录

}

}else{

//校验用户名失败，不允许登录

}

}

#### 5.3.2.5 返回参数示例

isAuthorized说明：

isAuthorized 是否已授权一账通

true:不需要显示强制升级一账通多选框

false：需要显示强制升级一账通多选框

{"isConflict":false,"isEmailAddress":false,"isMobileNumber":false,"isNikename":false,"isNotInitPage":false,"referisError":false,"whereFrom":"gome",“isAuthorized”,true}

错误码枚举：

401,"参数invokeFrom校验不通过"

500,"接口内部异常"

## 5.4普通登录

### 5.4.1接口名

com.gome.userCenter.facade.login.IUserLoginFacade.doLogin(String login,Stringpassword,RequestParamsrequestParams,Map map)

### 5.4.2Dubbo接口客户端配置

<dubbo:referenceinterface="com.gome.userCenter.facade.login.IUserLoginFacade " id="userLoginFacade" group="unifyLogin"/>

### 5.4.3需要统一登录系统分配的参数

* 调用渠道:RequestParams的invokeChannel
* 站点: map的loginSite
* 公司名称:map的companyName
* 登录的同时是否执行授权:map的isAuthorized

5.4.4接口说明及调用Demo  
**publicvoid**doLogin() {

IUserLoginFacadeuserLoginFacade = (IUserLoginFacade) context.getBean("userLoginFacade");

String login = "whqwer41";

String password = "a111111";

Map<String, Object> map = **new**HashMap<String, Object>();

map.put("loginSite", "gome");//统一登录系统会为每个公司分配各自的站点--必填

map.put("companyName", "gomeOnline");//统一登录系统会为每个公司分配自己的公司名称--必填

map.put("isAuthorized", true);//是否是授权登录，是传true，每个公司只能授权自己的用户。授权了以后该公司的用户将能在其他渠道登录。

RequestParamsrequestParams = **new**RequestParams();

requestParams.setInvokeChannel("web");//调用渠道，统一登录系统分配invokeChannel--必填

requestParams.setClientIp("127.0.0.1");//服务调用方ip—为用户ip，必填

requestParams.setRemotePort("8080");//服务调用方远程端口号--非必填

requestParams.setHostName("DEV");//服务器名称--非必填

requestParams.setHostPort("80");//服务器部署占用端口号--非必填

//第一个参数：用户名

//第二个参数：明文密码

UserLoginResult<UserInfo> result = userLoginFacade.doLogin(login, password, requestParams, map);

**if**(result.isSuccess()){//result一定会返回，不用判断null，登录成功

System.*out*.println("登录成功,用户信息"+result.getBuessObj());

System.*out*.println("各公司userId"+result.getExtraInfoMap().get("subCompanyUserId"));

System.*out*.println("各公司编号"+result.getExtraInfoMap().get("subCompanyCode"));

System.*out*.println(“SCN需要调用方保存到cookie"+result.getExtraInfoMap().get("SCN"));

}**else**{//登录失败

System.*out*.println("登录失败页面提示信息"+result.getMessage());

}

}

错误枚举类：

com.gome.userCenter.model.enu.UserLoginEnum

## 5.5第三方登录

### 5.5.1接口名

com.gome.userCenter.facade.login.ISNSLoginFacade.doSNSLogin(GomeSNSUser snsUser,RequestParamsrequestParams, Map map)

### 5.5.2Dubbo接口客户端配置

<dubbo:referenceinterface="com.gome.userCenter.facade.login.ISNSLoginFacade " id="isnsLoginFacade" group="unifyLogin"/>

### 5.5.3需要统一登录系统分配的参数

* 调用渠道:RequestParams的invokeChannel
* 站点: map的loginSite
* 第三方登录类别:GomeSNSUser的whereFrom
* 公司名称:map的companyName
* 说明：loginSite和invokeChannel可以一样

### 5.5.4接口说明及调用Demo

**Publicvoid**doSNSLogin() {

ISNSLoginFacadeisnsLoginFacade = (ISNSLoginFacade) context.getBean("isnsLoginFacade");

RequestParamsrequestParams=**new**RequestParams();

requestParams = **new**RequestParams();

requestParams.setInvokeChannel("web");//调用渠道，统一登录系统分配invokeChannel--必填

requestParams.setClientIp("127.0.0.1");//服务调用方ip-为用户ip，-必填

requestParams.setRemotePort("8080");//服务调用方远程端口号--非必填

requestParams.setHostName("DEV");//服务器名称--非必填

requestParams.setHostPort("80");//服务器部署占用端口号--非必填

GomeSNSUsersnsUser = **new**GomeSNSUser();

snsUser.setSNSUserId("o0zUGt2fkuKQ\_zZLyHNNt1jCpyr81");//第三方snsuserid--必填

snsUser.setSNSUserName("诺言");//第三方登录用户名snsusername--必填

snsUser.setWhereFrom("22");//--必填，参考—>6.1第三方登录WhereFrom参考值

Map<String, Object> map = **new**HashMap<String, Object>();

map.put("loginSite", "gome");//统一登录系统会为每个公司分配各自的站点--必填

map.put("companyName", "gome");//统一登录系统会为每个公司分配名字--必填

map.put("snsuser\_register","false");//校验第三方用户是否存在

UserLoginResult<UserInfo> result = isnsLoginFacade.doSNSLogin(snsUser, requestParams, map);

**if**(result.isSuccess()){//result一定会返回，不用判断null，注册成功

System.*out*.println("注册成功"+JSONObject.*toJSONString*(result));

System.*out*.println("各公司userId"+result.getExtraInfoMap().get("subCompanyUserId"));

System.*out*.println(“SCN需要调用方保存到cookie"+result.getExtraInfoMap().get("SCN"));

}**else**{//注册失败

System.*out*.println("注册失败页面提示信息"+result.getMessage());

}

}

错误码枚举：

6001 不存在第三方用户（跳转到关联手机号页面）

## 5.6登录状态相关功能

### 5.6.1需要导入的jar

<dependency>

<groupId>com.gome</groupId>

<artifactId>sso-model</artifactId>

<version>STABLE-1.0.9</version>

</dependency>

<dependency>

<groupId>com.gome</groupId>

<artifactId>sso-facade</artifactId>

<version>STABLE-1.0.9</version>

</dependency>

<dependency>

<groupId>com.gome</groupId>

<artifactId>sso-common</artifactId>

<version>STABLE-1.0.9</version>

</dependency>

<dependency>

<groupId>com.gome</groupId>

<artifactId>memberCore-lang</artifactId>

<version>STABLE-1.0.6</version>

</dependency>

<dependency>

<groupId>com.gome</groupId>

<artifactId>memberCore-utils</artifactId>

<version>STABLE-1.0.6</version>

</dependency>

### 5.6.2dubbo接口和redis配置

<bean name="cacheService" class="com.gome.memberCore.utils.tools.impl.CacheServiceDirect">

<property name="zookeeperIpAndPort" value="${zookeeperIpAndPort}"/>

<property name="business" value="SSO"/>

</bean>

<dubbo:service registry="memberZookeeper" protocol="memberGroup" interface="com.gome.sso.facade.IUserSsoFacade" ref="userSsoFacade"group="unifyLogin"/>

### 5.6.3登录或者注册后保存登录状态

1：登录或者注册后获取加密的SCN

2：添加SCN cookie

//第一个参数：response

//第二个参数：encryptCookie加密的SCN

//第三个参数：domain 域名

//第四个参数：isAutoLogin是否是自动登录，自动登录会默认保存cookie7天

public static boolean com.gome.sso.common.outside.SsoUserCookieToolsOutside.addUserCookieOnly(HttpServletResponse response,StringencryptCookie,Stringdomain,BooleanisAutoLogin)

### 5.6.4验证登录状态

有两种方案供选择：直接使用工具类校验状态，使用Filter

* 直接使用工具类验证
* 校验逻辑概述：先校验 cookie->再校验redis->如果redis没有->会请求sso接口校验
* 参数接口说明：

//第一个参数：request

//第二个参数：response

//第三个参数：cacheService: redis客户端

//第四个参数：userSsoFacade: ssodubbo服务

//第五个参数：invokeFrom : 为各组分配的invokeFrom,没有需要申请

//第六个参数：domain : 为调用方的域

com.gome.sso.common.outside.SsoUserCookieToolsOutside.checkUserIsLogin(HttpServletRequest request, HttpServletResponse response,CacheService cacheService,IUserSsoFacade userSsoFacade,String invokeFrom,String domain)

使用Filter，配置如下

<filter>

<filter-name>LoginCheckCacheFilter</filter-name>

<filter-class>com.gome.sso.common.outside.filter.LoginCheckCacheFilter</filter-class>

<init-param>

<param-name>cacheServiceName</param-name>

<param-value>cacheService</param-value><-- 分配的invokeFrom -->

</init-param>

<init-param>

<param-name>ssoInterfaceName</param-name>

<param-value>userSsoFacade</param-value><--ssodubbo服务-->

</init-param>

<init-param>

<param-name>invokeFrom</param-name><-- 分配的invokeFrom -->

<param-value>gomeHome</param-value>

</init-param>

<init-param>

<param-name>domain</param-name><-- 调用方的域 -->

<param-value>.atgsit.com.cn</param-value><-- 配置成调用方的域 -->

</init-param>

</filter>

<filter-mapping>

<filter-name>LoginCheckCacheFilter</filter-name>

<url-pattern>\*.no</url-pattern><-- 配置想要的后缀-->

</filter-mapping>

<filter-mapping>

<filter-name>LoginCheckCacheFilter</filter-name>

<url-pattern>\*.do</url-pattern><-- 配置想要的后缀-->

</filter-mapping>

<filter-mapping>

<filter-name>LoginCheckCacheFilter</filter-name>

<url-pattern>\*.jsp</url-pattern><-- 配置想要的后缀-->

</filter-mapping>

### 5.6.5延长登录状态

#### 5.6.5.1接口名

com.gome.sso.facade.IUserSsoFacade.extendLoginAuth(String scn,int extendSeconds,String invokeFrom)

#### 5.6.5.2Dubbo接口客户端配置

<dubbo:referenceinterface="com.gome.sso.facade.IUserSsoFacade" id="userSsoFacade" group="unifyLogin"/>

#### 5.6.5.3需要统一登录系统分配的参数

* 调用方标识：invokeFrom

#### 5.6.5.4接口说明及调用Demo

* 只能根据集团统一登录系统的userId获取用户信息，而不是根据各公司自己的用户id获取。
* Demo:

**Publicvoid**getItemById(){  
//第一个参数：登录或者注册成功后获取的SCN

//第二个参数：需要延长的时间，单位时间：秒

//第三个参数：调用方标识

UserResult<Boolean> result=ssoFacade.extendLoginAuth("scn",3600, "web");

**if**(result.isSuccess()){//result一定会返回，不用判断null获取成功

//延长登录状态成功

}**else**{//注册失败

//延长登录状态失败

}

}

### 5.6.6不需要登录或者注册即可保存登录状态

#### 5.6.6.1接口名

com.gome.sso.facade.IUserSsoFacade.saveUserToCache(UserInfoCache userInfo,String invokeFrom,String companyName,boolean isAutoLogin)

#### 5.6.6.2Dubbo接口客户端配置

<dubbo:referenceinterface="com.gome.sso.facade.IUserSsoFacade" id="userSsoFacade" group="unifyLogin"/>

#### 5.6.6.3需要统一登录系统分配的参数

* 调用方标识：invokeFrom
* 公司名称：companyName

#### 5.6.6.4接口说明及调用Demo

说明：

* 调用此接口需要知道用户id和login，login可以是用户的手机号，邮箱或者登录名。

Demo:

**Publicvoid**testSaveUserToCache(){  
 UserInfoCache info = **new** UserInfoCache();

//第一个参数：为userInfo，需要login和用户id

//第二个参数invokeFrom：给调用方分配的invokeFrom，接口全新调用使用

//第三个参数companyName：集团统一登录系统会根据companyName来匹配各公司登录需要保持的时间,不填默认保存1天

//第四个参数isAutoLogin：true/false是否自动登录，自动登录默认设置登录状态为7天

**MapResult**<SsoAuthModelSCN> result = userSsoFacade.saveUserToCache(userInfo,"test", "公司名称", **false**);

//获取SCN

String sCN = result.getBuessObj().getSCN();//SCN

}

## 5.7根据userId获取用户信息

### 5.7.1接口名

com.gome.userCenter.facade.impl.userservice.profile.UserInfoFacade.getItemByIdForUnify(String userId, String invokeFrom, Map<String, Object> param)

说明：

param必须传入为调用方分配的companyName,注意区分大小写，例如：

param.put(“companyName”,”test”);

### 5.7.2Dubbo接口客户端配置

<dubbo:referenceinterface="com.gome.userCenter.facade.userservice.profile.IUserInfoFacade" id="userInfoFacade" group="unifyLogin"/>

### 5.7.3需要统一登录系统分配的参数

* 调用方标识：invokeFrom

### 5.7.4接口说明及调用Demo

* 只能根据集团统一登录系统的userId获取用户信息，而不是根据各公司自己的用户id获取。
* Demo:

**Publicvoid**getItemById(){

Map<String,Object> param = new HashMap<String,Object>();

param.put(“companyName”,”test”);

MapResult<UnifyUserInfo>result=userInfoFacade.getItemByIdForUnify("25507682","test",param);

//参数1: userId，必须为数字

//参数2: 调用方标识参数，由统一登录系统为调用方分配

//参数3: 至少传入统一登录系统为调用方分配的companyName

**if**(result.isSuccess()){//result一定会返回，不用判断null获取成功

//获取用户信息

UnifyUserInfo user=result.getBuessObj();

System.*out*.println("老数据各公司自己的userId:"+result.getBuessObj().getCompanyUserId());

}**else**{//注册失败

System.*out*.println("获取失败信息"+result.getMessage());

}

}

错误枚举：

400,"用户信息不存在"

401,"参数invokeFrom校验不通过"

403,"传递的参数不正确"

500,"接口内部异常"

428,"companyName校验不通过"

508,"用户信息不存在或该用户未授权"

## 5.8修改用户信息-校验手机号或邮箱是否已被使用

### 5.8.1接口名

com.gome.userCenter.facade.impl.userservice.profile.EmailMobileFacade.isEmailOrMobileUsed(String emailOrMobile, String invokeFrom, Map<String, Object>param)

说明：

param必须传入为调用方分配的companyName,注意区分大小写，例如：

param.put(“companyName”,”test”);

### 5.8.2 Dubbo接口客户端配置

<dubbo:referenceinterface="com.gome.userCenter.facade.userservice.profile.IEmailMobileFacade " id="emailMobileFacade" group="unifyLogin"/>

### 5.8.3需要统一登录系统分配的参数

* 调用方标识：invokeFrom

### 5.8.4接口说明及调用Demo

**Publicvoid**isEmailOrMobileUsed(){

Map<String,Object> param = new HashMap<String,Object>();

param.put(“companyName”,”test”);

UserResult<Boolean> result = emailMobileFacade.isEmailOrMobileUsed("18101203018", "test",param);

//参数1：要查询的手机号

//参数2：调用方标识参数，由统一登录系统为调用方分配

**if**(result.isSuccess()){

//查询成功

**if**(result.getBuessObj()){

//存在

}**else**{

//不存在

}

}**else**{

//查询失败

}

}

## 5.9 重置密码

### 5.9.1接口名

com.gome.userCenter.facade.accountSecurity.IPasswordAuthFacade.resetPasswordForUnifyUser(String userId, String password, String confirm, String invokeFrom, Map<String, Object> param)

说明：

param必须传入为调用方分配的companyName,注意区分大小写，例如：

param.put(“companyName”,”test”);

### 5.9.2Dubbo接口客户端配置

<dubbo:referenceinterface="com.gome.userCenter.facade.accountSecurity.IPasswordAuthFacade" id="passwordAuthFacade" group="unifyLogin"/>

### 5.9.3需要统一登录系统分配的参数

* 调用方标识：invokeFrom

### 5.9.4接口说明及调用Demo

说明：统一登录系统会比较第二个和第三个参数是否一样，如果不一样会返回修改失败

**Publicvoid**resetPasswordForUnifyUser(){

Map<String,String> map = new HashMap<String,String>();

String userId = “1231”;

String password = “a111111”;

String confirm = “a111111”;

String invokeFrom = “gomeOnline”;

map .put(“companyName”,”test”);

**MapResult<Map<String,String>>** result = passwordAuthFacade.resetPasswordForUnifyUser(userId,password,confirm,invokeFrom，map );

//参数1：用户id

//参数2：未加密的密码

//参数3：确认密码，第二次输入的密码

//参数4：调用方标识参数，由统一登录系统为调用方分配

//参数5：至少传入统一登录系统为调用方分配的companyName

**if**(result.isSuccess){

//修改成功，

}**else**{

//调用失败或修改失败，返回错误code 和 message

}

}

错误枚举：

401,"参数invokeFrom校验不通过"

500,"接口内部异常"

1001,"新密码不能为空"

1002,"确认密码不能为空"

3001,"6-20个字符，字母、数字或符号的组合"

3002,"不能为同一字符"

3003,"不能全为数字"

3004,"非法字符，请使用字母加数字或符号的组合，6-20个字符"

3006,"两次密码不一致"

3007,"您的密码与账户信息太重合，有被盗风险，请换一个密码"

3008,"新密码和旧密码相同"

3009,"您的密码与账户信息太重合，有被盗风险，请换一个密码"

3012,"第三方用户不能修改密码" 注：指的是未完善过个人信息的第三方用户

6001,"用户id为空"

6002,"用户id不存在"

428,"companyName校验不通过"

111,"未授权用户，不支持跨渠道找回密码"

注：若接入方对密码有特殊要求，此错误枚举需相应进行扩充

## 5.10修改及激活手机号

### 5.10.1接口名

com.gome.userCenter.facade.impl.userservice.othergroup.CSCUserFacade.updateMobileAndIsActivated(String userId, String newMobile, String isMobileActivated, String invokeFrom, Map<String, Object> param)

说明：

param必须传入为调用方分配的companyName,注意区分大小写，例如：

param.put(“companyName”,”test”);

### 5.10.2Dubbo接口客户端配置

<dubbo:referenceinterface="com.gome.userCenter.facade.userservice.othergroup.ICSCUserFacade " id="cSCUserFacade" group="unifyLogin"/>

### 5.10.3需要统一登录系统分配的参数

* 调用方标识：invokeFrom

### 5.10.4接口说明及调用Demo

**Publicvoid**updateMobileAndIsActived(){

Map<String,Object> param = new HashMap<String,Object>();

param.put(“companyName”,”test”);

UserResult<Integer> result = cSCUserFacade.updateMobileAndIsActivated("25507682", "13098057568", "Y", "test",param);

//参数1：用户id

//参数2：要被修改成的手机号

//参数3：要修改的手机号激活状态，'Y'标识激活，'N'标识不激活

//参数4：调用方标识参数，由统一登录系统为调用方分配

//参数5：至少传入统一登录系统为调用方分配的companyName

**if**(result.isSuccess()){//result一定会返回，不用判断null，修改成功

System.*out*.println("修改成功"+JSONObject.*toJSONString*(result));

System.*out*.println("修改数量:"+result.getBuessObj());

}**else**{//注册失败

System.*out*.println("获取失败信息"+result.getMessage());

}

}

错误枚举：

428,"companyName校验不通过"

1025,"未授权用户，不支持跨渠道修改手机信息"

## 5.11修改及激活邮箱

### 5.11.1接口名

com.gome.userCenter.facade.impl.userservice.othergroup.CSCUserFacade.updateEmailAndIsActivated(String userId, String newEmail, String isEmailActivated, String invokeFrom, Map<String, Object> param)

说明：

param必须传入为调用方分配的companyName,注意区分大小写，例如：

param.put(“companyName”,”test”);

### 5.11.2Dubbo接口客户端配置

<dubbo:referenceinterface="com.gome.userCenter.facade.userservice.othergroup.ICSCUserFacade " id="cSCUserFacade" group="unifyLogin"/>

### 5.11.3需要统一登录系统分配的参数

* 调用方标识：invokeFrom

### 5.11.4接口说明及调用Demo

**Publicvoid**updateEmailAndIsActivated(){

Map<String,Object> param = new HashMap<String,Object>();

param.put(“companyName”,”test”);

UserResult<Integer> result = cSCUserFacade.updateEmailAndIsActivated("25507682", "yjp322@126.com", "Y", "test",param);

//参数1：用户id

//参数2：要被修改成的邮箱

//参数3：要修改的手机号激活状态，'Y'标识激活，'N'标识不激活

//参数4：调用方标识参数，由统一登录系统为调用方分配

//参数5：至少传入统一登录系统为调用方分配的companyName

**if**(result.isSuccess()){//result一定会返回，不用判断null，修改成功

System.*out*.println("修改成功"+JSONObject.*toJSONString*(result));

System.*out*.println("修改数量:"+result.getBuessObj());

}**else**{//注册失败

System.*out*.println("获取失败信息"+result.getMessage());

}

}

错误枚举：

428,"companyName校验不通过"

2023,"未授权用户，不支持跨渠道修改邮箱信息"

## 5.12修改手机号激活状态

### 5.12.1接口名

com.gome.userCenter.facade.impl.userservice.othergroup.CSCUserFacade.updateIsMobileActivated(String userId, String isMobileActivated, String invokeFrom, Map<String, Object> param)

说明：

param必须传入为调用方分配的companyName,注意区分大小写，例如：

param.put(“companyName”,”test”);

### 5.12.2Dubbo接口客户端配置

<dubbo:referenceinterface="com.gome.userCenter.facade.userservice.othergroup.ICSCUserFacade " id="cSCUserFacade" group="unifyLogin"/>

### 5.12.3需要统一登录系统分配的参数

* 调用方标识：invokeFrom

### 5.12.4接口说明及调用Demo

**Publicvoid**updateIsMobileActivated(){

Map<String,Object> param = new HashMap<String,Object>();

param.put(“companyName”,”test”);

UserResult<Integer> result = cSCUserFacade.updateIsMobileActivated("2561230","Y","test",param);

//参数1:用户id

//参数2:手机号激活状态'Y'激活，'N'不激活

//参数3:调用方标识参数，由统一登录系统为调用方分配

//参数4：至少传入统一登录系统为调用方分配的companyName

**if**(result.isSuccess()){//result一定会返回，不用判断null，修改成功

System.*out*.println("修改成功"+JSONObject.*toJSONString*(result));

System.*out*.println("修改数量:"+result.getBuessObj());

}**else**{//注册失败

System.*out*.println("获取失败信息"+result.getMessage());

}

}

错误枚举：

428,"companyName校验不通过"

1025,"未授权用户，不支持跨渠道修改手机信息"

## 5.13修改邮箱激活状态

### 5.13.1接口名

com.gome.userCenter.facade.impl.userservice.othergroup.CSCUserFacade.updateIsEmailActivated(String userId, String isEmailActivated, String invokeFrom, Map<String, Object> param)

说明：

param必须传入为调用方分配的companyName,注意区分大小写，例如：

param.put(“companyName”,”test”);

### 5.13.2Dubbo接口客户端配置

<dubbo:referenceinterface="com.gome.userCenter.facade.userservice.othergroup.ICSCUserFacade " id="cSCUserFacade" group="unifyLogin"/>

### 5.13.3需要统一登录系统分配的参数

* 调用方标识：invokeFrom

### 5.13.4说明及调用Demo

**Publicvoid**updateIsEmailActivated(){

Map<String,Object> param = new HashMap<String,Object>();

param.put(“companyName”,”test”);

UserResult<Integer> result = cSCUserFacade.updateIsEmailActivated("2561230","Y","test"，param);

//参数1:用户id

//参数2:手机号激活状态'Y'激活，'N'不激活

//参数3:调用方标识参数，由统一登录系统为调用方分配

//参数4：至少传入统一登录系统为调用方分配的companyName

**if**(result.isSuccess()){//result一定会返回，不用判断null，修改成功

System.*out*.println("修改成功"+JSONObject.*toJSONString*(result));

System.*out*.println("修改数量:"+result.getBuessObj());

}**else**{//注册失败

System.*out*.println("获取失败信息"+result.getMessage());

}

}

错误枚举：

428,"companyName校验不通过"

2023,"未授权用户，不支持跨渠道修改邮箱信息"

## 5.14第三方用户完善个人信息

### 5.14.1接口名

com.gome.userCenter.facade.userservice.profile.IUnifyUserInfoFacade.completeUserInfo(Map<String, Object>infoMap, String invokeFrom)

说明：

* 此接口用于第三方用户通过授权登录之后，如需完善资料时调用，可以完善的字段有：login、nickname、password、mobile、email等

### 5.14.2Dubbo接口客户端配置

<dubbo:referenceinterface="com.gome.userCenter.facade.userservice.profile.IUnifyUserInfoFacade" id="unifyUserInfoFacade" group="unifyLogin"/>

### 5.14.3需要统一登录系统分配的参数

* 调用方标识：invokeFrom

### 5.14.4接口说明及调用Demo

说明：调用此接口会把完善个人资料字段设置为“Y”,同时会设置个人信息；

此接口供第三方授权登录用户调用，用于完善个人资料，具体修改字段由接入方根据自身需求决定，此接口只能调用一次。

调用接口前，必须先调用5.18接口检查是否允许进行此操作，绑定手机、邮箱等信息，必须经过用户短信/邮箱验证码确认。

**Publicvoid**testCompleteUserInfo(){

Map<String,Object>param = **new**HashMap<String, Object>();

param.put("userId",onlineUserId);//必填

param.put("password","111222");//根据业务情况设置

param.put("mobile","18233560243");//根据业务情况设置

param.put("email","1823@qq.com");//根据业务情况设置

param.put("login","myaccount");//根据业务情况设置

param.put("nickname","hello");//根据业务情况设置

***param.put(“companyName”,”test”);//必传***

MapResult<Integer> result = unifyUserInfoFacade.completeUserInfo(param,"gomeOnline");

//参数1:用户信息map

//参数2:调用方标识参数，由统一登录系统为调用方分配

**if**(result.isSuccess()){

//修改成功

System.*out*.println("修改成功"+JSONObject.*toJSONString*(result));

}**else**{

//修改失败，返回错误code和msg

System.*out*.println("获取失败信息"+result.getMessage());

}

}

错误枚举：

403,"传递的参数不正确"

400,"用户信息不存在"

500,"接口内部异常"

409,"用户名已存在"

410,"手机号已存在"411,"邮箱已存在"

428,"companyName校验不通过"

5002, "未授权用户，不支持完善个人信息"

## 5.15校验用户是否可以找回密码

### 5.15.1接口名

com.gome.userCenter.facade.impl.accountSecurity.PasswordAuthFacade.getResetPasswordInfoWithDetailByLogin(String login, String invokeFrom, Map<String, Object> param)

说明：登录前用户找回密码使用，可以通过登录名，手机号或者邮箱找回；

此接口通过登录名，手机号或者邮箱校验用户是否可以找回密码。

**注意：如果输入的用户名，在不同渠道重复，此接口会将每个渠道的用户全部返回，供接入方自行选择处理**

### 5.15.2Dubbo接口客户端配置

<dubbo:referenceinterface="com.gome.userCenter.facade.accountSecurity.IPasswordAuthFacade" id="passwordAuthFacade" group="unifyLogin"/>

### 5.15.3需要统一登录系统分配的参数

* 调用方标识：invokeFrom
* 返回值说明：

允许修改时:

isSuccess为true，buessObj包含以下内容

userId-用户id;

mobile-手机号;

isMobileActivated-手机是否激活，取值Y或N;

email-邮件;

isEmailActivated-邮件是否激活，取值Y或N;

userName–用户输入的login

subCompanyCode – 用户所属渠道编号，枚举见：com.gome.userCenter.model.enu.CompanyNameEnum

不允许修改时：

isSuccess为false，code和msg返回错误信息

### 5.15.4接口说明及调用Demo

**Publicvoid**getUserResetPasswordInfoByLogin(){

Map<String,Object> param = new HashMap<String,Object>();

param.put(“companyName”,”test”);

**MapResult<Map<String,Object>>** result = passwordAuthFacade.getResetPasswordInfoWithDetailByLogin ("abc123456","test"，param);

//参数1：用户名

//参数2：调用方标识参数，由统一登录系统为调用方分配

//参数3：至少传入统一登录系统为调用方分配的companyName

**if**(result.isSuccess){

//可以找回密码，返回示例{"message":null,"code":0,"success":true,"buessObj":{"isEmailActivated":"N","email":"","whereFrom":null,"userName":"durng","isMobileActivated":"Y","mobile":"150\*\*\*\*0371","userId":"123456"}}

}**else**{

//不能找回密码，

**返回示例1**：同一渠道内冲突用户，提示用户数据错误

{"message":"数据错误，不能找回密码","code":501,"success":false,"buessObj":null}

**返回示例2**：国美和库巴冲突用户，提示用户进行选择

{"message":"库巴冲突用户","code":104,"success":false,"buessObj":{"userName":"123","item0":{"isEmailActivated":"N","email":"","whereFrom":null,"userName":"durng@coo8","isMobileActivated":"Y","mobile":"150\*\*\*\*0371"},"item1":{"isEmailActivated":"N","email":"","whereFrom":null,"userName":"durng","isMobileActivated":"Y","mobile":"130\*\*\*\*1211"}}}

**返回示例3**：不同渠道冲突用户，提示用户进行选择

{"extParams":null,"message":"不同渠道冲突用户","code":108,"success":false,"buessObj":{"item2":{"isEmailActivated":"N","nickName":null,"email":null,"userId":"100033555874","login":"guanjia0004","isMobileActivated":"Y","mobile":"15010013782","subCompanyCode":"3"},"item1":{"isEmailActivated":"Y","nickName":"汉译英01","email":"hu\*\*\*110@126.com","userId":"869473036","login":"hanxiying@yolo24.com","isMobileActivated":"Y","mobile":"15010013782","subCompanyCode":"1"},"item0":{"isEmailActivated":"N","nickName":null,"email":null,"userId":"100033555556","login":"test11130003","isMobileActivated":"Y","mobile":"15010013782","subCompanyCode":"2"},"userName":"15010013782"}}

**返回示例4**：未激活手机或邮箱，不支持找回密码

{"extParams":null,"message":"未激活手机或邮箱，不支持找回密码","code":109,"success":false,"buessObj":null}

}

}

错误码枚举：

104,"库巴冲突用户"

105,"用户不存在"

106,"门店用户，不能找回密码"

107,"第三方用户，不支持找回密码"

108,"不同渠道冲突用户"

109,"未激活手机或邮箱，不支持找回密码"

401,"参数invokeFrom校验不通过"

402,"非空参数不能传递空值"

500,"接口内部异常"

501,"数据错误，不能找回密码"

111,"未授权用户，不支持跨渠道找回密码"

428,"companyName校验不通过"

## 5.16校验用户是否可以修改密码

### 5.16.1接口名

com.gome.userCenter.facade.impl.accountSecurity.PasswordAuthFacade.checkUserAbleToUpdatePassword(String userId, String invokeFrom, Map<String, Object> param)

说明：

* 登录之后用户修改密码使用；通过userId校验用户是否可以修改密码。

param必须传入为调用方分配的companyName,注意区分大小写，例如：

param.put(“companyName”,”test”);

* 返回值说明：

允许修改时:

isSuccess为true，buessObj包含以下内容

mobile-手机号;

isMobileActivated-手机是否激活，取值Y或N;

email-邮件;

isEmailActivated-邮件是否激活，取值Y或N;

hasSetPassword - 是否已设置密码，取值Y或N

注：修改密码时必须先验证手机或邮箱，且当用户已设置密码时，必须输入正确的旧密码:

不允许修改时：

isSuccess为false，code和msg返回错误信息

### 5.16.2Dubbo接口客户端配置

<dubbo:referenceinterface="com.gome.userCenter.facade.accountSecurity.IPasswordAuthFacade" id="passwordAuthFacade" group="unifyLogin"/>

### 5.16.3需要统一登录系统分配的参数

* 调用方标识：invokeFrom

### 5.16.4接口说明及调用Demo

**Publicvoid**checkUserAbleToUpdatePassword(){

Map<String,Object> param = new HashMap<String,Object>();

param.put(“companyName”,”test”);

**MapResult<Map<String,String>>** result = passwordAuthFacade.checkUserAbleToUpdatePassword("123123123","test"，param);

//参数1：用户id

//参数2：调用方标识参数，由统一登录系统为调用方分配

//参数3：至少传入统一登录系统为调用方分配的companyName

**if**(result.isSuccess){

//允许修改，result.setBuessObj获取数据map,map中包含以下key：

\* mobile-手机号;

\* isMobileActivated-手机是否激活，取值Y或N;

\* email-邮件;

\* isEmailActivated-邮件是否激活，取值Y或N;

\* hasSetPassword - 是否已设置密码

}**else**{

//不允许修改，返回错误code 和 message

}

}

错误枚举：

400,"用户信息不存在"

401,"参数invokeFrom校验不通过"

403,"传递的参数不正确"

500,"接口内部异常"

3008, "第三方用户未完善个人资料，无法修改密码"

3009, "请先绑定手机或邮箱，再进行密码修改"

428,"companyName校验不通过"

3010, "未授权用户，不支持跨渠道修改密码"

## 5.17修改密码

### 5.17.1接口名

com.gome.userCenter.facade.impl.accountSecurity.PasswordAuthFacade.updatePasswordForUnifyUser(Map<String, String> paramMap, String invokeFrom)

### 5.17.2Dubbo接口客户端配置

<dubbo:referenceinterface="com.gome.userCenter.facade.accountSecurity.IPasswordAuthFacade" id="passwordAuthFacade" group="unifyLogin"/>

### 5.17.3需要统一登录系统分配的参数

* 调用方标识：invokeFrom

### 5.17.4接口说明及调用Demo

**Publicvoid**updatePasswordForUnifyUser(){

Map<String,String> map = new HashMap<String,String>();

map.put("userId","123123");

map.put("oldPassword","123456a");

map.put("password","a111111");

map.put("confirmPassword","a111111");

**map.put(“companyName”,”test”);//必传参数**

**MapResult<Map<String,String>>** result = passwordAuthFacade.updatePasswordForUnifyUser(map,"test");

//参数1：用户id、统一登录系统为调用方分配的companyName等

//参数2：调用方标识参数，由统一登录系统为调用方分配

**if**(result.isSuccess){

//修改成功，

}**else**{

//调用失败或修改失败，返回错误code 和 message

}

}

错误枚举：

401,"参数invokeFrom校验不通过"

403,"传递的参数不正确"

500,"接口内部异常"

1001,"新密码不能为空"

1002,"确认密码不能为空"

2001,"ele\_oldPassword","当前密码输入错误"

3001,"6-20个字符，字母、数字或符号的组合"

3002,"不能为同一字符"

3003,"不能全为数字"

3004,"非法字符，请使用字母加数字或符号的组合，6-20个字符"

3006,"两次密码不一致"

3007,"您的密码与账户信息太重合，有被盗风险，请换一个密码"

3008,"新密码和旧密码相同"

3012,"第三方用户不能修改密码" 注：指的是未完善过个人信息的第三方用户

6001,"用户id为空"

6002,"用户id不存在"

428,"companyName校验不通过"

3010, "未授权用户，不支持跨渠道修改密码"

## 5.18 查询第三方用户完善资料状态

### 5.18.1接口名

com.gome.userCenter.facade.userservice.profile.IUnifyUserInfoFacade.getUserInfoCompletedStatus(String userId, [String](eclipse-javadoc:%E2%98%82=userCenter-facade/src%5C/main%5C/java%3Ccom.gome.userCenter.facade.userservice.profile%7BIForgotPasswordFacade.java%E2%98%83IForgotPasswordFacade~updatePassword~QString;~QString;~QString;%E2%98%82String) invokeFrom，Map<String,Object> param)

说明：

* 调用5.14功能之前使用；通过userId校验第三方用户是否需要完善个人资料。
* 返回值说明：

需要完善时:

isSuccess为true

不需要完善时：

isSuccess为false，code和msg返回错误信息

### 5.18.2Dubbo接口客户端配置

<dubbo:referenceinterface="com.gome.userCenter.facade.userservice.profile.IUnifyUserInfoFacade" id="unifyUserInfoFacade" group="unifyLogin"/>

### 5.18.3需要统一登录系统分配的参数

* 调用方标识：invokeFrom

### 5.18.4接口说明及调用Demo

Publicvoid getUserInfoCompletedStatus(){

Map<String,String> map = new HashMap<String,String>();

**map.put(“companyName”,”test”);//必传参数**

**MapResult<Map<String,String>>** result = unifyUserInfoFacade.getUserInfoCompletedStatus(“123123”,"test"，map);

//参数1：用户id

//参数2：调用方标识参数，由统一登录系统为调用方分配

**if**(result.isSuccess){

//允许修改

}**else**{

//不允许修改，返回错误code 和 message

}

}

错误枚举：

400,"用户信息不存在"

403,"传递的参数不正确"

407,"非第三方用户不能完善个人资料

408,"已完善资料的用户不能重复调用"

500,"接口内部异常"

5002, "未授权用户，不支持完善个人信息"

428,"companyName校验不通过"

## 5.19校验用户是否可以找回密码，根据优先级返回用户(推荐)

### 5.19.1接口名

com.gome.userCenter.facade.impl.accountSecurity.PasswordAuthFacade.getResetPasswordInfoByPriority(String login, String invokeFrom, Map<String, Object> param)

说明：登录前用户找回密码使用，可以通过登录名，手机号或者邮箱找回；

此接口通过登录名，手机号或者邮箱校验用户是否可以找回密码。

**注意：如果输入的用户名，在不同渠道重复，此接口会按照渠道优先级筛选，并返回唯一的一个用户；此点和5.15不同，5.15是把所有的渠道冲突账号返回供各客户端调用**

### 5.19.2Dubbo接口客户端配置

<dubbo:referenceinterface="com.gome.userCenter.facade.accountSecurity.IPasswordAuthFacade" id="passwordAuthFacade" group="unifyLogin"/>

### 5.19.3需要统一登录系统分配的参数

* 调用方标识：invokeFrom
* 调用方公司名：companyName；放到Map<String, Object> param参数中
* 返回值说明：

允许修改时:

isSuccess为true，buessObj包含以下内容

userId-用户id;

mobile-手机号;

isMobileActivated-手机是否激活，取值Y或N;

email-邮件;

isEmailActivated-邮件是否激活，取值Y或N;

subCompanyCode – 用户所属渠道编号，枚举见：com.gome.userCenter.model.enu.CompanyNameEnum

不允许修改时：

isSuccess为false，code和msg返回错误信息

### 5.19.4接口说明及调用Demo

**Publicvoid**getUserResetPasswordInfoByLogin(){

Map<String,Object> param = new HashMap<String,Object>();

param.put(“companyName”,”test”);

**MapResult<Map<String,Object>>** result = passwordAuthFacade.getResetPasswordInfoByPriority("abc123456","test",param);

//参数1：用户名

//参数2：调用方标识参数，由统一登录系统为调用方分配

//参数3：至少传入统一登录系统为调用方分配的companyName

**if**(result.isSuccess){

//可以找回密码，返回示例{"message":null,"code":0,"success":true,"buessObj":{"isEmailActivated":"N","email":"","whereFrom":null,"userName":"durng","isMobileActivated":"Y","mobile":"150\*\*\*\*0371","userId":"123456"}}

}**else**{

//不能找回密码，

**返回示例1**：同一渠道内冲突用户，提示用户数据错误

{"message":"数据错误，不能找回密码","code":501,"success":false,"buessObj":null}

**返回示例2**：国美和库巴冲突用户，提示用户进行选择

**注：除国美在线外，其他接入方不需要考虑此场景**

{"message":"库巴冲突用户","code":104,"success":false,"buessObj":{"userName":"123","item0":{"isEmailActivated":"N","email":"","whereFrom":null,"userName":"durng@coo8","isMobileActivated":"Y","mobile":"150\*\*\*\*0371"},"item1":{"isEmailActivated":"N","email":"","whereFrom":null,"userName":"durng","isMobileActivated":"Y","mobile":"130\*\*\*\*1211"}}}

**返回示例3**：未激活手机或邮箱，不支持找回密码

{"extParams":null,"message":"未激活手机或邮箱，不支持找回密码","code":109,"success":false,"buessObj":null}

}

}

错误码枚举：

104,"库巴冲突用户"

105,"用户不存在"

106,"门店用户，不能找回密码"

107,"第三方用户，不支持找回密码"

108,"不同渠道冲突用户"

109,"未激活手机或邮箱，不支持找回密码"

401,"参数invokeFrom校验不通过"

402,"非空参数不能传递空值"

500,"接口内部异常"

501,"数据错误，不能找回密码"

111,"未授权用户，不支持跨渠道找回密码"

428,"companyName校验不通过"

## 5.20用户授权

### 5.20.1接口名

com.gome.userCenter.facade.userservice.othergroup.ICSCUserFacade.userAuthorize(String SCN,String companyName,String invokeFrom)

说明：调用此接口授权过的用户将可以在其他公司登录。

自己公司只能授权自己公司的用户。

### 5.20.2Dubbo接口客户端配置

<dubbo:referenceinterface="com.gome.userCenter.facade.userservice.othergroup.ICSCUserFacade" id="cSCUserFacade" group="unifyLogin"/>

### 5.20.3需要统一登录系统分配的参数

* 调用方标识：invokeFrom
* 调用方公司名：companyName
* 用户登录返回SCN:SCN
* 返回值说明：

授权成功：

isSuccess为true

授权失败：

isSuccess为false，code和msg返回错误信息

### 5.20.4接口说明及调用Demo

**publicvoid**userAuthorize (){

MapResult<Integer> result = cSCUserFacade.userAuthorize("asdadffffsdsdsdsd","gomeOnLine","test");

//参数1：SCN

//参数2：调用方公司标识

//参数3：调用方标识

**if**(result.isSuccess()){

//授权成功

}**else**{

//授权失败

}

}

错误码枚举：

503,"非法授权 "

504,"授权失败 "

505,"非法SCN"

## 5.21 查询用户是否可以升级一账通

### 5.21.1接口名

com.gome.userCenter.facade.userservice.profile.IUnifyUserInfoFacade

.checkAuthorize (String userId,String companyName,String invokeFrom)

说明：调用此接口可以判断一个用户是否可以升级为一账通

### 5.21.2Dubbo接口客户端配置

<dubbo:referenceinterface="com.gome.userCenter.facade.userservice.profile.IUnifyUserInfoFacade " id="unifyUserInfoFacade" group="unifyLogin"/>

### 5.21.3需要统一登录系统分配的参数

* 调用方标识：invokeFrom
* 调用方公司名：companyName
* 用户id:userId
* 返回值说明：

可以升级：

isSuccess为true

不能升级：

isSuccess为false，code和msg返回错误信息

### 5.21.4接口说明及调用Demo

**publicvoid**checkAuthorize (){

IUnifyUserInfoFacade unifyUserInfoFacade=(IUnifyUserInfoFacade)context.getBean("unifyUserInfoFacade");

MapResult<Boolean> result = unifyUserInfoFacade.checkAuthorize("112345", "gomeOnLine", "test");

//参数1：用户id

//参数2：调用方公司标识

//参数3：调用方标识

**if**(result.isSuccess()){

//可以升级

}**else**{

//不可以升级

}

}

错误码枚举：

428,"companyName校验不通过"

503: 非法升级

506,"第三方用户不能升级"

507,"该用户已升级"

## 5.22手机号是否为被其它用户绑定

### 5.22.1接口名

com.gome.userCenter.facade.register.IUserRegisterFacade.isBindMobile(String Mobile,String invokeFrom,Map extMap);

说明：调用此接口可以判断手机号是否为绑定手机号

### 5.22.2Dubbo接口客户端配置

<dubbo:referenceinterface="com.gome.userCenter.facade.register.IUserRegisterFacade" id="userRegisterFacade" group="unifyLogin"/>

### 5.22.3需要统一登录系统分配的参数

* 调用方标识：invokeFrom
* 手机号：mobile
* 拓展参数:extMap
* extMap.put("companyName", "gomeOnLine");
* 返回值说明：

可以升级：

isSuccess为true

不能升级：

isSuccess为false，code和msg返回错误信息

### 5.22.4接口说明及调用Demo

**publicvoidisBindMobile** (){

IUserRegisterFacade userRegisterFacade=(IUserRegisterFacade)context.getBean("userRegisterFacade");

Map<String, Object> extMap = new HashMap<String, Object>();

extMap.put("companyName", "gomeOnLine");

MapResult<Boolean> bindMobile= userRegisterFacade.isBindMobile("15311818899", "gomeOnLine", extMap);

//参数1：调用方标识

//参数2：手机号

//参数3：拓展参数

if (bindMobile.isSuccess())

{

} else

{

}

错误码枚举：

400,"用户不存在"

## 5.23第三方首次登录后绑定到现有手机号的账号上并登陆

### 5.23.1接口名

com.gome.userCenter.facade.login.IUserLoginFacade.mobileLogin(String mobile,RequestParams requestParams,Map map)

说明：

1：第三方首次登录后绑定到现有手机号的账号上

2：绑定到手机号的账号并登录，返回手机号所属账号信息

### 5.23.2Dubbo接口客户端配置

<dubbo:referenceinterface="com.gome.userCenter.facade.login.IUserLoginFacade" id="userInfoFacade" group="unifyLogin"/>

### 5.23.3需要统一登录系统分配的参数

* 手机号：mobile
* 请求参数：requestParams
* 拓展map:map
* 公司名：companyName
* 是否绑定第三方用户：isBindSnsUser
* 第三方用户信息：gomeSnsUser
* 第三方唯一id：SNSUnionId
* 注册源：registerSource
* 返回值说明：

绑定成功：

isSuccess为true

绑定不成功：

isSuccess为false，code和msg返回错误信息

### 5.23.4接口说明及调用Demo

Map<String, Object> extMap = new HashMap<String, Object>();

extMap.put("companyName", "gomeOnLine");

extMap.put("isBindSnsUser", "true");

GomeSNSUser snsUser = new GomeSNSUser(null, whereFrom, snsUserId, snsUserName);

extMap.put("gomeSnsUser", snsUser);

extMap.put(“SNSUnionId”, snsUserId);

extMap.put(“registerSource”, “gomeOnline”);

RequestParams requestParams = controllerTools.initRequestParams(request); UserLoginResult<UserInfo> mobileLogin = userLoginFacade.mobileLogin(phoneNum, requestParams, extMap);

if (mobileLogin.isSuccess()) {

} else {

}

## 5.24根据用户id查询用户授权状态以及授权时间

### 5.24.1接口名

com.gome.userCenter.facade.unify.ISsoUnifyFacade.queryAuthInfoById(String userId, String invokeFrom, Map<String, Object> extMap)

**功能说明：此接口用于根据授权与否、授权时间决定送优惠券等，各公司自行根据结果处理。**

### 5.24.2Dubbo接口客户端配置

<dubbo:reference interface="

com.gome.userCenter.facade.unify.ISsoUnifyFacade" id="ssoUnifyFacade" group="unifyLogin"/>

### 5.24.3需要统一登录系统分配的参数

* userId : 用户ID
* invokeFrom : 调用来源
* 拓展参数:extMap

如：extMap.put("companyName", "gomeOnLine");

### 5.24.4接口说明及调用Demo

Map<String, Object> extMap = new HashMap<String, Object>();

extMap.put("companyName", "gomeOnLine");

MapResult<Map<String,Object>>result=ssoUnifyFacade.queryAuthInfoById(“10000001521”,”gomeOnLine”,extMap );

if( result.isSuccess()){

//调用成功,获取授权信息集合

Map<String,Object> authInfo= result.getBuessObj();

String isAuthorized = (String) authInfo.get(“isAuthorized”);

If("Y".equalsIgnoreCase(isAuthorized )){

//已授权用户，获取授权时间

String authorizedDate= (String) authInfo.get(“authorizedDate”);

}else{

//未授权用户

}

}else{

//调用失败

}

# 6各功能所用到的接口及调用顺序

## 6.1注册功能相关接口调用说明

* 说明：web端1,2,3步需要页面ajax异步调用接口

1：注册校验用户名是否存在，用户名存在不允许注册，调用5.1.1中说明的接口(com.gome.userCenter.facade.register.IUserRegisterFacade.existUser(String login,String invokeFrom,Map extMap))

2：如果是手机注册，校验手机号是否存在，手机号存在不允许注册，调用5.1.1中说明的接口(com.gome.userCenter.facade.register.IUserRegisterFacade.existUser(String login,String invokeFrom,Map extMap))；发送和校验短信验证码各公司自行实现

3：如果是邮箱注册，校验邮箱是否存在，邮箱存在不允许注册，调用5.1.1中说明的接口(com.gome.userCenter.facade.register.IUserRegisterFacade.existUser(String login,String invokeFrom,Map extMap))；发送和校验邮件各公司自行实现

4：以上3步全部校验通过，走注册流程，调用5.2中说明的注册接口

(com.gome.userCenter.facade.register.IUserRegisterFacade. registerUnifyUser (GomeUnifyRegisterUser gomeUser,Map<String,Object>extMap) )

## 6.2普通登录功能相关接口调用说明

* 说明：
* web端前台页面输入用户名后ajax异步同时调用登录前校验用户名接口和是否显示验证码接口
* App或者wap需要同步调用5.3.1中的是否显示验证码接口；如果有从app或者web接收验证码则校验验证码，没有接收到验证码则调用5.3.1中的是否显示验证码接口

1：登录前校验用户名是否可以登录，调用5.3.2中的接口（com.gome.userCenter.facade.login.IUserLoginFacade.isConflictOrSnsOrNikeName(String login,String invokeFrom, Map<String, Object> extMap)）

2：登录前校验是否显示验证码，如果需要显示验证码，前台页面要显示，调用5.3.1中的接口com.gome.userCenter.facade.login.IUserLoginFacade.isShowAuthenticCodeWithGrade(String login,String ipAddress,String invokeFrom,Map<String, String>captExtMap)；显示和校验验证码各公司自行实现。

3：如果1和2校验通过，会调用5.4普通登录接口（com.gome.userCenter.facade.login.IUserLoginFacade.doLogin(String login,Stringpassword,RequestParamsrequestParams,Map map)）

## 6.3修改手机号功能相关接口调用说明

* 说明：web端第2步需要页面ajax异步调用接口，app或wap同步调用做手机号是否已经被使用校验，如果已经被使用则终止整个流程。

1：查询用户信息，获取用户当前手机号、手机激活状态，调用5.7中说明的接口：com.gome.userCenter.facade.impl.userservice.profile.UserInfoFacade.getItemByIdForUnify(String userId, String invokeFrom, Map<String, Object> param)

2：查询要绑定的手机号是否已经被他人占用，如果被占用此手机号不允许使用，调用5.8中说明的接口：

com.gome.userCenter.facade.userservice.profile.IEmailMobileFacade.isEmailOrMobileUsed(String emailOrMobile,StringinvokeFrom, Map<String, Object> param)

3：发送校验短信验证码等功能由各公司自行实现

4：以上步骤全部校验通过后，进行修改手机的操作，调用5.10中说明的接口：

com.gome.userCenter.facade.userservice.othergroup.ICSCUserFacade.updateMobileAndIsActivated(String userId,StringnewMobile,StringisMobileActivated,StringinvokeFrom, Map<String, Object> param)

## 6.4修改邮箱功能相关接口调用说明

* 说明：web端第2步需要页面ajax异步调用接口，app或wap同步调用做邮箱是否已经被使用校验，如果已经被使用则终止整个流程。

1：查询用户信息，获取用户当前邮箱、邮箱激活状态，调用5.7中说明的接口：com.gome.userCenter.facade.impl.userservice.profile.UserInfoFacade.getItemByIdForUnify(String userId, String invokeFrom, Map<String, Object> param)

2：查询要绑定的邮箱是否已经被他人占用，如果邮箱被占用此邮箱不允许使用，调用5.8中说明的接口：

com.gome.userCenter.facade.userservice.profile.IEmailMobileFacade.isEmailOrMobileUsed(String emailOrMobile,StringinvokeFrom, Map<String, Object> param)

3：发送校验邮件验证码等功能由各公司自行实现

4：以上步骤全部校验通过后，进行修改邮箱的操作，调用5.11中说明的接口：

com.gome.userCenter.facade.userservice.othergroup.ICSCUserFacade.updateEmailAndIsActivated(String userId,String newEmail,String isEmailActivated,StringinvokeFrom, Map<String, Object> param)

## 6.5找回密码功能相关接口调用说明

1：查询用户信息，根据login校验是否可以找回密码，调用5.15中说明的接口：

com.gome.userCenter.facade.accountSecurity.IPasswordAuthFacade.getResetPasswordInfoWithDetailByLogin([String](eclipse-javadoc:%E2%98%82=userCenter-facade/src%5C/main%5C/java%3Ccom.gome.userCenter.facade.userservice.profile%7BIForgotPasswordFacade.java%E2%98%83IForgotPasswordFacade~updatePassword~QString;~QString;~QString;%E2%98%82String)login, [String](eclipse-javadoc:%E2%98%82=userCenter-facade/src%5C/main%5C/java%3Ccom.gome.userCenter.facade.userservice.profile%7BIForgotPasswordFacade.java%E2%98%83IForgotPasswordFacade~updatePassword~QString;~QString;~QString;%E2%98%82String)invokeFrom, Map<String, Object> param)

2：发送校验短信或邮件验证码等功能由各公司自行实现

3：以上步骤完成且符合条件之后，进行重置密码的操作，调用5.9中说明的接口：

com.gome.userCenter.facade.accountSecurity.IPasswordAuthFacade.resetPasswordForUnifyUser(String userId,String password,String confirm,String invokeFrom, Map<String, Object> param)

## 6.6修改密码功能相关接口调用说明

说明：修改密码需要用户是已经登录状态，登录时已经知道userId

1：查询，根据id获取用户是否允许修改密码的状态信息，调用5.16中说明的接口：

com.gome.userCenter.facade.accountSecurity.IPasswordAuthFacade. checkUserAbleToUpdatePassword ([String](eclipse-javadoc:%E2%98%82=userCenter-facade/src%5C/main%5C/java%3Ccom.gome.userCenter.facade.userservice.profile%7BIForgotPasswordFacade.java%E2%98%83IForgotPasswordFacade~updatePassword~QString;~QString;~QString;%E2%98%82String)userId, [String](eclipse-javadoc:%E2%98%82=userCenter-facade/src%5C/main%5C/java%3Ccom.gome.userCenter.facade.userservice.profile%7BIForgotPasswordFacade.java%E2%98%83IForgotPasswordFacade~updatePassword~QString;~QString;~QString;%E2%98%82String)invokeFrom, Map<String, Object> param)

2：发送校验短信或邮件验证码等功能由各公司自行实现

3：以上步骤完成且符合条件之后，进行修改密码的操作，调用5.17中说明的接口：

com.gome.userCenter.facade.accountSecurity.IPasswordAuthFacade.updatePasswordForUnifyUser(Map<String, String> paramMap, [String](eclipse-javadoc:%E2%98%82=userCenter-facade/src%5C/main%5C/java%3Ccom.gome.userCenter.facade.userservice.profile%7BIForgotPasswordFacade.java%E2%98%83IForgotPasswordFacade~updatePassword~QString;~QString;~QString;%E2%98%82String)invokeFrom, Map<String, Object> param)

## 6.7第三方用户完善资料功能相关接口调用说明

说明：第三方用户完善资料前用户必须是登录状态并获取userId

1：查询，根据id获取第三方用户是否需要完善用户资料状态信息，调用5.18中说明的接口：

com.gome.userCenter.facade.userservice.profile.IUnifyUserInfoFacade.getUserInfoCompletedStatus(String userId, [String](eclipse-javadoc:%E2%98%82=userCenter-facade/src%5C/main%5C/java%3Ccom.gome.userCenter.facade.userservice.profile%7BIForgotPasswordFacade.java%E2%98%83IForgotPasswordFacade~updatePassword~QString;~QString;~QString;%E2%98%82String) invokeFrom，Map<String,Object> param)

2：如需要用户设置手机或邮箱，当用户输入手机或邮箱之后，需先检查输入的手机或邮箱是否已被占用，异步调用5.1中说明的接口：

com.gome.userCenter.facade.register.IUserRegisterFacade.existUser(String login)

2：发送校验短信或邮件验证码等功能由各公司自行实现

3：以上步骤完成且符合条件之后，进行完善第三方用户资料的操作，调用5.14中说明的接口：

com.gome.userCenter.facade.userservice.profile.IUnifyUserInfoFacade.completeUserInfo(Map<String, Object>infoMap, String invokeFrom)

## 6.8第三方首次登录后，绑定到现有手机号的账号上并登陆说明

1：查询第三方用户是否是新用户，调用5.5第三方登录的接口：

com.gome.userCenter.facade.login.ISNSLoginFacade.doSNSLogin(GomeSNSUser snsUser,RequestParamsrequestParams, Map map)，接口调用demo中有说明：

**map.put("snsuser\_register","false");//校验第三方用户是否存在**

如果返回结果为6001 不存在第三方用户则表示第三方用户为新用户，执行手机绑定操作。

2： 检测手机号是否存在用户：调用5.22接口：

com.gome.userCenter.facade.register.IUserRegisterFacade.isBindMobile(String Mobile,String invokeFrom,Map extMap);

根据返回值走不同接口：

3.如果返回第三方用户信息不存在时(code:400)，表示该手机号没有对应的用户，跳转注册流程，引导用户注册,调用5.2的用户普通注册流程完成注册，并绑定第三方用户。

4.如果返回成功，①发短信校验用户各自实现。②调用5.23接口：

com.gome.userCenter.facade.login.IUserLoginFacade.mobileLogin(String mobile,RequestParams requestParams,Map map)完成手机号对应的用户和第三方用户的绑定。

# 7参考说明

## 7.1第三方登录WhereFrom参考值

*gome* (0),

*alipay* (1),

*baidu* (2),

*kaixin* (3),

*qq* (4),

*renren* (5),

*sina* (6),

*unionpay* (8),

*netease* (9),

*qihoo* (10),

## 7.2用户登录名login的生成规则

说明：如果对接方公司没有用户登录名login时，统一登录系统对login的生成规则如下：

1、手机号注册，生成的用户名规则：“gm”+“\_”+“手机号”，例如：gm\_13811111111。如果还有重复，生成的用户名规则：“gm”+“\_”+“手机号”+“随机生成三位小写字母”，例如：gm\_13811111111abc；

2、非手机号注册，生成的用户名规则：“gm”+“\_”+“随机生成十位数字和小写字母混合”，例如：gm\_8674hu868t；